Predicting Microbial Gene Fitness with a Biological Knowledge Graph

Winston E. Anthony[1*], Sam Donald[1*], Adam Deutschbauer[2], Sumit Purohit[1], Kaizad Patel[1] and Robert Egbert[1†]

[1]*Integrated Discovery Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA, USA.* [2]*Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

[†]*E-mail: robert.egbert@pnnl.gov*

[*]*These authors contributed equally to this work.*

Predicting the condition-dependent fitness effects of gene inactivation remains a central challenge in microbial phenomics. We introduce GIMME-KG (**G**raph **I**nference for **M**icrobial Dark **M**atter **E**xploration with **K**nowledge **G**raphs), a knowledge graph (KG) integrating 7.94 million RB-TnSeq fitness measurements spanning 55,421 genes, 1,770 experiments, and 10 Pseudomonas strains with experimental metadata, metabolic annotations, and pretrained text and protein-sequence representations. We train GFM1 (GIMME-KG Fitness Model 1), a heterogeneous graph attention network (GAT) to predict gene-inactivation fitness effects under three held-out evaluations: experiments, genes, and ortholog clusters. Message passing over a subset of the knowledge graph increases performance: In the held-out-gene setting, message passing reduced mean squared error (MSE) by 10.1% and increased Pearson correlation by 12.0% relative to a multi-layer perceptron (MLP) using the same multimodal node-encoder design and decoder template. Regression gains were larger among well-connected genes and partly reflected their enrichment for strong fitness defects. On held-out genes, GFM1 had higher precision than the MLP, but lower recall when predicting strong fitness effects. It also outperformed a genome-scale metabolic model in a matched *Pseudomonas putida* benchmark. Together, these results indicate that message passing can leverage knowledge-graph structure to improve gene-fitness prediction when informative relationships to training genes are available. GIMME-KG and GFM1 represent a step forward in predictive inference of microbial metabolism and a new tool for *in silico* phenotype prediction. Appendix is available here: `https://github.com/PNNL-Predictive-Phenomics/GIMME-KG-manuscript`

Keywords: Knowledge graph; microbial fitness; graph neural network; fitness prediction; Pseudomonas.

1. Introduction

The fitness effect of a genetic perturbation depends on the environment in which it is measured. Characterizing these genotype–environment interactions is important for understanding microbial physiology and predicting phenotypes under unmeasured conditions. Predicting these interactions could reduce the experimental burden of mapping genotype–phenotype fitness landscapes and enable *in silico* inference for unmeasured genes and conditions.[1,2] High

© 2026 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

1 INTRODUCTION

throughput genome-wide fitness assays provide a powerful method for rapid experimental assessment of the function and metabolic plasticity of microbes. One such technique, randomly barcoded transposon sequencing (RB-TnSeq), uses barcoded transposon insertions to generate pooled mutant libraries containing disruptions across much of the genome.[3] RB-TnSeq assays have been used to identify conditionally important genes, generate hypotheses about gene function, and associate poorly characterized genes with biological pathways through correlated fitness profiles.[4] The Fitness Browser provides a large collection of RB-TnSeq measurements from 9,532 experiments spanning nutrient and stress conditions across diverse bacterial lineages.[5] These measurements provide a foundation for training models that predict fitness effects for gene–condition combinations not observed during training.

The Gram-negative genus *Pseudomonas* includes environmentally, industrially, and clinically important species with diverse metabolic capabilities.[6,7] Despite extensive experimental and computational characterization, a substantial fraction of genes in *Pseudomonas putida* remain without characterized functions.[8] The combination of large fitness datasets, established metabolic models, and persistent annotation gaps makes *P. putida* and related pseudomonads useful systems for evaluating predictive models of genotype–environment interactions.

Conventional computational annotation commonly relies on sequence similarity to characterized homologs.[9] Profile hidden Markov models extend this approach by capturing position-specific patterns of conservation and variation within protein families.[10] Although these methods are valuable for functional annotation, functional similarity alone does not determine the fitness effect of gene disruption under a particular environmental condition. Many poorly characterized proteins, sometimes described as "protein dark matter," also have little detectable similarity to experimentally characterized proteins.[11] Genome-scale metabolic models (GSMMs) provide a mechanistic approach by representing gene–reaction relationships and using constraint-based optimization to simulate metabolic flux and growth.[12,13] GSMMs can simulate the effects of single-gene disruptions, but their predictions are restricted to genes and processes represented in the metabolic network.[14]

Protein language models (PLMs) for protein sequences are promising candidates to address the "dark matter problem".[15] PLMs encode protein sequences as dense representations that capture statistical patterns associated with structural and functional properties. These representations can be combined with fitness measurements, experimental metadata, and metabolic annotations to predict condition-specific phenotypes.[16] Knowledge graphs (KGs) provide a structured framework for integrating such heterogeneous information by representing biological entities and their relationships under an explicit schema.[17,18] Graph neural networks (GNNs) can then aggregate information across relationships such as orthology, pathway membership, gene–reaction associations, and shared experimental context.[19–21] Message passing could therefore allow a predictive model to use relational information integrated from resources such as KEGG,[22] MetaCyc,[23] and Swiss-Prot.[24] Whether this graph structure improves fitness prediction beyond the underlying sequence, text, and metadata features, however, remains an empirical question.

Here, we introduce GIMME-KG (**G**raph **I**nference for **M**icrobial Dark **M**atter **E**xploration with **K**nowledge **G**raphs), a heterogeneous biological knowledge graph integrating RB-TnSeq

fitness measurements with gene, experiment, metabolic, text-derived, and protein-sequence information. We train GFM1 (GIMME-KG Fitness Model 1), a heterogeneous graph neural network for predicting condition-specific gene fitness. We compare graph-based models with a multimodal multilayer perceptron (MLP) that uses the same node-encoder design and decoder template but does not perform message passing. Models are evaluated under three held-out settings: experiments, genes, and family-style gene clustering, to test generalization to unobserved gene–experiment combinations. We examine how the GAT–MLP performance difference varies with network connectivity, annotation coverage, gene-family conservation, conditional essentiality, and Enzyme Commission class, and compare GFM1 with a GSMM in a matched *P. putida* benchmark. Message passing improves prediction for held-out experiments and genes, but its advantage is not significant during the cluster-aware split. Together, these analyses demonstrate biological knowledge-graph structure provides predictive information beyond multi-modal node features and establishes a framework for graph-based microbial fitness prediction.

2. Related Work

Prior machine-learning approaches to microbial gene function and fitness have primarily used either protein-sequence representations or experimental phenotype data. PLMs such as ESM-3 encode amino acid sequences as dense representations, and recent work has combined PLM embeddings with GNNs to predict gene essentiality.[25,26] Sequence-only models, however, do not explicitly represent the environmental conditions under which fitness is measured. Conversely, phenotype-based methods capture relationships within observed assay data. Borchert et al., for example, identified 84 co-fitness modules from a 179-condition *P. putida* fitness matrix using independent component analysis,[27] but such matrix-based approaches do not directly generalize to genes, strains, or conditions absent from the observed data.

Knowledge graphs can integrate sequence-derived, experimental, and functional information. Heterogeneous GNNs have been applied to biomedical knowledge graphs for drug–target, drug–drug interaction, and essentiality prediction tasks;[28–31] FuseLinker additionally combines frozen language-model text embeddings with GNN-based link prediction.[32] Our framework instead targets condition-specific microbial gene fitness by integrating text-derived annotations, protein-sequence representations, experimental context, and heterogeneous graph structure.

3. Methodology

3.1. *Datasets*

We obtained 7,937,507 gene–experiment fitness measurements spanning 55,421 genes, 1,770 experiments, and 10 *Pseudomonas* strains from the Fitness Browser including growth media composition and experimental metadata such as pH and temperature. Gene-level fitness scores are calculated from changes in relative mutant-barcode abundance and reported on a $\log_2$ scale.[5] We define a conditional fitness defect using the "extreme fitness value" of ≤ -2.[27]

The Fitness Browser gene set is integrated with four external reference databases when membership is known: SEED,[33] MetaCyc,[23] KEGG,[22] and UniProt annotations, including Enzyme Commission (EC) assignments. We integrated the full EC hierarchy using the EC

RDF,[34] and added MetaCyc reactions and compounds. The released KG contains 141,947 nodes across 19 node types, 884,487 structural edges across 27 relationship types, and 7,937,507 `FITNESS` supervision edges. Model node types and features are documented in Appendix Tables 2-3, together with the architecture sweep and feature-ablation results (Appendix C and D). Figure 1 shows the schema of the released KG, including the `FITNESS` supervision edges. The split-specific training subgraph is derived from this inventory for each evaluation protocol and excludes `FITNESS` edges from structural message passing. No text or annotations embedded in the training data mention essentiality or fitness outcomes.

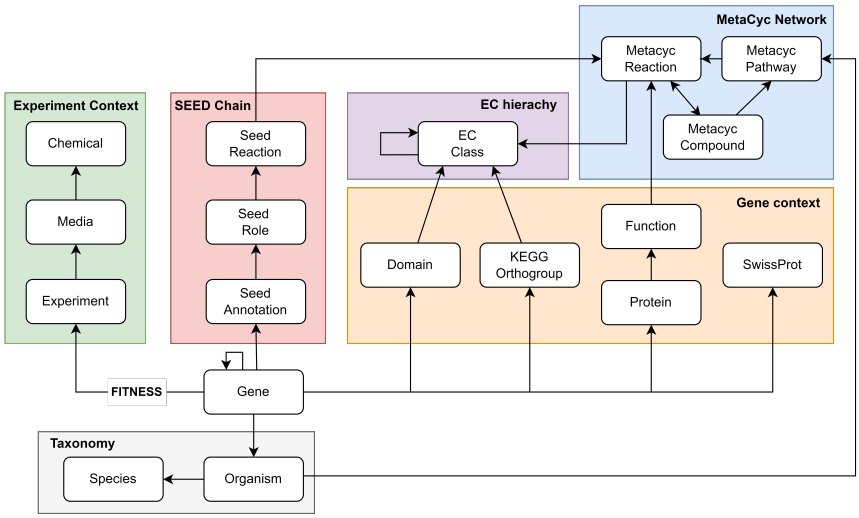


Fig. 1. Schema of the released GIMME-KG knowledge graph, including `FITNESS` supervision edges connecting gene and experiment subnetworks. Split-specific training subgraphs apply additional relation filtering before bidirectional structural message passing.

3.2. *Data Processing*

To reduce the influence of extreme fitness values, we fit a quantile transformer on the training fold that maps the target distribution to an approximate standard normal using 10,000 quantile bins. Transformed targets were used for model optimization and validation-based early stopping. All predictions were inverse-transformed before computing the reported metrics using the original $\log_2$ fitness scale. We evaluated three split protocols representing generalization to held-out experiments, genes, and gene clusters. Split counts below are based on assignment seed 42 and are available in Appendix section A.

- **Experiments:** We assigned the 1,770 experiments and their associated fitness cells to one partition.
- **Genes:** We assigned the 55,421 genes and their associated fitness cells to one fold.
- **Clusters:** To reduce homology across partitions, we assigned each gene to exactly one cluster using the following priority order: KEGG orthogroup, Swiss-Prot group, connected component of `ORTHOLOGOUS` edges, or singleton, emulating CAFA gene clustering.[22,35]

3.3. *Multi-Modal Node Encoder*

As shown in Figure 2, we encode each node within the heterogeneous graph by projecting every available feature type through its own branch, then fusing the results to a common hidden vector of width H. Each branch b produces a raw vector $\mathbf{x}_v^{(b)}$ for node v, and we define $\mathcal{A}_k$ as the set of branches active on node type k. The four supported branch types are:

- **Text** ($\mathbb{R}^{d_T}$). A frozen text representation embedded with `gemini-embedding-001`[36] into a fixed 3072-dimensional vector.
- **Numeric** ($\mathbb{R}^{2n_k}$). For a node type k with n_k scalar properties, the branch takes the vector of z-normalized values paired with n_k binary flags indicating if each value is present.
- **Categorical** ($\mathbb{R}^{\sum_p e_p}$). Each categorical property p has a learned embedding table $\mathbf{E}_p$ of width $e_p = \max\big(2, \lceil\sqrt{|c_p|}\rceil\big)$, where $|c_p|$ is the property's cardinality, and are concatenated to form the branch input.
- **Vector** ($\mathbb{R}^{d_V}$). Pre-trained ESM-3[25] embeddings using sequence-length hidden states extracted from the last transformer layer and compressed into a single, 1536-d vector via mean-pooling across the amino acid dimension to return a fixed length.

The output vectors are combined into a final node representation in two stages, both using the same reusable module $\mathcal{B}$ consisting of a layer-normalization operator LN, a learned linear layer with weights $\mathbf{W}$ and bias $\mathbf{b}$, a nonlinearity $\phi = \mathrm{GELU}$, and an optional dropout layer D_p with probability p during training. For numeric and categorical input branches this layer-normalization operation is disabled, as their low dimensionality makes normalization degenerate, with it remaining active for text and ESM-3 vector branches and in the post-concatenation fusion block.

$$\mathcal{B}(\mathbf{x}) = D_p\big(\,\phi\big(\,\mathbf{W}\,\mathrm{LN}(\mathbf{x}) + \mathbf{b}\,\big)\,\big). \tag{1}$$

Each branch is first projected by its own instance of the module, $\mathcal{B}_b$, differing in both input and output width based on branch type. The branch outputs are then concatenated and passed through $\mathcal{B}_k$ (per node type as $\mathcal{A}_k$ varies across nodes), mapping the concatenation to a hidden vector of width H:

$$\mathbf{h}_{k,v}^{(0)} = \mathcal{B}_k\Big(\,\big[\,\mathcal{B}_b(\mathbf{x}_v^{(b)})\,\big]_{b\in\mathcal{A}_k}\,\Big) \in \mathbb{R}^H. \tag{2}$$

3.4. *Heterogeneous Message Passing*

Stacking the encoded representations across nodes gives $\mathbf{H}_k^{(0)} \in \mathbb{R}^{N_k\times H}$ per node type k. As shown in Figure 2, we propagate information across the split-specific training subgraph using L typed heterogeneous convolution layers. Structural relations are represented in both directions during message passing. The `FITNESS` relation is retained as the supervision signal but excluded from message passing. For the cluster-aware split, cross-partition `ORTHOLOGOUS` and `BEST_HIT` relations are also removed. Our final convolutional layers use the multi-head GATv2 attention[37] backbone. The exploratory screen also evaluated GraphSAGE as an alternative message-passing backbone, with the complete comparison provided in Appendix Table 4.

3 METHODOLOGY

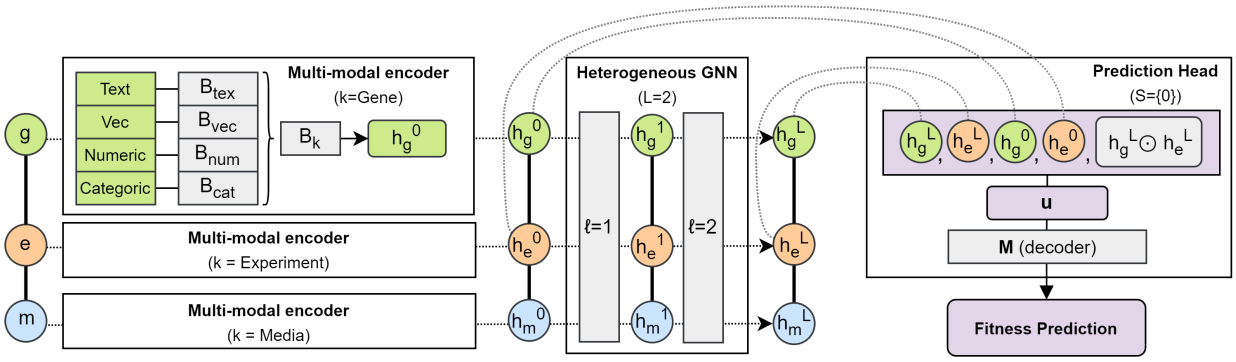


Fig. 2. Architecture schematic, detailing the Multi-modal encoders, variable depth GNN, and prediction head. Note that within the prediction head, the concatenation (cat) fusion method is displayed, and that the larger KG is represented by a simple three node graph containing a single gene, experiment and media node.

3.5. *Prediction Head*

For an input gene-experiment pair (g, e), let $\mathbf{h}_g^{(\ell)}, \mathbf{h}_e^{(\ell)}$ denote the gene and experiment representations at depth ℓ, with $\ell = 0$ being the encoder output (Equation (2)) and $\ell = L$ the final message-passing output. Let $\mathcal{S} \subseteq \{0, 1, \ldots, L-1\}$ be an optional subset of retained intermediate depths, $\mathcal{D} = \{L\} \cup \mathcal{S}$ be the full set of retained depths, and for $t \in \{g, e\}$ the stack of depth-ℓ be $\mathbf{H}_t$, corresponding with Equation (3).

$$\mathbf{H}_t = \left[\mathbf{h}_t^{(\ell)} \right]_{\ell \in \mathcal{D}} \in \mathbb{R}^{|\mathcal{D}| \times H} \tag{3}$$

The final models use two prediction-head fusion variants:

- *Concatenation (**cat**).* $\mathbf{H}_g$ and $\mathbf{H}_e$ are flattened, giving a head input whose width grows linearly with $|\mathcal{S}|$.

$$\mathbf{u} = \left[\mathrm{vec}(\mathbf{H}_g) \,;\, \mathrm{vec}(\mathbf{H}_e) \,;\, \mathbf{h}_g^{(L)} \odot \mathbf{h}_e^{(L)} \right] \tag{4}$$

- *Cross-attention (**xa**).* Each side's final representation $\mathbf{h}_t^{(L)}$ queries the *other* side's $\mathbf{H}_{t'}$ through an independent multi-head attention block.[38] Cross attention is weighted independently across depths. Final representations, cross-attention contexts, and the Hadamard interaction are then concatenated to produce a fixed width vector of size $5H$.

The final GAT models use cross-attention (**xa**), while the matched MLP uses concatenation (**cat**). The exploratory screen also evaluated skip-attention (**sa**), GraphSAGE backbones, and alternative depths, with these variants defined and compared in Appendices B and C.

Finally, the head input $\mathbf{u}$ is passed through a decoder $\mathcal{M} : \mathbb{R}^{|\mathbf{u}|} \to \mathbb{R}$, a two layer MLP with input LayerNorm, GELU nonlinearities, and dropout, producing a scalar prediction $\hat{y} = \mathcal{M}(\mathbf{u})$ on the quantile-transformed target scale. We note that the MLP-only baseline is recovered as the $L = 0$ special case of the **cat** fusion method. It uses the same multi-modal node-encoder design and decoder template as the graph models but does not perform message passing. A complete description of the architecture is provided in Appendix section B.

3.6. *Training and Evaluation*

All models use hidden width 256, dropout 0.2, and minibatches of 16,384 gene–experiment fitness pairs. We optimize the mean squared error (MSE) on the quantile-transformed targets using the Adam optimizer with peak learning rate 10^{-3}, weight decay 10^{-4}, and a 5% linear warm-up. The learning rate is reduced by a factor of 0.3 after eight validation epochs without improvement, to a minimum of 10^{-5}. Training wall times, graph/cache footprint, and inference benchmarks for the selected GAT and matched MLP models are reported in Appendix section F. An architecture screen selected GAT/$L = 2$/xa for the experiments and genes splits and GAT/$L = 6$/xa for the cluster-aware split. The full architecture sweep is provided in Appendix C. To separate model selection from confirmation, these screening runs were excluded from primary inference, with fourteen additional GAT and MLP runs then trained for each split. Model checkpoints and analysis artifacts are provided in the supplementary data release. For feature ablation assays, described in Appendix section D, we retrained the selected genes-split GAT after disabling one feature branch while leaving the corresponding nodes and graph edges intact. Performance comparisons in Table 2, Figure 3, Appendix Figure 2, and Table 3 use the best model for each split based on the lowest validation MSE on the transformed target scale. These comparisons evaluate individual selected runs, with the bootstrap intervals in the structural analysis measuring variation across test genes. For the performance over coverage analysis we compared per-organism GAT/MLP MSE across the three evaluation splits using annotation coverage, defined as the fraction of organism genes linked to at least one of SEED, MetaCyc, KEGG, or UniProt in the original KG. 95% confidence intervals (CI) for MSE differences were estimated using 1,000 paired bootstrap resamples of genes or gene families.

3.7. *GSMM performance comparison*

The GSMM iJN1463 was downloaded from the BiGG database `http://bigg.ucsd.edu/`. FBA was computed across all genes and carbon source experiments and we define essential where growth_ratio to control 0.25 or gene knockout growth approaches 0. The condition and detailed comparison description is provided within Appendix section E, along with an inference runtime analysis (Appendix section F, table 17).

3.8. *Graph Structure Analysis*

Using test predictions, we evaluated when GAT outperformed MLP by comparing test-set MSE across genes grouped by graph connectivity and family prevalence. We adjusted comparisons for differences in fitness distributions, assessed whether access to measured neighbors improved performance, and examined the relationship between neighbor and test-gene fitness. Statistical uncertainty and significance were estimated using paired bootstrapping and multiple-testing correction. Full methods are provided in Appendix section E.

Table 1. Mean and standard deviation of test performance for the GAT and MLP baseline.

Split	Model	n	MSE↓	MAE↓	ρ↑
Experiments	MLP	14	0.1361 ± 0.0016	0.2127 ± 0.0006	0.7457 ± 0.0029
	GAT/$L = 2$/xa	14	$\mathbf{0.1289 \pm 0.0016}$	$\mathbf{0.2100 \pm 0.0007}$	$\mathbf{0.7657 \pm 0.0024}$
Genes	MLP	14	0.2410 ± 0.0054	0.2551 ± 0.0015	0.5324 ± 0.0113
	GAT/$L = 2$/xa	14	$\mathbf{0.2166 \pm 0.0102}$	$\mathbf{0.2484 \pm 0.0027}$	$\mathbf{0.5960 \pm 0.0189}$
Clusters	MLP	14	0.2965 ± 0.0092	0.2719 ± 0.0017	0.4237 ± 0.0250
	GAT/$L = 6$/xa	14	$\mathbf{0.2931 \pm 0.0090}$	$\mathbf{0.2704 \pm 0.0016}$	$\mathbf{0.4263 \pm 0.0300}$

Table 2. Comparison of best MLP and GAT performance across data splits in the high essentiality stratum. "Ess." refers to *conditionally essential* fitness pairs per split.

Split	Model	Truly ess.	Pred. ess.	TP	Recall	Precision	$\mathbf{F}_1$
Experiments	GAT	10,468	6,794	**5,919**	**0.565**	**0.871**	**0.686**
	MLP	10,468	**6,951**	5,815	0.556	0.837	0.668
Genes	GAT	17,742	5,687	5,194	0.293	**0.913**	0.443
	MLP	17,742	**6,575**	**5,661**	**0.319**	0.861	**0.466**
Clusters	GAT	13,748	2,347	**2,007**	**0.146**	**0.855**	**0.249**
	MLP	13,748	**2,524**	1,990	0.145	0.788	0.245

4. Results

4.1. *Model Predictive Performance*

For interpretability, relative MSE reduction is reported as $(\overline{\text{MSE}}_{\text{MLP}} - \overline{\text{MSE}}_{\text{GAT}})/\overline{\text{MSE}}_{\text{MLP}}$. The addition of message passing through the GAT layer improved the observed mean on all three metrics for held-out experiments and genes (Table 1). Relative to the MLP, mean MSE decreased by 5.3% for experiments and 10.1% for genes, corresponding to absolute reductions of 0.0072 (p=0.000366, Wilcoxon signed-rank test, Benjamini-Hochberg (BH) correction) and 0.0244 (p=0.000366, Wilcoxon signed-rank test, BH correction), respectively. By contrast, the difference in cluster-split MSE was not significant, and GAT had lower MSE in only 7 of 14 pairs. While increasing GAT layer depth to 6 during the more restrictive cluster-aware split enabled deeper exploration of the knowledge graph's integrated metabolic networks, there was no significant improvement over the MLP model.

We conduct an exploratory analysis to determine which node features support the increase in performance observed under gene-split fitness prediction for graph-observed nodes. Each run was compared with the full GAT using the same seed. The experiment-text ablation resulted in the largest observed change in performance (+0.0106 ΔMSE, paired-t $p = 0.00627$, BH-adjusted $p = 0.10659$); feature coverage, including field-level coverage for numeric inputs, and the complete ablation sweep are reported in Appendix D.

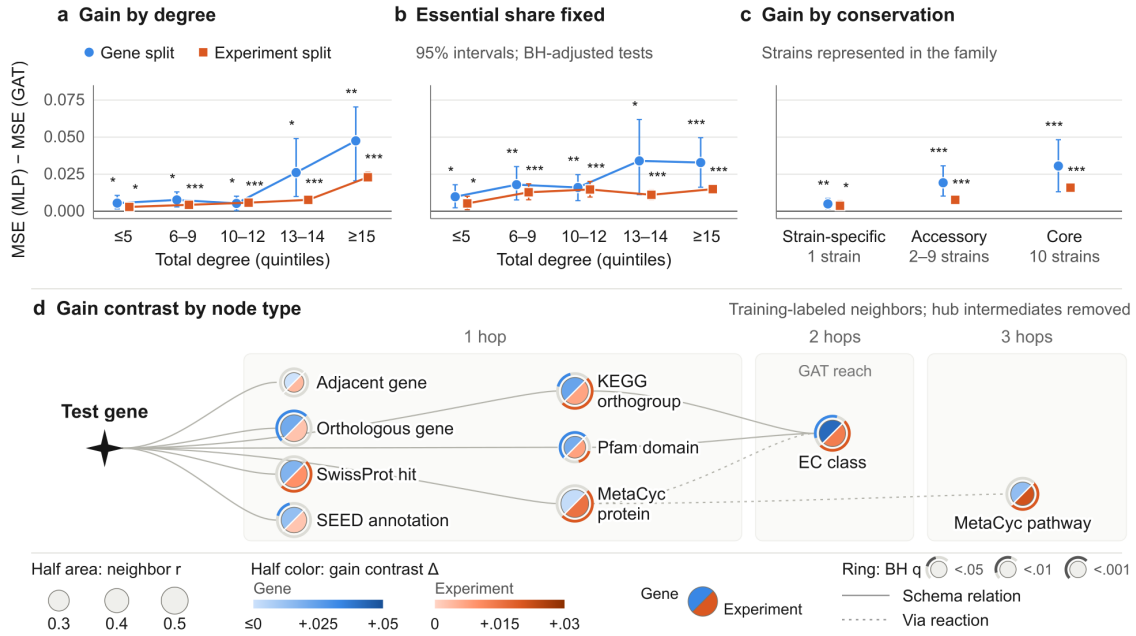


Fig. 3. GAT performance gains in relation to knowledge-graph structure. Gains over MLP are shown across test-gene connectivity before and after adjusting for fitness composition (a–b), and by gene-family conservation (c). Panel d relates gain to the availability and fitness similarity of training-labeled neighbors across graph node types and path lengths. Nodes diagonally display gene and experiment splits, shade denotes delta gain contrast and area neighbor r. Ring length around each half encodes the contrast BH-adjusted q at specified levels; an empty grey track indicates q $\leq$ 0.05. All panels use 0.05. 0.01, and 0.001 as confidence levels and 95% paired bootstrap intervals.

4.2. *Graph Structure, Conservation, and Strong-Defect Prediction*

Graph structure is not homogeneous at the gene level, with different levels of gene annotation, experimental validation, and integration into biological and metabolic knowledge bases. 39.7% of gene nodes belong to families represented in all ten sampled strains, 45.7% to families represented in two to nine strains, and 14.6% to strain-specific families (Appendix Table 10). Core genes are more connected, with mean degree 15.0 versus 3.8 for strain-specific genes. GAT's MSE gain increased from the lowest to highest degree quintile in both the experiment split (0.0029 to 0.0229) and the gene split (0.0055 to 0.0475; Figure 3a). Conditionally essential pairs were enriched in high-degree bins and holding their share fixed flattened the experiment-split curve above the lowest quintile, whereas a larger gain persisted in the upper gene-split quintiles (Figure 3b). Gains also increased from strain-specific to core families, from 0.0037 to 0.0159 in experiments and from 0.0049 to 0.0305 in genes (Figure 3c). In the gene split, labeled-neighbor coverage through orthology, KEGG, Pfam, SEED, and EC node types were associated with larger mean per-gene gains (BH q<0.05; Figure 3d). Regression gains did not translate into a uniform improvement in other performance metrics (Table 2). In the experiment split, GAT recovered 104 more true-essential pairs than MLP, with higher precision and recall over MLP (0.871 and 0.565 versus 0.837 and 0.556, respectively). In the gene split, GAT traded higher precision (0.913 versus 0.861) for lower recall (0.293 versus 0.319) and F1 (0.443 versus

Table 3. Comparison of flux balance analysis (FBA) and GFM1 predictions against experimental RB-TnSeq measurements for cold-start genes in *Pseudomonas putida* KT2440.

Model	AUROC	AUPRC	F_1	Precision	Recall	Accuracy	Pearson
FBA (iJN1463)	0.733	0.294	0.455	0.392	0.544	0.932	0.435
GFM1 + xa	**0.983**	**0.874**	**0.712**	**0.919**	**0.581**	**0.976**	**0.823**

0.466). In the cluster split, GAT precision was higher (0.855 versus 0.788), while recall and F1 differed only slightly.

Test genes with level-1 Enzyme Commission (EC) class (1–7) exhibited MAE gains varied across split protocols. GAT achieved lower MAE in all seven classes under experiment holdout, 5/7 under gene holdout, and 6/7 under cluster holdout. Isomerases and ligases favoured MLP under gene holdout, while lyases favored MLP under cluster holdout. GAT exhibited strong performance for the conditional essentiality set as well, with lower MAE in 7/7, 4/7, and 5/7 categories (Appendix Figure 2). Across the observed species gene annotation-coverage range (80.8–90.4%), GAT achieved strong performance in experiment and gene splits: the corresponding paired 95% CI for the MSE difference excluded zero in favor of GAT for 8/10, 6/10 organisms, while no organism had CI clearing zero for the cluster split.

4.3. *GFM1 outperforms a GSMM when predicting fitness growth defects*

GFM1 substantially outperforms iJN1463 derived fitness predictions across both classification and continuous prediction metrics (Table 3). Its average precision was 0.874 versus 0.294, approximately 3.0 times the FBA value and 16.8 times the 0.05195 positive prevalence. At the specified thresholds, GFM1 produced 7 false positives versus 115 for FBA, a reduction of 108 (93.9%), while recalling 79 of 136 strong-defect pairs (58.1%) versus 74 (54.4%). Pearson correlation with experimental fitness was 0.823 for GFM1 and 0.435 for FBA.

5. Discussion

5.1. *Fitness prediction under cluster-aware generalization remains challenging*

Our results show that microbial gene-knockout effects can be predicted in two settings: generalization from known genes to held-out experiments and cold-start prediction for held-out genes. Performance declined under the gene-cluster split as similar genes were withheld during training. In the experiment and gene splits, pretrained embeddings and biological context allow the model to transfer information across related conditions and genes. Much of this information is unavailable or restricted in the cluster split, reducing message passing performance gain over the non-graph baseline.

Class imbalance may contribute to this result: most gene–experiment pairs have little or no measured fitness effect, so a model can minimize average error by predicting values near the center of the distribution. Conditionally essential pairs comprised only 1.71% of the cluster test set but had a highly uncertain MSE gap, 0.483 (95% family-bootstrap interval -1.178, 2.044).

The gap on other pairs was smaller and positive, 0.0050 (0.0005, 0.0113), indicating that rare, large defects contribute disproportionately to uncertainty in the aggregate comparison.

5.2. *Relational structure improves prediction of conditionally essential phenotypes*

Message passing improves precision over matched non-graph baselines, whereas recall and F1 improve only in the experiment and cluster splits as listed in Table 2. GAT is numerically higher in seven of the nine split-by-metric comparisons, and Figure 3 links larger regression gains to connectivity, conservation, and selected annotation nodes. Fixing the conditionally essential share shows that test composition explains part of the connectivity association. These findings support the addition of message passing by adding more precise strong fitness defect calls and context-dependent regression benefits. They also demonstrate that large-scale RB-TnSeq measurements can serve as training data for predicting phenotypes in unmeasured gene–condition combinations.

Under the tested architectures and optimization settings, two-layer GATs performed best in the experiment and gene splits, so it is possible that local graph context is more useful than additional rounds of message passing. However, encoder depth alone does not establish that deeper resources, such as the EC hierarchy or MetaCyc compound network, are uninformative: their contributions may also depend on graph connectivity, feature quality, sampling, and optimization. The experiment-text ablation had the largest observed MSE increase, but its reported effect did not survive BH correction (adjusted p=0.10659). Weak or uncertain effects of individual ablations could reflect redundancy among node features, compensation through graph topology, or limited statistical sensitivity. Still, GFM1 showed strong MSE performance across the range of species gene-annotation coverage, indicating that coverage as low as 80% does not result in diminished performance. Future analysis of novel or poor coverage gene sets is needed to fully enumerate GAT performance in that regime.

5.3. *GFM1 complements mechanistic genome-scale metabolic models*

GFM1 showed stronger agreement with experimental RB-TnSeq measurements than flux balance analysis with iJN1463 for the evaluated *Pseudomonas putida* KT2440 benchmark. In particular, GFM1 achieved higher classification performance and stronger correlation with continuous fitness measurements while producing fewer false-positive growth-defect predictions (Table 3). These results support the value of learned models for predicting bacterial gene fitness effect.

The approaches nevertheless provide different forms of information. Genome-scale metabolic models impose stoichiometric and mass-balance constraints and generate reaction- and flux-level hypotheses, whereas GFM1 produces empirical fitness predictions without mechanistic explanation. Their strengths are therefore complementary rather than directly interchangeable. GFM1 could prioritize gene–condition pairs with predicted fitness defects, after which a genome-scale metabolic model could be used to investigate candidate metabolic mechanisms.

5.4. *Limitations and future work*

Several limitations should be considered. First, the training data are taxonomically restricted to ten *Pseudomonas* strains. An organism-level split would more closely represent prediction for a previously unmeasured species or strain and would test whether gene and experiment representations transfer across phylogenetic lineages. Expanding training to the broader Fitness Browser collection could improve the learning of conserved sequence-to-function relationships, although differences in experimental design, annotation quality, and phylogenetic representation would require careful control.

Structural graph metrics such as degree, annotation coverage, and conservation are correlated, and their associations with prediction gain do not identify causal edge contributions. The non-significant cluster-split comparison and the restricted organism/condition scope of the FBA benchmark limit evaluation of how the models generalize beyond tested conditions.

Finally, alternative architectures may also improve information flow through the heterogeneous graph. Because the GAT and MLP also use different prediction heads (**xa** vs **cat**), their performance difference reflects the complete model configurations and cannot be attributed solely to message passing. For example, knowledge-source-specific encoders or mixture-of-experts modules could process individual subnetworks before combining their representations through a learned gate. Protein structures, protein–protein interactions, transcriptomics, and proteomics could provide additional and potentially complementary signals. Future work should evaluate these modalities alongside alternative heterogeneous-graph encoders and more stringent phylogenetic generalization protocols.

6. Conclusion

We developed GFM1, a heterogeneous graph neural network for predicting condition-specific microbial fitness following gene inactivation. Using the GIMMME-KG microbial metabolic knowledge graph, GFM1 improved mean regression performance over the MLP in experiment and gene holdouts, while the cluster-split mean MSE advantage was not significant. Larger regression gains were associated with graph connectivity and conservation, with conditionally essential enrichment explaining part of the connectivity pattern. At a fixed strong-defect threshold, GFM1 improved precision in all splits but traded recall for precision in held-out genes. Generalization to withheld gene clusters and recovery of rare, large fitness defects remain important challenges.

GFM1 also showed stronger agreement with experimental RB-TnSeq measurements than flux balance analysis with iJN1463 on the evaluated *Pseudomonas putida* KT2440 benchmark. These results demonstrate that protein and text embeddings can be integrated with biological knowledge graphs to learn condition-dependent genotype–phenotype relationships. Future work should expand the taxonomic and experimental diversity of the training data, improve transfer to novel gene families and explore architecture changes, and evaluate the resulting representations in experimental-design and functional-annotation workflows. Together, these results highlight GFM1's potential to prioritize informative gene–condition combinations for experimental validation and accelerate microbial functional characterization.

Funding Statement

This work was supported in part by the U.S. Department of Energy (DOE), Office of Biological and Environmental Research (BER), as part of BER's Genomic Science Program (GSP), and is a contribution of the Pacific Northwest National Laboratory (PNNL) Secure Biosystems Design Science Focus Area "Persistence Control of Engineered Functions in Complex Soil Microbiomes." It is also part of the Predictive Phenomics Initiative and conducted under the Laboratory Directed Research and Development Program. Pacific Northwest National Laboratory is a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under Contract No. DE-AC05-76RL01830.

Competing Interests

No competing interest is declared.

References

1. A. Wagner, Genotype sampling for deep-learning assisted experimental mapping of a combinatorially complete fitness landscape, *Bioinformatics* **40**, p. btae317 (2024).
2. S. Valavarasu, Y. Sangu and T. Mahapatra, Prediction of antibiotic resistance from antibiotic susceptibility testing results from surveillance data using machine learning, *Scientific Reports* **15**, p. 30509 (2025).
3. K. M. Wetmore, M. N. Price, R. J. Waters, J. S. Lamson, J. He, C. A. Hoover, M. J. Blow, J. Bristow, G. Butland, A. P. Arkin *et al.*, Rapid quantification of mutant fitness in diverse bacteria by sequencing randomly bar-coded transposons, *MBio* **6**, 10 (2015).
4. A. J. Borchert, A. Bleem and G. T. Beckham, Experimental and analytical approaches for improving the resolution of randomly barcoded transposon insertion sequencing (rb-tnseq) studies, *ACS Synthetic Biology* **11**, 2015 (2022), PMID: 35657709.
5. M. N. Price, K. M. Wetmore, R. J. Waters, M. Callaghan, J. Ray, H. Liu, J. V. Kuehl, R. A. Melnyk, J. S. Lamson, Y. Suh *et al.*, Mutant phenotypes for thousands of bacterial genes of unknown function, *Nature* **557**, 503 (2018).
6. M. Rajkumar, L. B. Bruno and J. R. Banu, Alleviation of environmental stress in plants: The role of beneficial pseudomonas spp., *Critical Reviews in Environmental Science and Technology* **47**, 372 (2017).
7. M. G. Wilson and S. Pandey, Pseudomonas aeruginosa, in *StatPearls [Internet]*, (StatPearls Publishing, 2023)
8. H. Asif, D. J. Studholme, A. Khan, M. Aurongzeb, I. A. Khan and M. K. Azim, Comparative genomics of an endophytic pseudomonas putida isolated from mango orchard, *Genetics and Molecular Biology* **39**, 465 (2016).
9. T. Joshi and D. Xu, Quantitative assessment of relationship between sequence similarity and function similarity, *BMC genomics* **8**, p. 222 (2007).
10. Y. Ma, H. Chen, J. Kang, X. Guo, C. Sun, J. Xu, J. Tao, S. Wei, Y. Dong, H. Tian *et al.*, The hidden markov model and its applications in bioinformatics analysis, *Genes & Diseases* **13** (2026).
11. G. A. Pavlopoulos, F. A. Baltoumas, S. Liu, O. Selvitopi, A. P. Camargo, S. Nayfach, A. Azad, S. Roux, L. Call, N. N. Ivanova *et al.*, Unraveling the functional dark matter through global metagenomics, *Nature* **622**, 594 (2023).
12. J. Nogales, J. Mueller, S. Gudmundsson, F. J. Canalejo, E. Duque, J. Monk, A. M. Feist, J. L. Ramos, W. Niu and B. O. Palsson, High-quality genome-scale metabolic modelling of pseu-

domonas putida highlights its broad metabolic capabilities, *Environmental microbiology* **22**, 255 (2020).
13. A. Antonakoudis, R. Barbosa, P. Kotidis and C. Kontoravdi, The era of big data: Genome-scale modelling meets machine learning, *Computational and structural biotechnology journal* **18**, 3287 (2020).
14. A. M. Abdel-Haleem, H. Hefzi, K. Mineta, X. Gao, T. Gojobori, B. O. Palsson, N. E. Lewis and N. Jamshidi, Functional interrogation of plasmodium genus metabolism identifies species-and stage-specific differences in nutrient essentiality and drug targeting, *PLoS computational biology* **14**, p. e1005895 (2018).
15. Z. Harteveld, A. Van Hall-Beauvais, I. Morozova, J. Southern, C. Goverde, S. Georgeon, S. Rosset, M. Defferrard, A. Loukas, P. Vandergheynst *et al.*, Exploring "dark matter" protein folds using deep learning, *bioRxiv* , 2023 (2023).
16. J. Zhong, L. Li, R. Dannenfelser and V. Yao, Benchmarking gene embeddings from sequence, expression, network, and text models for functional prediction tasks, *bioRxiv* , 2025 (2025).
17. S. Ji, S. Pan, E. Cambria, P. Marttinen and P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, *IEEE Transactions on Neural Networks and Learning Systems* **33**, 494 (2022).
18. C. Peng, F. Xia, M. Naseriparsa and F. Osborne, Knowledge graphs: Opportunities and challenges, *Artificial intelligence review* **56**, 13071 (2023).
19. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov and M. Welling, Modeling relational data with graph convolutional networks (2017).
20. X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. Yu and Y. Ye, Heterogeneous graph attention network (2021).
21. Z. Ye, Y. J. Kumar, G. O. Sing, F. Song and J. Wang, A comprehensive survey of graph neural networks for knowledge graphs, *IEEE Access* **10**, 75729 (2022).
22. M. Kanehisa, The kegg database, in *'In silico'simulation of biological processes: Novartis Foundation Symposium 247*, 2002.
23. R. Caspi, R. Billington, I. M. Keseler, A. Kothari, M. Krummenacker, P. E. Midford, W. K. Ong, S. Paley, P. Subhraveti and P. D. Karp, The metacyc database of metabolic pathways and enzymes-a 2019 update, *Nucleic acids research* **48**, D445 (2020).
24. A. Bairoch and R. Apweiler, The swiss-prot protein sequence database and its supplement trembl in 2000, *Nucleic acids research* **28**, 45 (2000).
25. T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert *et al.*, Simulating 500 million years of evolution with a language model, *Science* **387**, 850 (2025).
26. J. Ma, J. Song, N. D. Young, B. C. Chang, P. K. Korhonen, T. L. Campos, H. Liu and R. B. Gasser, 'bingo'—a large language model-and graph neural network-based workflow for the prediction of essential genes from protein data, *Briefings in Bioinformatics* **25**, p. bbad472 (2024).
27. A. J. Borchert, A. C. Bleem, H. G. Lim, K. Rychel, K. D. Dooley, Z. A. Kellermyer, T. L. Hodges, B. O. Palsson and G. T. Beckham, Machine learning analysis of rb-tnseq fitness data predicts functional gene modules in pseudomonas putida kt2440, *Msystems* **9**, e00942 (2024).
28. M. Li, X. Cai, S. Xu and H. Ji, Metapath-aggregated heterogeneous graph neural network for drug-target interaction prediction, *Briefings in Bioinformatics* **24**, p. bbac578 (January 2023).
29. Z. Li, X. Tu, Y. Chen and W. Lin, HetDDI: A pre-trained heterogeneous graph neural network model for drug-drug interaction prediction, *Briefings in Bioinformatics* **24**, p. bbad385 (September 2023).
30. Q. Ye, Y. Zeng, L. Jiang, Y. Kang, P. Pan, J. Chen, Y. Deng, H. Zhao, S. He, T. Hou and C.-Y. Hsieh, A knowledge-guided graph learning approach bridging phenotype- and target-based drug discovery, *Advanced Science* **12**, p. e2412402 (April 2025).

31. H. Wen, S. Carpenter, K. McGinnis, A. Nelson, K. Smith and T. Hong, Esssubgraph improves performance and generalizability of mammalian essential gene prediction with large networks, *GigaScience* **14**, p. giaf136 (01 2025).
32. Y. Xiao *et al.*, FuseLinker: leveraging LLM-generated text and domain knowledge for enhanced link prediction on biomedical knowledge graphs, *Journal of Biomedical Informatics* **158**, p. 104735 (2024).
33. R. Overbeek, R. Olson, G. D. Pusch, G. J. Olsen, J. J. Davis, T. Disz, R. A. Edwards, S. Gerdes, B. Parrello, M. Shukla, V. Vonstein, A. R. Wattam, F. Xia and R. Stevens, The SEED and the rapid annotation of microbial genomes using subsystems technology (RAST), *Nucleic Acids Research* **42**, D206 (2014).
34. A. G. McDonald and K. F. Tipton, Enzyme nomenclature and classification: the state of the art, *The FEBS Journal* **290**, 2214 (2023).
35. P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, A large-scale evaluation of computational protein function prediction, *Nature methods* **10**, 221 (2013).
36. Google, Gemini api: Embeddings `https://ai.google.dev/gemini-api/docs/embeddings`, Accessed: 2026-01-19.
37. S. Brody, U. Alon and E. Yahav, How attentive are graph attention networks?, in *International Conference on Learning Representations*, 2022.
38. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in *Advances in neural information processing systems*, 2017.