Graph Neural Network based Hierarchy-Aware Embeddings of Knowledge Graphs: Applications to Yeast Phenotype Prediction[†]

Filip Kronström FILIPKRO@CHALMERS.SE
Alexander H. Gower
Daniel Brunnsåker
Department of Computer Science and Engineering,
Chalmers University of Technology and University of Gothenburg, Sweden

Ievgeniia A. Tiukova
Department of Life Sciences, Chalmers University of Technology, Sweden
Department of Industrial Biotechnology, KTH Royal Institute of Technology, Sweden

Ross D. King
Department of Computer Science and Engineering,
Chalmers University of Technology and University of Gothenburg, Sweden
Department of Chemical Engineering and Biotechnology, University of Cambridge, United Kingdom

Abstract

We present a method for finding hierarchy-aware embeddings of knowledge graphs (KGs) using graph neural networks (GNNs) enriched with a semantic loss derived from underlying ontologies. This method yields embeddings that better reflect domain knowledge. To demonstrate their utility, we predict and interpret the effects of gene deletions in the yeast *Saccharomyces cerevisiae* and learn box embeddings for KGs in the absence of a prediction task. We further show how box embeddings can serve as the basis for evaluating KG revisions.

Our yeast KG is constructed from community databases and ontology terms. Low-dimensional box embeddings combined with GNNs are used to predict cell growth for double gene knockouts. Over 10-fold cross validation, these predictions have a mean R^2 score of 0.360, significantly higher than baseline comparisons, demonstrating that high-level qualitative knowledge is informative about experimental outcomes. Incorporating semantic loss terms in the training of the models improves their predictive performance (R^2=0.377) by aligning embeddings with ontology structure. This shows that class hierarchies from ontologies can be exploited for quantitative prediction. We also test the trained models on triple gene knockouts, showing they generalise to data beyond those seen in training.

Additionally, by identifying co-occurring relations in the yeast KG important for the cell-growth predictions, we construct hypotheses about interacting traits in yeast. A biological experiment validates one such finding, revealing an association between inositol utilisation and osmotic stress resistance, highlighting the model's potential to guide biological discovery.

[†] This is an extended version of the paper "Ontology-based box embeddings and knowledge graphs for predicting phenotypic traits in *Saccharomyces cerevisiae*" (Kronström et al., 2025), presented at NeSy 2025.

1. Introduction

Knowledge graphs (KGs) are widely used to represent structured knowledge as sets of triples of the form (*subject, predicate, object*). In many domains, including the life sciences, KGs are enriched with ontological information, where formally defined vocabularies describe classes of entities and relations between them. In particular, hierarchical[1] class information expressed through the '`subClassOf`' relation is prevalent, as domain experts can organise concepts, without requiring formal expertise in logic or ontology engineering. As a result, such hierarchies constitute a pragmatic and high-impact source of background knowledge for representation learning.

Representation learning enables the transformation of KGs into a form that works as an input to a program class that does not accept the KG itself as an input. A form of representation learning that has proven useful for downstream tasks, such as link prediction and property prediction, is embedding KGs into an n-dimensional vector space. Many KG embedding (KGE) methods are trained primarily on observed triples, using the relational structure of the graph as the main learning signal. However, when ontological knowledge is present, an embedding should ideally reflect not only the connectivity of the graph but also background semantic constraints. Rewarding compatibility between learned representations and known domain structure can provide additional inductive bias during training, improve generalisation beyond observed data, and support more interpretable predictions (Gutiérrez-Basulto and Schockaert, 2018).

Incorporating semantic constraints into continuous representations raises the question of how ontological concepts should be represented geometrically. Hierarchical relations impose inclusion constraints, which can be captured using point embeddings in non-Euclidean spaces, such as hyperbolic embeddings (Nickel and Kiela, 2017). However, when the goal is to explicitly encode class inclusion and concept-level constraints, volumetric representations provide a natural alternative. In this paradigm, concepts are modelled as subsets of a latent space using geometric objects, such as hyperspheres (Kulmanov et al., 2019), hypercones (Ganea et al., 2018), or hyperrectangles (boxes) (Vilnis et al., 2018; Peng et al., 2022; Xiong et al., 2022; Jackermeier et al., 2024; Yang et al., 2025), enabling hierarchical relations to be expressed directly through geometric containment.

While volumetric representations are well suited for capturing ontological structure, it is challenging to integrate them with complex heterogeneous graphs, and to train them end-to-end for task-specific prediction. Many real-world KGs are large, heterogeneous, and feature-rich. This motivates the use of graph neural networks (GNNs) as a framework for representation learning through aggregation of information from local neighbourhoods and node attributes (Kipf and Welling, 2017; Hamilton et al., 2017). GNN-based approaches have been successfully applied to KGs and relational data for tasks such as node classification, link prediction, and property prediction (Schlichtkrull et al., 2018; Ye et al., 2022).

However, standard GNN architectures do not explicitly enforce global semantic constraints derived from ontologies, and may therefore learn representations that violate known hierarchical relationships, even when such information is available as background knowledge. This motivates approaches that combine data-driven representation learning with

[1] Section 5.1 provides a detailed definition of what is meant by the terms *hierarchy* and *hierarchical* in this work.

explicit semantic constraints derived from ontologies. We also hypothesise that, particularly in settings with low-dimensional or scarce data, that the inductive bias introduced by using ontology constraints during representation learning will improve performance on downstream prediction tasks.

In this paper, we address this limitation by combining GNN-based KG embeddings with box-based representations of ontological concepts by introducing Hierarchy-aware GNNs. Our methodological contribution is a framework in which box embeddings are used to encode hierarchical background knowledge, while GNNs are used to learn features from the graph structure.

We evaluate this approach in the context of biological knowledge about the yeast *Saccharomyces cerevisiae*. Yeast is among the most extensively studied model organisms and plays a central role in both basic research and industrial applications (Parapouli et al., 2020). Decades of experimental work have produced large amounts of structured data, available through curated databases such as the Saccharomyces Genome Database (Engel et al., 2024). These resources integrate experimental observations with ontological annotations describing biological processes, phenotypes, chemical compounds, and interactions.

Despite the depth of scientific research on yeast, our understanding of its biology remains incomplete: many genes are poorly annotated, and current models fail to predict many phenotypic outcomes of complex genetic interactions (Wood et al., 2019; Costanzo et al., 2019). Experimental investigation is therefore essential for advancing biological knowledge. However, biological experimentation is costly, and the space of possible experiments is vast. As a result, computational methods that support hypothesis generation at scale are of high value (King et al., 2004; Brunnsåker et al., 2025).

We construct an ontology-enriched KG describing yeast biology and apply our embedding framework to predict gene fitness and to generate hypotheses about interactive properties. This application serves both as a realistic use case and as an empirical evaluation of whether incorporating hierarchical semantic constraints into KG embeddings improves predictive performance and supports biologically meaningful knowledge discovery.

The remainder of the paper is structured as follows. Section 2 presents related work relevant for this paper. Sections 3 and 4 introduce background, and define terms and losses used throughout the paper. In Section 5 we present Hierarchy-aware GNNs, a *Saccharomyces cerevisiae* KG, and applications of the Hierarchy-aware GNNs on this KG. These results are presented in Section 6, and discussion and future work can be found in Section 7.

2. Related work

Knowledge graph embedding methods aim to represent entities and relations in a continuous vector space such that observed facts are preserved geometrically. In their most simple formulation, KGs are treated as collections of triples of the form (*subject, predicate, object*) or (*head, relation, tail*), and embeddings are learned by optimising a scoring function over such triples.

One approach is to model relations using bilinear interactions between entity embeddings. RESCAL (Nickel et al., 2011) represents each relation as a full matrix, enabling expressive modelling of many-to-many relations but incurring high computational cost. DistMult (Yang et al., 2015) simplifies this formulation by restricting relation matrices to be diagonal, yielding a model that performs well for symmetric relations but cannot capture asymmetry. ComplEx (Trouillon et al., 2016) extends DistMult to a complex-valued space, allowing asymmetric relations to be modelled while retaining computational efficiency.

Another prominent family of approaches models relations as transformations applied to entity embeddings. TransE (Bordes et al., 2013) represents relations as translations between entity vectors such that $s+p \approx o$, offering simplicity and interpretability, but struggling with complex relations. RotatE (Sun et al., 2019) generalises this idea by interpreting relations as rotations in the complex plane, enabling the modelling of symmetry, antisymmetry, inversion, and composition while achieving strong empirical performance.

Incorporating hierarchical information. Despite differences in formulation and expressivity, the models mentioned so far share a common focus on instance-level relational structure derived from observed triples. While they can capture relational patterns and implicit structural regularities, ontological constraints such as class subsumption or disjointness are typically not enforced explicitly. This has motivated extensions of KGE methods that seek to incorporate hierarchical information while preserving the efficiency and scalability of triple-based learning.

One such approach is HAKE (Zhang et al., 2020), which augments standard KG embeddings with hierarchy awareness. It introduces a polar coordinate representation in which hierarchical depth and lateral similarity are modelled separately, enabling improved link prediction in graphs with pronounced hierarchical structure. Conceptually, HAKE shares similarities with hyperbolic and Poincaré embedding approaches in how it captures hierarchical structure, despite operating in a Euclidean space. While these approaches capture hierarchical patterns through geometry, they do not explicitly represent ontological concepts or logical axioms.

A semantics-driven line of work focuses on embedding ontologies formulated in description logics by constructing geometric representations that approximate model-theoretic interpretations. ELEm (Kulmanov et al., 2019) introduces a volumetric embedding approach in which classes and nominals are represented as hyperspheres, and relations are modelled using TransE-style translations. Subsumption and existential restrictions are enforced via geometric containment, demonstrating that $\mathcal{EL}^{++}$ axioms can be satisfied approximately in a continuous space.

ELBE (Peng et al., 2022) builds directly on earlier volumetric ontology embeddings by replacing spherical concept regions with axis-aligned hyperrectangles (boxes), ensuring closure under intersection. This geometric choice enables a more faithful encoding of $\mathcal{EL}^{++}$ axioms while retaining a volumetric interpretation of concepts. Subsequent work further extends box-based ontology embeddings within $\mathcal{EL}^{++}$: BoxEL (Xiong et al., 2022) represents concepts as boxes and roles as affine transformations, providing soundness guarantees with respect to $\mathcal{EL}^{++}$-semantics; Box^{2}EL (Jackermeier et al., 2024) represents both concepts and roles as boxes to support many-to-many relations; and TransBox (Yang et al., 2025) provides an $\mathcal{EL}^{++}$-closed construction that enables compositional modelling of complex concepts and many-to-many roles through box-based sets of translations.

To address issues of noise and uncertainty, probabalistic approaches to KG embeddings have also been developed, where embedding parameters are modelled as random variables. Vilnis et al. (2018) introduces probabilistic box representations to model uncertainty and inclusion relationships, while Dasgupta et al. (2020) addresses optimisation and identifiability issues arising in such models.

Graph neural network approaches. While the approaches above focus on how entities, relations, or concepts are represented geometrically, they typically learn embeddings either in isolation or directly from triples or axioms. In contrast, many real-world knowledge graphs are large and heterogeneous, motivating methods that explicitly aggregate information from local graph structure. Graph neural networks (GNNs) provide such a framework by learning node representations through iterative message passing over graph neighbourhoods, naturally supporting end-to-end optimisation for specific downstream prediction tasks.

Relational Graph Convolutional Networks (R-GCNs) (Schlichtkrull et al., 2018) were developed specifically with knowledge graphs in mind and have been applied to tasks such as link prediction and node classification. R-GCNs extend standard GCNs to multi-relational settings by introducing relation-specific message transformations, an idea that can be combined with a wide range of message-passing architectures.

More generally, the integration of symbolic knowledge into neural models has been explored through semantic loss functions, which penalise predictions that violate logical constraints during training (Xu et al., 2018). While originally proposed for feed-forward neural networks, this idea provides a general mechanism for incorporating prior knowledge into end-to-end learning.

Box embeddings have also been combined with GNNs in specific application settings. BoxGNN (Lin et al., 2024) integrates box-based representations into a GNN architecture for recommendation tasks by redefining message passing and aggregation operations directly in box space, using geometric operations such as intersection and union. In this setting, boxes serve as the primary representation for users, items, and attributes. This approach is tailored to recommendation and does not aim to enforce semantic constraints in knowledge graph embeddings.

Knowledge graph embeddings in the natural sciences. KGs have been widely adopted in the biomedical domain as a means of integrating heterogeneous experimental and curated data into a unified representation. For example BioKG (Walsh et al., 2020) and SPOKE (Morris et al., 2023) combine information from different databases to create one large heterogeneous graph with information about, for example, genes and drugs. There are also graphs describing more narrow phenomena such as the protein-protein associations and the drug-drug interactions in the Open Graph Benchmark (Hu et al., 2020).

Memariani et al. (2025) propose a box-embedding framework for extending the ChEBI chemical ontology from molecular data. In their model, chemical classes are represented as boxes, and molecules, encoded from SMILES strings using an ELECTRA-based encoder (Clark et al., 2019), are represented as points. Molecules are trained to lie inside the boxes of their annotated classes (including all superclasses), so that box containment and overlap reflect subsumption and disjointness relations between classes. This enables evaluation not only of molecular classification, but also of how well the learned box geometry recovers the class taxonomy.

Predicting biological properties from structured background knowledge can be approached in several ways. Ma et al. (2018) encode GO-annotations together with the GO hierarchy in a neural network to predict cellular growth in *S. cerevisiae*. By mining patterns from a Datalog knowledge base containing facts from databases, Brunnsåker et al. (2024) connect qualitative biological concepts to quantitative intracellular protein abundance measurements. Gualdi et al. (2024) predict gene-disease associations using embeddings derived from a protein interaction knowledge graph, evaluating a range of methods including translational KG embeddings (e.g. TransE and RotatE), random-walk-based approaches, and GNN-based models. In their approach, embeddings are learned independently of the prediction task and subsequently used as features for supervised classifiers such as support vector machines and tree-based models, resulting in a two-stage pipeline rather than end-to-end training.

3. Background

Graph neural networks

Graph neural networks (GNNs) are neural network models that take graphs as inputs, and learn vector representations of nodes. In a GNN, the structure of the graph informs the architecture, with information being passed from the neighbourhood of each node. At each message passing layer of the GNN, information is aggregated from a node's neighbourhood to update its representation.

Each message passing layer therefore propagates information from the immediate neighbourhood of a node, so by increasing the number of message passing layers we increase the distance in the graph that information propagates—this is referred to as the receptive field of the GNN. Message passing layers are often interspersed with pooling layers to increase the receptive field. An example of a message aggregation scheme is a convolutional layer, the basic component of a graph convolutional network (GCN). In a GCN layer, updates to node embeddings at each layer are calculated by multiplying the embeddings from the previous layer by a scaled adjacency matrix (including self-adjacency) and a weight matrix (Kipf and Welling, 2017).

While a very powerful tool, particularly when learning representations for homogeneous graphs, GCNs do not use information about edge type when learning on heterogeneous graphs, which most KGs are. R-GCNs apply the same ideas as GCNs, but instead allow for separate weight matrices depending on edge type, and therefore are much better suited to representation learning on heterogeneous KGs than GCNs (Schlichtkrull et al., 2018).

GraphSAGE

GraphSAGE (Hamilton et al., 2017) is an alternative GNN framework for learning node embeddings that, instead of directly learning embeddings for nodes in a static graph, learns a function that can generate embeddings. As a result, GraphSAGE is able to generate embeddings for new nodes that were not in the graph the model was trained on, provided these new nodes share the same attribute schema as the original graph. GraphSAGE uses an aggregator to collect information from the neighbourhood of a given node; concatenates this aggregated information with the current embedding state of the node; and then passes this through a weighted function to propagate the information. The aggregator can be directly encoded, for example a simple mean, or it could be learned during training. The receptive field of the network can be increased by adding more layers.

Description logics

Description logics (DLs) are fragments of first-order logic that allow only for constants (individuals), unary predicates (concepts), and binary predicates (roles). The symbol $\sqsubseteq$ is used to represent concept inclusion, and $\equiv$ is used for equivalence. We use DL to express axioms to describe a domain, resulting in a knowledge graph. These axioms are generally split into terminological axioms (TBox)—regarding general knowledge about concepts and roles in the domain—and assertional axioms (ABox)—which make statements about individuals (Baader et al.). For example, we can represent TBox axioms for the statements: (1) "proteins are molecules"; (2) "carbohydrates are molecules"; (3) "carbohydrates are disjoint from proteins"; and (4) "enzymes are molecules that catalyze a biochemical reaction" in DL:

$$\begin{aligned}
\texttt{Protein} &\sqsubseteq \texttt{Molecule} && (1)\\
\texttt{Carbohydrate} &\sqsubseteq \texttt{Molecule} && (2)\\
\texttt{Carbohydrate} \sqcap \texttt{Protein} &\sqsubseteq \bot && (3)\\
\texttt{Enzymes} &\equiv \texttt{Molecule} \sqcap \exists \texttt{Catalyzes.Reaction} && (4)
\end{aligned}$$

Two examples of ABox axioms are:

$$\begin{aligned}
&\texttt{Protein(hexokinase)}.\\
&\texttt{Catalyzes(hexokinase,glucose\_phosphorylation)}.
\end{aligned}$$

From the TBox and ABox, we could deduce that hexokinase is an enzyme, because it catalyzes the phosphorylation of glucose.

Individuals, concepts, and roles in DLs can be represented in Web Ontology Language (OWL) using individual, class, and property statements.

$\mathcal{EL}$++ is a lightweight DL designed for efficient reasoning over large ontologies, allowing conjunctions, existential restrictions, and role hierarchies while ensuring polynomial-time reasoning (Baader et al.). It underpins the OWL 2 EL profile and is widely used in biomedical ontologies, including the ones introduced below.

Ontologies and data

Decades of research on *Saccharomyces cerevisiae* have resulted in extensive knowledge that is available both in the scientific literature and in curated biological databases. The Saccharomyces Genome Database (SGD) (Engel et al., 2024) provides a central resource aggregating curated information about *S. cerevisiae* genes, including Gene Ontology annotations, experimentally observed phenotypes, and regulatory as well as genetic interaction data. Complementary information about biochemical reactions, events in which substrates are transformed into products, and pathways, sets of interconnected reactions driving cellular functions, is available in pathway-oriented resources such as BioCyc (Karp et al., 2019).

The contents of these databases are represented using ontologies and controlled vocabularies to ensure consistency and interoperability. The Gene Ontology (GO) (Ashburner et al., 2000) defines classes describing molecular functions, biological processes, and cellular components. Phenotypic information in SGD is formalised using the Ascomycete Phenotype Ontology (APO) (Costanzo et al., 2009), which captures observable characteristics arising from interactions between genotype and environment, such as growth defects or resistance to chemical perturbations. Chemical compounds are described using Chemical Entities of Biological Interest (ChEBI) (Hastings et al., 2016). Genetic, physical, and regulatory interactions are modelled using the Interaction Network Ontology (INO) (Hur et al., 2015) and Molecular Interactions (MI) (Hermjakob et al., 2004). Commonly used relations shared across ontologies are defined in the Relations Ontology (RO) (Mungall et al., 2020). Among these resources, only GO and ChEBI define relations between classes, whereas the remaining ontologies provide taxonomies of domain-specific entities.

4. Preliminaries

Box embeddings

Axis-aligned hyperrectangles, or "boxes" as they often are referred to, are defined as the Cartesian product of closed intervals,

$$\text{Box} = \prod_{i=1}^{n} [z_i, Z_i], \tag{5}$$

where z_i and Z_i correspond to the lower and upper coordinate along dimension i. To fulfil the criteria that the upper coordinate should be greater than or equal to the lower coordinate, $Z_i \geq z_i$, we create boxes from latent variables, θ, using the `MinDeltaBoxTensors` constructor introduced by Chheda et al. (2021). The upper and lower box coordinates are defined as follows:

$$z_i(\theta_i) = \theta_i^z, \qquad Z_i(\theta_i) = z_i + \text{softplus}(\theta_i^Z). \tag{6}$$

The equation for transforming latent variables, θ into box representations is therefore:

$$\text{Box}(\theta) = \prod_{i=1}^{n} [z_i(\theta_i), Z_i(\theta_i)] \tag{7}$$

Boxes can also be represented by their centre-point, c_i, and offset, o_i, along dimension i, found from z and Z as follows:

$$c_i = \frac{z_i + Z_i}{2}, \qquad o_i = Z_i - c_i \tag{8}$$

A knowledge graph (KG) can be represented using box embeddings by embedding classes and individuals (nodes) as axis-aligned boxes in a low-dimensional space. As discussed in Section 2, this idea has been explored in several forms, with its popularity largely driven by a favourable trade-off between expressivity and computational efficiency. Central to box embeddings is the modelling of transitive relations through geometric containment, where the box of a head entity is constrained to lie within the box of a tail entity. In this work, we use box embeddings to represent '`subClassOf`' relationships between classes, rewarding containment of each subclass box within its corresponding superclass box.

Semantic losses

To learn box embeddings we consider two different types of loss functions for concept inclusions of the form $C \sqsubseteq D$. The first loss, here called $\mathcal{L}_{distance}$, has previously been used by, for example, Peng et al. (2022) and Jackermeier et al. (2024). Using the nomenclature presented above it is calculated by first finding the element-wise distance between the two boxes,

$$d(C_i, D_i) = |c_i^C - c_i^D| - o_i^C - o_i^D \tag{9}$$

In the loss function, this quantifies how far a subclass box is from being completely contained within the superclass box,

$$\mathcal{L}_{distance}(C, D) = \Big|\Big|\Big(\max(0, d(C_i, D_i) + 2o_i^C)\Big)_{i=1}^{n}\Big|\Big| \tag{10}$$

Note here that Jackermeier et al. (2024) used the loss in (10) in the case $D \neq \emptyset$, and a separate loss if $D = \emptyset$. In this work, we don't consider that it is reasonable to assume that any of the concepts included in the knowledge graph is empty. Mostly, this is because the concepts in the knowledge graphs used here come directly from scientific ontologies, where their inclusion implicitly represents a belief of their existence, but also relies on an open world assumption. It could be that the loss as defined in Jackermeier et al. (2024) is necessary for complex or compound concepts in other applications, where the empty set is a possibility for concept D. We also omit the margin parameter from Jackermeier et al. (2024) as we want the loss to be zero when the subclass axiom is satisfied in the embedding space. To keep disjoint classes, $C \sqcap D \sqsubseteq \bot$, apart we penalise overlap of the boxes by the following loss (where $\mathbb{I}$ is the indicator function):

$$\mathcal{L}^{-}_{distance}(C, D) = \Big|\Big|\Big(\max(0, -d(C_i, D_i))\Big)_{i=1}^{n}\Big|\Big| \cdot \prod_{i=1}^{n} \mathbb{I}\big[d(C_i, D_i) < 0\big] \tag{11}$$

The second loss type considers the overlap between boxes and for the subsumption $C \sqsubseteq D$ it is calculated as

$$\mathcal{L}_{overlap} = -\log\left(\frac{\mathsf{Vol}(\mathrm{Box}(C) \cap \mathrm{Box}(D))}{\mathsf{Vol}(\mathrm{Box}(C))}\right) \tag{12}$$

For disjoint classes we instead use the following loss:

$$\mathcal{L}^{-}_{overlap} = -\log\left(1 - \frac{\mathsf{Vol}(\mathrm{Box}(C) \cap \mathrm{Box}(D))}{\min(\mathsf{Vol}(\mathrm{Box}(C)), \mathsf{Vol}(\mathrm{Box}(D)))}\right) \tag{13}$$

To avoid large flat regions in the loss landscape, for example when two boxes are completely disjoint, Dasgupta et al. (2020) proposed that boxes and intersections of boxes are interpreted as Gumbel random variables. They showed that the volume of such boxes and intersections are determined by Bessel functions which can be reasonably approximated by softplus functions. In practice, this produces smooth intersections between boxes and ensures non-zero gradients, also in cases such as disjoint boxes.

Throughout this work we use $\mathcal{L}_{\sqsubseteq}$ and $\mathcal{L}^{-}_{\sqsubseteq}$ as placeholders for either of the inclusion and disjointness losses introduced above. To learn box embeddings the positive and negative losses are simply added together, possibly weighted differently. We present ways of doing this in more detail in Sections 5.3 and 5.5.

Regularisation losses

Patel et al. (2020) proposes to regularise the volume of boxes when training box embeddings, we use the implementation by Chheda et al. (2021) which does this by applying the L2-norm to all sides of the box,

$$R = \sum_{i}^{n} \|Z_i - z_i\|^2 \tag{14}$$

We also found that regularising excessively small boxes can be beneficial in certain cases. This was implemented as

$$R_{\text{small}} = \sum_{i}^{n} \max\left(0, \frac{1}{\|Z_i - z_i\|} - l_0\right), \tag{15}$$

with l_0 being a threshold determining below what size the box is penalised.

5. Material and methods

In this section, we begin by introducing the hierarchy-aware GNN framework used throughout this work. We then describe the construction of the *S. cerevisiae* KG and detail how the hierarchy-aware GNN is applied to predict gene deletion fitness using this graph. Next, we present a KG embedding approach based on the same hierarchy-aware GNN architecture, trained without an explicit prediction task, to study its representational properties. Finally, we introduce a link evaluation method based on the resulting embeddings, which is used to assess the impact of adding or modifying links in the KG.

5.1. Hierarchy-aware GNN

The losses presented in Section 4 have primarily been used for training shallow embeddings of class hierarchies, but they can also be applied to GNNs whose outputs parameterise box embeddings. In this work, we represent KGs as TBox-style axioms by encoding class assertions, $C(a)$, as subsumption axioms, $\{a\} \sqsubseteq C$, and role assertions, $r(a, b)$, as existential restrictions, $\{a\} \sqsubseteq \exists r.\{b\}$. The resulting graph consists only of class- and nominal-level axioms of the form

$$A \sqsubseteq B \tag{16}$$

$$A \sqcap B \sqsubseteq \bot \tag{17}$$

$$A \sqsubseteq \exists r.B. \tag{18}$$

We refer to axioms of the form (16) and (17) as the class hierarchy. The existential restrictions in (18) define all edges of the graph, meaning that relations between individuals and relations between classes are represented in a uniform way within the graph.

This TBox-style graph is then used as the computational graph for the GNN: nodes correspond to classes and nominals, while directed edges correspond to axioms of the form in (18). Message passing therefore propagates information along logical relations in the KG, while the hierarchy axioms will provide geometric constraints on the embeddings.

Each layer, $l \in \{1, \dots, L\}$, of a GNN learns weights w_l that parameterise a function, $B_l : \mathbb{R}^{n_{l-1}} \longrightarrow \mathbb{R}^{n_l}$. Treating the output of each layer in a GNN, $\theta_l = B_l(\theta_{l-1}; w_l)$, as the latent representation of a class or nominal, box embeddings are generated via the transformation in (7) and optimised using the loss functions in (10-13). An illustration of this architecture is shown in Figure 1.

For heterogeneous knowledge graphs spanning multiple domains whose class hierarchies can be embedded independently, we can use separate embedding spaces and hierarchy losses for each domain.

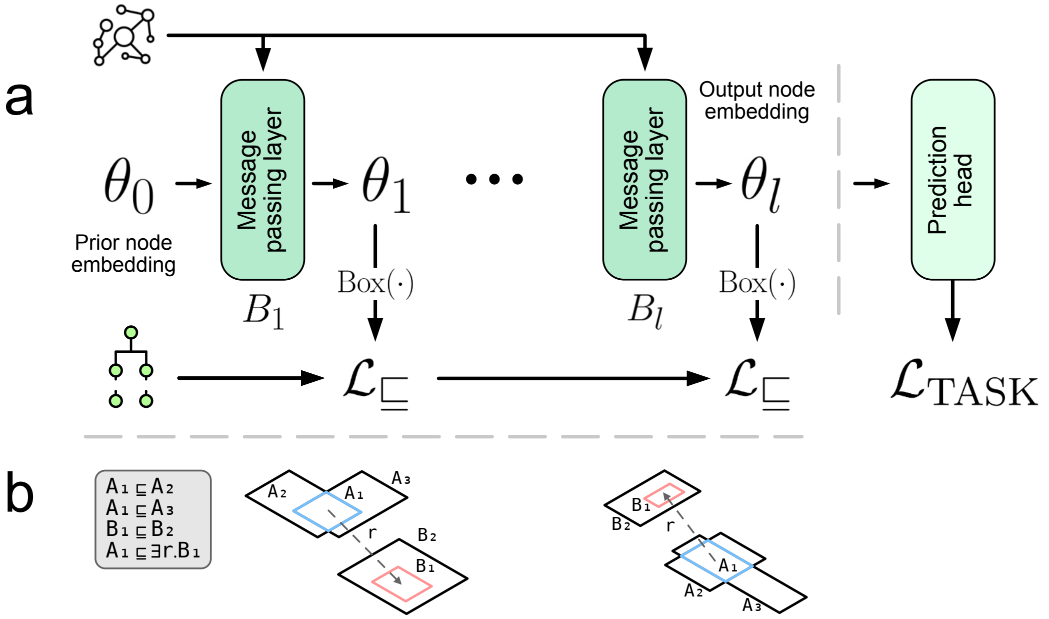


Figure 1: Overview of the Hierarchy-aware GNN. **a** shows how the output of each message passing layer, which aggregates information between neighbours in the KG, is treated as a latent variable that is converted into boxes through the box transformation in (6). The boxes are trained to fulfil specified class hierarchies using the losses in (10-13), which can also be applied to the prior node embedding. The output of the GNN can be fed to some prediction head, trained by jointly minimising a task-specific loss. **b** illustrates how a box embedding is transformed throughout the GNN. Note that the relation, r, is drawn as an arrow in this embedding, but in the model it is represented as a message passing edge.

The taxonomy loss for a multilayer, possibly multidomain, GNN is calculated as the sum of the individual losses,

$$\mathcal{L} = \sum_{l=0}^{L} \sum_{d=0}^{D} \sum_{(A,B)\in\mathcal{S}_d} \mathcal{L}_{\sqsubseteq}(\text{Box}(\theta_l^A), \text{Box}(\theta_l^B)), \tag{19}$$

with $\mathcal{S}_d$ being the set of subclass axioms for domain d:

$$\mathcal{S}_d = \{(A, B) : A \sqsubseteq B \in H^{(d)}\} \tag{20}$$

and where l specifies the layers of the GNN and d the node domains, and $H^{(d)}$ is the class hierarchy from the ontologies describing domain d. $\mathcal{L}_{\sqsubseteq}$ corresponds to the loss in (10) or (12). Likewise, a negative loss can be computed to prevent the embeddings from collapsing into identical boxes for all classes. The loss is obtained either from disjointness information in the taxonomy or from randomly sampled negative examples, following the formulations in (11) or (13).

The approach is flexible in the sense that it is architecture agnostic and can be used either on its own for box embeddings of KGs using GNNs, or as a semantic loss together with another loss term, for example, when training prediction models. The rationale behind this approach is that class hierarchies often contain information that is not necessarily modelled in the graph edges, and can be especially useful to improve the representation of poorly connected nodes in the graph.

Training prediction models end-to-end with semantic losses, rather than first finding a separate semantic embedding of the ontology or KG, allows for extraction of the edges important for the task at hand while still adhering to the class taxonomy. The flexibility of neural networks enables this to be used for various prediction tasks, including edge, node, and graph level predictions. Depending on the prediction head it can be used for both classification and regression.

The loss function for such a model would simply combine the task specific and semantic loss, possibly along with regularisation of the parameters as

$$\mathcal{L} = \mathcal{L}_{\text{TASK}}(y, \hat{y}) + \alpha(\mathcal{L}_{\sqsubseteq} + \beta\mathcal{L}_{\sqsubseteq}^{-}) + \lambda \left\| w \right\|^2 , \tag{21}$$

where α weights the semantic loss, β determines the impact of negative examples in the semantic loss, and λ controls the regularisation of the network parameters.

In Section 5.3, we illustrate this approach on an edge-weight prediction (link regression) task, where the GNN output is fed into an neural network (NN) prediction head. The same architecture naturally extends to other edge-level tasks, as well as node- and graph-level tasks; the only difference lies in how the appropriate representations are selected and supplied to the NN.

To represent boxes and implement box-related operations, such as intersection and volume calculations, we use the `box-embeddings` Python package (v0.1.0) (Chheda et al., 2021).

5.2. Knowledge graph

We have created a heterogeneous knowledge graph describing genes in the yeast *Saccharomyces cerevisiae* by combining facts expressed in classes and relations from multiple ontologies. We represent the graph using TBox axioms, as discussed in Section 5.1, by rewriting class assertions, $C(a)$, as $\{a\} \sqsubseteq C$ and role assertions, $r(a, b)$, as $\{a\} \sqsubseteq \exists r.\{b\}$. In this way, we get the same representation of asserted facts from databases as we have for terminological statements from ontologies, like GO or ChEBI. This simplifies the interface between KG, box embeddings, and GNNs.

The knowledge graph is created from data in SGD, where the information is defined using terms from several different ontologies. A high level overview of the graph, showing how different node types are connected, can be seen in Figure 2a. Figure 2b shows examples of the hierarchies classes instantiating these nodes are represented in. Characteristics of these hierarchies are shown in Table 1.

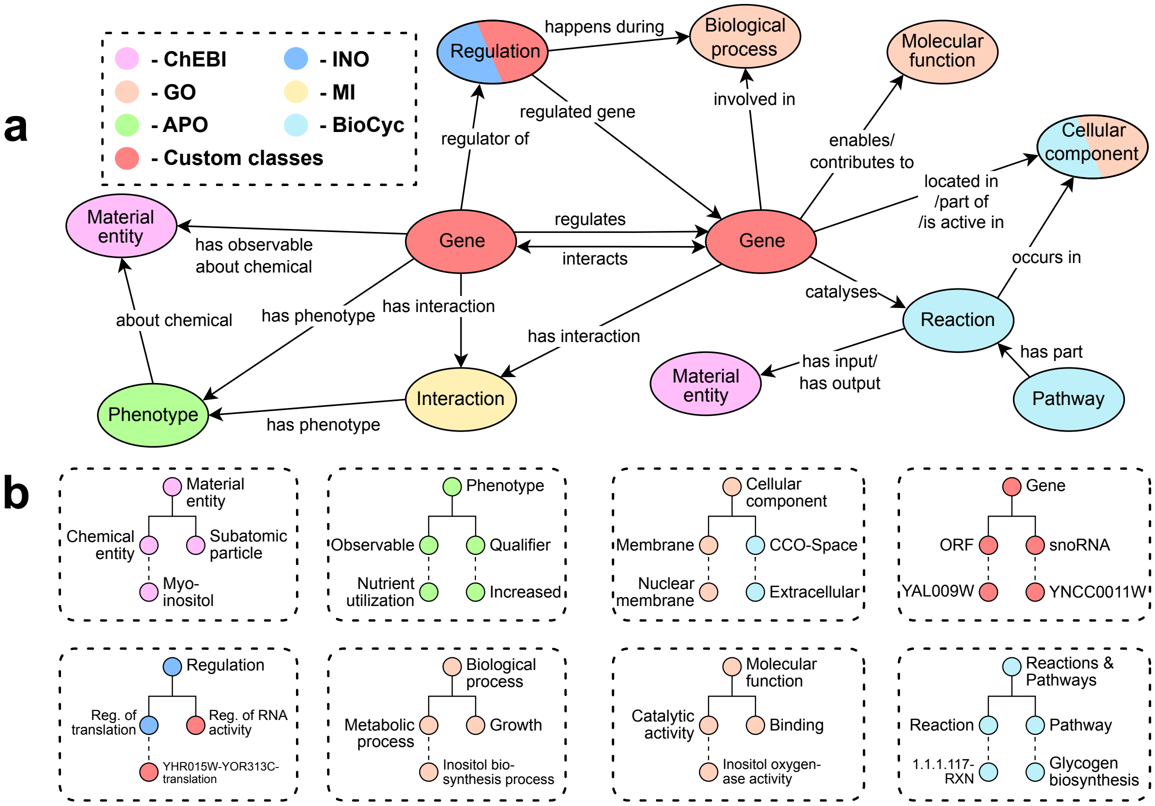


Figure 2: An overview of the different types of classes and how they are connected in the knowledge graph is shown in **a**. The colour of the nodes specifies where the classes are defined. **b** shows examples from the hierarchies defining classes in the domains introduced in Section 5.3.

The GO-annotations in SGD are naturally described by classes in the Gene Ontology and relations from the OBO Relations Ontology, which are specified in the database. Phenotypes are described using terms from APO where a phenotype is represented by an '`observable`', for example '`heat sensitivity`', and possibly a '`qualifier`', for example '`increased`'. We represent the phenotype as the subclass of the intersection of these two types of classes, and phenotypes are linked to genes using the RO relation '`has phenotype`'. Some phenotypes describe observables related to specific chemicals, in such cases the chemical class in ChEBI is linked with a custom relation, '`aboutChemical`'. To form a closer connection between genes and chemicals related to phenotypes, which proved useful for downstream tasks (see Section 5.3), a link specific to the type of observable was added between the gene and the chemical. An example of how this is implemented in description logic can be seen in (26) in Appendix A.

Gene regulation in SGD is a directed relationship between two genes that can be positive, negative, or unspecified, and of different types, for example, regulation of protein activity or expression. In some instances, a biological process from GO specifies under which condi-

Table 1: Characteristics for the hierarchies representing the different domains in the KG. The "Origin" column indicates in which ontology or taxonomy the hierarchy is defined, ★ indicates we have defined it, or parts of it.

Domain	Number of classes	Maximum depth	Number of leaf nodes	Median leaf node depth	Origin
Material entities	217,381	24	200,368	10	ChEBI
Biological processes	28,337	16	14,949	6	GO
Phenotypes	3,949	9	3,731	4	APO
Molecular functions	11,201	12	9,181	5	GO
Regulation	20,563	7	20,536	5	INO, ★
Reactions	3,101	8	2,734	4	BioCyc
Cellular components	4,184	11	3,225	4	GO
Genes	7,383	7	7,322	3	★

tions the regulation occurs. We introduce custom relations describing regulation type and direction, which we use to link the two genes in the graph. When a biological process is specified we also link the genes to a gene-specific subclass of the '`regulation`' class from INO, which in turn is linked to the GO-term. The description logic implementation of such a regulation can be seen in (27) in Appendix A.

Interactions between genes are represented as undirected relationships, as the available interaction data captures symmetric associations rather than directional or causal effects between genes. These interactions may also be associated with a phenotype observed alongside the interaction. Similarly to regulation this is modelled as a link between the involved genes and a gene specific subclass of either a '`protein-protein interaction`' from INO or a '`genetic interaction`' from MI, which is linked to the phenotype.

Beyond the data from SGD we have also included information about reactions and pathways from BioCyc, which uses its own controlled vocabulary. In the graph, reactions are linked to their input and output chemicals, as well as, when specified, genes they are catalysed by and locations in the cell where they take place. We link pathways to their involved reactions, as well as to the compounds that are consumed and produced.

5.3. Predicting gene deletion fitness

To demonstrate the usefulness of our KG, and how the method described in Section 5.1 can be used in practice, we trained GNNs to predict phenotypic traits in *S. cerevisiae*. We use data from Costanzo et al. (2016) where cell growth is measured when pairs of genes are deleted (digenic deletions) from the genome. By comparing this growth to that of cells with no gene deletions, a fitness score could be determined that describes the impact of deleting the two genes. A subset of this data, grown under the same standard experimental conditions (30°C), is used to train our model. This results in a dataset with 10,085,183 examples of deleted gene pairs and a corresponding fitness. Note that the genetic interaction relation from SGD describes similar phenomena, often derived from the same dataset. These relations are thus removed from the graph before training to avoid data leakage.

The prediction task can be formulated as estimating positive real-valued weights on undirected edges between gene nodes. Given a knowledge graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, and the set of genes $\mathcal{V}_g \subset \mathcal{V}$ we aim to learn a symmetric function,

$$f_{\mathcal{G}} : \mathcal{V}_g \times \mathcal{V}_g \to \mathbb{R}_{\geq 0}, \tag{22}$$

where $f_{\mathcal{G}}(i, j) = f_{\mathcal{G}}(j, i)$ predicts the fitness of the double-gene deletion (i, j) based on the information encoded in $\mathcal{G}$.

We divided the classes in the KG into the eight domains in Figure 2b and Table 1. This split was beneficial for performance and for stability in training (see Table 12 in Appendix C). These splits were done manually, but generally they align well with the ontologies the classes are from, or disjoint branches in the same ontology. The reasoning behind this is that these domains represent non-overlapping concepts, so not much is gained by representing them in the same embedding space. Doing this also allows us to reduce the dimensionality of the embedding space and vary it depending on the number of classes in the domain, thereby reducing the overall computational complexity.

After adding reverse links to enable message passing in both directions and removing infrequent edges (fewer than 1,000 occurrences), the resulting graph contains 72 distinct link types[2] and nodes from eight different domains used for prediction. Ignoring infrequent edges was found to increase predictive performance, likely by reducing overfitting, which can be seen in Table 9 in Appendix C.

Prior shallow node embeddings were trained using the *overlap* losses in (12) and (13), representing the classes as Gumbel boxes. Large boxes were penalised using the regularisation in (14) and negative examples are generated by drawing random classes, $\bar{p}$, that are not in $\{p | c \sqsubseteq^* p\}$, where $\sqsubseteq^*$ refers to chains of the '`subClassOf`' relation, i.e., negative examples are not in the set of all ancestors to c. Parameters used to train the box embeddings and the dimensions of the different domains are reported in Appendix B.1.

For predicting the gene-pair fitness we use a heterogeneous GNN together with a fully connected neural network; an overview of the architecture can be seen in Figure 3a. The GNN used is based on the max-aggregated GraphSAGE embedding algorithm (Hamilton et al., 2017) briefly introduced by in Section 4. In our heterogeneous setting, each source-edge-target type has its own SAGEConv-module as proposed by Schlichtkrull et al. (2018), whose outputs are combined using mean aggregation to create the node embeddings from each layer. We found it beneficial to adjust the dimensionality of the message-passing modules based on the type of source-edge-target triple. Domains with a high degree of incoming connectivity, such as 'Material entities' or 'Genes', are assigned higher dimensional feature spaces. This can be interpreted as increasing the expressivity for domains with high degree connectivity and the performance gain can be observed observed in Table 11 in Appendix C. The resulting class embeddings, generated by the GNN, capture aggregated neighbourhood information. By applying box-losses as introduced in Section 5.1, the training will also aim to represent the class hierarchy as box embeddings. Figure 3b illustrates how information is propagated from the initial box embeddings across different domains to the gene embeddings.

[2] Filtering out links with fewer than 1,000 examples removes 160 edge types, of which 118 have fewer than 100 examples and 68 have fewer than 10.

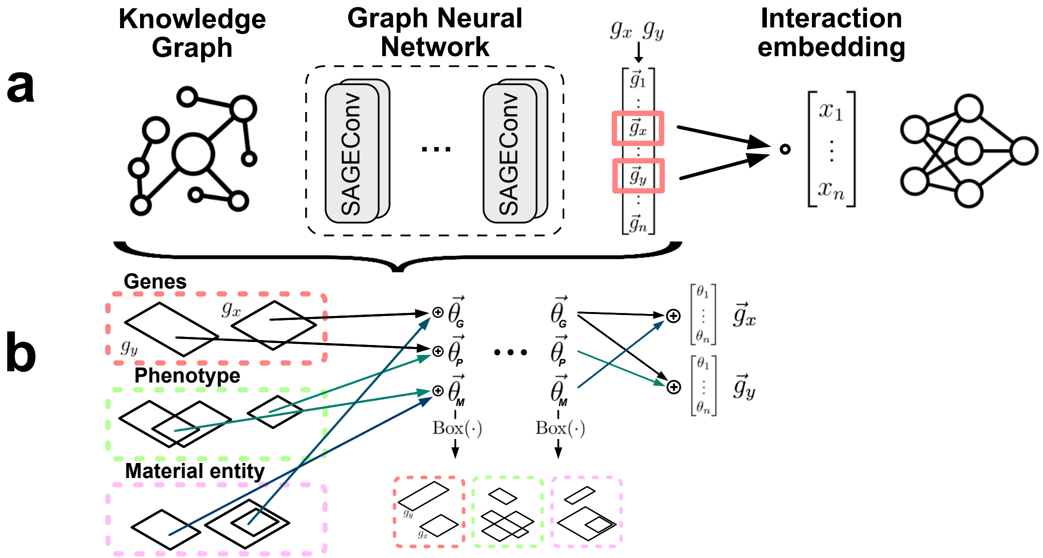


Figure 3: An overview of the system predicting the fitness when deleting pairs of genes is shown in **a**. A GNN using GraphSAGE message passing layers, acting on the KG from Section 5.2, generates node embeddings. The embeddings of pairs of genes are combined through element-wise multiplication and fed to a NN predicting the fitness of the gene deletion. **b** shows how classes in the different domains are represented by boxes and how information is aggregated in the GNN, as well as how the node embeddings throughout the network are interpreted as boxes using the box transformation in (6). These boxes are optimised to represent the class hierarchies from the underlying ontologies. Arrows represent learnable GraphSAGE modules, different for each `source domain-edge-target domain` type.

Table 2: Node feature combinations for edge-level prediction task.

	Formulation	Symmetric	Learnable
Product	$x_1 \odot x_2$	✓	
Bilinear	$x_1 \odot Wx_2 + x_2 \odot Wx_1$	✓	✓
Intersection	$\text{Box}(x_1) \cap \text{Box}(x_2)$	✓	
Concatenation	$\texttt{concat}(x_1, x_2)$		

Because the task operates at the edge level, the embeddings of the deleted genes must be combined prior to being fed into the regressor network. This can be done in a number of ways, in this work we have considered four ways of doing this: element-wise product, vector-valued bilinear symmetric transformation, box intersection, and concatenation, defined in Table 2. Among these, the product, bilinear, and intersection methods are symmetric, and the bilinear transformation is the only learnable combination method. As can be seen in Table 10 in Appendix C, we found the element-wise product to perform the best. The node-pair representation is fed to a fully connected neural network outputting a real valued prediction.

We train the model by minimising, using the Adam optimiser, the following loss function,

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}(y, \hat{y}) + \alpha(\mathcal{L}_{\sqsubseteq} + \beta\mathcal{L}^{-}_{\sqsubseteq}) + \lambda \|w\|^2 \tag{23}$$

$\mathcal{L}_{\text{MSE}}$ denotes the mean squared errors of the fitness predictions and $\mathcal{L}_{\sqsubseteq}$ and $\mathcal{L}^{-}_{\sqsubseteq}$ are defined in (19). α and β are weights determining the impact of the semantic loss, measuring how well the box embeddings represents the class hierarchies. λ controls regularisation of the parameters in the network.

The models are trained and evaluated using 10-fold cross validation where the data split is based on the genes. Any gene pairs that include genes from both the training and validation sets are discarded. This ensures that no pairs involving validation-set genes are seen during training, so the learned representations of genes in the training set do not influence the predictions being evaluated.

Hyperparameters, including learning rate, regularisation (λ), the depth and width of the fully connected neural network, the depth of the GNN, and embedding dimensions throughout the GNN, are tuned using Bayesian optimisation. Tuning is performed on a separate data split from the one evaluated in Section 6.1. This tuning is done for a model using box embeddings as prior node representations, but without semantic loss during training, the same parameters are then used for all evaluated models. The used hyperparameters are reported in Appendix B.2.

5.4. Hypothesis generation

Because the predictions are derived from a knowledge graph in which each edge encodes domain-relevant semantics, they can be exploited to identify patterns among the most informative relationships in the graph. Furthermore, because the predicted measures can be interpreted as arising from interactions between the two deleted genes, we examine which gene-associated traits are jointly influential in driving the model's predictions.

The procedure for identifying such interactions is given in Algorithm 1. Using gradient-based post-hoc interpretability methods, we assign importance scores to all nodes connected to the genes involved in each deletion. For each pair of gene-associated nodes, we compute an interaction score as the product of their individual importance values, and then aggregate these scores across deletions to obtain a set of globally important interacting traits. Note that this method does not rely on hierarchy-aware GNN embeddings.

Algorithm 1: Edge pair importance attribution algorithm

```
Model # trained prediction model
Explainer(Model) # importance attribution algorithm
importances ← {}
foreach $\{g_1, g_2\} \in \{\textit{Deleted genes}\}$ do
    $(\textit{links\_g}_1, \textit{links\_g}_2)$ ← Explainer$(g_1, g_2)$ # importance scores for $(s, p)$ pairs from
      triples where the object is $g_1$ and $g_2$ respectively
    foreach $l_1 \in \textit{links\_g}_1$ do
        foreach $l_2 \in \textit{links\_g}_2$ do
            importances$[l_1 l_2]$ ← importances$[l_1 l_2]$ + $l_1 * l_2$
        end
    end
end
```

We used the *input* × *gradient* method (Shrikumar et al., 2017) to assign importance scores, implemented in Captum (Kokhlikyan et al., 2020), but this approach is not specific to any attribution method.

5.5. Learning GNN box embeddings without a prediction task

To study the effect that semantic losses have on embedding representations, and to demonstrate how they can be used to train box embeddings in the absence of a prediction task, we use the same KG as above with some modifications. Following the same methodology as Section 5.2 we rewrite role and class assertions as TBox axioms. '`subClassOf`' relations are used as the positive examples for $\mathcal{L}_{\sqsubseteq}$, and negative examples for $\mathcal{L}^{-}_{\sqsubseteq}$ are taken from disjointness axioms from two sources. Firstly, we create additional TBox statements for disjointness between the three subclasses in the molecular function domain, and between each of these classes and the subclasses of the other two. Secondly, for randomly drawn pairs to distinguish between individual classes in the graph.

In the same way as the models described in Section 5.3, the GNN is constructed from SAGEConv modules for each edge type. For the purposes of demonstration, we learn embeddings in two dimensions so they can easily be visualised. We simultaneously train initial box embeddings (randomly initialised) and a GNN by minimising, again using the Adam optimiser, the loss function:

$$\mathcal{L} = \mathcal{L}_{\sqsubseteq} + \beta \mathcal{L}^{-}_{\sqsubseteq} + \lambda_s R_{\text{small}} + \lambda \left\| w \right\|^2 , \tag{24}$$

where R is the regularisation loss from (15), penalising small boxes with an l_0 of 1, and β, λ and λ_s are weights determining the impact of the negative semantic loss, weight regularisation loss, and small box regularisation loss respectively. Note the absence of the mean squared error term present in the prediction task above. The regularisation term was included as disjointness tended to make boxes extremely small along one or more dimensions during training rather than move position in the space. The negative semantic loss is decomposed into

$$\mathcal{L}^{-}_{\sqsubseteq} = \mathcal{L}^{-}_{\sqsubseteq\text{data}} + \gamma \mathcal{L}^{-}_{\sqsubseteq\text{random}}, \tag{25}$$

which enables us to tune the contribution of the randomly selected disjointness axioms. For each loss type (*distance* or *overlap*) we separately tuned the hyperparameters, and the used values are reported in Table 6 in Appendix B.3.

5.6. Link evaluation

A potential application for the semantic losses defined above, in addition to the training of box embedding models for quantitative prediction tasks, is to rank proposed revisions to a knowledge graph based on the resultant changes to embeddings and losses. Distance based approaches to link prediction have been attempted before, for example with HAKE (Zhang et al., 2020) where they replaced either subject or object in existing triples. But the use of a GNN in our method means we can theoretically assess completely unseen additions to the graph by evaluating their global effect. We test this by adding single edges to the graph according to the following scheme.

Say that for a given ontology we construct a graph $\mathcal{G} = (V, E)$, where each edge vertex $v \in V$ is a class in the ontology and each edge $e \in E$ represents a role assertion. Following the methodology outlined above, we use $\mathcal{G}$ as the basis for a GNN, and train box embeddings and the weights of the GNN using the semantic loss. Introducing new role assertions to the graph results in graph $\tilde{\mathcal{G}}$, and passing the prior box embeddings through the GNN using these additional edges will change the final box embeddings. We calculate the distance between the original learned box embeddings and those after the changes to the graph, giving us a measure of the change to the embeddings from the graph revision. This process is described in Algorithm 2.

```
Algorithm 2: Link evaluation algorithm
Train embedding parameters $\theta_l$ on $\mathcal{G}_{\text{train}}$
$\delta \leftarrow \emptyset$
foreach $e \in E_{test}$ do
    $\tilde{\mathcal{G}} \leftarrow \mathcal{G}_{\text{train}} \cup \{e\}$
    $B \leftarrow \text{Box}(\text{GNN}_\theta(\mathcal{G}_{\text{train}}))$
    $\tilde{B} \leftarrow \text{Box}(\text{GNN}_\theta(\tilde{\mathcal{G}}))$
    $\delta \leftarrow \delta \cup (e, \langle B, \tilde{B} \rangle)$ # distance between the generated box embeddings
end
Sort $\delta$ to get ranked revisions
```

To evaluate this proposed method, we split the edges in the KG into training and test data using a 80:20 training and test split, stratified by relation type. This results in a training graph $\mathcal{G}_{\text{train}} = (V, E_{\text{train}})$ and a test graph $\mathcal{G}_{\text{test}} = (V, E_{\text{test}})$. The embeddings and GNN are trained as per Section 5.5. We run Algorithm 2, going through each edge in the test data. We also perform the same steps with randomly generated edges with source and target drawn from the same classes as the test edge, and with completely randomly drawn source and target nodes. The distance metric used is defined in (9).

6. Results

6.1. Gene deletion fitness prediction

In Table 3 we present the coefficient of determination (R^2) for different versions of the model described in Section 5.3. We evaluate a model without any information from class hierarchies, a model with the prior node embeddings in box form, and models using both prior node embeddings and the semantic loss in (23). Both the *overlap* and *distance* version of the losses are evaluated. The model not using any hierarchy information (c in Table 3) learns shallow embeddings specifically for this task to represent the nodes in the KG. For the model only using prior box embeddings (d in Table 3), these are not modified during training. For the two models with the semantic loss (e and f), we apply it to all domains except the one embedding the genes. The gene hierarchy builds on a rudimentary SGD gene categorisation, offering very little information, with over 90% of genes falling into the same category. Hence we deem the hierarchy in this domain uninformative.

The GNN-based models are compared to Light Gradient Boosting Machines (LightGBMs) (Ke et al., 2017). We considered the instantiation of the phenotype information from the KG as gene representation. The phenotypes describe observable characteristics of the genes and are the part of the KG we expect to be most informative for this task (further support for this is seen when considering feature importances for the GNN, mentioned in Section 6.2, which are dominated by phenotypes). The instantiation of the phenotypes is sparse, with 2680 features.

We also considered a ComplEx (Trouillon et al., 2016) embedding (64 dimensions) of the KG as an alternative gene representation. In contrast to the phenotype instantiation, which relies on a hand-selected subset of biologically relevant relations, ComplEx provides a dense, low-dimensional embedding learned from the full multi-relational structure of the KG. This allows LightGBM to exploit global relational patterns that are not explicitly encoded in the phenotype feature vectors. We also evaluated ComplEx embedding with an MLP prediction head, as well as a Box^{2}EL-based predictor. They did not achieve the same predictive performance, and are reported in Table 8 in Appendix C.

From the results it is clear that the GNN generates gene embeddings which can be used for predicting this fitness to a reasonable degree, given the amount of noise typically present in biological measurements (Li et al., 2021). Using the box embeddings to represent classes rather than learning them from scratch results in a significant (p<0.05, paired t-test) improvement. Enforcing the hierarchical class structure through the semantic loss throughout the model improves the results further, the model trained with the *distance*-loss performs significantly (p<0.05, paired t-test) better than the models not using the semantic loss. The instantiated phenotype information seems to be somewhat useful for prediction, but is not as informative as the full KG. We also see that the ComplEx KG embedding captures information useful for this task, however not as informative as the task specific embedding found by the GNNs.

The parity plots for the predictions are shown in Figure 4(a) and 4(c). From this we can see that most double gene deletions do not have major impact on the fitness. We can also see a clear shrinkage effect where the model mispredicts extreme values, especially deletions with low fitness are overestimated. Comparing the predictions from the model trained with

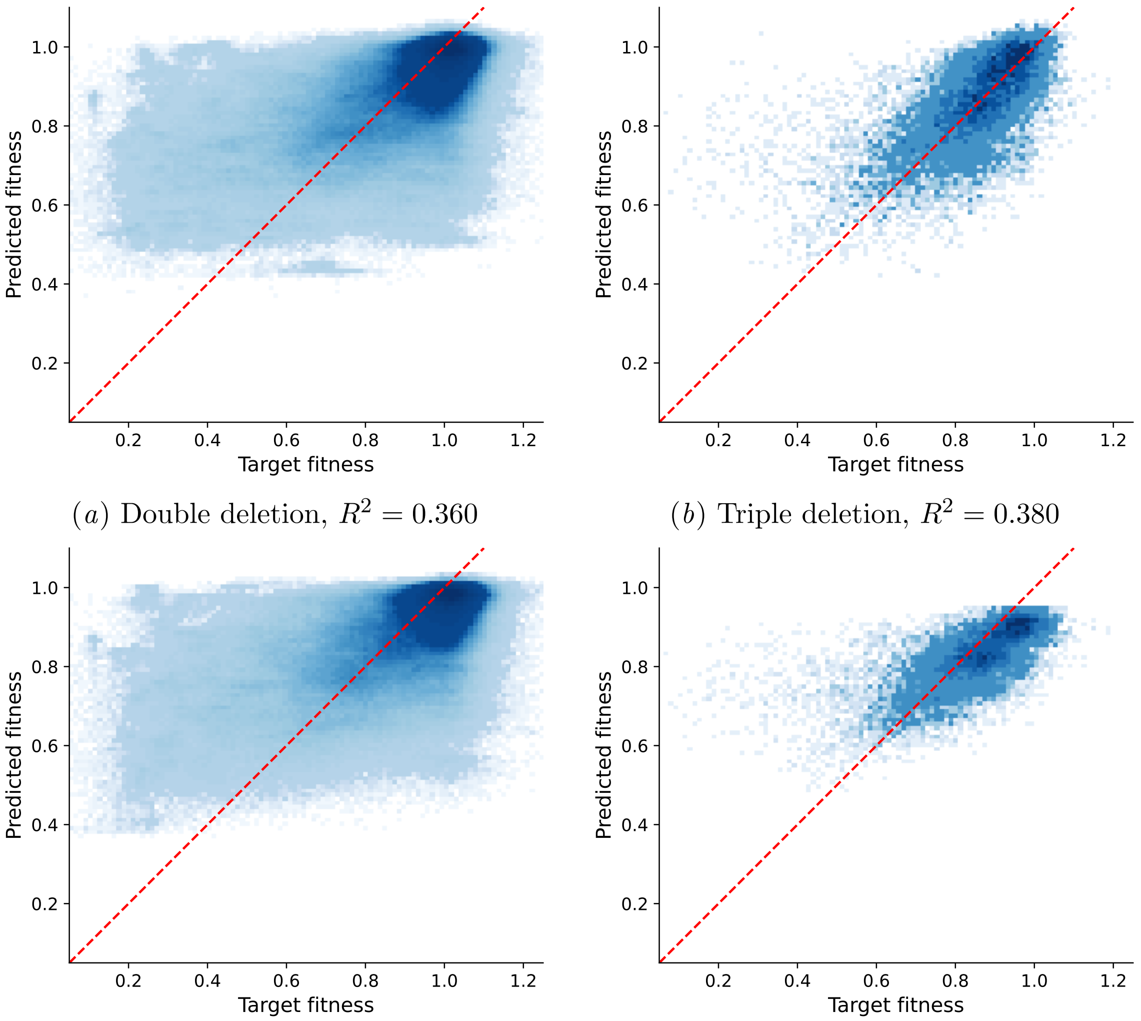
Table 3: Results from 10-fold cross-validation of digenic deletion fitness. The GNN without box embeddings (*c*) learns task-specific shallow embeddings as the prior node representations. The other three GNNs (*d*-*f*) uses pre-trained box embeddings and the semantic loss in (23) is applied to two of them (*e* and *f*). All GNN models share the same architecture. The instantiation model (*a*) uses a sparse feature matrix with non-zero entries for phenotype annotations from the KG. The ComplEx based model (*b*) first embeds the KG in 64 dimensions and predicts the growth from this. Significant pairwise differences are indicated by ↑ and ↓ (p<0.05, paired, one-sided t-test).

	Description	Mean R^2	SD	a	b	c	d	e	f
a	ComplEx + LightGBM	0.191	0.039	-		↓	↓	↓	↓
b	Instantiations + LightGBM	0.211	0.022		-	↓	↓	↓	↓
c	GNN without box embeddings	0.348	0.050	↑	↑	-		↓	↓
d	GNN with prior box embeddings	0.360	0.043	↑	↑		-		↓
e	GNN + $\mathcal{L}_{overlap}$	0.368	0.038	↑	↑	↑		-	
f	GNN + $\mathcal{L}_{distance}$	**0.377**	0.046	↑	↑	↑	↑		-

the semantic loss we can see that they in general are rather similar, but that the semantic loss model seems to have fewer large underestimations.

Figure 5 show the semantic losses in the different domains, introduced in Figure 2b and Table 1, for the best performing model, using $\mathcal{L}_{distance}$. A similar pattern is observed for all domains where both the positive loss for the first layer (the pretrained box embeddings) and the negative loss for the second layer is low and fairly constant. The positive losses for the second layers decreases across all domains throughout training, while the negative loss for the first layer does not change much and is substantially higher than the others. This suggests that, even though they result in a significant improvement in prediction performance, the initial box embeddings have a lot of overlap between classes. On the other hand, the embeddings generated by the GNN discriminate very well between classes, already at the first epoch and the hierarchical structure is learnt throughout training.

To evaluate our model on a slightly modified version of the original task we used data from Kuzmin et al. (2018), who performed a study similar to the one used for training our models, but focused on trigenic deletion fitness. This dataset comprises a total of 15,095 triple deletion datapoints. For this we use one model trained on the full dataset from Costanzo et al. (2016), but instead perform the element-wise product between the three involved genes. Notably we achieve an R^2 of 0.380 for a model using box embeddings as prior node representations, and 0.415 for a model using the same prior node embeddings, but trained with the *distance*-based semantic loss. These values are slightly higher than the average performance observed in the cross-validation of digenic deletions. Figure 4(b) and 4(d), shows parity plots for these predictions. Again, the prediction patterns for the two models look similar, but the model trained with the semantic loss does not predict as high fitness. An important note on this experiment is that, unlike the double deletion experiment, the individual genes making up the triple deletions are now seen as parts of double deletion examples in training.



(a) Double deletion, $R^2 = 0.360$

(b) Triple deletion, $R^2 = 0.380$

(c) Double deletion with semantic loss, $R^2 = 0.377$

(d) Triple deletion with semantic loss, $R^2 = 0.415$

Figure 4: Parity plots for double, (a) and (c), and triple, (b) and (d), gene deletions. (a) and (b) shows the parity plot for the model using box embeddings as prior node representations only, while (c) and (d) shows the predictions from a model also trained with the *distance*-based semantic losses, $\mathcal{L}_{distance}$. For the double deletion, the predictions from all validation sets in the cross validation are shown.

6.2. Hypothesis generation and experimental evaluation

In this section, we present a case study evaluating the interaction-discovery procedure described in Section 5.4, identifying candidate trait interactions for experimental testing. To focus on patterns corresponding to viable experiments for our laboratory setup, we filtered for edges related to nutrient utilisation phenotypes. A more detailed description of the filtering process can be found in Appendix D.1. The top ten most important edge pairs are shown in Figure 6(a) and detailed in Table 13 and 14 in Appendix D.2. The highest-

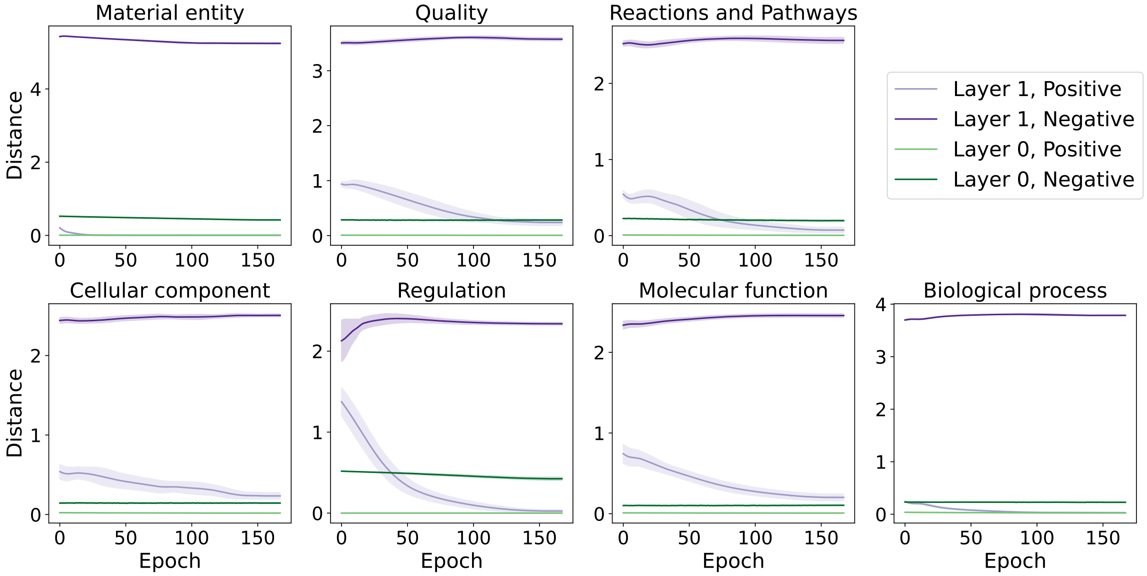
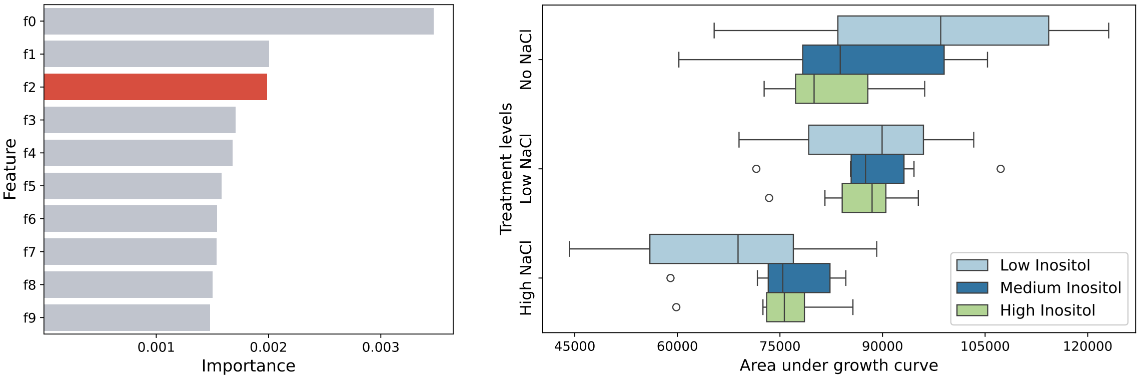


Figure 5: Average $\mathcal{L}_{distance}$ and $\mathcal{L}^{-}_{distance}$ losses per class for the different domains in the KG, during training of the best performing model (f) in Table 3. The line is the average loss across the 10 folds and the shaded area shows ± one standard deviation. Note that this loss is not applied to the gene-domain, since its class hierarchy is not deemed to be informative.

weighted, safely testable pair was selected and highlighted in red in Figure 6(a), linking one of the involved genes to inositol (vitamin B8) utilisation and the other to NaCl stress resistance, suggesting a potential interaction between these traits.

To experimentally test this hypothesis, a perturbation experiment was performed in an automated laboratory cell (Williams et al., 2015), in which inositol and NaCl was supplied in a range of concentrations, details about the experimental design and cultivation methods can be found in Appendix D.3. An $\Delta ino1$ mutant (INOsitol requiring) was used for all subsequent experiments, as it is unable to synthesise inositol on its own, ensuring that any intracellular accumulation was acquired only through transport from the media. The growth dynamics of the cells in the different experimental conditions were summarised with the area under curve (AUC) of the growth curves, providing a single-valued measure of the biomass accumulation over the course of the experiment. The full growth dynamics can be seen in Figure 9 in Appendix D.4 and summarising boxplots are shown in Figure 6(b). Statistical testing for interaction effects was done with a Gaussian generalised linear model (GLM), further details can be found in Appendix D.4.

These empirical results, seen in Figure 6(b) and Table 18 in Appendix D.4, indicate a significant interaction between inositol supplementation and induced NaCl stress, verifying that the proposed edge-interactions are consistent with experimental data. Specifically, supplementing with inositol rescued cells from NaCl-induced stress, indicating that inositol availability enhances their ability to withstand salt stress. Inositol has previously been



(a) Feature importances

(b) AUC growth boxplots

Figure 6: An overview of the selection and results of the experiment we performed. (a) shows the highest ranked importances of edge-pairs and the pair selected for the experiment, nutrient utilisation of inositol and stress resistance to NaCl, is highlighted in red. *f0* and *f1*, which have a higher assigned weight, are discarded due to safety and lab constraints as it involves the chemical bleomycin. (b) Box plot showing the distribution of AUC for all of the experimental conditions tested. Inositol supplementation significantly impacts growth dynamics in high doses ($p < 0.05$). NaCl stress changes the impact of inositol in a dose dependent manner, suggesting an interactive effect ($p < 0.05$).

implicated in biosynthesis and integrity of cell membranes (Culbertson and Henry, 1975). Since NaCl can disrupt osmotic balance, enhanced membrane stability is likely to have a protective effect for the cells.

6.3. Demonstration of embeddings for the molecular function domain

For the knowledge graph constructed according to the method in Section 5.5, after the hyperparameter search for *distance*-loss and *overlap*-loss, we constructed final box embeddings in two dimensions using a GNN with one message passing layer. In Figure 7 we plot box embeddings for each loss, before input into the GNN and then the final embeddings, for a subset of classes in the molecular function domain. We see clearly that the GNN is performing a transformation of the prior embeddings. Furthermore, both losses result in learned embeddings that begin to capture semantic concepts from the KG, in particular that '`structural molecule activity`' and '`molecular function regulator activity`' are disjoint, and that each of these is disjoint from '`small molecule sensor activity`'. Both losses also responded to the randomly drawn negative loss contributions ($\mathcal{L}^{-}_{\sqsubseteq\text{random}}$) in attempting to separate individuals, though this is more evident with the *overlap*-loss. These semantic constraints are more evidently respected in the embeddings before the message passing in the GNN. This is perhaps understandable, as the message passing includes no additional information about the hierarchy of the ontology, but does include information from the relations in the graph. Also, in this KG, the concepts in the molecular function domain have relations only to concepts in the gene domain (see Figure 2). The hierarchy in the gene domain, constructed from the gene categorisation in SGD, is fairly flat. The majority of concepts in this domain are immediate subclasses of 'ORF'.

Aside from this observation, comparing Figures 7(a) and 7(b), we see that prior to being passed through the GNN, the boxes corresponding to different 'structural molecule activity', 'molecular function regulator activity', and 'small molecule sensor activity' subclasses are of quite varied shape and size, and somewhat spread through the embedding space. Some clustering is apparent, which corresponds to structure in the hierarchy. The loss from randomly drawn disjointness axioms keeps them somewhat separated. Having passed through the GNN, the boxes take on dramatically different shapes and relative positions. The semantic loss of these embeddings is low on average, but subclasses of the same superclass have now very similar embeddings, and there are some clear violations of the hierarchy. By contrast, comparing Figures 7(c) and 7(d), we see that the shapes of the boxes trained with *overlap* losses are often longer and thinner, minimising the volume of each intersection. And the final embeddings do not differ from the prior embeddings to the same extent as with *distance* loss. The overall position of the parent classes is broadly the same, with some minor translations. With both losses, the box volume increased after passing through the GNN, and the difference between height and width decreased.

6.4. Evaluation of graph revisions

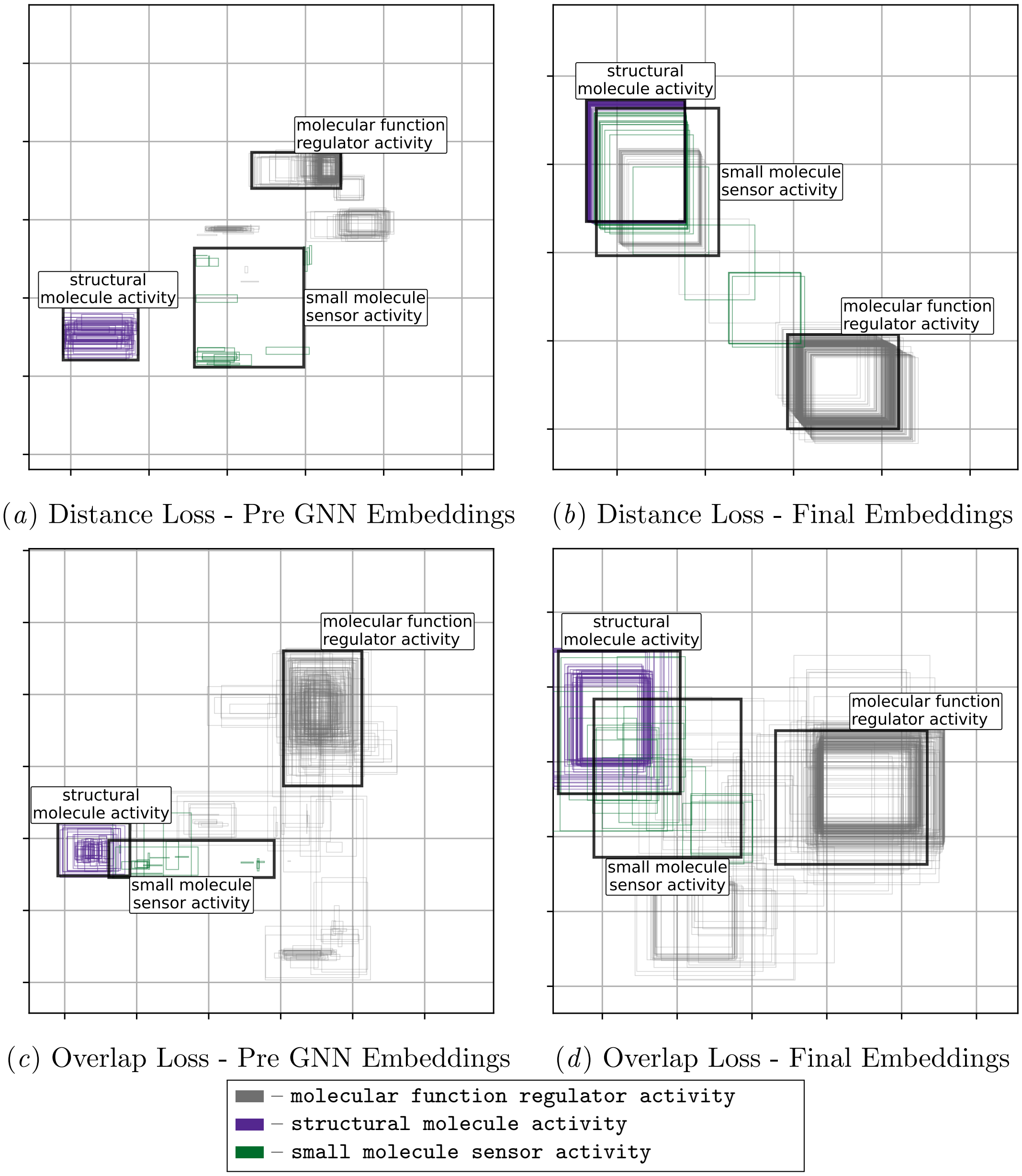
The median distances and loss changes for individual edge revisions to the graph were small, and these measures overall had very large variance. However there were signs in these data which suggest that using these values could be used to rank candidate revisions to a knowledge base. For the majority of the relations, when constraining the random draw to appropriate classes, the distance rank distribution of these randomly drawn edges was significantly different to the rank distribution of the test edges (p <0.05, two-sided Mann–Whitney U test). Figure 8 shows a subset of the relation types in the graph. For some relation types, as is the case for 'hasChemStressResistance' and hasChemStressResistance_Increased, there is no appreciable difference in the embeddings after the new edges are added. Summary statistics for these distance evaluations are provided in Table 19 in Appendix E.

7. Discussion

In this work we have presented a method generating KG embeddings using GNNs, taking hierarchical class information as well as graph structure into account. We do this by introducing a semantic loss term to the training acting on box transformations of the node embeddings. We have seen that it can be used on its own to generate KG embeddings adhering to subsumptions defined in ontologies to some degree, but more importantly this method shows promise when used together with another, task-specific prediction loss.

The main advantage of our approach is that signals from semantic information encoded in class hierarchies, and signals from predictive tasks can jointly be used to train graph neural networks.

We demonstrated the power of using both these signals together by predicting digenic deletion fitness from a KG describing *S. cerevisiae* genes which we constructed. While the predictive R^2 of 0.377 may seem low, biological data is inherently noisy, and even



(a) Distance Loss - Pre GNN Embeddings

(b) Distance Loss - Final Embeddings

(c) Overlap Loss - Pre GNN Embeddings

(d) Overlap Loss - Final Embeddings

Figure 7: Learned box embeddings in two dimensions for the molecular function domain. 7(a) and 7(b) box embeddings prior to input into GNN; 7(b) and 7(d) show final embeddings for *distance* and *overlap* loss respectively.

replicating experiments is challenging (Roper et al., 2022). Moreover, our model predicts quantitative outcomes from high-level qualitative information. What is more interesting is how the prediction performance was improved by introducing more hierarchical information to the models. One explanation we envision for the improved performance is that enforcing the class hierarchies has a regularising effect, while also providing semantic grounding for the modelled concepts. A weakness of the ontologies used, for example ChEBI and GO,

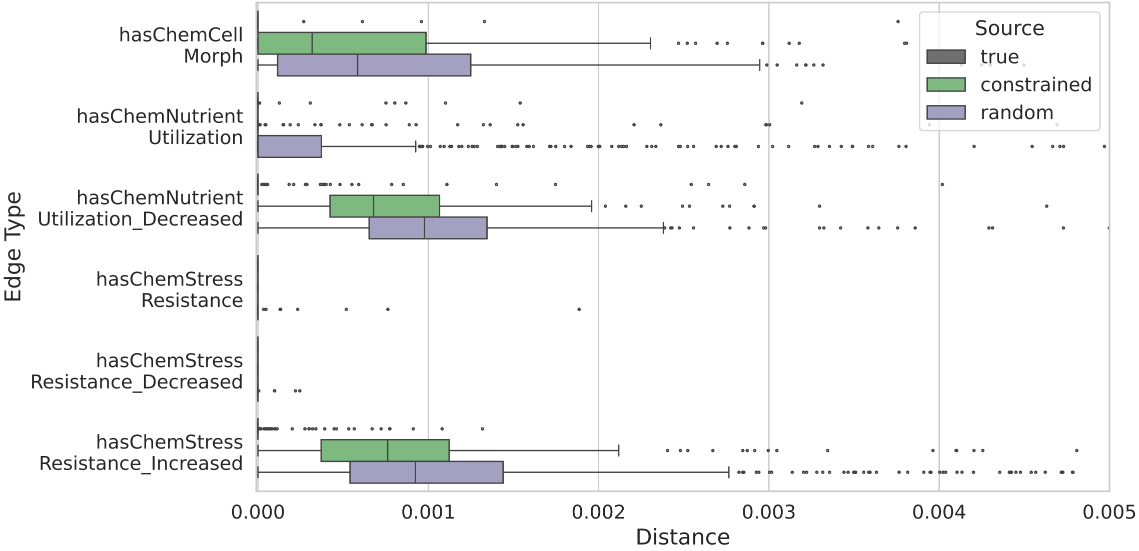


Figure 8: Distribution of distances of box embeddings learned from revised graphs $\tilde{\mathcal{G}}$ to the original embeddings learned from $\mathcal{G}_{\text{train}}$, shown by relation type, for a subset of the relation types in the graph. (The method for calculating these differences is described in 5.6). For most relations, the distance ranks of randomly drawn edges (constrained to the appropriate class) were significantly different to the ranks of the test edges (p <0.05, two-sided Mann–Whitney U test). Embeddings were learned with inclusion losses.

is that they do not contain axioms for disjointness between concepts (such as those we introduced in Section 5.5). This is possibly a consequence of them having been designed primarily for constructing databases, rather than for automated reasoning. Despite this, the improved performance on the prediction task also suggests that the ontologies used are, at least somewhat, good models of the domains.

KG embeddings that, at least to some extent, adhere to their underlying ontologies can potentially be used for several tasks, even if they were trained with a particular problem in mind. Our trigenic gene deletion experiments are one example of applying the model slightly outside its original domain. The increased performance in this task compared to the digenic deletion is likely, at least partly, due to the individual genes involved no longer being unseen during training. The fitness will depend heavily on the traits of the individual genes, which will be better represented for genes in the training data. The embeddings could potentially be applied to a broader range of tasks, such as GO annotation of genes, which is typically addressed by integrating multiple knowledge sources (Merino et al., 2022).

We have not utilised any sequence information for our fitness predictions, despite it being the most informative data about genes and fully available for *S. cerevisiae*. Representing the initial gene embeddings as some encoding of their sequence would provide richer and more meaningful gene embeddings and most likely result in better predictions. However, our current setup will put more emphasis on using the information in the KG as the basis of the predictions. In this way this helps demonstrate both the knowledge in the KG and the usefulness of our embedding method.

The capability of making predictions from qualitative facts enabled interpretability techniques to guide experiment selection, underscoring the value of structured data representation and computational methods in accelerating research. Our edge filtering for viable experiments introduces biases regarding the type of hypotheses generated. Leveraging large language models could be one approach to automatically refine this selection and reveal overlooked experiments.

We suggested two different loss functions for learning box embeddings. The first is based on the volume of overlap between boxes, rewarding overlap for subclasses, and penalising overlaps in the case of disjointness. The second was based on the distance from the boxes fulfilling the subsumption axioms. Studying the 2-dimensional embeddings of in Figure 7, the embedding learnt through the *overlap* loss seems to have slightly better captured the semantics of the ontology. Boxes for subclasses of '`structural molecule activity`' and '`molecular function regulator activity`' are mostly contained within their respective superclass, but there were some inconsistencies with the hierarchical structure of the ontology, especially after the message passing of the GNN. The embedding learnt through the *distance*-based loss instead places the boxes for each class along a diagonal.

Another property of the embeddings learned in this example is the variation in position among instances, which is better for the *overlap*-loss. Figure 7(*b*) also shows some variation, primarily among instances of '`small molecule sensor activity`', but there are some large disagreements with the semantics of the ontology. It is probably the case that a 2-dimensional embedding is not sufficient to capture the complexity of this domain.

Interestingly, when class hierarchies were enforced in the gene deletion fitness predictions, the *distance*-based loss yielded better predictive performance. One potential explanation for this finding is that the *distance*-based loss may be particularly well suited as a semantic loss to complement a task-specific loss. By introducing a semantic loss component to our total loss, we of course want to capture how faithfully a given embedding adheres to the semantics of the source ontology. But to have smooth training, a desirable feature of a semantic loss measure is that its gradients are informative when the constraints are not fulfilled. This is exactly what $\mathcal{L}_{distance}$ does. To obtain useful gradients with $\mathcal{L}_{overlap}$, one could for example use Gumbel boxes as in Dasgupta et al. (2020), but a result of the introduced smoothing is that losses can remain nonzero even when the semantic constraints are fulfilled. With another loss term primarily guiding the training, in this case $\mathcal{L}_{\text{MSE}}$, the issue of $\mathcal{L}_{distance}$ not discriminating between classes that we observed in Section 6.3 is not as pressing, as the primary loss will likely also push towards being able to discriminate.

Our proposed method for evaluating link revisions to a KG can be seen as an interesting application and direction for future research. Evaluating only the distance in the generated embeddings is, in this setting, not enough to discriminate between true and random edges. It could possibly work better for a more heterogeneous graph, with a richer class hierarchy. A measure of distance, combined with the semantic losses, could represent a measure of surprise. In a scientific discovery context, surprise can be used as part of an active learning algorithm, where a learning agent selects hypotheses that are expected to generate the highest amount of information. This opens up opportunities to create and evaluate scientific hypotheses based on the effect they have on KG embeddings in the domain.

Future work

We identify several directions for future work based on this paper. First, the hierarchy-aware GNN approach should be evaluated on a broader range of problems, for example tasks from the Open Graph Benchmark. While the results presented here indicate promise for a specific link-level prediction task on a particular KG, we do not assess how well the approach generalises to other settings. We expect that its effectiveness will depend on the availability and informativeness of class hierarchies, as well as on the extent to which they provide information not already encoded in the graph structure.

Representing all graph elements as TBox axioms constitutes a simplification of the underlying semantics. For example, modelling individuals as boxes allows intersections between instances, which lack a clear semantic interpretation. In contrast, instances in knowledge graph embedding models are more commonly represented as points. Similarly, existential restrictions between classes are currently modelled in the same manner as relations between instances, whereas such restrictions could potentially be handled more directly within the semantic loss formulation.

The box representation could also be more tightly integrated into the message-passing process through box-level message aggregation, as in Lin et al. (2024), where messages are propagated via geometric composition of boxes. For hierarchy-aware GNNs on knowledge graphs, this could enable message passing that preserves class hierarchies and ontological constraints directly in the embedding space, rather than enforcing them only through auxiliary loss terms.

We present a method for generating scientific hypotheses and demonstrate its potential through the successful experimental validation of one such hypothesis. Although promising, this constitutes only an initial proof of concept, and substantially more extensive evaluation is required to assess the reliability and broader applicability of the approach. In addition to experimental correctness, future work should evaluate generated hypotheses according to their scientific interest, novelty, and relevance as perceived by domain experts.

Finally, the link evaluation method introduced in Section 5.6, while rudimentary in the context of link prediction, may provide an alternative mechanism for hypothesis evaluation. By evaluating sets of added links jointly, hypotheses can be assessed not only in isolation but also in terms of their global impact on the learned representation. Changes in the embedding space can be interpreted as a measure of surprise, while corresponding changes in the semantic loss provide a quantitative signal of consistency with existing knowledge. Together, these signals offer a principled basis for evaluating hypotheses with respect to both novelty and plausibility.

8. Conclusion

In this work we have presented a method generating KG embeddings using GNNs, taking hierarchical class information as well as graph structure into account. We show that enforcing the class hierarchies as semantic losses throughout the model can help predictive performance while also producing internal representations which better correspond to our knowledge of the domain. This is demonstrated on a KG we have created from publicly available data about the yeast *S. cerevisiae*. Based on this KG we can, not only predict biological measurements, but also use interpretability tools to form a hypothesis about phenotype interactions. One such hypothesis was tested and supported by performing a biological experiment, uncovering an association between inositol utilisation and NaCl stress. This illustrates how models with semantic grounding can help in scientific discovery.

The code and data for this project are available at https://github.com/filipkro/kg-box-emb.

Acknowledgments

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the UK Engineering and Physical Sciences Research Council (EPSRC) [EP/R022925/2, EP/W004801/1 and EP/X032418/1], the Chalmers AI Research Centre, and Swedish Research Council Formas [2020-01690].

References

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1): 25–29, 2000. ISSN 1546-1718. doi: 10.1038/75556. URL https://www.nature.com/articles/ng0500_25. Number: 1 Publisher: Nature Publishing Group.

Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press. ISBN 978-0-521-87361-1. doi: 10.1017/9781139025355.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, volume 2 of *NIPS'13*, pages 2787–2795. Curran Associates Inc., 2013.

Daniel Brunnsåker, Filip Kronström, Ievgeniia A Tiukova, and Ross D King. Interpreting protein abundance in saccharomyces cerevisiae through relational learning. *Bioinformatics*, 40(2):btae050, 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae050. URL https://doi.org/10.1093/bioinformatics/btae050.

Daniel Brunnsåker, Alexander H. Gower, Prajakta Naval, Erik Y. Bjurström, Filip Kronström, Ievgeniia A. Tiukova, and Ross D. King. Agentic AI integrated with scientific knowledge: Laboratory validation in systems biology, 2025. URL https://www.biorxiv.org/content/10.1101/2025.06.24.661378v3. ISSN: 2692-8205 Pages: 2025.06.24.661378 Section: New Results.

Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. Box embeddings: An open-source library for representation learning using geometric structures, 2021. URL http://arxiv.org/abs/2109.04997.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. 2019. URL https://openreview.net/forum?id=r1xMH1BtvB.

Maria C. Costanzo, Marek S. Skrzypek, Robert Nash, Edith Wong, Gail Binkley, Stacia R. Engel, Benjamin Hitz, Eurie L. Hong, J. Michael Cherry, and the Saccharomyces Genome Database Project. New mutant phenotype data curation system in the saccharomyces genome database. *Database: The Journal of Biological Databases and Curation*, 2009: bap001, 2009. ISSN 1758-0463. doi: 10.1093/database/bap001.

Michael Costanzo, Benjamin VanderSluis, Elizabeth N. Koch, Anastasia Baryshnikova, Carles Pons, Guihong Tan, Wen Wang, Matej Usaj, Julia Hanchard, Susan D. Lee, Vicent Pelechano, Erin B. Styles, Maximilian Billmann, Jolanda van Leeuwen, Nydia van Dyk, Zhen-Yuan Lin, Elena Kuzmin, Justin Nelson, Jeff S. Piotrowski, Tharan Srikumar, Sondra Bahr, Yiqun Chen, Raamesh Deshpande, Christoph F. Kurat, Sheena C. Li, Zhijian Li, Mojca Mattiazzi Usaj, Hiroki Okada, Natasha Pascoe, Bryan-Joseph San Luis, Sara Sharifpoor, Emira Shuteriqi, Scott W. Simpkins, Jamie Snider, Harsha Garadi Suresh, Yizhao Tan, Hongwei Zhu, Noel Malod-Dognin, Vuk Janjic, Natasa Przulj, Olga G. Troyanskaya, Igor Stagljar, Tian Xia, Yoshikazu Ohya, Anne-Claude Gingras, Brian Raught, Michael Boutros, Lars M. Steinmetz, Claire L. Moore, Adam P. Rosebrock, Amy A. Caudy, Chad L. Myers, Brenda Andrews, and Charles Boone. A global genetic interaction network maps a wiring diagram of cellular function. *Science*, 353(6306):aaf1420, 2016. doi: 10.1126/science.aaf1420. URL https://www.science.org/doi/10.1126/science.aaf1420. Publisher: American Association for the Advancement of Science.

Michael Costanzo, Elena Kuzmin, Jolanda van Leeuwen, Barbara Mair, Jason Moffat, Charles Boone, and Brenda Andrews. Global genetic networks and the genotype-to-phenotype relationship. *Cell*, 177(1):85–100, 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.01.033. URL https://www.sciencedirect.com/science/article/pii/S0092867419300960.

Michael R Culbertson and Susan A Henry. INOSITOL-REQUIRING MUTANTS OF SACCHAROMYCES CEREVISIAE. *Genetics*, 80(1):23–40, 1975. ISSN 1943-2631. doi: 10.1093/genetics/80.1.23. URL https://doi.org/10.1093/genetics/80.1.23.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 182–192. Curran Associates Inc., 2020. ISBN 978-1-7138-2954-6.

Stacia R. Engel, Suzi Aleksander, Robert S. Nash, Edith D. Wong, Shuai Weng, Stuart R. Miyasato, Gavin Sherlock, and J. Michael Cherry. Saccharomyces genome database: Advances in genome annotation, expanded biochemical pathways, and other key enhancements. *Genetics*, page iyae185, 2024. ISSN 1943-2631. doi: 10.1093/genetics/iyae185.

María Andreína Francisco Rodríguez, Jordi Carreras Puigvert, and Ola Spjuth. Designing microplate layouts using artificial intelligence. *Artificial Intelligence in the Life Sciences*, 3:100073, 2023. ISSN 2667-3185. doi: 10.1016/j.ailsci.2023.100073. URL https://www.sciencedirect.com/science/article/pii/S266731852300017X.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1646–1655. PMLR, 2018. ISSN: 2640-3498.

Francesco Gualdi, Baldomero Oliva, and Janet Piñero. Predicting gene disease associations with knowledge graph embeddings for diseases with curtailed information. *NAR Genomics and Bioinformatics*, 6(2):lqae049, 2024. ISSN 2631-9268. doi: 10.1093/nargab/lqae049. URL https://doi.org/10.1093/nargab/lqae049.

Víctor Gutiérrez-Basulto and S. Schockaert. From knowledge graph embedding to ontology embedding? An analysis of the compatibility between vector space representations and rules. In *International Conference on Principles of Knowledge Representation and Reasoning*, 2018.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 1025–1035. Curran Associates Inc., 2017. ISBN 978-1-5108-6096-4.

Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic acids research*, 44:D1214–9, 2016. ISSN 1362-4962. doi: 10.1093/nar/gkv1031. URL https://europepmc.org/articles/PMC4702775.

Henning Hermjakob, Luisa Montecchi-Palazzi, Gary Bader, Jérôme Wojcik, Lukasz Salwinski, Arnaud Ceol, Susan Moore, Sandra Orchard, Ugis Sarkans, Christian von Mering, Bernd Roechert, Sylvain Poux, Eva Jung, Henning Mersch, Paul Kersey, Michael Lappe, Yixue Li, Rong Zeng, Debashis Rana, Macha Nikolski, Holger Husi, Christine Brun, K. Shanker, Seth G. N. Grant, Chris Sander, Peer Bork, Weimin Zhu, Akhilesh Pandey, Alvis Brazma, Bernard Jacq, Marc Vidal, David Sherman, Pierre Legrain, Gianni Cesareni, Ioannis Xenarios, David Eisenberg, Boris Steipe, Chris Hogue, and Rolf Apweiler. The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. *Nature Biotechnology*, 22(2):177–183, 2004. ISSN 1087-0156. doi: 10.1038/nbt926.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 22118–22133. Curran Associates Inc., 2020. ISBN 978-1-7138-2954-6.

Junguk Hur, Arzucan Özgür, Zuoshuang Xiang, and Yongqun He. Development and application of an interaction network ontology for literature mining of vaccine-associated gene-gene interactions. *Journal of Biomedical Semantics*, 6(1):2, 2015. ISSN 2041-1480. doi: 10.1186/2041-1480-6-2. URL https://doi.org/10.1186/2041-1480-6-2.

Mathias Jackermeier, Jiaoyan Chen, and Ian Horrocks. Dual box embeddings for the description logic EL++. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pages 2250–2258. Association for Computing Machinery, 2024. ISBN 979-8-4007-0171-9. doi: 10.1145/3589334.3645648. URL https://dl.acm.org/doi/10.1145/3589334.3645648.

Peter D. Karp, Richard Billington, Ron Caspi, Carol A. Fulcher, Mario Latendresse, Anamika Kothari, Ingrid M. Keseler, Markus Krummenacker, Peter E. Midford, Quang Ong, Wai Kit Ong, Suzanne M. Paley, and Pallavi Subhraveti. The BioCyc collection of microbial genomes and metabolic pathways. *Briefings in Bioinformatics*, 20(4):1085–1093, 2019. ISSN 1477-4054. doi: 10.1093/bib/bbx085.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Ross D. King, Kenneth E. Whelan, Ffion M. Jones, Philip G. K. Reiser, Christopher H. Bryant, Stephen H. Muggleton, Douglas B. Kell, and Stephen G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427 (6971):247–252, 2004. ISSN 1476-4687. doi: 10.1038/nature02236. URL https://www.nature.com/articles/nature02236. Publisher: Nature Publishing Group.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

Filip Kronström, Daniel Brunnsåker, Ievgeniia A. Tiukova, and Ross D. King. Ontology-based box embeddings and knowledge graphs for predicting phenotypic traits in saccharomyces cerevisiae. In *19th International Conference on Neurosymbolic Learning and Reasoning*, 2025.

Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. EL embeddings: Geometric construction of models for the description logic EL++. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6103–6109. International Joint Conferences on Artificial Intelligence Organization, 2019. ISBN 978-0-9992411-4-1. doi: 10.24963/ijcai.2019/845. URL https://www.ijcai.org/proceedings/2019/845.

Elena Kuzmin, Benjamin VanderSluis, Wen Wang, Guihong Tan, Raamesh Deshpande, Yiqun Chen, Matej Usaj, Attila Balint, Mojca Mattiazzi Usaj, Jolanda van Leeuwen, Elizabeth N. Koch, Carles Pons, Andrius J. Dagilis, Michael Pryszlak, Zi Yang Wang, Julia Hanchard, Margot Riggi, Kaicong Xu, Hamed Heydari, Bryan-Joseph San Luis, Ermira Shuteriqi, Hongwei Zhu, Nydia Van Dyk, Sara Sharifpoor, Michael Costanzo, Robbie Loewith, Amy Caudy, Daniel Bolnick, Grant W. Brown, Brenda J. Andrews, Charles Boone, and Chad L. Myers. Systematic analysis of complex genetic interactions. *Science*, 360(6386):eaao1729, 2018. doi: 10.1126/science.aao1729. URL https://www.science.org/doi/10.1126/science.aao1729. Publisher: American Association for the Advancement of Science.

Gang Li, Jan Zrimec, Boyang Ji, Jun Geng, Johan Larsbrink, Aleksej Zelezniak, Jens Nielsen, and Martin KM Engqvist. Performance of regression models as a function of experiment noise. *Bioinformatics and Biology Insights*, 15:11779322211020315, 2021. ISSN 1177-9322. doi: 10.1177/11779322211020315. URL https://doi.org/10.1177/11779322211020315. Publisher: SAGE Publications Ltd STM.

Fake Lin, Ziwei Zhao, Xi Zhu, Da Zhang, Shitian Shen, Xueying Li, Tong Xu, Suojuan Zhang, and Enhong Chen. When box meets graph neural network in tag-aware recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 1770–1780. Association for Computing Machinery, 2024. ISBN 979-8-4007-0490-1. doi: 10.1145/3637528.3671973.

Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, 15(4):290–298, 2018. ISSN 1548-7105. doi: 10.1038/nmeth.4627. URL https://www.nature.com/articles/nmeth.4627. Publisher: Nature Publishing Group.

Adel Memariani, Martin Glauer, Simon Flügel, Fabian Neuhaus, Janna Hastings, and Till Mossakowski. Box embeddings for extending ontologies: a data-driven and interpretable approach. 17(1):138, 2025. ISSN 1758-2946. doi: 10.1186/s13321-025-01086-1.

Gabriela A Merino, Rabie Saidi, Diego H Milone, Georgina Stegmayer, and Maria J Martin. Hierarchical deep learning for predicting GO annotations by integrating protein knowledge. *Bioinformatics*, 38(19):4488–4496, 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac536. URL https://doi.org/10.1093/bioinformatics/btac536.

John H Morris, Karthik Soman, Rabia E Akbas, Xiaoyuan Zhou, Brett Smith, Elaine C Meng, Conrad C Huang, Gabriel Cerono, Gundolf Schenk, Angela Rizk-Jackson, Adil Harroud, Lauren Sanders, Sylvain V Costes, Krish Bharat, Arjun Chakraborty, Alexander R Pico, Taline Mardirossian, Michael Keiser, Alice Tang, Josef Hardi, Yongmei Shi, Mark Musen, Sharat Israni, Sui Huang, Peter W Rose, Charlotte A Nelson, and Sergio E Baranzini. The scalable precision medicine open knowledge engine (SPOKE): a massive knowledge graph of biomedical information. *Bioinformatics*, 39(2):btad080, 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad080. URL https://doi.org/10.1093/bioinformatics/btad080.

Chris Mungall, David Osumi-Sutherland, James A. Overton, Jim Balhoff, Clare72, pgaudet, Matthew Brush, Nico Matentzoglu, Vasundra Touré, Damion Dooley, Michael Sinclair, Anthony Bretaudeau, Scott Cain, Melissa Haendel, diatomsRcool, Jen Hammock, Marie-Angélique Laporte, Mark Jensen, and Martin Larralde. oborel/obo-relations: 2020-07-21, 2020. URL https://zenodo.org/records/3955125.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6341–6350. Curran Associates Inc., 2017. ISBN 978-1-5108-6096-4.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 809–816. Omnipress, 2011. ISBN 978-1-4503-0619-5.

Maria Parapouli, Anastasios Vasileiadis, Amalia-Sofia Afendra, and Efstathios Hatziloukas. Saccharomyces cerevisiae and its industrial applications. *AIMS Microbiology*, 6(1):1–31, 2020. ISSN 2471-1888. doi: 10.3934/microbiol.2020001. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7099199/.

Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Li, Luke Vilnis, and Andrew McCallum. Representing joint hierarchies with box embeddings. In *Automated Knowledge Base Construction (AKBC)*, 2020. doi: 10.24432/C5KS37.

Xi Peng, Zhenwei Tang, Maxat Kulmanov, Kexin Niu, and Robert Hoehndorf. Description logic EL++ embeddings with intersectional closure, 2022. URL http://arxiv.org/abs/2202.14018.

Katherine Roper, A. Abdel-Rehim, Sonya Hubbard, Martin Carpenter, Andrey Rzhetsky, Larisa Soldatova, and Ross D. King. Testing the reproducibility and robustness of the cancer biology literature by robot. *Journal of The Royal Society Interface*, 19(189): 20210821, 2022. doi: 10.1098/rsif.2021.0821. URL https://royalsocietypublishing.org/doi/10.1098/rsif.2021.0821. Publisher: Royal Society.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 593–607. Springer International Publishing, 2018. ISBN 978-3-319-93417-4. doi: 10.1007/978-3-319-93417-4_38.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences, 2017. URL http://arxiv.org/abs/1605.01713.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2071–2080. PMLR, 2016. ISSN: 1938-7228.

Cornelis Verduyn, Erik Postma, W. Alexander Scheffers, and Johannes P. Van Dijken. Effect of benzoic acid on metabolic fluxes in yeasts: A continuous-culture study on the regulation of respiration and alcoholic fermentation. *Yeast*, 8(7):501–517, 1992. ISSN 1097-0061. doi: 10.1002/yea.320080703. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/yea.320080703. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/yea.320080703.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1025.

Brian Walsh, Sameh K. Mohamed, and Vít Nováček. BioKG: A knowledge graph for relational learning on biological data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 3173–3180. Association for Computing Machinery, 2020. ISBN 978-1-4503-6859-9. doi: 10.1145/3340531.3412776. URL https://dl.acm.org/doi/10.1145/3340531.3412776.

Kevin Williams, Elizabeth Bilsland, Andrew Sparkes, Wayne Aubrey, Michael Young, Larisa N. Soldatova, Kurt De Grave, Jan Ramon, Michaela de Clare, Worachart Sirawaraporn, Stephen G. Oliver, and Ross D. King. Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases. *Journal of The Royal Society Interface*, 12(104):20141289, 2015. doi: 10.1098/rsif.2014.1289. URL https://royalsocietypublishing.org/doi/10.1098/rsif.2014.1289. Publisher: Royal Society.

Valerie Wood, Antonia Lock, Midori A. Harris, Kim Rutherford, Jürg Bähler, and Stephen G. Oliver. Hidden in plain sight: what remains to be discovered in the eukaryotic proteome? *Open Biology*, 9(2):180241, 2019. ISSN 2046-2441. doi: 10.1098/rsob.180241. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6395881/.

Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. Faithful embeddings for EL++ knowledge bases. In *The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings*, pages 22–38. Springer-Verlag, 2022. ISBN 978-3-031-19432-0. doi: 10.1007/978-3-031-19433-7_2.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5502–5511. PMLR, 10–15 Jul 2018.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Hui Yang, Jiaoyan Chen, and Uli Sattler. TransBox: EL++-closed ontology embedding. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pages 22–34. Association for Computing Machinery, 2025. ISBN 979-8-4007-1274-6. doi: 10.1145/3696410.3714672.

Zi Ye, Yogan Jaya Kumar, Goh Ong Sing, Fengyan Song, and Junsong Wang. A comprehensive survey of graph neural networks for knowledge graphs. 10:75729–75741, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3191784.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. 34(3):3065–3072, 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i03.5701.

Appendix A. Examples of description logic in KG

A description logic example of how a phenotype with a qualifier and chemical are specified in the KG. This example is about decreased (APO_0000003) utilisation of carbon source (APO_0000096) of lactate (CHEBI_16004), observed for the gene 'YBL030C'. The gene is linked to the phenotype by the has phenotype relation (RO_0002200) and to lactate with hasChemNutrientUtilization_Decreased.

$$\begin{aligned}
&\texttt{APO\_0000098-APO\_0000003-CHEBI\_16004} \sqsubseteq \texttt{APO\_0000098} \sqcap \texttt{APO\_0000003}\\
&\texttt{APO\_0000098-APO\_0000003-CHEBI\_16004} \sqsubseteq \exists\texttt{aboutChemical.CHEBI\_16004}\\
&\texttt{YBL030C} \sqsubseteq \exists\texttt{RO\_0002200.APO\_0000098-APO\_0000003-CHEBI\_16004}\\
&\texttt{YBL030C} \sqsubseteq \exists\texttt{hasChemNutrientUtilization\_Decreased.CHEBI\_16004}.
\end{aligned} \tag{26}$$

A description logic example of how a gene (YCR073C) is positively regulating the protein activity (INO_0000104) of another gene (YLR113W). This regulation happens during (RO_0002092) cellular response to heat (GO_0034605).

$$\begin{aligned}
&\texttt{YCR073C-YLR113W-protein\_activity-positive} \sqsubseteq \texttt{INO\_0000104}\\
&\texttt{YCR073C} \sqsubseteq \exists\texttt{positive\_regulator\_of.YCR073C-YLR113W-protein\_activity-positive}\\
&\texttt{positive\_regulator\_of.YCR073C-YLR113W-protein\_activity-positive}\\
&\qquad \sqsubseteq \exists\texttt{regulated\_gene.YLR113W}\\
&\texttt{positive\_regulator\_of.YCR073C-YLR113W-protein\_activity-positive}\\
&\qquad \sqsubseteq \exists\texttt{RO\_0002092.GO\_0034605}\\
&\texttt{YCR073C} \sqsubseteq \exists\texttt{positively\_regulating.YLR113W}.
\end{aligned} \tag{27}$$

Appendix B. Hyperparameters

B.1. Box embedding parameters

Table 4: Parameters used for the box embeddings of the different domains

Domain	Dimensions	Epochs	Lr	Regularisation	Gumbel temperature	Neg. ex. ratio
Material entity	10	1,000	1e-2	1e-3	0.25	2.0
Genes	8	1,000	1e-2	1e-3	0.25	4.0
Regulations	5	800	1e-2	1e-3	0.25	2.0
Molecular functions	5	800	1e-2	1e-3	0.25	2.0
Biological processes	5	800	1e-2	1e-3	0.25	2.0
Phenotypes	5	800	1e-2	1e-3	0.25	2.0
Reactions & Pathways	5	800	1e-2	1e-3	0.25	2.0
Cellular components	5	800	1e-2	1e-3	0.25	2.0

B.2. Prediction models

The best performing model was trained for 160 epochs, with a learning rate of 1e-4, and L2 regularisation weight of 0.1. The depth of the GNN was 2 and the embedding dimensions for the domains are listed in Table 5 and are the same throughout the GNN. The fully connected neural network predicting the interaction from the embeddings is of depth 2 with 64, and 1 neurons respectively. For models trained with the semantic loss in (23) we used $\alpha = 0.1$ and $\beta = 0.05$.

Table 5: Embedding dimensions for the different domains throughout the GNN.

Embedding dimensions	32	64	128
Domains	Cellular components Molecular functions Reactions Regulations	Biological processes Phenotypes	Material entities Genes

B.3. Box embeddings for training without prediction task

Table 6: The embedding models, trained with both $\mathcal{L}_{distance}$ and $\mathcal{L}_{overlap}$ used the same hyperparameters. The learning rate was after each epoch multiplied with (1 − Lr decay) and the regularisation used is presented in (15), penalising small boxes, with $l_0 = 1$. λ_s, λ, β, and γ refer to weights in the losses in (24) and (25).

Epochs	Initial lr	Lr decay	Regularisation, λ	Small Box Regularisation, λ_s	Negative weight, β	Negative weight, γ
500	0.1	0.001	0.001	0.01	0.5	1.0

Appendix C. Prediction model comparisons

Table 7: The architecture described in Section 5.3 with different levels of hierarchy integration. These results are also presented in Table 3. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
c	GNN without box embeddings	0.348	0.050
d	GNN with prior box embeddings	0.360	0.043
e	GNN with prior box embeddings + $\mathcal{L}_{overlap}$	0.368	0.038
f	GNN with prior box embeddings + $\mathcal{L}_{distance}$	**0.377**	0.046

Table 8: Baseline performance. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
a	ComplEx + LightGBM	0.191	0.039
b	Instantiations + LightGBM	0.211	0.022
g	ComplEx + MLP	0.165	0.032
h	Box^{2}EL + LightGBM	0.016	0.007

Table 9: Impact on predictive performance when ignoring rare edges in the graph. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
i	Ignoring edges with fewer than 100 examples	0.333	0.043
j	Ignoring edges with fewer than 500 examples	0.365	0.045
f	Ignoring edges with fewer than 1,000 examples	**0.377**	0.046
k	Ignoring edges with fewer than 5,000 examples	0.368	0.045

Table 10: Impact on predictive performance when combining gene embeddings using the methods presented in Table 2. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
f	Element-wise product	**0.377**	0.046
l	Bilinear	0.363	0.042
m	Concatenation	0.349	0.062
n	Intersection	0.347	0.039

Table 11: Impact on predictive performance when varying the dimensionalities of the embeddings of the model. Varying initial dimensions means the box embeddings in Table 4 are used. For the same initial dimensions, box embeddings in 10 dimensions are used, found with the hyperparameters in Table 4. Varying GNN dimensions refers to the parameters in Table 5. For the same GNN dimensions all embeddings are in 128 dimensions. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
f	Varying initial dimensions, varying GNN dimensions	**0.377**	0.046
o	Varying initial dimensions, same GNN dimensions (128)	0.361	0.046
p	Same initial dimension (10), same GNN dimensions (128)	0.372	0.053
q	Same initial dimensions (10), varying GNN dimensions	0.368	0.046

Table 12: Comparison of different model architectures. Model f refers to the architecture presented in Section 5.3, r uses a Transformer based GNN, and s uses the same architecture as f, but do not split the graph into distinct domains which are embedded separately. Results from 10-fold cross-validation of digenic deletion fitness.

	Description	Mean R^2	SD
f	GraphSAGE, NN prediction head, split domains	**0.377**	0.046
r	Transformer based GNN, NN prediction head, split domains	0.253	0.030
s	GraphSAGE, NN prediction head, one domain	0.128	0.184

Appendix D. Model-driven experiment

D.1. Edge filtering

Table 13: We filter for co-occurring edge pairs in which at least one edge connects a gene to a node that is a subclass of one of the following APO classes, related to nutrient utilisation.

APO Class	Description
`APO_0000096`	General nutrient utilisation
`APO_0000097`	Auxotrophy
`APO_0000099`	Utilisation of nitrogen source
`APO_0000100`	Nutrient uptake
`APO_0000125`	Utilisation of phosphorous source
`APO_0000219`	Utilisation of sulfur source

Table 14: We also allow edge pairs where at least one of the edges links a gene to a chemical through any of the following relations.

`hasChemNutrientUtilization`
`hasChemNutrientUtilization_Increased`
`hasChemNutrientUtilization_Decreased`

D.2. Top edge pairs

Table 15: The 10 edge pairs with the highest importance weight after filtering for the criteria specified in Appendix D.1. The edge pair selected for the experiment is highlighted. Ch.Nutr.Util. is short for 'hasChemNutrientUtilization', Ch.Nutr.Util.Dec. is short for hasChemNutrientUtilization_Decreased, and Ch.StressRes. is short for 'hasChemStressResistance'. Clarifications of terms can be found in Table 16.

Importance	Relation1	Class1	Relation2	Class2
0.003471	Ch.Nutr.Util.	CHEBI_17268	Ch.StressRes.	CHEBI_22907
0.002002	Ch.StressRes.	CHEBI_22907	has_phenotype	APO_0000099- APO_0000245- CHEBI_14321
0.001985	Ch.Nutr.Util.	CHEBI_17268	Ch.StressRes.	CHEBI_26710
0.001705	Ch.Nutr.Util.Dec.	CHEBI_23414	Ch.StressRes.	CHEBI_22907
0.001679	hasChemCellMorph	CHEBI_26710	has_phenotype	APO_0000099- APO_0000245- CHEBI_14321
0.001580	Ch.Nutr.Util.	CHEBI_17268	Ch.StressRes.	CHEBI_50145
0.001541	Ch.Nutr.Util.Dec.	CHEBI_77995	Ch.StressRes.	CHEBI_49470
0.001537	Ch.StressRes.	CHEBI_22907	has_phenotype	APO_0000099- APO_0000245- CHEBI_26271
0.001500	Ch.Nutr.Util.	CHEBI_17268	has_phenotype	APO_0000059- APO_0000002- CHEBI_26710
0.001477	Ch.Nutr.Util.Dec.	CHEBI_16236	Ch.StressRes.	CHEBI_22907

Table 16: Clarifications for terms in the Table 15.

Identifier	Label
CHEBI_17268	Myo-inositol
CHEBI_22907	Bleomycin
APO_0000099	Util. of nitrogen source
APO_0000245	Decreased
CHEBI_14321	Glutamate
CHEBI_26710	Sodium chloride
CHEBI_23414	Cpper sulfate
CHEBI_50145	Fenpropimorph
CHEBI_77995	Diphenyl, phenanthroline
CHEBI_26271	Proline
APO_0000059	Vacuolar morphology
APO_0000002	Abnormal
CHEBI_16236	Ethanol

D.3. Cultivation method

The Δ*ino1* deletion mutant was taken from the EUROSCARF deletion collection, with the strain background being BY4741, genotype: MATa, *his3*Δ1, *leu2*Δ0, *met15*Δ0, *ura3*Δ0 (Y01272).

The Δ*ino1* mutant was pre-cultured overnight in minimally buffered delft media containing the following: 5g/L (NH4)2SO4, 3g/L KH2PO4, 0.5g/L MGSO4. 7H2O, and 1mL/L trace metal and vitamin solutions as described by Verduyn et al. (1992), 25 mg/L myo-inositol and 2% glucose (w/v) in 30°C, and 220rpm. The pre-culture was adjusted to 0.5 OD600, and robotically dispensed with a 1:20 dilution into a 96-well microculture plate using a Hamilton Microlab Star liquid handling robot. A negative control was also included to assess the baseline growth of the Δ*ino1* mutant without any supplementation of myo-inositol. Additionally, myo-inositol-free media with 0.25% (w/v) glucose, myo-inositol (Sigma aldrich 57570-100G), Sodium chloride (Merck 1064041000) and MilliQ-water was robotically dispensed, resulting in a total volume of 250μL and the concentrations defined in Table 17.

Table 17: The concentrations of inositol and NaCl used for the experiment.

Inositol	NaCl
0.00 mMolar	0.0 Molar
0.01 mMolar	0.0 Molar
0.01 mMolar	0.3 Molar
0.01 mMolar	0.6 Molar
0.05 mMolar	0.0 Molar
0.05 mMolar	0.3 Molar
0.05 mMolar	0.6 Molar
0.25 mMolar	0.0 Molar
0.25 mMolar	0.3 Molar
0.25 mMolar	0.6 Molar

A robust plate layout was generated with PLAID (Francisco Rodríguez et al., 2023). The processed plate was cultivated in the automated laboratory cell Eve. The plate was transferred from an automated incubator (30°C) to a Teleshaker Magnetic Shaking System, where it was shaken for 30s at 800 rpm, divided evenly between clockwise and counter-clockwise double-orbital shaking. After shaking, the plate was transferred to a BMG Polarstar plate reader, where it underwent optical density measurements at 600 nM (the temperature in the plate reader was kept at a constant 30°C). After measuring, the plate was returned to the incubator. The protocol was automatically repeated every 20 min for up to 24 h.

D.4. Growth data processing and statistical testing

Outliers in the growth curves (measured through optical density at 600nm) were identified and filtered using the interquartile range (IQR), where any data points outside the range of [Q1-1.5 IQR, Q3+1.5 IQR] were excluded from the dataset. The filtered curves were then subsequently smoothed using a rolling mean of window size 3. The resulting averaged growth curves can be seen in Figure 9. Area under curve was calculated using `numpy.trapz` (`v1.26.4`). To assess the effects of inositol and NaCl on AUC, a generalised linear model was employed (`statsmodels v0.14.4`). The model was fitted using a Gaussian family distribution. Choice α-value was set at 0.05. We modelled all factors as categorical to avoid imposing any assumptions on linearity. The model is specified as follows:

$$\mathrm{AUC} \backsim C(Inositol) \times C(NaCl). \tag{28}$$

Table 18: Estimated parameters from the GLM examining the effects of myo-Inositol-supplementation and NaCl treatment on growth dynamics. The table presents coefficient estimates, p-values and confidence intervals for the main effects and interaction terms. Significant interactions indicate that the effect of myo-inositol supplementation changes depending on treatment levels. The two highlighted rows indicate the significant interaction effect.

	Coefficient ($\times 10^3$)	Confidence interval ($\times 10^3$)	p-value
Intercept	97.29	[88.20, 106]	0.000
Medium inositol	-11.84	[-24.7, 1.02]	0.071
High inositol	-14.56	[-27.9, -1.24]	**0.032**
Low NaCl	-9.50	[-22.4, 3.37]	0.148
High NaCl	-30.61	[-43.5, -17.8]	**0.000**
Medium inositol × Low NaCl	13.11	[-5.40, 31.6]	0.165
High inositol × Low NaCl	13.38	[-5.45, 32.2]	0.164
Medium inositol × High NaCl	20.92	[2.41, 39.4]	**0.027**
High inositol × High NaCl	22.64	[3.40, 41.9]	**0.021**

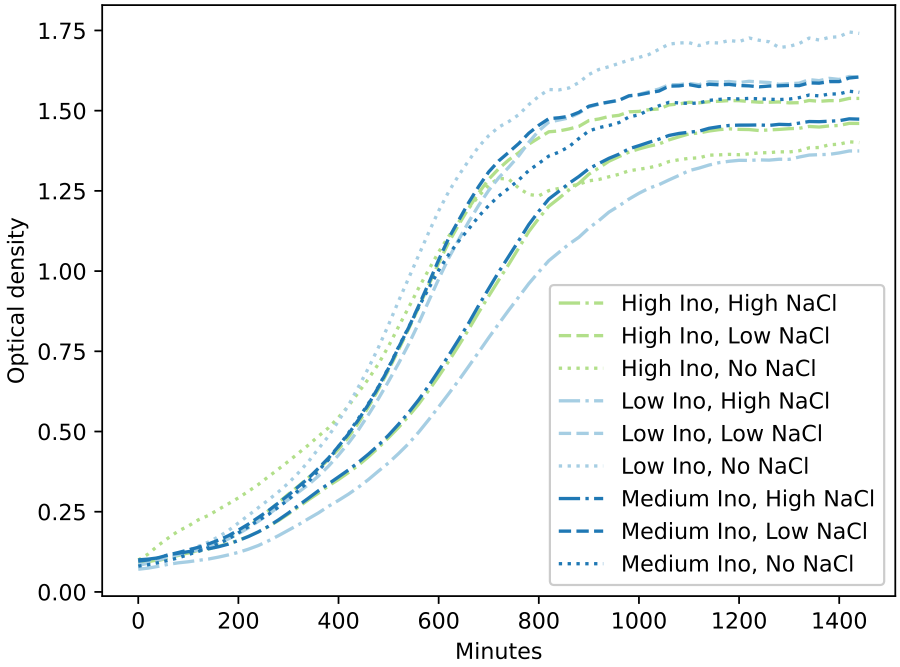


Figure 9: Growth curves showing the mean optical densities of the 6-8 repetitions for the different experimental groups. Optical density (at 600nM) is a unitless measurement typically used as an indirect measure of cell density and biomass.

Appendix E. Link evaluation results per edge type

Table 19: Link evaluation results per edge type.

Edge Type	#Test Edges	Mean Dist. (real)	Mean Dist. (constrained)	Mean Dist. (random)	M-W U (Real vs. Random)	p-value (Real vs. Random)	M-W U (Real vs. Constrained)	p-value (Real vs. Constrained)
BFO_0000050	500	2.578e-04	6.271e-04	5.339e-04	48417.5	2.790e-64	82420.0	8.367e-22
BFO_0000051	500	2.114e-04	1.941e-04	2.123e-04	129566.0	2.239e-01	125009.5	9.981e-01
BFO_0000066	154	3.423e-04	1.812e-03	3.121e-03	1318.0	2.676e-44	3515.5	3.007e-29
RO_0000057	268	9.756e-04	8.101e-04	1.568e-03	32214.0	3.917e-02	40732.5	7.176e-03
RO_0000087	500	1.487e-07	7.363e-08	5.085e-08	125250.0	5.641e-01	125250.0	5.641e-01
RO_0001025	500	1.882e-06	4.538e-05	2.369e-04	39615.0	9.627e-106	103855.0	5.874e-19
RO_0002092	500	3.008e-06	2.505e-03	1.581e-03	21128.5	8.319e-142	42882.0	5.331e-102
RO_0002200	500	1.465e-05	3.496e-04	7.656e-04	28532.5	2.766e-122	61920.0	3.883e-66
RO_0002211	500	6.292e-05	1.723e-04	9.614e-04	16370.5	8.205e-141	105081.5	6.095e-11
RO_0002212	500	6.503e-04	1.363e-03	5.717e-04	149994.0	2.525e-11	113514.0	6.415e-03
RO_0002213	500	8.264e-04	1.319e-03	6.458e-04	152742.5	3.195e-10	118608.0	1.861e-01
RO_0002233	500	1.520e-03	3.474e-03	5.530e-03	34860.0	3.657e-90	60270.5	1.124e-48
RO_0002234	500	1.755e-03	3.260e-03	4.422e-03	41317.0	9.300e-78	65275.5	1.096e-41
RO_0002326	123	1.689e-04	2.285e-04	4.176e-04	7129.0	2.712e-01	7034.5	1.902e-01
RO_0002327	500	6.910e-05	2.628e-04	8.388e-04	93425.0	1.358e-20	97765.0	3.737e-16
RO_0002331	500	4.410e-08	2.594e-06	3.656e-04	110971.5	3.377e-14	124999.5	1.000e+00
has_functional_parent	500	7.659e-05	1.070e-04	1.603e-04	107705.0	2.071e-10	119350.5	1.271e-02
has_parent_hydride	365	1.983e-04	6.620e-04	8.974e-04	26468.0	6.410e-49	42928.0	1.521e-19
is_conjugate_acid_of	500	6.061e-05	5.847e-05	9.118e-05	125726.0	6.824e-01	124774.0	9.897e-01
is_conjugate_base_of	500	9.977e-04	9.682e-04	1.105e-03	124591.0	9.287e-01	155845.0	1.416e-11
is_enantiomer_of	500	1.896e-03	1.450e-03	1.868e-03	136175.5	1.440e-02	158041.5	3.806e-14
is_substituent_group_from	258	7.831e-04	6.807e-04	1.012e-03	24170.0	7.363e-08	32937.5	8.388e-01
is_tautomer_of	389	2.202e-04	2.026e-04	2.099e-04	74080.5	5.379e-01	79678.5	8.296e-02
aboutChemical	500	1.244e-05	4.705e-05	1.061e-03	89850.0	5.328e-34	123477.5	2.087e-01
catalyzedBy	500	5.007e-04	8.114e-04	1.368e-03	70205.0	3.460e-33	82218.0	7.053e-21
catalyzedByGene	410	3.229e-04	5.789e-04	8.031e-04	54720.5	5.104e-18	51335.0	4.838e-22
encodedBy	500	9.142e-04	3.046e-04	8.949e-04	192462.0	2.198e-49	214997.0	1.863e-86
hasChemCellMorph	500	1.386e-05	5.905e-04	9.649e-04	22913.0	3.254e-137	50092.5	6.066e-89
hasChemNutrientUtilization	500	1.740e-05	8.846e-05	6.804e-04	89428.0	4.286e-34	119228.0	3.233e-04
hasChemNutrientUtilization_Decreased	293	1.588e-04	1.105e-03	1.455e-03	3409.0	4.242e-91	6908.0	6.862e-78
hasChemStressResistance	500	0.000e+00	0.000e+00	7.486e-06	123000.0	4.545e-03	125000.0	1.000e+00
hasChemStressResistance_Decreased	500	0.000e+00	0.000e+00	1.330e-06	123750.0	2.511e-02	125000.0	1.000e+00
hasChemStressResistance_Increased	500	2.258e-05	9.645e-04	1.439e-03	6243.0	3.008e-167	21798.5	1.434e-133
hasGeneProductPart	500	8.593e-04	9.468e-04	1.363e-03	108666.5	3.481e-04	117656.0	1.078e-01
hasRightParticipant	103	2.895e-04	1.905e-03	2.359e-03	1814.0	1.259e-16	3750.0	1.650e-04
negative_regulator_of	149	4.904e-04	1.686e-04	1.135e-03	4601.0	2.371e-18	19438.0	3.664e-29
negatively_regulating_transcription	218	3.719e-04	6.226e-04	5.868e-04	19703.0	2.035e-03	14560.0	2.661e-12
positive_regulator_of	252	5.027e-04	2.059e-04	1.089e-03	20142.0	1.233e-12	53510.0	2.045e-40
positively_regulating_transcription	376	2.299e-04	4.084e-04	5.691e-04	51171.0	5.630e-11	42113.0	8.380e-22
regulating_gene	500	9.037e-05	1.770e-04	8.146e-04	19748.5	1.030e-118	94922.0	1.715e-11
regulating_protein_activity	108	5.941e-04	6.883e-04	1.004e-03	6566.0	1.102e-01	5433.0	3.856e-01
regulating_transcription	500	3.737e-05	1.064e-04	8.277e-04	6146.0	9.529e-153	105635.0	5.863e-06
regulator_of	500	4.440e-04	1.253e-04	8.625e-04	122335.0	5.596e-01	223930.0	4.458e-104