# gGN: representing the Gene Ontology as low-rank Gaussian distributions

Alejandro A. Edera*[1], Georgina Stegmayer[1], and Diego H. Milone[1]

[1]Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET,Ciudad Universitaria UNL 3000, Santa Fe, Argentina

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

**Abstract**

Computational representations of knowledge graphs are critical for several tasks in bioinformatics, including large-scale graph analysis and gene function characterization. In this study, we introduce gGN, an unsupervised neural network for learning node representations as Gaussian distributions. Unlike prior efforts, where the covariance matrices of these distributions are simplified to diagonal, we propose representing them with a low-rank approximation. This representation not only maintains manageable learning complexity, allowing for scaling to large graphs, but is also more effective for modeling the structural features of knowledge graphs, such as their hierarchical and directional relationships between nodes. To learn the low-rank Gaussian distributions, we introduce a semantic-based loss function that effectively preserves these structural features. Systematic experiments reveal that gGN preserves structural features more effectively than existing approaches and scales efficiently on large knowledge graphs. Furthermore, applying gGN to represent the Gene Ontology, a widely used knowledge graph in bioinformatics, outperformed multiple baseline methods in ubiquitous gene characterization tasks. Altogether, the

*Corresponding author: aedera@sinc.unl.edu.ar

proposed low-rank Gaussian distributions not only effectively represent knowledge graphs but also open new avenues for enhancing bioinformatics tasks. gGN is publicly available as an easily installable package at `https://github.com/aedera/ggn`.

# 1 Introduction

Representing facts about the world as knowledge graphs, specifically directed acyclic graphs (DAGs), enables the computational manipulation of this information, accelerating automation across diverse fields [1, 2, 3, 4]. A notable instance is the Gene Ontology (GO), a widely-used knowledge graph in bioinformatics that has significantly accelerated our understanding of cell biology [5, 6, 7, 8]. Recently, the use of neural networks has advanced further manipulation of graphs [9, 10, 11, 12] by representing nodes as point vectors, effectively encoding node relationships through vector positions [13, 14, 15]. However, investigations into the representation potential of point vectors have highlighted limitations in adequately expressing typical structural features present in knowledge graphs such as hierarchical relationships [16, 17, 18, 19] and directional relationships between nodes [20, 21, 22, 23].

An effective alternative to address these limitations involves representing nodes as Gaussian distributions [24]. In this approach, nodes are represented by the distribution means, while relationships between nodes are encoded by intersection patterns between the ellipsoidal regions defined by the covariance matrices of the distributions [23]. This alternative representation has shown to be effective to encode graphs accurately, as illustrated by Graph2Gauss [25]. However, to reduce learning time and data requirements, the covariance matrices employed by these Gaussian representations are generally defined as simple diagonal matrices [24] at the cost of overly limiting the representational power of the Gaussian distributions [26]. For example, Graph2Gauss and analogous methods employ diagonal matrices to handle large-scale graphs efficiently [27, 28]. However, the representational cost of using this covariance simplification remains unclear. Some studies suggest that modeling intricate node dependencies with Gaussian distributions may require covariance matrices with greater representational power than diagonal ones can provide [29, 30]. Despite their representational benefits, the use of non-diagonal covariance matrices are underexplored in knowledge graph representation, likely due to the challenges associated with scaling these matrices to the size of large

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

graphs.

We hypothesize that utilizing Gaussian distributions with enhanced representation power can yield more accurate representations of knowledge graphs. To investigate this, we propose gGN, a neural network designed to represent graphs using low-rank factorized Gaussian distributions. This approach employs low-rank factorization to represent covariance matrices, offering greater representation power than the diagonal approximation, while substantially reducing the number of parameters compared to arbitrary covariance matrices, thus allowing for scaling to large graphs. Additionally, we drew inspiration from seminal works on semantic similarity analysis [31, 32] to design a loss function to train gGN. Experiments on synthetic and real-world knowledge graphs demonstrate that, compared to their diagonal and point vector counterparts, the low-rank Gaussian distributions are more effective representations, better preserving hierarchical and directional relationships between nodes. Furthermore, applying gGN to multiple gene characterization tasks using the Gene Ontology outperformed diverse methods commonly used for those tasks. gGN is publicly available as an easily installable package at `https://github.com/aedera/ggn`.

# 2 Material and methods

## 2.1 Low-rank Gaussian distributions

A graph can be represented as a set of Gaussian distributions [24]. Each node $i = 1, \ldots, n$ is modeled by a $d$-dimensional Gaussian distribution $\mathcal{N}_i = \mathcal{N}(\mu_i, \Sigma_i)$, where $\mu_i \in \mathbb{R}^d$ is the mean and $\Sigma_i \in \mathbb{R}^{d \times d}$ is the covariance matrix. Commonly, $\Sigma_i = D_i \in \mathbb{R}^d$ is assumed, where $D_i$ is a diagonal matrix. This diagonal approximation reduces space complexity ($\mathbb{R}^{d \times d}$ to $\mathbb{R}^d$) and consequently learning time and training data volume. Instead, we propose to represent $\Sigma_i$ using a low-rank factorization

$$\Sigma_i = D_i + P_i P_i^T, \tag{1}$$

where $P_i$ is a covariance factor represented by a real-valued matrix with dimensions $\mathbb{R}^{d \times r}$, where $r$ is known as the matrix rank, and $r \ll d$.

Our proposal balances the simplicity of a diagonal matrix with the expressiveness of a full matrix. The low-rank factorization is much sparser



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

than a full square matrix ($d + r \times d \ll d \times d$), promoting efficient learning, while it incorporates more parameters compared to the diagonal counterpart ($d + r \times d > d$), enhancing representation power. The rank $r$ can be adjusted to trade off between learning complexity and representation power. When $r = 0$, the low-rank factorization reduces to a diagonal matrix. As the rank increases, the number of parameters grows, yielding richer representations at the cost of increasing learning complexity. We will empirically demonstrate that a rank slightly greater than zero substantially outperforms the diagonal approximation in representing knowledge graphs, while maintaining affordable learning complexity.

We propose to use a neural network named gGN to model the low-rank Gaussian distributions. This network, denoted $f(\cdot)$, projects a given node $i$ into the space $\mathbb{R}^{d(r+2)}$. That is,

$$f(i) = (\mu_i, D_i, P_i), \tag{2}$$

thus defining the parameters of a low-rank Gaussian distribution $\mathcal{N}_i = \mathcal{N}(f(i))$, where $D_i$ and $P_i$ determines $\Sigma_i$ as shown in Eq. 1.

## 2.2 Learning low-rank Gaussian distributions

To represent nodes as low-rank Gaussian distributions, we propose the workflow depicted in Figure 1. Starting with a given graph, a batch $B$ of nodes is randomly selected and given as input to gGN to obtain low-rank Gaussian distributions. The parameters of gGN are learned in an unsupervised manner by minimizing a loss function $\mathcal{L}$, which is introduced in the following section, that preserves the shortest path lengths ($S$) of the input graph. Recall that a path from node $i$ to $j$ is a sequence of nodes, such that consecutive nodes in this sequence are linked by (directed) edges and the first and last elements in the sequence correspond to $i$ and $j$, respectively. The number of edges in the sequence is the length of the path.

We employed the Floyd-Warshall algorithm to calculate the shortest path lengths, which are represented by a sparse matrix $S \in \mathbb{R}_+^{n \times n}$, where $S_{ij}$ denotes the shortest path length from node $i$ to node $j$, or $\infty$ if nodes are unreachable. The Floyd-Warshall algorithm is performed only once per graph resulting in a manageable time complexity of $O(n^3)$, which can be further reduced with GPU implementations [33, 34] and diverse approximations [35, 36].



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

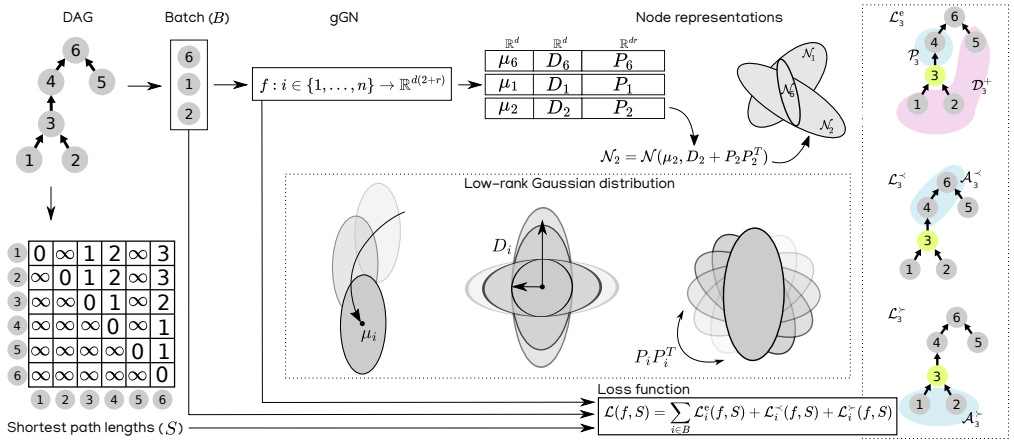


Figure 1: The gGN model workflow for representing nodes as low-rank Gaussian distributions, along with a schematic illustration of a low-rank Gaussian distribution and the loss function.

## 2.3 Loss function

To learn the low-rank Gaussian distributions, we defined the following loss function

$$\mathcal{L}(f, S, E) = \sum_{i \in B} \mathcal{L}_i^{\mathrm{e}}(f, S, E) + \mathcal{L}_i^{\prec}(f, S, E) + \mathcal{L}_i^{\succ}(f, S, E), \tag{3}$$

where $f$ is the neural network gGN, $S$ is the sparse matrix representing the input graph and $E$ is an energy function.

The loss function $\mathcal{L}$ comprises three loss components, each of which measuring how accurately the Gaussian distribution $\mathcal{N}_i$ generated by $f$ represents the node $i \in B$ as described by $S$. To evaluate the accuracy of the representation, each component uses the energy function $E$ to assess whether $\mathcal{N}_i$ encodes specific hierarchical and directional relationships between $i$ and the rest of the nodes. In the first component, $\mathcal{L}_i^{\mathrm{e}}$, the relationships between node $i$ and its immediate neighbors are analyzed. In contrast, the second component, $\mathcal{L}_i^{\prec}$, evaluates the relationships between node $i$ and its ancestors. Conversely, the final component, $\mathcal{L}_i^{\succ}$, measures the relationships between node $i$ and its descendants. More details of each component are provided in A.4.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

## 2.4 Energy funcion: Kullback-Leibler divergence

The energy function $E$ is used to measure how similar two Gaussian representations are (the lower, the more similar). The idea is that if nodes $i$ and $j$ are connected, their Gaussian representations $\mathcal{N}_i$ and $\mathcal{N}_j$ should have a relatively lower energy, $E_{ij} = E(\mathcal{N}_i, \mathcal{N}_j)$, than the energy between the Gaussian representations of unconnected nodes. We propose to define $E_{ij}$ as the Kullback-Leibler (KL) divergence: $E_{ij} = \mathrm{KL}(\mathcal{N}_j \| \mathcal{N}_i)$ (details in A).

The choice of the KL divergence as the energy function offers a number of benefits for representing knowledge graphs. The KL divergence generalizes the Pythagoras' theorem for square distances [37], allowing asymmetric energies (i.e., $E_{ij} \neq E_{ji}$). This generalization is relevant for modeling directed edges [23], as edge directionality can be interpreted as an asymmetric distance/energy between two nodes [38]. Additionally, the KL divergence can be calculated analytically, avoiding computationally intensive Monte Carlo approximations [39]. Although the cost of this analytical calculation is not negligible due to the involvement of determinant and matrix inversion operations, it can be substantially reduced by applying the matrix determinant lemma [40] and the Woodbury matrix identity [41]. As demonstrated in A, the cost of the determinant and matrix inversion operations boil down to $O(d + r^3)$ and $O(r^3)$, respectively. In addition, further computational savings can be obtained by leveraging redundant computations (details in A).

## 2.5 Knowledge graphs

We used the GO (release 2020-10-06) as a case study due to its instrumental role in bioinformatics [19]. This ontology is composed of three DAGs [19], each describing different areas of cell biology: BP (Biological Process), CC (Cellular Component) and MF (Molecular Function). Table 4 shows key structural features of each DAG in the GO. Following common practices, we constructed the DAGs using the *is-a* relationship. Additionally, we also used synthetic DAGs generated by the random-dag-generator-go package (`https://github.com/laser/random-dag-generator-go`).

## 2.6 Predicting protein-protein interactions

To assess the capacity of our proposed representation to predict protein-protein interactions (PPIs), we employed the best-match average approach [42] on a



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

previously published dataset [14] containing interacting and non-interacting pairs of proteins annotated with GO terms. Given two proteins, $A$ and $B$, the best-match average was used as a predictive indicator of their interaction:

$$BMA(A,B) = \frac{1}{2|A|}\sum_{a\in A}\max_{b\in B} sim(a,b) + \tag{4}$$

$$\frac{1}{2|B|}\sum_{b\in B}\max_{a\in A} sim(a,b), \tag{5}$$

where $a \in A$ and $b \in B$ denote GO terms annotating each protein, respectively, and $sim$ is a similarity measure. For Gaussian representations, we used the KL divergence as the similarity measure, while the cosine similarity was used for point-vector representations. The best-match average takes values from zero to one, indicating non-interaction and interaction, respectively.

To evaluate the predictive performance, we used the receiver operating characteristic (ROC) curve, in which the true positive rate is compared against the false positive rate at various thresholds. To compare the performance between different predictive methods, we calculated the areas under their ROC curves (AUCs).

# 3 Results

## 3.1 A case study on synthetic graphs

To clearly illustrate the benefits of using low-rank Gaussian distributions, we employed them to represent four synthetic DAGs as 2-dimensional ($d = 2$) Gaussian distributions with rank 2 ($r = 2$), using diagonal counterparts ($r = 0$) as controls. In Figure 2, the resulting node representations are depicted as ellipses showing points within one standard deviation from their corresponding means. The ellipses of the low-rank distributions show encapsulation patterns that capture the hierarchical and directional relationships between nodes. For example, the encapsulation patterns in Figures 2A and B show that ellipses associated with parent nodes are within those corresponding to their children, which is not clearly observed when using diagonal covariance matrices. Moreover, Figures 2C and D illustrates how this encapsulation pattern relies on edge directionality, as the encapsulation of the root inside the leaves (Figure 2C) is disrupted when the directionality of edges is flipped (Figure 2D).

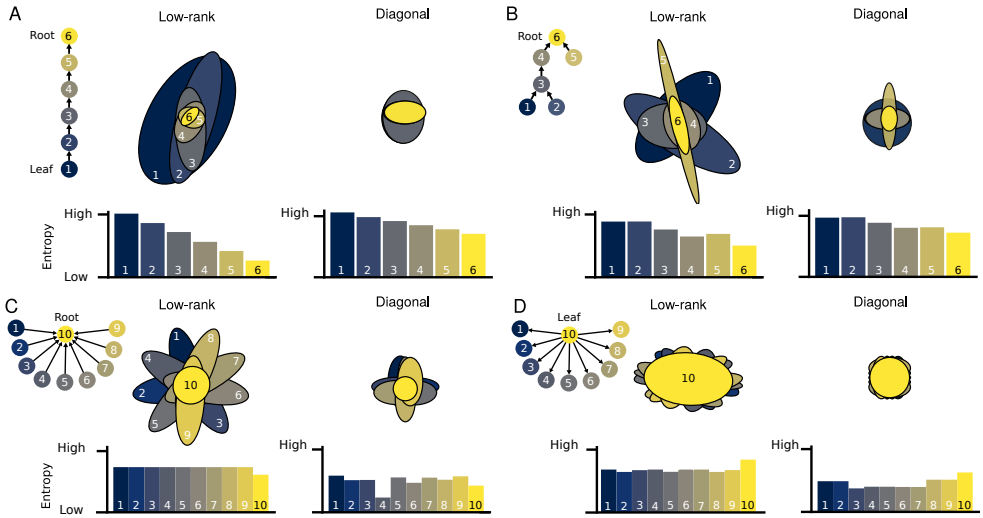


Figure 2: Gaussian representations. The nodes of four synthetic graphs (A-D) are represented by low-rank or diagonal Gaussian distributions. Ellipses depict points within one standard deviation from their corresponding means. The encapsulation patterns between ellipses represent the hierarchical and directional relationships between their associated nodes. Bar plots (bottom) depict the entropy associated with each node representation. Entropy reflects the expected information content of a Gaussian distribution.

These results illustrate the benefits of using low-rank covariance matrices for representing knowledge graphs.

To quantify the observed encapsulation patterns, we calculated the entropies of the distributions, as entropy reflects the expected information content of a random variable [43]. As shown in the bar plots in Figure 2, the entropy of child nodes tends to be higher than those of their parents. Since leaf nodes are associated with concrete concepts while those close to the root with abstract concepts in knowledge graphs [5], this finding indicates that Gaussian distributions are representing the information content of nodes similarly to well-known semantic similarity measures [44]. Specifically, concrete concepts (i.e., leaves) are associated with representation with high information content while abstract concepts (i.e., roots) display the opposite pattern [45, 46, 31, 32]. This is exemplified in Figure 2, particularly in panel D where all the roots have associated lower entropy compared to the leaf. This codification of the nodes' information content is due to the use of the

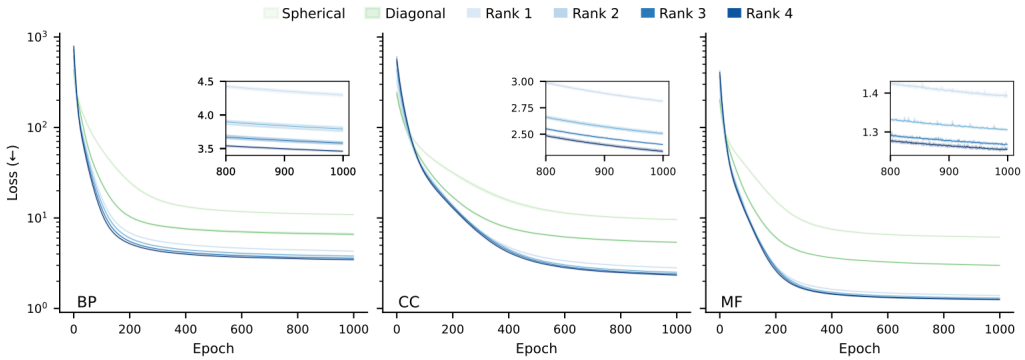


Figure 3: Loss curves from learning Gaussian representations with different types of covariance matrices. The shaded regions indicate dispersion across different seeds.

reverse KL divergence in the loss function, as using the forward version leads to codifying the information content in the opposite manner [24, 25, 23].

## 3.2 Stably learning of low-rank node representations on large-scale knowledge graphs

To understand whether the low-rank representations can be applied on large-scale knowledge graphs, we analyzed the stability of learning 10-dimensional, low-rank representations by monitoring the loss values yielded on the GO (B). As controls, we also included 10-dimensional Gaussian distributions parameterized by either diagonal covariance matrices and spherical ones (identical values on the main diagonal).

The results revealed that learning was stable even when the rank $r$ varied from 1 to 4 (Figure 3). Compared to the spherical and diagonal approximations (green), the low-rank representations (blue) achieved better (lower) loss, indicating that the underlying structure of the data is better approximated by low-rank covariance matrices. Low ranks displayed almost similar loss values compared to high ones while maintaining relatively reduced learning complexity (Figure 4A). A similar trend was observed with synthetic random graphs (Figure 4B), where lower rank representations also required less GPU memory (Figure 9). This suggests that low ranks offer sufficient representational power at relatively low computational cost for representing large-scale knowledge graphs, such as those included in the GO.

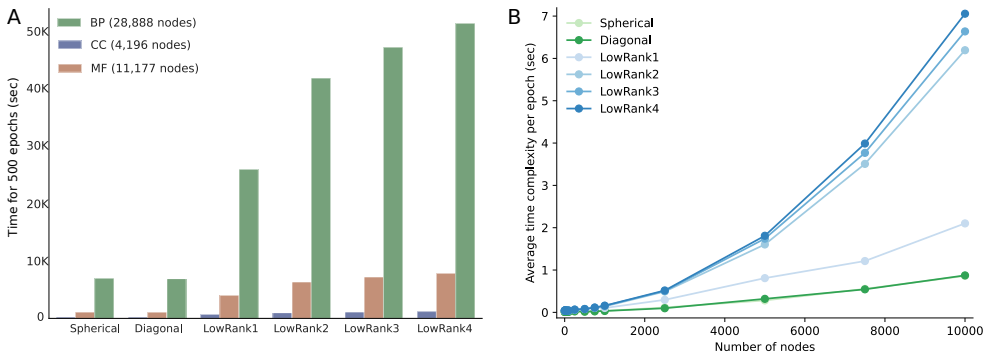


Figure 4: Time complexity from learning Gaussian representations with different types of covariance matrices. A) Time complexity for the BP, CC and MF. B) Time complexity for two-dimensional representations learned from synthetic random DAGs with increasing number of nodes.

To understand the impact of node dimensionality $d$ on learning stability, we additionally learned rank-1 node representation for dimensionalities ranging from 10 to 50. This revealed that the learning process was still stable and scaled for higher dimensions (Figure 5). Increasing dimensionality ($d > 10$) just resulted in representations yielding relatively marginal loss gains, indicating that the GO structure can be accurately represented with 10 dimensions. Together, these results indicate that gGN can stably learn node representations, and its low-rank approximation scales in large knowledge graphs.

## 3.3 Low-rank approximation better preserves directional and hierarchical information

To evaluate the ability of the low-rank Gaussian distributions to represent knowledge graphs, we used them to visualize the BP sub-ontology, the largest and most structurally complex knowledge graph evaluated (Table 4). Compared to their spherical and diagonal counterparts, the low-rank representations exhibited a more complex spatial arrangement (Figure 6A). To determine whether this spatial arrangement was associated with the directional information of BP, we first assessed how effectively the low-rank representations preserved the shortest path lengths between nodes, an important feature governed by edge directions that reflects the underlying geometry of graphs [47].

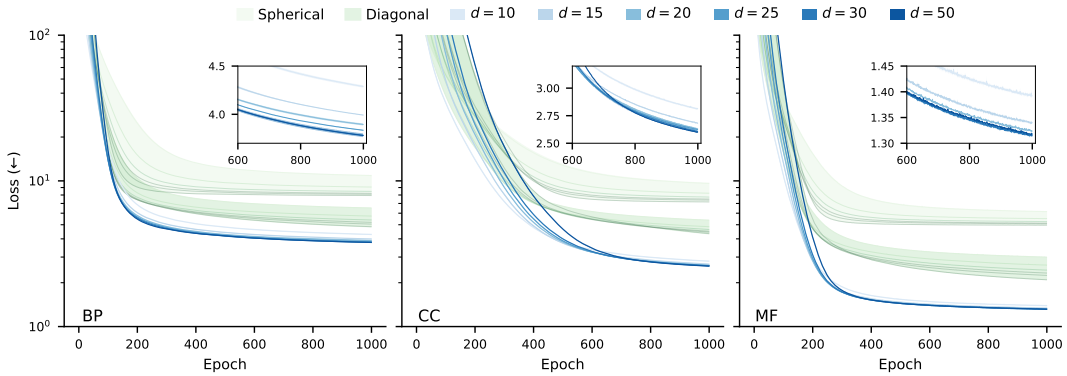


Figure 5: Loss curves from learning representations with increasing dimension $d$. The shaded regions indicate dispersion across different seeds.

By comparing the correlation between the similarities of the Gaussian distributions and the shortest path lengths of the corresponding nodes (C), we found that the low-rank representations exhibited a stronger positive correlation compared to the spherical and diagonal representations (Figure 6B). Additionally, we evaluated Graph2Gauss, which is analogous to our diagonal representations but uses a different loss function [25]. The results indicated that the low-rank distributions outperformed Graph2Gauss, highlighting the representational benefits of gGN. Since the calculation of shortest path lengths depend on edge directions, we found that, unlike spherical and diagonal distributions, the low-rank representations strongly correlated not only with the path lengths derived from the original edge directions ($\succ$) but also with the lengths obtained when flipping the edge directions ($\prec$) (Figure 6C). This is a remarkable result demonstrating the ability of our approach to preserve directional information.

By performing a similar analysis on the other two sub-ontologies (CC and MF), we found that the low-rank representations consistently outperformed both spherical and diagonal counterparts in preserving shortest path lengths in both directions (Table 1). We also observed that the Gaussian representations were more effective than diverse point-vector representations (E). Although previous research has shown that these representations are able to capture shortest path lengths [48], a suitable training using semi-supervised approaches is required. The spherical representations outperformed the diagonal Gaussian distributions in preserving shortest path lengths in the case of direction $\succ$ in

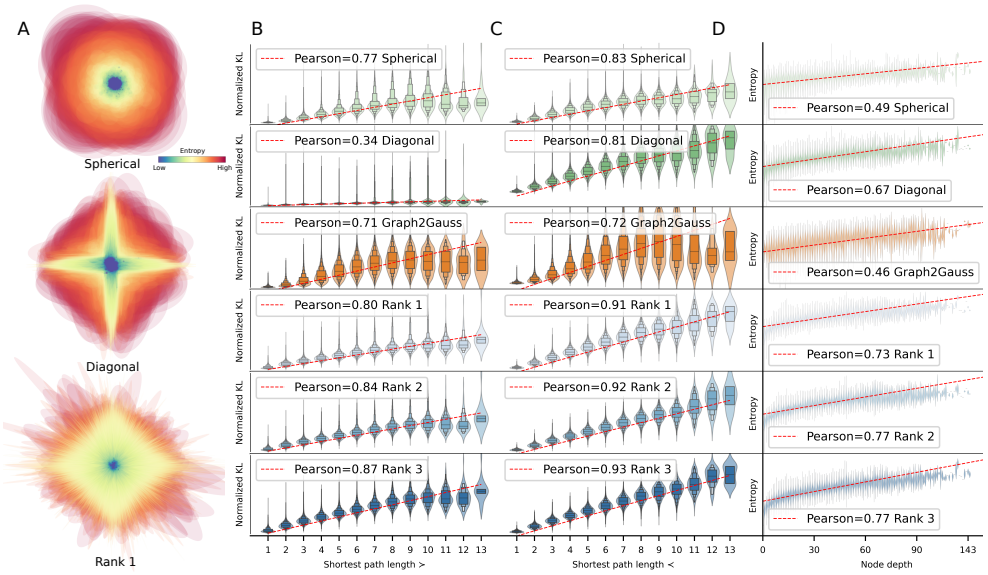


Figure 6: Correlations between KL divergence and structural features captured by Gaussian representations on biological processes (BP). A) Visualization of BP using two-dimensional Gaussian representations obtained by three different methods: spherical, diagonal and low rank. B-D) Violins depict KL divergences (or entropy), y-axis, calculated between 10-dimensional representations of nodes at a given short path length (or node depth), x-axis.

BP. Further experiments indicated that, unlike the spherical and low-rank representations, the diagonal Gaussian distributions are more sensitive to the number of parent nodes (F).

The visualization of BP also showed that Gaussian distributions with low entropy were clusterized on the center and encapsulated by distributions with higher entropy (Figure 6A). To determine whether this encapsulation pattern was associated with the hierarchical information of BP, we analyzed whether the entropy of Gaussian distributions correlated with the depths of their corresponding nodes (C). The results revealed that, unlike all the alternative Gaussian representations evaluated, the low-rank representations showed the highest positive correlations with node depth in BP (Figure 6D). The correlations were also larger than those obtained by Graph2Gauss, indicating that low-rank covariances are better suited than diagonal ones for representing hierarchical information. Additionally, the entropies of our di-

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

Table 1: Preserving graph features using representations with $d = 10$. The best correlation is boldfaced and the second best is underlined.

| | | Pearson correlation (↑) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Shortest path length $\succ$ | | | Shortest path length $\prec$ | | | Node depth | | |
| | Model | BP | CC | MF | BP | CC | MF | BP | CC | MF |
| Point vector | AROPE | -0.09±0.00 | -0.08±0.00 | -0.09±0.00 | -0.31±0.00 | -0.36±0.00 | -0.33±0.00 | 0.00±0.00 | -0.04±0.01 | 0.00±0.00 |
| | GraRep | -0.01±0.00 | -0.03±0.00 | -0.01±0.00 | -0.01±0.01 | 0.05±0.01 | -0.05±0.00 | 0.00±0.01 | -0.02±0.00 | 0.00±0.00 |
| | SVD_anc | -0.07±0.00 | -0.07±0.00 | -0.01±0.00 | -0.24±0.00 | -0.13±0.00 | -0.35±0.00 | 0.00±0.01 | 0.00±0.01 | -0.01±0.00 |
| | SVD_des | -0.05±0.00 | -0.02±0.01 | -0.02±0.00 | -0.20±0.00 | -0.11±0.00 | -0.12±0.00 | 0.00±0.00 | 0.00±0.02 | -0.01±0.01 |
| | DeepWalk | 0.00±0.00 | -0.01±0.00 | 0.00±0.00 | 0.00±0.00 | 0.05±0.00 | -0.02±0.00 | 0.00±0.00 | 0.00±0.02 | 0.00±0.00 |
| | LINE | -0.01±0.00 | -0.04±0.00 | -0.05±0.00 | -0.05±0.01 | 0.02±0.01 | -0.23±0.00 | 0.00±0.00 | 0.01±0.02 | 0.01±0.01 |
| | node2vec | -0.04±0.00 | -0.03±0.00 | -0.02±0.00 | -0.22±0.00 | -0.13±0.00 | -0.13±0.01 | 0.00±0.00 | 0.01±0.02 | 0.01±0.01 |
| | VERSE | -0.04±0.00 | -0.03±0.00 | -0.03±0.00 | -0.24±0.00 | -0.12±0.00 | -0.24±0.00 | 0.00±0.00 | 0.00±0.01 | 0.00±0.01 |
| | onto2vec | -0.05±0.00 | -0.04±0.01 | -0.04±0.00 | -0.22±0.00 | -0.09±0.00 | -0.22±0.01 | 0.00±0.01 | -0.01±0.01 | -0.01±0.01 |
| | anc2vec | -0.04±0.00 | -0.04±0.00 | -0.05±0.00 | -0.21±0.00 | -0.07±0.00 | -0.33±0.00 | 0.00±0.00 | 0.00±0.01 | 0.00±0.01 |
| | neig2vec | -0.03±0.00 | -0.02±0.01 | -0.02±0.01 | -0.30±0.00 | -0.11±0.00 | -0.24±0.00 | 0.00±0.01 | 0.01±0.03 | 0.01±0.01 |
| Gaussian | Spherical | 0.77±0.00 | 0.77±0.10 | 0.68±0.01 | 0.83±0.06 | 0.72±0.08 | 0.84±0.08 | 0.48±0.06 | 0.07±0.08 | -0.2±0.06 |
| | Diagonal | 0.34±0.09 | 0.73±0.06 | 0.85±0.01 | 0.81±0.03 | 0.65±0.19 | 0.83±0.01 | 0.66±0.01 | 0.58±0.01 | 0.58±0.03 |
| | Rank 1 | 0.80±0.02 | 0.71±0.05 | 0.87±0.01 | 0.91±0.00 | 0.90±0.00 | 0.95±0.00 | 0.74±0.01 | 0.72±0.00 | 0.68±0.01 |
| | Rank 2 | 0.84±0.01 | 0.78±0.01 | 0.88±0.01 | 0.92±0.00 | 0.92±0.00 | 0.94±0.00 | **0.76±0.01** | 0.76±0.01 | **0.72±0.01** |
| | Rank 3 | **0.87±0.01** | **0.81±0.01** | **0.91±0.01** | 0.93±0.00 | 0.93±0.00 | 0.95±0.00 | **0.76±0.01** | 0.76±0.01 | 0.69±0.01 |
| | Rank 4 | **0.87±0.01** | **0.80±0.01** | **0.91±0.00** | **0.94±0.00** | **0.94±0.00** | **0.96±0.00** | 0.75±0.01 | **0.78±0.02** | 0.66±0.02 |
| | G2G | 0.71±0.01 | 0.69±0.00 | 0.75±0.02 | 0.73±0.01 | 0.72±0.01 | 0.79±0.01 | 0.43±0.03 | 0.27±0.07 | 0.42±0.06 |

agonal representations were more correlated with node depth compared to those of Graph2Gauss, suggesting that the proposed loss function is more effective in capturing hierarchical information. Similar results were observed on the other two knowledge graphs (Table 1), highlighting the benefits of using low-rank covariances for representing the hierarchical information of knowledge graphs.

To investigate whether the superior representational power of low-rank Gaussian distributions was independent of the number of parameters used for representing nodes, we devised the following experiment. We compared the performance of rank-1 representations against higher-dimensional diagonal ones, ensuring both types used the same number of parameters for representing nodes. For instance, both 30-dimensional diagonal representations and 15-dimensional rank-1 representations result in 30 parameters per node. The results showed that, although both representations performed similarly on preserving directional information, the low-rank approximation consistently outperformed the diagonal one in preserving node depths (Table 2), even when the diagonal representations used the double of parameters for representing nodes. We confirmed that this trend persists with higher-order rank representations (Table 5), highlighting that the superior performance is driven by the use of ranks greater than zero. Altogether, these findings indicate that the superior representational power of low-rank Gaussian distributions is independent of the number of parameters, supporting that its greater



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

representation power comes from its capacity to better model knowledge graphs.

Table 2: Diagonal and rank-1 representations using the same number of parameters per node. Significant correlations are boldfaced.

| | | Pearson correlation (↑) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30 params per node | | 40 params per node | | 50 params per node | |
| Feature | Graph | Diag ($d$=30) | Rank1 ($d$=15) | Diag ($d$=40) | Rank1 ($d$=20) | Diag ($d$=50) | Rank1 ($d$=25) |
| Path length $\succ$ | BP | **0.90±0.01** | 0.88±0.01 | 0.91±0.01 | 0.90±0.01 | 0.91±0.01 | 0.90±0.01 |
| | CC | **0.85±0.01** | 0.79±0.02 | **0.87±0.01** | 0.81±0.02 | **0.86±0.02** | 0.81±0.03 |
| | MF | 0.88±0.01 | 0.87±0.01 | 0.88±0.01 | 0.87±0.01 | **0.89±0.01** | 0.86±0.01 |
| Path length $\prec$ | BP | 0.93±0.01 | 0.93±0.01 | 0.93±0.01 | 0.93±0.01 | 0.93±0.01 | 0.94±0.01 |
| | CC | 0.90±0.01 | 0.90±0.02 | 0.89±0.01 | 0.91±0.02 | 0.91±0.02 | 0.91±0.02 |
| | MF | 0.93±0.01 | **0.96±0.01** | 0.93±0.01 | **0.96±0.01** | 0.93±0.01 | **0.96±0.01** |
| Node depth | BP | 0.66±0.02 | **0.73±0.03** | 0.63±0.02 | **0.71±0.01** | 0.59±0.02 | **0.70±0.02** |
| | CC | 0.63±0.02 | **0.70±0.02** | 0.64±0.01 | **0.69±0.01** | 0.62±0.02 | **0.69±0.03** |
| | MF | 0.62±0.05 | **0.67±0.05** | 0.58±0.02 | **0.68±0.01** | 0.63±0.02 | **0.68±0.02** |

## 3.4 Ablation study on the loss function

To understand the contribution of each loss component (Eq. 3) in preserving the hierarchical and directional relationships between nodes, we conducted an analysis by learning low-rank representations after ablating each loss component at a time. These "ablated" representations were then compared to those obtained without ablation. If a loss component significantly contributes to the preservation of one graph feature, the unablated representations should outperform their ablated equivalents in preserving that feature.

The results of this ablation analysis are shown in Figure 7. Here, the dots depict the Pearson correlation coefficients that indicate how well the unablated (x-axis) and ablated (y-axis) representations preserved the shortest paths (A-F) or the node depths (G-I). Dots falling below the main diagonal indicate cases where the unablated representations performed better (higher correlation) than the ablated ones. These results show that the ablation of loss components yielded node representations unable to properly preserve the graph features. When the loss component $\mathcal{L}^{\prec}$ was ablated (Figure 7A-C), the resulting node representations showed worse correlations compared to their unablated counterparts (Figure 7C). This indicates that the ablation impaired the proper preservation of the shortest path length for the original ($\prec$) edge directions. Similarly, ablation of $\mathcal{L}^{\succ}$ (Figure 7D-F) led to worse correlations for the shortest path length for the reverse ($\succ$) edge directions

(panel E). Thus, this result indicates that the component $\mathcal{L}^{\succ}$ is essential for properly preserving the shortest path lengths in the $\succ$ direction. Likewise, the ablation of the loss component $\mathcal{L}^{e}$ resulted in lower correlations (Figure 7G-I), particularly producing node representations with weaker correlations against node depth information compared to their unablated counterparts (Figure 7G). This shows that node depth information is not well preserved when the loss component $\mathcal{L}^{e}$ is ablated. The loss component $\mathcal{L}^{e}$ impacted the three evaluated features, suggesting that the preservation of immediate node relationships is essential to preserve higher-order relationships, such as those defined by the shortest path lengths between nodes. Overall, this ablation analysis highlights the efficacy of the proposed loss function in preserving the shortest path lengths and node depths of knowledge graphs.

## 3.5 Node classification

Functional profiling of genes relies on semantic similarity computations based on structural features extracted from the GO [49, 50, 51, 52, 53, 54]. To evaluate whether the node representations learned by gGN are effective in representing the GO, we used them for node classification (D), using as controls 17 representative baseline methods typically used for extracting features from graphs (E).

The results showed that the best classifications were obtained by the low-rank representations, displaying the lowest Davies-Bouldin indexes compared to those obtained by the other methods (Table 3). This indicates that the low-rank representations are better for representing the complex structure of the Gene Ontology.

## 3.6 Predicting protein-protein interactions

The semantic information of the GO has been demonstrated to be effective in predicting protein-protein interactions (PPIs) [13], which is a fundamental task for gene characterization [55]. This effectiveness arises from the fact that interacting proteins are more likely to be found in similar cellular compartments or biological processes. Consequently, GO terms annotating these proteins exhibit higher similarity compared to those annotating non-interacting proteins, providing a predictive indicator of their potential interactions. Therefore, we used the task of predicting PPIs from GO terms as a valuable setting to evaluate the benefits of our low-rank representation.

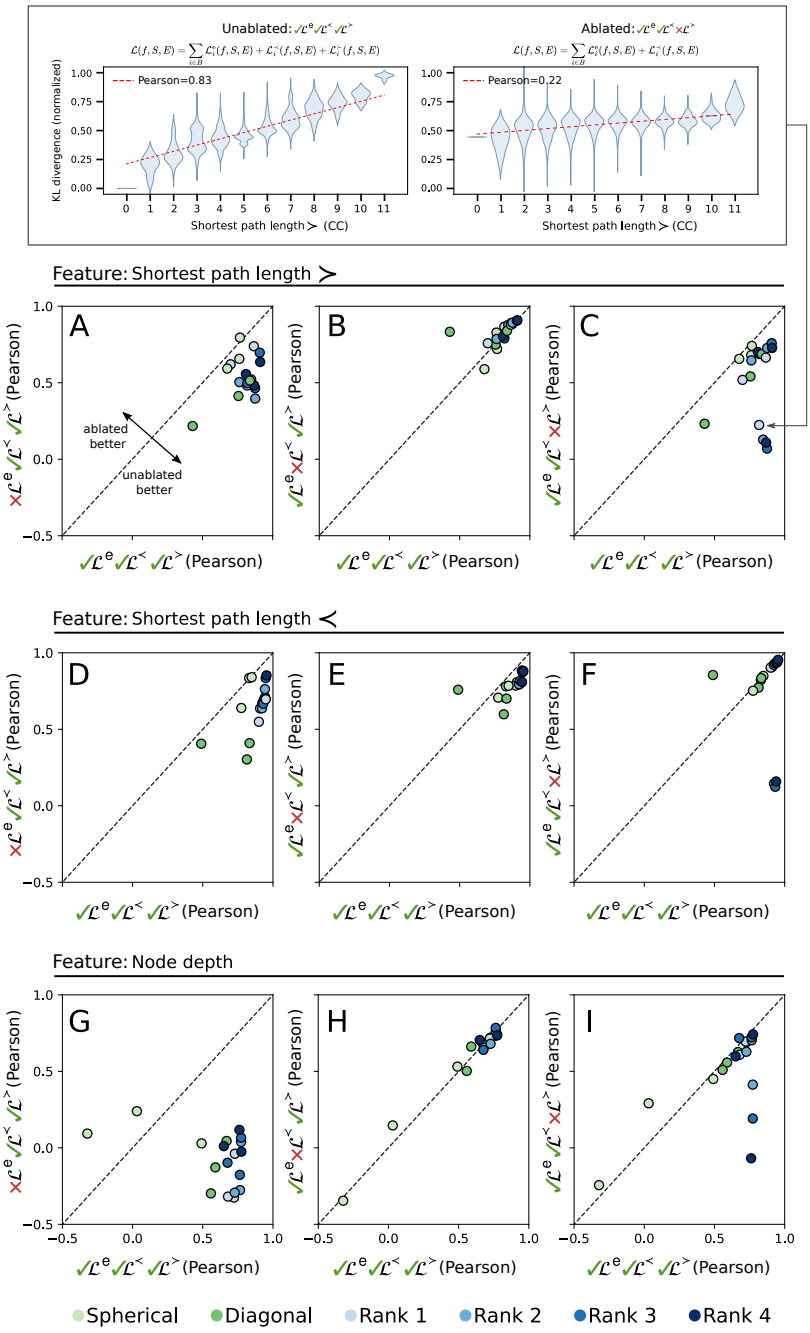


Figure 7: Ablation study. Each dot represents the Pearson correlation between unablated and ablated representations learned from a specific GO graph using a particular type of covariance matrix. Pearson correlations were calculated for three different structural features.



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

Table 3: Node classification performance (lower Davies-Bouldin index/better performance). Infinite Davies-Bouldin indices indicate inability of classifying nodes correctly.

| | Davies–Bouldin index (↓) | | |
|---|---|---|---|
| Method | BP | CC | MF |
| Lin | ∞ | ∞ | ∞ |
| Resnik | ∞ | ∞ | ∞ |
| AIC | ∞ | ∞ | ∞ |
| Wang | 452.0 | 11.8 | 12.6 |
| GOGO | 225.5 | 8.8 | 12.6 |
| GraRep | 172.5 | 6.7 | 12.4 |
| AROPE | 394.2 | 27.3 | 21.8 |
| SVD-anc | 69.2 | 11.5 | 15.5 |
| SVD-des | 290.3 | 23.2 | 37.6 |
| DeepWalk | 1465.8 | 6.8 | 12.9 |
| LINE | 39.3 | 8.4 | 19.3 |
| node2vec | 505.1 | 21.1 | 35.1 |
| VERSE | 301.9 | 8.4 | 14.3 |
| onto2vec | 282.6 | 8.2 | 13.0 |
| anc2vec | 46.5 | 10.0 | 21.3 |
| neig2vec | 66.3 | 11.4 | 37.8 |
| Graph2Gauss | 5.0 | 3.6 | 39.7 |
| Spherical | 2.7 | 13.2 | 2.2 |
| Diagonal | 1.6 | 1.9 | 1.7 |
| Rank 1 | 1.0 | 0.9 | 0.8 |
| Rank 2 | 0.9 | 0.8 | **0.7** |
| Rank 3 | **0.8** | **0.7** | **0.7** |
| Rank 4 | **0.8** | **0.7** | **0.7** |

Specifically, we represented the GO terms annotating pairs of proteins, labeled as interacting and non-interacting, as low-rank Gaussian distributions, and then we evaluated their performance in predicting interactions using the best-match average approach [42]. As baseline methods, we included ten different point-vector representations.

As shown in Figure 8, the low-rank representations consistently outperforms the evaluated point-vector representations across the three evaluated graphs, achieving the highest AUC values: 0.77 (BP), 0.69 (CC) and 0.66 (MF). This superior performance indicates that the use of Gaussian distributions to represent GO terms better capture their relationships than using point-vector representations, highlighting the benefits of our proposed representations on gene characterization tasks that rely on the Gene Ontology.

# 4 Conclusions

Gaussian distributions have demonstrated to be an effective approach to represent graphs. However, its parameterization typically relies on diagonal



sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

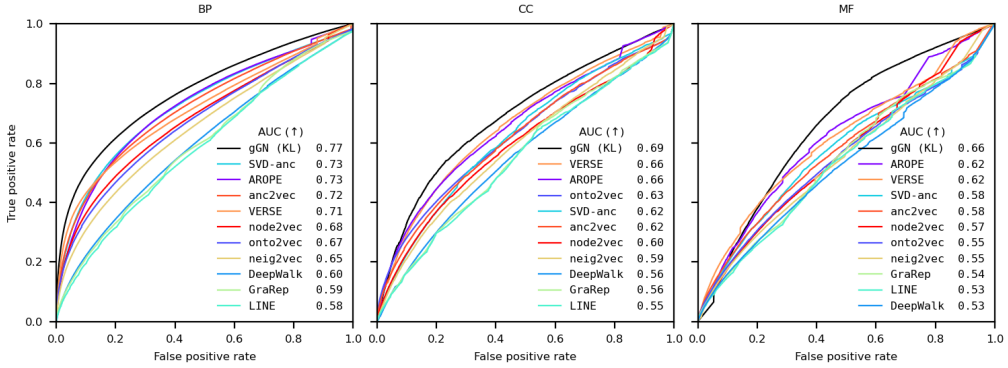


Figure 8: Comparing gGN (KL divergence) and cosine similarity measures on predicting PPIs.

covariance matrices to reduce computational complexity at the cost of degrading representation power. To address this limitation, we have proposed gGN, a neural network that uses a low-rank factorization to enhance the representation power of covariance matrices while keeping computational complexity low. Our theoretical analysis and empirical experiments demonstrated that our proposal leads to better representation of knowledge graphs. This superior performance lies in the ability of low-rank representations to express correlations between embedding dimensions, which is critical for modeling directional and hierarchical relationships between nodes. To learn these correlations between embedding dimensions from data, we designed a loss function that employs the reverse KL divergence. Unlike previous approaches, the reverse KL divergence leads to encode nodes' information content similarly as proposed by foundational studies on semantic similarity calculation. Experiments on the Gene Ontology, a representative knowledge graph in bioinformatics, showed that our approach achieves higher predictive performance compared to traditional methods used on gene characterization tasks. The results highlight the benefits of using the low-rank approximation on representing knowledge graphs.

# Data availability

The data and source code used in the experiments are available in a Github repository at `https://github.com/aedera/ggn`.



sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

# Funding

This work was supported by ANPCyT (PICT 2018 3384) and UNL (CAI+D 2020 115).


# A KL divergence and loss components

Let $\mathcal{N}_i$ and $\mathcal{N}_j$ be two multivariate Gaussian distributions with covariance matrices following Eq. 1 in the main text. The energy $E_{ij} = E(\mathcal{N}_i, \mathcal{N}_j)$ is defined as the (reverse) KL divergence:

$$E_{ij} = \mathrm{KL}(\mathcal{N}_j|\mathcal{N}_i) = \frac{1}{2}\Bigg( \underbrace{\log \frac{|\Sigma_i|}{|\Sigma_j|}}_{\text{Term 1}} + \underbrace{\mathrm{tr}(\Sigma_i^{-1}\Sigma_j)}_{\text{Term 2}} + \underbrace{(\mu_i - \mu_j)^T \Sigma_i^{-1}(\mu_i - \mu_j)}_{\text{Term 3}} - d \Bigg), \tag{6}$$

where $|\cdot|$ is the determinant, with $E_{ij}$ being non-negative and equals zero when both distributions are equal.

The KL divergence between Gaussian distributions has a closed-form expression [39], and algebraic manipulations can be applied to its terms to substantially reduce the computational cost, as described below.

## A.1 Calculating KL Term 1

This term involves calculating the determinant of two covariance matrices

$$\log \frac{|\Sigma_i|}{|\Sigma_j|} = \log|\Sigma_i| - \log|\Sigma_j|,$$

where the matrix determinant lemma can be used to calculate the determinant

$$
\begin{aligned}
\log|\Sigma| &= \log|D + PP^T| \\
&= \log\left(|D||I + P^T D^{-1} P|\right) \\
&= \log|D| + \log|\underbrace{I + P^T D^{-1} P}_{\text{capacitance matrix } C}| \\
&= \log|D| + \log|C|.
\end{aligned}
$$

Here, $\log|D| = \log\prod_k D_{kk} = \sum_k \log D_{kk}$. On the other hand, by using the Cholesky decomposition, the determinant of the capacitance matrix $C \in \mathbb{R}^{r\times r}$ is

$$
\log|C| = \log|LL^T|,
$$

where $L \in \mathbb{R}^{r\times r}$ is a lower triangular matrix. Here, calculating the Cholesky decomposition takes $O(r^3)$ time, where $r \ll d$. Thus, the determinant can be calculated as follows

$$
\begin{aligned}
\log|LL^T| &= \log|L|^2 \\
&= 2\log|L| \\
&= 2\log\prod_k L_{kk} \\
&= 2\sum_k \log L_{kk}.
\end{aligned}
$$

Taken together, Term 1 can be expressed as

$$
\begin{aligned}
\log\frac{|\Sigma_i|}{|\Sigma_j|} &= \log|\Sigma_i| - \log|\Sigma_j| \\
&= \log|D_i| + \log|C_i| - (\log|D_j| + \log|C_j|) \\
&= \log|D_i| + 2\log|L_i| - (\log|D_j| + 2\log|L_j|)\,,
\end{aligned}
$$

where computing the determinant of a covariance matrix boils down to computing the determinants of a diagonal and a triangular matrices, which only involves the product of their diagonal values.

## A.2 Calculating KL Term 2

Using the low-rank form, the second term is

$$\begin{aligned}\mathrm{tr}(\Sigma_i^{-1}\Sigma_j) &= \mathrm{tr}[(D_i + P_iP_i^T)^{-1}(D_j + P_jP_j^T)] \\ &= \mathrm{tr}[(D_i + P_iIP_i^T)^{-1}(D_j + P_jP_j^T)],\end{aligned}$$

and using the Woodbury matrix identity, the first factor can be re-written as

$$\begin{aligned}&= \mathrm{tr}[(D_i^{-1} - D_i^{-1}P_i(\underbrace{I + P_i^TD_i^{-1}P_i}_{C_i})^{-1}P_i^TD_i^{-1})(D_j + P_jP_j^T)] \\ &= \mathrm{tr}[(D_i^{-1} - D_i^{-1}P_iC_i^{-1}P_i^TD_i^{-1})(D_j + P_jP_j^T)],\end{aligned}$$

now, we can re-used the Cholesky decomposition of $C_i$

$$= \mathrm{tr}[(D_i^{-1} - D_i^{-1}P_i(L_iL_i^T)^{-1}P_i^TD_i^{-1})(D_j + P_jP_j^T)]$$

by algebraically manipulating the factorization of $C_i$, a symmetric structure $(A)$ can be exposed

$$\begin{aligned}&= \mathrm{tr}[(D_i^{-1} - \underbrace{D_i^{-1}P_iL_i^{-T}}_{A^T}\underbrace{L_i^{-1}P_i^TD_i^{-1}}_{A})(D_j + P_jP_j^T)] \\ &= \mathrm{tr}[(D_i^{-1} - A^TA)(D_j + P_jP_j^T)] \\ &= \mathrm{tr}[(D_i^{-1}D_j + D_i^{-1}P_jP_j^T - A^TAD_j - A^TAP_jP_j^T)].\end{aligned}$$

Due to the linearity of the trace operator, the latter equation is equivalent to

$$= \mathrm{tr}(D_i^{-1}D_j) + \mathrm{tr}(D_i^{-1}P_jP_j^T) - \mathrm{tr}(A^TAD_j) - \mathrm{tr}(A^TAP_jP_j^T),$$

to further simplify this expression, we can use the Cholesky decomposition of some diagonal matrices

$$= \mathrm{tr}(D_i^{-1}D_j) + \mathrm{tr}(D_i^{-\frac{1}{2}}D_i^{-\frac{1}{2}}P_jP_j^T) - \mathrm{tr}(A^TAD_j^{\frac{1}{2}}D_j^{\frac{1}{2}}) - \mathrm{tr}(A^TAP_jP_j^T).$$

By using the cyclic property of the trace operator

$$\begin{aligned}
&= \mathrm{tr}(D_i^{-1}D_j) + \mathrm{tr}(D_i^{-\frac{1}{2}}P_jP_j^TD_i^{-\frac{1}{2}}) - \mathrm{tr}(D_j^{\frac{1}{2}}A^TAD_j^{\frac{1}{2}}) \\
&\quad - \mathrm{tr}(AP_jP_j^TA^T) \\
&= \mathrm{tr}(D_i^{-1}D_j) + \mathrm{tr}(\underbrace{D_i^{-\frac{1}{2}}P_j}_{E}\underbrace{(D_i^{-\frac{1}{2}}P_j)^T}_{E^T}) - \\
&\mathrm{tr}(\underbrace{D_j^{\frac{1}{2}}A^T}_{Z}\underbrace{(D_j^{\frac{1}{2}}A^T)^T}_{Z^T}) - \mathrm{tr}(\underbrace{AP_j}_{K}\underbrace{(AP_j)^T}_{K^T}) \\
&= \mathrm{tr}(D_i^{-1}D_j) + \mathrm{tr}(EE^T) - \mathrm{tr}(ZZ^T) - \mathrm{tr}(KK^T).
\end{aligned}$$

Since the matrix multiplications in the traces involve transposed matrices, these multiplications be further reduced: $\mathrm{tr}(XX^T) = \sum_{k\ell} X_{k\ell}^2$.

## A.3 Calculating KL Term 3

The third term can be expressed as

$$\begin{aligned}
(\mu_i - \mu_p)^T\Sigma_i^{-1}(\mu_i - \mu_p) &= \Delta^T\Sigma_i^{-1}\Delta \\
&= \Delta^T(D_i + P_iP_i^T)^{-1}\Delta \\
&= \Delta^T(D_i + P_iIP_i^T)^{-1}\Delta,
\end{aligned}$$

where $\Delta = (\mu_i - \mu_p)$.

The intermediate factor can be re-expressed using the Woodbury matrix identity

$$
\begin{aligned}
&\Delta^T (D_i + P_i I P_i^T)^{-1} \Delta \\
&\quad = \Delta^T (D_i^{-1} - D_i^{-1} P_i (I + P_i^T D_i^{-1} P_i)^{-1} P_i^T D_i^{-1}) \Delta \\
&\quad = \Delta^T (D_i^{-1} - D_i^{-1} P_i C_i^{-1} P_i^T D_i^{-1}) \Delta \\
&\quad = \Delta^T (D_i^{-1} - D_i^{-1} P_i L_i^{-T} L_i^{-1} P_i^T D_i^{-1}) \Delta \\
&\quad = \Delta^T (D_i^{-1} - A^T A) \Delta \\
&\quad = \Delta^T D_i^{-1} \Delta - \Delta^T A^T A \Delta \\
&\quad = \Delta^T D_i^{-\frac{1}{2}} D_i^{-\frac{1}{2}} \Delta - \Delta^T A^T A \Delta \\
&\quad = (D_i^{-\frac{1}{2}} \Delta)^T D_i^{-\frac{1}{2}} \Delta - (A\Delta)^T A \Delta.
\end{aligned}
$$

This last line is cheap to compute as it involves diagonal matrices and the matrix $A$, which was previously computed in the Term 2.

## A.4 The three components of the loss function

In Eq. 3, the component $\mathcal{L}_i^{\mathrm{e}}$ measures how well $f(i)$ represents the relationships between node $i$ and its immediate neighbors, that is

$$
\mathcal{L}_i^{\mathrm{e}}(f, S, E) = \sum_{j \in \mathcal{P}_i} E_{ij}^2 + \sum_{j \in \mathcal{D}_i^+} e^{-E_{ij}}, \tag{7}
$$

where $\mathcal{P}_i = \{j \neq i \,|\, S_{ij} = 1\}$ is the set of parent nodes of $i$, and $\mathcal{D}_i^+ = \{j \neq i \,|\, S_{ji} = 1 \text{ or } S_{ji} = \infty\}$ denotes the set of immediate descendants of $i$ ($S_{ji} = 1$) along with unreachable nodes ($S_{ji} = \infty$). The energy function $E_{ij} \equiv E(f(i), f(j))$ measures the similarity between $\mathcal{N}_i$ and $\mathcal{N}_j$ (the lower the more similar) by using the KL divergence. During optimization, $\mathcal{L}_i^{\mathrm{e}}$ is minimized, thereby increasing the similarity between $\mathcal{N}_i$ and $\mathcal{N}_j$ when $j \in \mathcal{P}_i$ while reducing it when $j \in \mathcal{D}_i^+$.

In contrast, the component $\mathcal{L}_i^{\prec}$ in Eq. refeq:loss measures how well $f(i)$ represents the relationships between node $i$ and its ancestors. It is defined as

$$
\mathcal{L}_i^{\prec}(f, S, E) = \sum_{j \in \mathcal{A}_i^{\prec}} \left( \frac{S_{ij}}{S_{ia}} - \frac{E_{ij}}{E_{ia}} \right)^2, \tag{8}
$$

Table 4: Structural features of each graph composing the Gene Ontology.

| | GO graphs | | |
|---|---|---|---|
| Statistic | BP | CC | MF |
| Number of nodes | 28,888 | 4,196 | 11,177 |
| Number of edges | 67,238 | 6,904 | 13,604 |
| Avg number of parents | $2 \pm 1$ | $2 \pm 1$ | $1 \pm 1$ |
| Max number of parents | 9 | 5 | 6 |
| Avg shortest path length | $4 \pm 2$ | $3 \pm 2$ | $3 \pm 2$ |
| Network diameter | 13 | 10 | 11 |
| Avg number of ancestors | $24 \pm 17$ | $11 \pm 6$ | $7 \pm 3$ |
| Max number of ancestors | 146 | 40 | 32 |
| Avg clustering coefficient | 4.01e−10 | 2.64e−02 | 7.00e−04 |

where $\mathcal{A}_i^{\prec} = \{j \,|\, S_{ij} > 0, S_{ij} \neq \infty\}$ denotes the ancestors of $i$, and $a = \text{argmax}_j \, S_{ij}$. This component computes the degree of matching between the shortest path length $S_{ij}$ and the energy $E_{ij}$ for each node $j \in \mathcal{A}_i^{\prec}$. To mitigate scale heterogeneity, the maximum values $S_{ia}$ and $E_{ia}$ are used for normalization. The minimization of $\mathcal{L}_i^{\prec}$ yields similarities between $\mathcal{N}_i$ and the Gaussian distributions representing the ancestors of $i$ that are ranked according to how far the ancestors are from $i$. For instance, $\mathcal{N}_i$ would be more similar to $\mathcal{N}_j$ than to $\mathcal{N}_k$ when $j$ and $k$ are the parent and the grandfather of $i$, respectively.

Similarly, the final component, $\mathcal{L}_i^{\succ}$, in Eq. refeq:loss measures how well $f(i)$ represents the relationships between node $i$ and its descendants

$$\mathcal{L}_i^{\succ}(f, S, E) = \sum_{j \in \mathcal{A}_i^{\succ}} \left( \frac{S_{ji}}{S_{ai}} - \frac{E_{ji}}{E_{ai}} \right)^2, \tag{9}$$

where $\mathcal{A}_i^{\succ} = \{j \,|\, S_{ji} > 0, S_{ji} \neq \infty\}$ is the set of descendants of $i$, and $a = \text{argmax}_j \, S_{ji}$. Therefore, during optimization, the minimization of $\mathcal{L}_i^{\prec}$ yields similarities between $\mathcal{N}_i$ and the Gaussian distributions representing the descendants of $i$ that are ranked according to the shortest path lengths of the descendants.

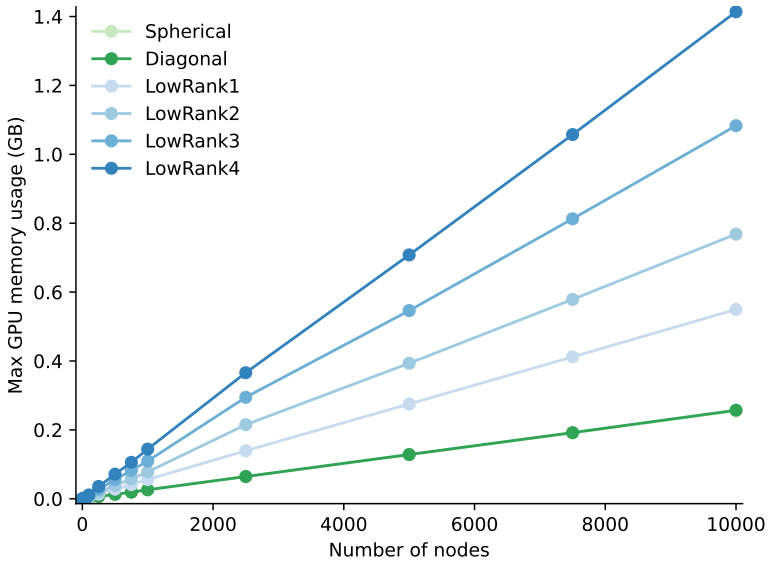


Figure 9: Maximum GPU usage when learning synthetic random DAGs ($d = 10$, epochs=500).

# B Learning node representations with gGN (resource requirements)

In our experiments, node representations were obtained by training gGN during 500 epochs using batches of 128 random nodes with three different seeds. The training set contained all the nodes of a given graph, in line with previous studies [56, 13]. The Adam optimizer [57] with default parameters was used for training. All the experiments conducted on this study were run on an i7-5960X processor equipped with eight 3-GHz dual cores in a server with 64GB of RAM, an Nvidia Titan X GPU and a 3-TB NVMe disk. The parameters $\mu_i$ and $P_i$ were initialized randomly using a standard normal distribution, while $D_i$ was initialized with 1s, following previous recommendations [23]. During the learning process, $D_i$ was constrained to be positive definite by clipping it to lie within the hypercube $[0.01, \infty]^d$.

Table 5: Diagonal and low-rank representations using the same number of parameters per node. Significant correlations are boldfaced.

| Params | $d$ | Model | Pearson correlation (↑) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Short path length $\succ$ | | | Short path length $\prec$ | | | Node depth | | |
| | | | BP | CC | MF | BP | CC | MF | BP | CC | MF |
| 30 | 30 | Diagonal | **0.90**±0.01 | **0.85**±0.01 | **0.88**±0.01 | **0.93**±0.01 | 0.90±0.01 | 0.93±0.01 | 0.66±0.02 | 0.63±0.02 | 0.62±0.05 |
| | 15 | Rank1 | 0.88±0.01 | 0.79±0.02 | 0.87±0.01 | **0.93**±0.01 | 0.90±0.02 | **0.96**±0.01 | 0.73±0.03 | 0.70±0.02 | 0.67±0.05 |
| | 10 | Rank2 | 0.85±0.02 | 0.76±0.02 | **0.88**±0.03 | 0.92±0.03 | **0.92**±0.02 | 0.94±0.01 | **0.77**±0.01 | **0.77**±0.02 | **0.73**±0.03 |
| | 6 | Rank3 | 0.73±0.01 | 0.68±0.01 | 0.85±0.02 | 0.88±0.01 | 0.86±0.02 | 0.90±0.01 | 0.64±0.01 | 0.66±0.01 | 0.56±0.01 |
| 40 | 40 | Diagonal | **0.91**±0.01 | **0.87**±0.01 | 0.88±0.01 | **0.93**±0.01 | 0.89±0.01 | 0.93±0.01 | 0.63±0.02 | 0.64±0.01 | 0.58±0.02 |
| | 20 | Rank1 | 0.90±0.01 | 0.81±0.02 | 0.87±0.01 | **0.93**±0.01 | 0.91±0.02 | **0.96**±0.01 | 0.71±0.01 | 0.69±0.01 | **0.68**±0.01 |
| | 10 | Rank3 | 0.87±0.01 | 0.82±0.02 | **0.91**±0.01 | **0.93**±0.02 | **0.93**±0.01 | 0.95±0.01 | **0.77**±0.02 | **0.76**±0.03 | **0.68**±0.01 |
| 50 | 50 | Diagonal | **0.91**±0.01 | **0.86**±0.02 | 0.89±0.01 | 0.93±0.01 | 0.91±0.02 | 0.93±0.01 | 0.59±0.02 | 0.62±0.02 | 0.63±0.02 |
| | 25 | Rank1 | 0.90±0.01 | 0.81±0.03 | 0.86±0.01 | **0.94**±0.01 | 0.91±0.02 | **0.96**±0.01 | 0.70±0.02 | 0.69±0.03 | **0.68**±0.02 |
| | 10 | Rank4 | 0.87±0.01 | 0.81±0.03 | **0.91**±0.02 | **0.94**±0.01 | **0.94**±0.03 | 0.95±0.02 | **0.76**±0.01 | **0.77**±0.02 | 0.65±0.01 |



# C Shortest path length and node depth

To determine whether Gaussian distributions can capture the directional structure of a knowledge graph, we evaluated their capacity to encode the shortest path lengths within the graph. This property is relevant in knowledge graphs, as these lengths are determined by the directionality of the edges. Given that asymmetric distance/energy between two nodes can be used to encode edge directionality [38], we evaluated whether Gaussian distributions could encode the shortest path lengths for both the original directionality of edges ($\prec$) and the reverse directions ($\succ$).

For a pair of nodes, $i$ and $j$, we compared the shortest path length between them, $S_{ij}$, with the distance/energy between their corresponding Gaussian representations, $\mathrm{KL}(\mathcal{N}_i \| \mathcal{N}_j)$. We summarized the comparisons for all node pairs by calculating the Pearson correlation coefficient $r = \mathrm{cov}(X, Y)/(\sigma_X \sigma_Y)$, where cov is the covariance, $X = (\mathrm{KL}(\mathcal{N}_i \| \mathcal{N}_j) \mid S_{ij} \neq \infty)$, $Y = (S_{ij} | S_{ij} \neq \infty)$, and $\sigma$ is the standard deviation. To assess edge directionality, we performed this correlation considering the original directionality of edges ($\prec$) and the reverse directions ($\succ$).

For nodes represented as point vectors, we used $X = (s_{\cos}(v_i, v_j) \mid S_{ij} \neq \infty)$, where

$$s_{\cos}(v_i, v_j) = \frac{\langle v_i, v_j \rangle}{|v_i||v_j|}. \tag{10}$$

Here, $v_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$ are point vector representations for nodes $i$ and $j$,

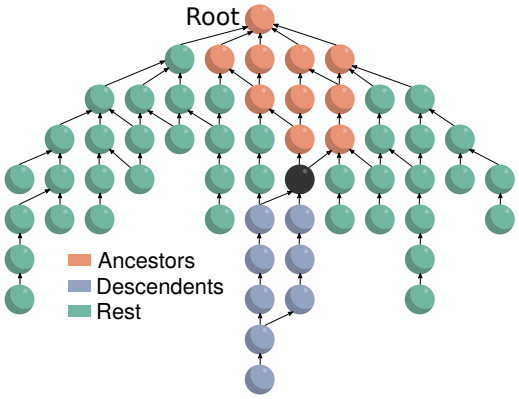


Figure 10: Given any node (black), the remaining nodes can be classified into three disjoint sets.

respectively, and $|\cdot|$ is the Euclidean norm.

To determine whether Gaussian distribution can capture the hierarchical information of a knowledge graph, we evaluated their ability to preserve the depth of the nodes. To this aim, we compared the depth of a given node $i$ with the entropy of its corresponding Gaussian representation, $H(\mathcal{N}_i)$. We summarized the comparisons for all nodes by calculating the Pearson correlation using $X = (H(\mathcal{N}_i) \mid i = 0, \ldots, n-1)$ and $Y = (|\mathcal{A}_i^{\prec}| \mid i = 0, \ldots, n-1)$. The entropy is defined as

$$H(\mathcal{N}_i) = \frac{d}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\Sigma_i| \tag{11}$$

In the case of nodes represented by point vectors, we used $X = (s_{\cos}(v_i, v_{\text{root}}) \mid i = 0, \ldots, n-1)$, where $v_{\text{root}}$ is the point vector representing the root node.

# D Node classification

Given a knowledge graph as shown in Figure 10, any node $i$ (black) can be used to classify the remaining nodes into three disjoint sets: ancestors (orange), descendants (blue) or neither of them (green). We investigated whether node representations were able to discriminate between the ancestors and descendants of a node. To this aim, we calculated the similarity between the representation of node $i$ and the representations of the other nodes. Then

we assessed whether these similarities could effectively discriminate between the ancestors and descendants of a node. For this assessment, we used the Davies-Bouldin (DB) index [58] to measure the degree of overlap between similarities calculated from ancestors and those from descendant nodes. Given $C_{\mathcal{A}^\prec} = \bigcup_{i=0}^{n-1}\{s(i,j) : j \in \mathcal{A}^\prec\}$ and $C_{\mathcal{A}^\succ} = \bigcup_{i=0}^{n-1}\{s(i,j) : j \in \mathcal{A}^\succ\}$ be two sets of similarities calculated by a function $s(\cdot,\cdot)$, the DB index is defined as $\text{DB} = (s_{\mathcal{A}^\prec} + s_{\mathcal{A}^\succ})/d$, where $s_{\mathcal{A}^\prec}$ is the average over $\{|c - \bar{C}_{\mathcal{A}^\prec}|\}_{c \in C_{\mathcal{A}^\prec}}$, $s_{\mathcal{A}^\succ}$ is similarly defined using $C_{\mathcal{A}^\succ}$, and $d = |\bar{C}_{\mathcal{A}^\prec} - \bar{C}_{\mathcal{A}^\succ}|$, with $\bar{C}$ indicating the mean value. The lower the DB index is, the more separated the two sets are, with zero being the lowest possible DB index.

For Gaussian representations, the KL divergence (Eq. 6) was used as the similarity function $s(\cdot,\cdot)$, while we used the standard cosine similarity when nodes were represented as point vectors [13, 59], as shown in Eq. 10. We also included well-known methods for calculating the semantic similarity between nodes: Lin [31], Resnik [32] and AIC [60] that used the intrinsic information content [50]. The intrinsic information content (IC) of a node $i$ is

$$\text{IC}(i) = 1 - \frac{\log(\mathcal{D}_i + 1)}{\log n}, \tag{12}$$

where $\mathcal{D}_i$ is the set of descendants of $i$ and $n$ is the total number of nodes in the graph.

# E Baseline methods for node classification

Table 6 shows the methods selected as baselines for node classification. They are categorized in five groups: classic methods, based on matrix factorization (MF), random walks (RW), neural networks (NN) and Gaussian distributions. For all classic methods, we used default parameters as proposed by their authors. All methods that learned representations used 200 dimensions. For GrapRep, 20 orders were used, each of them with 10 dimensions. For AROPE, default parameters were used and 49 dimensions for each order. For SVD-anc (and SVD-desc), an $r$-dimensional representation was obtained for each node by truncating the left singular matrix U at dimension $r$, and multiplying it by the truncated diagonal matrix $D$: $U_{1:r}D_{1:k,1:k}$. The matrices $D$ and $U$ were obtained by computing the SVD decomposition [41] $M = UDV^T$; here, $M$ is the matrix of ancestors (SVD-anc) or descendants (SVD-desc), which

Table 6: Baseline methods.

| Method | Category | Repository |
|---|---|---|
| Lin [31] | Classic | - |
| Resnik [32] | Classic | - |
| AIC [60] | Classic | - |
| Wang [51] | Classic | `https://github.com/tanghaibao/goatools` |
| GOGO [53] | Classic | `https://github.com/zwang-bioinformatics/GOGO` |
| GraRep [61] | MF | `https://github.com/benedekrozemberczki/GraRep` |
| AROPE [62] | MF | `https://github.com/ZW-ZHANG/AROPE` |
| SVD-anc | MF | - |
| SVD-des | MF | - |
| DeepWalk [63] | RW | `https://github.com/phanein/deepwalk` |
| LINE [64] | RW | `https://github.com/tangjianpku/LINE` |
| node2vec [65] | RW | `https://github.com/aditya-grover/node2vec` |
| VERSE [66] | RW | `https://github.com/xgfs/verse.git` |
| onto2vec [13] | NN | `https://github.com/bio-ontology-research-group/onto2vec` |
| anc2vec [14] | NN | `https://github.com/sinc-lab/anc2vec` |
| neigh2vec [14] | NN | `https://github.com/sinc-lab/anc2vec` |
| Graph2Gauss [25] | Gaussian | `https://github.com/abojchevski/graph2gauss` |

were built from BP, CC and MF, respectively. For DeepWalk, 80 random walks per node with a maximum length of 40 were extracted. To construct representations, the extracted walks were treated as sentences, and nodes were treated as words, to be used as input to word2vec (`https://github.com/tmikolov/word2vec`) using the skip-gram objective with parameters: window=5, min-count=0 and iter=200. For LINE, we used 2 orders, 5 negatives, 100 samples and $\rho = 0.025$. In node2vec, $p = 1$ and $q = 1$ were used. We used VERSE with $\alpha = 0.85$ and 3 samples. From the axioms extracted by onto2vec, it was given as input to word2vec using the skip-gram objective with parameters: window=5, min-count=0 and iter=200. The node representations of anc2vec and neigh2vec were downloaded from their public repositories. For Graph2Gauss, we used as input an adjacency matrix built for each graph to learn node representations with default parameters. It is worth noting that the capacity of the representations learned by Graph2Gauss is theoretically equivalent to those obtained by gGN when using covariance matrices with rank zero (diagonal representations). However, both methods differ in the loss function employed for learning node representations.

# F Number of parent nodes impacts the performance of diagonal representations

Figure 6A showed that spherical representations outperformed the diagonal ones in preserving the shortest path lengths $\succ$ in BP, the largest and most structurally complex graph evaluated (Table 4). Unlike the other two knowledge graphs evaluated (CC and MF), BP has a more complex hierarchical structure. The highest average number of parents per node is 2.32 in BP (CC=1.64 and MF=1.21), with some nodes having up to nine parents (CC=5, MF=6). To investigate whether the number of parent nodes affects the performance of diagonal representations, we analyzed how well these representations preserved the shortest path lengths $\succ$ when varying the number of parent nodes in BP. Specifically, the maximum number of parents ($K$) for each node was randomly selected. This analysis showed that the performance of diagonal representations deteriorated as the number of parent nodes increased (Figure 11), a trend not clearly observed with spherical and low-rank representations. Additionally, this performance degradation when increasing the number of parent nodes was specific for the shortest path lengths $\succ$, showing no substantial effects on the other two features evaluated: the shortest path lengths $\prec$ and node depths. In these two features, the performance of diagonal representations was roughly equivalent to or better than that of the spherical representations independently of the number of parent nodes. Altogether, these results suggest that the spherical constraint on Gaussian distributions acts as a regularizing factor, facilitating the preservation of the shortest path lengths $\succ$ while limiting their ability to preserve other features, particularly node depth.

# References


[1] M. Khodak, A. Risteski, C. Fellbaum, S. Arora, Automated wordnet construction using word embeddings, in: Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications, 2017, pp. 12–23.

[2] L. J. Martin, P. Ammanabrolu, X. Wang, S. Singh, B. Harrison, M. Dhuliawala, P. Tambwekar, A. Mehta, R. Arora, N. Dass, et al., Improvisational

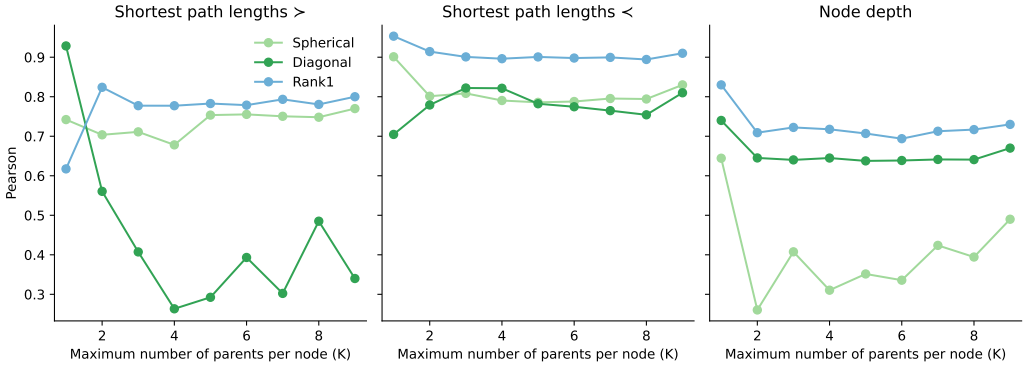


Figure 11: Pearson correlations calculated from 10-dimensional Gaussian representations learned from perturbations of BP with an increasing number ($K$) of parents per node. Since BP has a maximum of nine parent nodes, $K = 9$ represents the original BP without perturbations.

storytelling agents, in: Workshop on Machine Learning for Creativity and Design (NeurIPS 2017), Long Beach, CA, 2017.

[3] B. Barz, J. Denzler, Hierarchy-based image embeddings for semantic image retrieval, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 638–647.

[4] Z. Wang, Z. Wei, Pt-kgnn: A framework for pre-training biomedical knowledge graphs with graph neural networks, Computers in Biology and Medicine 178 (2024) 108768.

[5] G. O. Consortium, The gene ontology resource: 20 years and still going strong, Nucleic acids research 47 (D1) (2019) D330–D338.

[6] U. Consortium, Uniprot: a worldwide hub of protein knowledge, Nucleic acids research 47 (D1) (2019) D506–D515.

[7] H. Mi, A. Muruganujan, D. Ebert, X. Huang, P. D. Thomas, Panther version 14: more genomes, a new panther go-slim and improvements in enrichment analysis tools, Nucleic acids research 47 (D1) (2019) D419–D426.

[8] L. Zhang, Y. Jiang, Y. Yang, Gnngo3d: Protein function prediction based on 3d structure and functional hierarchy learning, IEEE Transactions on Knowledge & Data Engineering (01) (2023) 1–12.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE transactions on neural networks and learning systems 32 (1) (2020) 4–24.

[10] X. Meng, T. Zou, Clinical applications of graph neural networks in computational histopathology: A review, Computers in Biology and Medicine 164 (2023) 107201.

[11] Y. Gu, S. Zheng, Q. Yin, R. Jiang, J. Li, Redda: Integrating multiple biological relations to heterogeneous graph neural network for drug-disease association prediction, Computers in biology and medicine 150 (2022) 106127.

[12] M. U. Rehman, I. Hussain, H. Tayara, K. T. Chong, et al., A graph neural network approach for predicting drug susceptibility in the human microbiome, Computers in Biology and Medicine 179 (2024) 108729.

[13] F. Z. Smaili, X. Gao, R. Hoehndorf, Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations, Bioinformatics 34 (13) (2018) i52–i60.

[14] A. A. Edera, D. H. Milone, G. Stegmayer, Anc2vec: embedding gene ontology terms by preserving ancestors relationships, Briefings in Bioinformatics 23 (2) (2022).

[15] Y. Lu, D. Yang, P. Wang, P. Rosso, P. Cudre-Mauroux, Schema-aware hyper-relational knowledge graph embeddings for link prediction, IEEE Transactions on Knowledge & Data Engineering (01) (2023) 1–15.

[16] I. Vulić, D. Gerz, D. Kiela, F. Hill, A. Korhonen, Hyperlex: A large-scale evaluation of graded lexical entailment, Computational Linguistics 43 (4) (2017) 781–835.

[17] M. Nickel, D. Kiela, Poincaré embeddings for learning hierarchical representations, Advances in neural information processing systems 30 (2017).

[18] J. Kim, D. Kim, K.-A. Sohn, Hig2vec: hierarchical representations of gene ontology and genes in the poincaré ball, Bioinformatics 37 (18) (2021) 2971–2980.

[19] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al., Gene ontology: tool for the unification of biology, Nature genetics 25 (1) (2000) 25–29.

[20] S. Mežnar, M. Bevec, N. Lavrač, B. Škrlj, Ontology completion with graph-based machine learning: A comprehensive evaluation, Machine Learning and Knowledge Extraction 4 (4) (2022) 1107–1123.

[21] I. Vendrov, R. Kiros, S. Fidler, R. Urtasun, Order-embeddings of images and language, arXiv preprint arXiv:1511.06361 (2015).

[22] M. Ou, P. Cui, J. Pei, Z. Zhang, W. Zhu, Asymmetric transitivity preserving graph embedding, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 1105–1114.

[23] B. Athiwaratkun, A. G. Wilson, Hierarchical density order embeddings, arXiv preprint arXiv:1804.09843 (2018).

[24] L. Vilnis, A. McCallum, Word representations via gaussian embedding, arXiv preprint arXiv:1412.6623 (2014).

[25] A. Bojchevski, S. Günnemann, Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking, arXiv preprint arXiv:1707.03815 (2017).

[26] M. Monteiro, L. Le Folgoc, D. Coelho de Castro, N. Pawlowski, B. Marques, K. Kamnitsas, M. van der Wilk, B. Glocker, Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty, Advances in Neural Information Processing Systems 33 (2020) 12756–12767.

[27] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with gaussian embedding, in: Proceedings of the 24th ACM international on conference on information and knowledge management, 2015, pp. 623–632.

[28] D. Zhu, P. Cui, D. Wang, W. Zhu, Deep variational network embedding in wasserstein space, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 2827–2836.

[29] G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, I. Simpson, Structured uncertainty prediction networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5477–5485.

[30] G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, I. Simpson, Training vaes under structured residuals, arXiv preprint arXiv:1804.01050 (2018).

[31] D. Lin, et al., An information-theoretic definition of similarity., in: Icml, Vol. 98, 1998, pp. 296–304.

[32] P. Resnik, Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language, Journal of artificial intelligence research 11 (1999) 95–130.

[33] P. Harish, P. J. Narayanan, Accelerating large graph algorithms on the gpu using cuda, in: International conference on high-performance computing, Springer, 2007, pp. 197–208.

[34] O. Taştan, O. C. Eryüksel, A. Temizel, Accelerating johnson's all-pairs shortest paths algorithm on gpu, A report available at https://github.com/ouzan19/JohnsonAlgoCUDA. P (2017) 1–6.

[35] X. Zhao, H. Zheng, Orion: shortest path estimation for large social graphs, in: 3rd Workshop on Online Social Networks (WOSN 2010), 2010.

[36] F. S. Rizi, J. Schloetterer, M. Granitzer, Shortest path distance approximation using deep learning techniques, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 1007–1014.

[37] S.-i. Amari, Information geometry and its applications, Vol. 194, Springer, 2016.

[38] Z. Tan, B. Liu, G. Yin, Asymmetric graph representation learning, arXiv preprint arXiv:2110.07436 (2021).

[39] J. Duchi, Derivations for linear algebra and optimization, Berkeley, California 3 (1) (2007) 2325–5870.

[40] J. Ding, A. Zhou, Eigenvalues of rank-one updated matrices with some applications, Applied Mathematics Letters 20 (12) (2007) 1223–1226.

[41] K. B. Petersen, M. S. Pedersen, et al., The matrix cookbook, Technical University of Denmark 7 (15) (2008) 510.

[42] C. Pesquita, D. Faria, H. Bastos, A. E. Ferreira, A. O. Falcão, F. M. Couto, Metrics for go based protein semantic similarity: a systematic evaluation, in: BMC bioinformatics, Vol. 9, BioMed Central, 2008, pp. 1–16.

[43] T. M. Cover, Elements of information theory, John Wiley & Sons, 1999.

[44] P. W. Lord, R. D. Stevens, A. Brass, C. A. Goble, Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation, Bioinformatics 19 (10) (2003) 1275–1283.

[45] Z. Wu, M. Palmer, Verb semantics and lexical selection, arXiv preprint cmp-lg/9406033 (1994).

[46] J. J. Jiang, D. W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, arXiv preprint cmp-lg/9709008 (1997).

[47] M. Boguna, I. Bonamassa, M. De Domenico, S. Havlin, D. Krioukov, M. Á. Serrano, Network geometry, Nature Reviews Physics 3 (2) (2021) 114–135.

[48] F. Salehi Rizi, J. Schloetterer, M. Granitzer, Shortest path distance approximation using deep learning techniques, arXiv e-prints (2020) arXiv–2002.

[49] C. Dessimoz, N. Škunca, The gene ontology handbook, Springer Nature, 2017.

[50] N. Seco, T. Veale, J. Hayes, An intrinsic information content metric for semantic similarity in wordnet, in: Ecai, Vol. 16, 2004, p. 1089.

[51] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, C.-F. Chen, A new method to measure the semantic similarity of go terms, Bioinformatics 23 (10) (2007) 1274–1281.

[52] A. Warren, J. Setubal, Using entropy estimates for dag-based ontologies, arXiv preprint arXiv:1403.4887 (2014).

[53] C. Zhao, Z. Wang, Gogo: An improved algorithm to measure the semantic similarity between gene ontology terms, Scientific reports 8 (1) (2018) 1–10.

[54] A. B. Kamran, H. Naveed, Gontosim: a semantic similarity measure based on lca and common descendants, Scientific reports 12 (1) (2022) 1–10.

[55] D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo, et al., The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest, Nucleic acids research 51 (D1) (2023) D638–D646.

[56] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[57] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[58] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE transactions on pattern analysis and machine intelligence (2) (1979) 224–227.

[59] X. Zhong, R. Kaalia, J. C. Rajapakse, Go2vec: transforming go terms and proteins to vector representations via graph embeddings, BMC genomics 20 (9) (2019) 1–10.

[60] X. Song, L. Li, P. K. Srimani, S. Y. Philip, J. Z. Wang, Measure the semantic similarity of go terms using aggregate information content, IEEE/ACM transactions on computational biology and bioinformatics 11 (3) (2013) 468–476.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
A. Edera, G. Stegmayer & D. H. Milone; "gGN: Representing the Gene Ontology as low-rank Gaussian distributions"
Computers in Biology and Medicine, Vol. 183, pp. 109234, 2024.

[61] S. Cao, W. Lu, Q. Xu, Grarep: Learning graph representations with global structural information, in: Proceedings of the 24th ACM international on conference on information and knowledge management, 2015, pp. 891–900.

[62] Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, W. Zhu, Arbitrary-order proximity preserved network embedding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2778–2786.

[63] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701–710.

[64] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 1067–1077.

[65] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.

[66] A. Tsitsulin, D. Mottin, P. Karras, E. Müller, Verse: Versatile graph embeddings from similarity measures, in: Proceedings of the 2018 world wide web conference, 2018, pp. 539–548.