Journal Title Here, 2022, pp. 1–11

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

Benchmarking the Impact of Data Leakage on the Performance of Knowledge Graph Embedding Models for Biomedical Link Prediction

Galadriel Brière,[1,*] Thomas Stosskopf,[1,2] Benjamin Loire[1,3] and Anaïs Baudot[1,4,*]

[1]Aix Marseille Univ, INSERM, MMG, Marseille, France, [2]Present address: Aix Marseille Univ, INSERM, TAGC, Marseille, France, [3]Neurology Therapeutic Area, R&D Servier Paris-Saclay Institut, Gif-sur-Yvette, France and [4]CNRS, Marseille, France
[*]Corresponding author. marie-galadriel.briere@univ-amu.fr, anais.baudot@univ-amu.fr
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

In recent years, Knowledge Graphs (KGs) have received increasing attention for their ability to organize complex biomedical knowledge into structured representations of entities and relations. Knowledge Graph Embedding (KGE) models facilitate efficient exploration of KGs by learning compact data representations. These models are increasingly applied to biomedical KGs for link prediction, for instance to uncover new therapeutic uses for existing drugs.
While numerous KGE models have been developed and benchmarked for link prediction, existing evaluations often overlook the critical issue of data leakage. Data leakage leads the model to learn patterns it would not encounter when deployed in real-world settings, artificially inflating performance metrics and compromising the overall validity of benchmark results. In machine learning, data leakage can arise when (1) there is redundancy between the training and test sets, (2) the model leverages illegitimate features, or (3) the test set does not accurately reflect real-world inference scenarios. In this study, we demonstrate the impact of train-test redundancies in KGE-based link prediction and implement a systematic control procedure to detect and remove those redundancies. In addition, through permutation experiments, we investigate whether node degree acts as an illegitimate predictive feature, and find no evidence that models rely on it. Finally, we evaluate how well common test set sampling strategies reflect the challenges of real-world inference in drug repurposing. We compare random and cold-start data splits with a real-world inference set derived from the Orphanet database. We find that performance drops significantly on the real-world inference set, indicating that current benchmarking practices may overestimate how well KGE models generalize to practical applications. Overall, our findings highlight the importance of rigorous benchmark design and careful evaluation of the generalization ability of KGE models for biomedical link prediction.

Key words: Knowledge Graph Embedding, Data Leakage, Link Prediction, Drug Repurposing, Rare Diseases

Introduction

Biomedical knowledge encompasses diverse associations across multiple scales of biological organization. At the *molecular level*, biomolecules such as genes, proteins, or metabolites are organized into networks that capture intricate biomolecular interactions, such as gene regulatory networks, protein–protein interaction (PPI) networks or metabolic pathways. These molecular networks are further structured at the *cellular level*, where they coordinate functions within and across cell populations. At the *tissue and organ levels*, interactions between cell populations and tissues play a key role in regulating physiological processes. At the *clinical level*, relationships between diseases, phenotypes, and therapeutic compounds reflect shared pathophysiological properties. Biomedical knowledge also encompasses domain-specific ontologies such as the Gene Ontology (1) or the Human Phenotype Ontology (2), which provide standardized terminologies and hierarchical organizations for this complex, heterogeneous biological knowledge. Taken altogether, this large-scale and heterogeneous domain knowledge coalesce into Knowledge Graphs (KGs), which represent entities and their relationships using triples in the form $(head, relation, tail)$. KGs provide an intuitive and unified representation of multi-scale biological knowledge.

By combining the structured knowledge stored in KGs with Machine Learning (ML) approaches, Knowledge Graph Embedding (KGE) methods encode entities and relations into continuous, low-dimensional latent vector spaces while preserving the fundamental structural, topological, and semantic characteristics of KGs. These latent representations facilitate data exploration and enable efficient inference of novel interactions (3), a task known as link prediction.

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Drug repurposing, which involves finding new therapeutic uses for existing drugs, stands out as a critical application of link prediction where KGE can have a significant impact (4). In a knowledge graph perspective, this translates to predicting missing indication triples (triples of the form (drug, indication, disease)) that connect drugs to the diseases they can treat. By leveraging the ability of KGE to predict such novel indications, researchers can identify potential drug candidates more efficiently, reducing the time and cost associated with traditional drug discovery processes.

Despite the significant potential of KGE-based link prediction in biomedicine, challenges remain in evaluating the reliability and generalizability of these models. One key issue is data leakage. Data leakage can artificially inflate any ML predictor performance and undermine the validity of the evaluation process. Data leakage is defined as "*the relationship between (i) the reported performance scores of an ML-based predictor, (ii) the data used to train or validate the predictor, (iii) the data used to generate the reported performances of the ML-based predictor and (iv) the data on which the ML-based predictor should be run at inference time once it is deployed in its intended use case*" (5; 6).

Data leakage can be caused by multiple mechanisms. First, improper split of KG triples can result in data redundancy between the training and test sets, compromising the separation required between these datasets. *Akrami et al.* (7) demonstrated this issue through the study of widely-used non-biomedical benchmark KGs (*FB15k*, *WN18*, and *YAGO3-10*). Their findings revealed that eliminating these redundancies caused a significant decrease in the reported performance of the KGE models. The authors concluded that, under realistic conditions, KGE models do not provide an effective solution for the task of link prediction.

Second, ML models may exploit illegitimate features as proxies for prediction. This phenomenon has been observed in the context of sequence-based PPI prediction using deep learning models (8). In this study, the authors showed that prediction performance could be driven by features unintentionally exposed to the model during training, such as node degree in the PPI network, rather than the actual protein sequences the models were intended to learn from. By randomizing the training PPI network via degree-preserving network permutations, they demonstrated that the models retained significant predictive power despite the absence of biologically meaningful edges. This experiment highlighted the model reliance on topological shortcuts rather than on biologically relevant features. Such exploitation of illegitimate features has not specifically been demonstrated in KGE approaches. However, in theory, in KGE, access to the graph structure encoded through the set of triples can enable models to rely on properties such as node degree as shortcuts for predicting novel links. This highlights the importance of investigating whether KGE models rely on such structural properties for their prediction, as failing to do so may lead to overestimating their true predictive capabilities.

Finally, data leakage can occur when both training and test data share similar data distributions, which differ from the distribution of inference data encountered at model deployment. This type of leakage often stems from sampling bias, where the test set fails to reflect real-world inference conditions, leading to inflated performance metrics that do not represent actual deployment capabilities (5). In a KG, entities may exhibit varying levels of connectivity, with some entities having numerous documented interactions and others having only sparse connections. When test triples are randomly sampled from the KG, their entity connectivity distribution is likely to mirror that of the training data. However, in practice, inference might involve entities with connectivity patterns that differ significantly from those observed in the training set, such as rare diseases or chemical compounds with limited representation in the training data. This potential discrepancy motivates the need to evaluate model performance under conditions that reflect realistic inference scenarios by designing test sets that better resemble real deployment settings and explicitly limiting unintentional similarities with the training data.

To the best of our knowledge, existing benchmarks for biomedical KG analysis have not examined data leakage issues thoroughly. Some benchmarks implemented measures to mitigate contamination of the test set by redundancies with the training set. For instance, (9) ensured that known reverse relations, such as "is_a" and "is_a_reverse," were allocated within the same split. In another example, (10) proposed an evaluation protocol based on "cold-start" triples, which ensured that the models were evaluated on entirely new and unseen combinations of entities and relations. Similarly, (11) presented an evaluation framework termed "free-setting", in which all triples involving the relation to be predicted were removed from the training set. Lastly, (12) highlighted potential biases in inferring links for underrepresented entities in biomedical KGs, but these biases were not empirically assessed for their impact on model performance. Overall, while these efforts addressed certain aspects of data leakage, they did not systematically identify and mitigate the full range of leakage sources. Addressing this gap is essential for improving the robustness and reliability of models and benchmarks in the biomedical domain.

In this study, we propose strategies to systematically evaluate the impact of the three sources of data leakage described in (5). Our framework intends to enable comprehensive assessments of KGE models' performance under controlled data leakage conditions, with particular emphasis on drug repurposing scenarios. We develop specific experimental protocols for evaluating each source of data leakage:

- **Data Leakage 1 (DL1): Redundancy between Training and Test Sets.** We introduce a data leakage mitigation procedure that removes redundancies between training and test sets. By comparing KGE model performance with and without this data leakage mitigation procedure, we quantify the extent to which test set results are artificially inflated due to redundant information across the splits.
- **Data Leakage 2 (DL2): Use of Illegitimate Features.** We investigate whether KGE models exploit node degree as a proxy feature for link prediction. To this end, we train models on two versions of the dataset: the original KG and a permuted version where indication triples are randomly shuffled while each node's original degree is maintained. By comparing model performance across these two settings, we assess the extent to which node degree alone can drive predictions, despite the disruption of meaningful biological associations.
- **Data Leakage 3 (DL3): Sampling Bias in Test Set Construction.** We examine how test set sampling strategies can lead to performance evaluations that do not reflect real-world inference challenges in drug repurposing. To this end, we define three sets on which we evaluate KGE models: (i) a random test set, built by randomly sampling indication triples from the KG; (ii) a cold-start test set, also sampled from the KG, but constructed such that test drugs (or diseases) do not appear in any indication triples in

the training data; and (iii) an external real-world inference set, composed of curated drug-rare disease indications from the Orphanet database (13). By comparing model performance across these three sets, we assess whether commonly used test set constructions provide realistic conditions for evaluating drug repurposing models.

Material and Methods

Definitions

Knowledge Graph and Knowledge Graph Embedding

A Knowledge Graph (KG) is a directed multi-relational graph, where entities and their relationships are represented as triples (h, r, t). In this representation, h and t are the head and tail entities, while r is the directed relation connecting them.

Knowledge Graph Embeddings (KGE) models encode entities and relations from a KG into continuous vector spaces that capture latent patterns in the KG, facilitating downstream tasks such as link prediction. The embeddings are learned by comparing the scores of positive and negative triples, with negative triples generated through a process called Negative Sampling. A detailed formalization of KG and KGE within the encoder-decoder framework is available in the Supplementary Material.

Link prediction task

In the link prediction task, the goal is to complete a missing triple within the KG. Specifically, link prediction aims to predict either t in $(h, r, ?)$ or h in $(?, r, t)$. Each KGE model employs its own scoring function, which is used to quantify the plausibility of a triple (h, r, t). During evaluation, given a head h (resp. tail t) and a relation r, the decoder evaluates all possible candidates for the tail t (resp. head h) and ranks them based on the scores. The highest-ranked entity can then be selected to complete the triple, thus allowing the inference of missing links in the KG.

Training, Test, and Inference Sets

In this study, we define the following sets of triples:

- *Training Set:* The training set refers to triples sampled from the KG and used during the model training phase. In practice, this set is composed of two subsets: the training subset, which is directly used for optimizing the model parameters, and the validation subset, which is used to tune hyperparameters and prevent overfitting. For simplicity, we collectively refer to these subsets as the training set, unless the distinction is explicitly made.
- *Test Set:* The test set consists of triples sampled from the KG that were set aside and hidden during the training phase. For each triple in the test set, either the head or tail entity is masked, and the model's ability to rank the correct entity among all possible entities is evaluated.
- *Inference Set:* In real-world deployment, an inference set would consist of incomplete triples in which either the head or tail entity is unknown and needs to be predicted. Here, we leverage an external dataset containing known drug-rare disease indications curated from Orphanet (Orphanet Drug-Rare Disease Indications). We use this dataset as an inference set with a ground-truth to evaluate the performance of the models. We expect this analysis to reflect realistic drug repurposing applications for rare diseases.

Cold-Start Split

In KGE for link prediction, a cold-start split refers to an evaluation setting in which models are tested on entities that do not appear in any training triples involving the relation of interest (hereafter referred to as the target relation). This is typically implemented by constraining all KG triples involving a given head–relation or tail–relation pair (e.g., all $(\text{drug}, \text{indication}, \cdot)$ triples for a given drug, or all $(\cdot, \text{indication}, \text{disease})$ for a given disease) to be assigned to the same split. By enforcing this constraint on the test set, models are evaluated under more realistic conditions, where predictions must be made for entity-relation combinations with no prior associations in the training data.

Data

Knowledge Graph

We used ShepherdKG, a biomedical KG dedicated to rare disease diagnosis (14). This KG was derived from PrimeKG (15), a KG developed for precision medicine applications, and includes additional knowledge from Orphanet (13), a comprehensive database for rare diseases. ShepherdKG contains 10 entity types and 30 relationship types (Supplementary Tables S1 and S2). The KG dataset was divided as follows: 90% of the triples were assigned to the training set (including 10% of the triples reserved as the validation subset); the remaining 10% of the triples were allocated to the test set.

Orphanet Drug-Rare Disease Indications

To create the inference set, we obtained access to Orphanet's (13) dataset of Orphan drugs indicated for rare diseases, comprising 4915 drug-rare disease associations. Of these, 829 associations involved drug and disease entities present in the ShepherdKG. These 829 associations were retained as the inference set. The majority of the excluded associations involved gene therapies, which are entities not included in ShepherdKG, explaining the significant reduction. Among the 829 retained drug-disease associations, 164 were already included in ShepherdKG and were removed from the KG to ensure they were only present in the inference dataset.

Models and Implementation

Models

We used the TorchKGE Python library (16), which proposes implementations for several state-of-the-art KGE models. We investigated two types of KGE models, namely translational models (TransE (17) and TransD (18)) and bilinear models (RESCAL (19), DistMult (20), ANALOGY (21), ComplEx (22), and HolE (23)). Translational models represent relations as transformations in the embedding space, typically aiming to capture the relationship between entities through geometric operations like translations or rotations in the embedding space. In contrast, bilinear models use multiplicative interactions between embeddings, often through tensor operations (24).

Negative Sampling

Three negative sampling strategies are implemented in TorchKGE:

- In *Uniform Negative Sampling*, either the head or the tail of a triple is replaced with a randomly selected entity, following a uniform probability distribution across all entities in the KG;

- In *Bernoulli Negative Sampling*, the likelihood of replacing the head or tail of a triple depends on the mapping characteristics of the relation (i.e., one-to-many, many-to-one or many-to-many), aiming to minimize the generation of false negatives (25);
- In *Positional Negative Sampling*, the head or tail entity is replaced by an entity sampled from the set of entities that have already appeared at the same position (i.e., head or tail) in a triple involving the same relation. This strategy has the advantage of producing more realistic negative triples than Uniform Negative Sampling (26).

To take advantage of each negative sampling approach, we implemented a strategy that generates multiple types of negative triples for each positive triple. For each positive triple, we generated five negative triples using Uniform Negative Sampling, five negative triples using Bernoulli Negative Sampling, and one negative triple using Positional Negative Sampling.

Evaluation metrics

To evaluate the models on link prediction tasks on the validation set (i.e. the subset of the training set), test set, and inference set, we used the filtered Mean Reciprocal Rank (MRR), a metric that measures the quality of ranked predictions by assessing the position of the first correct entity among all candidates ranked by the decoder. Higher MRR values indicate better performance, reflecting the model's effectiveness in prioritizing correct links. Additionally, we occasionally report the filtered Hit@10 metric, which measures the proportion of correct entities ranked among the top 10 predictions.

Model Training

All models were trained using PyTorch (27) and PyTorch-Ignite (28) libraries. The training process employed the Adam optimizer, configured with a learning rate of 0.001 and a weight decay of 0.001. All models were trained with an embedding dimension of 200 for both entities and relations, and a batch size of 4096. For translational models, we used the Margin Loss with the margin set to 1. For bilinear models, we used the Binary Cross-Entropy Loss. All models were trained in two settings:

- **Standard training (run 1)**: Models were evaluated every 20 epochs on the validation set using the MRR metric. Early stopping was triggered if the validation MRR did not improve for 10 consecutive evaluations (i.e., 20 × 10 epochs), with training limited to a maximum of 500 epochs. The model with the highest validation MRR during training was saved for evaluation on the test set.
- **Scheduled training (run 2)**: Identical to standard training, but with the addition of a Cosine Annealing Warm Restarts learning rate scheduler (T_0=10, T_mult=2) to dynamically adjust the learning rate during training.

For all experiments, we reported the best MRR of the two trainings on the test set. Full MRR and Hit@10 results are available in Supplementary Tables 3–6.

Hardware and computational environment

All experiments were conducted on the Jean Zay supercomputer using a single NVIDIA V100 GPU with 32 GB of memory.

DL1 - Redundancy between Training and Test Sets

In KGE-based link prediction tasks, redundancy between training and test sets due to inadequate data separation (called here Data Leakage 1, DL1) can arise from two distinct scenarios, as described by *Akrami et al.* (7).

First, when splitting the dataset into training and test sets, overlapping information between the splits may go undetected. This happens when the KG contains semantically related relations, known as near-duplicate or near-reverse-duplicate relations (7). For example, in a biomedical KG, relations like "(protein)-binds to-(protein)" and "(protein)-interacts with-(protein)" can be considered *near-duplicates* as they convey similar meanings. Similarly, reverse relations, such as "(drug)-inhibits-(protein)" and "(protein)-is inhibited by-(drug)" can be considered as *near-reverse-duplicates* as they convey symmetric semantic meanings. When these semantically related relations are randomly distributed across training and test sets without proper control, they create unwanted information leakage. Overall, failing to identify and account for semantically redundant relations can compromise the separation of the training and test data. As proposed by *Akrami et al.*, semantically redundant relations can be detected by investigating whether, for two given relations, the heads and tails overlap exceeds predefined thresholds θ_1 and θ_2. These two thresholds account for asymmetric redundancy, where one relation may be redundant with respect to another without implying mutual redundancy (7). The complete mathematical formulation of this redundancy detection process is described in the Supplementary Materials (Equations S1 and S2).

Second, the KG may include *Cartesian product* relations, in which a large proportion of possible head-tail entity pairs are connected. In a biomedical KG, an example of such Cartesian product relation could be "(gene)-expressed in-(cell type)" for ubiquitously expressed genes (i.e. housekeeping genes). Cartesian product relationships can be identified by comparing the number of observed connections between the heads and tails covered by the relation with the total number of possible connections, and determining whether this proportion exceeds a predefined threshold θ (7). The specific mathematical formulation for this identification is provided in the Supplementary Materials (Equation S3).

DL1 Detection and Control Procedure

In order to detect and control DL1, we designed a preprocessing procedure to identify semantically redundant and Cartesian product relations within the KG, and ensure appropriate separation between training and test sets. This protocol is divided into several steps described below.

Step 1: Filtering Reverse Triples For every relationship type r, if a triple (a, r, b) exists in the KG, its reverse triple (b, r, a) is removed from the KG. This step addresses cases where the same relation r is used in both directions, which can be problematic since some KGE models (such as TransE and TransD) cannot encode symmetric relationships, while others (like DistMult) inherently impose symmetry on all relations. For undirected relationships, triple directionality is handled separately in Step 3 of this procedure.

Step 2: Identifying Semantically Redundant and Cartesian Product Relations For every possible pair of relations (r_1, r_2), we compute their semantic redundancy to identify near-duplicate and near-reverse-duplicate relations, using thresholds θ_1 and θ_2 set to 0.8. Additionally, for every relationship r, Cartesian product relations are identified using a threshold θ set to 0.8. Relations identified as semantically redundant are flagged for subsequent processing.

Step 3: Handling Directionality for Undirected Relations While most KGE models operate on directed triples, biomedical KGs often contain inherently undirected relationships. To accommodate these undirected interactions in our models, we systematically generate reverse triples with a new relation name. For any undirected relation r, we add a reverse counterpart r_{rev}. Specifically, for every triple (a, r, b), we add (b, r_{rev}, a). Because r and r_{rev} convey the same semantic information, the pair (r, r_{rev}) is flagged and added to the list of semantically redundant relations identified in Step 2.

Step 4: KG Split with Controlled Train-Test Separation At this stage, semantically redundant relations and Cartesian product relations have been identified and flagged. The KG is split into training and test sets, with the following considerations:

- For semantically redundant relations flagged in Step 2, as well as relationships made directed by the addition of their reverse in Step 3, redundant triples are removed from the training set. Specifically, for any triple placed in the validation or test set, its redundant counterparts are excluded from the training set.
- For Cartesian product relations flagged in Step 2, we ensure that all triples involving a given head (or tail) with the Cartesian relation r are placed in the same split.

Evaluation of DL1

To evaluate the impact of DL1 on KGE model performance, we compared link prediction results obtained on ShepherdKG with and without the application of the DL1 detection and control procedure described above. To create the dataset with DL1, steps 1 to 3 of the DL1 detection and control procedure were applied, but instead of using step 4's controlled splitting procedure, the data was randomly split into training and test sets. To create the dataset without DL1, the full DL1 detection and control procedure was applied, including step 4, thereby ensuring that no redundant triples remained between the training and test sets.

For both datasets, KGE models were trained using the same hyperparameters and evaluation metrics for consistency. Link prediction performance was measured for the prediction of all relations (i.e., global performance), for the prediction of relations flagged in the DL1 detection and control procedure (i.e., performance on semantically redundant and Cartesian product relations), and for the prediction of the indication relation.

DL2 - Investigating Node Degree as an Illegitimate Predictive Feature

To evaluate whether KGE models rely on node degree as an illegitimate feature for predicting new links (called here Data Leakage 2, DL2), we designed a permutation experiment. The underlying rationale is that if models utilize node degree as a predictive shortcut rather than learning meaningful semantic patterns, their performance should remain relatively stable when connections are randomized while preserving entity degrees. Formally, for a given relation r, we randomly permuted the tail entities while preserving the degrees of the heads and tails. Specifically, for a relation r connecting the set of heads H and the set of tails T, each tail in a triple is replaced by selecting a new entity randomly from the set T.

For this experiment, we used the ShepherdKG where DL1 had been controlled using the DL1 detection and control procedure described in DL1 Detection and Control Procedure. The permutation was applied to the indication relation between steps 3 and 4 of the DL1 detection and control pipeline. We compared the performance of KGE models trained on the permuted versus original version of the KG.

DL3 - Sampling Bias in Test Set Construction

To investigate the presence and impact of sampling bias in test set construction (called here Data Leakage 3, DL3) in the context of drug repurposing, we compared KGE model performance on indication triples from the test set (i.e., triples sampled from the KG) and on an external inference set of drug-rare disease indications (Orphanet Drug–Rare Disease Indications) reflecting realistic deployment scenarios. This comparison aimed to reveal the impact of dependencies that may exist between training and test sets, but are not expected between training and inference sets. We conducted this analysis using two data splitting strategies: a standard random split and a cold-start split.

Standard Random Split

We used the same models as those employed for the DL1 experiment, trained on ShepherdKG where DL1 had been controlled through the procedure described in DL1 Detection and Control Procedure. In this experiment, we reported only model performance on predicting indication triples from the test set (i.e. indication triples sampled randomly from the original KG). We compared these performances with the performances obtained in the inference set composed of curated drug-rare disease indications described in Orphanet Drug–Rare Disease Indications. This setup allowed us to evaluate whether models trained and evaluated on indication triples randomly sampled from ShepherdKG can generalize to predict drug-rare disease indications.

Cold-Start Split

We implemented cold-start splitting procedures applied to the indication relation in two distinct settings: a drug-based cold-start, and a disease-based cold-start.

In the drug-based cold-start, all indication triples involving a given drug were assigned to the same split. Similarly, in the disease-based cold-start, all indication triples involving a given disease were assigned to the same split. These procedures ensured that models were required to predict indications for drugs or diseases that had not appeared in any indication triples during training.

To maintain a balanced evaluation, we stratified the target entities (drugs or diseases) into frequency bins based on the number of associated indications, and distributed them proportionally across the training, validation, and test sets. This preserved a representative mix of frequent and rare entities across splits, while ensuring that each entity appeared in indication triples within only one split. To verify that this stratification yields comparable levels of difficulty across splits, we analyzed the distribution of known indications per entity in each subset. As shown in Supplementary Figure 1, the distributions were well balanced, indicating that frequent and rare entities were evenly represented across training and test sets.

We then trained KGE models separately on the drug-based and disease-based cold-start splits, and evaluated each model on its corresponding test set and on the orphan drug inference set. This allowed us to assess whether removing dependencies between training and test improved the alignment between test and inference performance—i.e., whether cold-start splitting

led to more realistic and reliable evaluation of generalization for drug repurposing tasks.

Availability

The ShepherdKG used in this study is available at https://zitniklab.hms.harvard.edu/projects/SHEPHERD/.

The dataset of orphan drugs is available upon request from Orphanet at https://www.orphadata.com/samples/.

All code and results are openly available on GitHub at https://github.com/galadrielbriere/drug_repurposing_kge_benchmark. The repository provides both the implementation to reproduce the results obtained in this study and a reusable framework for assessing data leakage in any KG, including the DL1 detection and control procedure, the DL2 permutation procedure, the DL3 cold-start split procedure, and the complete models and training implementations.

Results

Experiment 1: Assessing the Impact of Redundancy between Training and Test Sets (DL1) on Model Performance

Existing benchmarks on biomedical KGs using similar KGE models and reporting MRR values show MRR ranging from 0.1 (12) to 0.8 (9), and even up to 0.9 (29). This large range in performance may stem from differences in KG structure and model training protocols. However, DL1 (i.e., presence of redundancy between training and test data) has not been systematically investigated in these benchmarks, leaving open the question of whether data leakage contributes to these observed performance disparities.

To assess the impact of DL1 on model performance, we trained and evaluated seven KGE models (Material and Methods) on two variants of ShepherdKG: the ShepherdKG where only the steps 1 and 3 of the DL1 detection and control procedure (Material and Methods) were applied, followed by random train-test splitting (i.e., KG with DL1), and the ShepherdKG where all four steps of the DL1 detection and control procedure were applied to remove any redundancy between training and test sets (i.e., KG without DL1).

The application of the DL1 detection and control procedure on the ShepherdKG resulted in the removal of $1,336,637$ reverse triples at Step 1. Most of these removed triples $(1,336,314)$ were drug-drug relations, as for every $(h, \text{drug-drug}, t)$, the reverse triple $(t, \text{drug-drug}, h)$ was also included in the KG. At Step 2, we did not identify any semantically redundant or Cartesian product relations (using θ_1, θ_2, and θ set to 0.8). Because most KGE models consider directed relations, three undirected relations were made directed at Step 3: drug-drug similarity (*drug_drug* relation), disease-disease similarity (*disease_disease* relation), and protein-protein interactions (*protein_protein* relation). For these relations, the reverse was included with a different relation name (e.g., *drug_drug_rev*) to avoid introducing symmetric relationships. Including these reverse triples created a semantic redundancy in the KG, which was removed at Step 4 of the DL1 detection and control procedure for the KG controlling for data leakage but retained for the KG where data leakage was not controlled.

Our results revealed that redundancies between training and test data could substantially inflate model performance, as measured by global MRR (Figure 1-a). However, the extent of this inflation varied across models and may depend both on the intrinsic capacity of the model and its ability to exploit the leakage. Among the best performing models, ComplEx showed the most pronounced impact of DL1: the global MRR dropped from 0.46 to 0.30 (a 34.7% decrease), when DL1 was removed. ANALOGY and HolE also exhibited performance decreases when DL1 was controlled (dropping from 0.41 to 0.32 and 0.39 to 0.32 respectively).

In contrast, RESCAL maintained high performance regardless of DL1 (with an MRR dropping from 0.41 to 0.38, making it the best-performing model when DL1 was removed). TransE also showed only marginal difference (from 0.23 to 0.22) but with lower overall performance.

Finally, DistMult and TransD both exhibited low overall performance and minimal change when DL1 was controlled.

We hypothesize that the apparent insensitivity of DistMult, TransE, and TransD to DL1 control might be due to their limited capacity to capture complex patterns in the data. Indeed, both DistMult and TransE were evaluated in Akrami et al.'s study (7), and were found to be impacted by DL1. However, the KG considered in our study, with approximately 5.5 million triples, is substantially larger than the non-biomedical KGs evaluated in Akrami's study, containing between 150000 and 1 million triples. This might indicate that in our study, these models reached a performance plateau relatively early, potentially making them less sensitive to DL1 control.

The relatively strong performance of RESCAL with or without DL1 control suggests that this model might be less impacted by DL1 compared to other high-performing methods. RESCAL computes embeddings through tensor factorization, capturing global relationships between entities (30). This collective learning mechanism might reduce the tendency to overfit specific patterns in the training data, thereby mitigating the effects of DL1.

In addition to analyzing the global MRR, which considers all the KG's relations, we computed the MRR for relations that were semantically redundant between the training and test sets. These semantically redundant relations are causing DL1 (Figure 1-b). For these relations, the gap between the reported performance of models trained with and without DL1 was even more pronounced, as expected. For ComplEx, ANALOGY and HolE, the MRR decreased by 46%, 32%, and 20%, respectively, when DL1 was removed. RESCAL, however, continued to report nearly equal MRR values across the two settings (decreasing from 0.52 to 0.47, and remaining the best performing model when DL1 is controlled), similar to its performance for the global MRR. DistMult, TransE, and TransD showed poor performances, regardless of the presence of DL1, with MRR values below 0.15.

These results clearly indicate that some KGE models are sensitive to DL1, and that the top-performing models are particularly affected. Indeed, our findings highlight the significant impact of DL1 on the evaluation of KGE models, particularly for semantically redundant relations, where its presence can lead to artificially high evaluation metrics. These findings underscore the critical importance of controlling DL1 during model evaluation.

In addition, we wondered whether DL1-affected relations could potentially help the prediction of non-DL1 affected relations. Specifically, the indication relation was not semantically redundant in the KG (i.e. not affected by DL1). We tested if the presence of DL1-affected relations in the KG might, in some way, assist in predicting the non-DL1 affected indication triples. The results in Figure 1-c were inconclusive. The prediction performance was overall low in all models, reaching 0.27 at

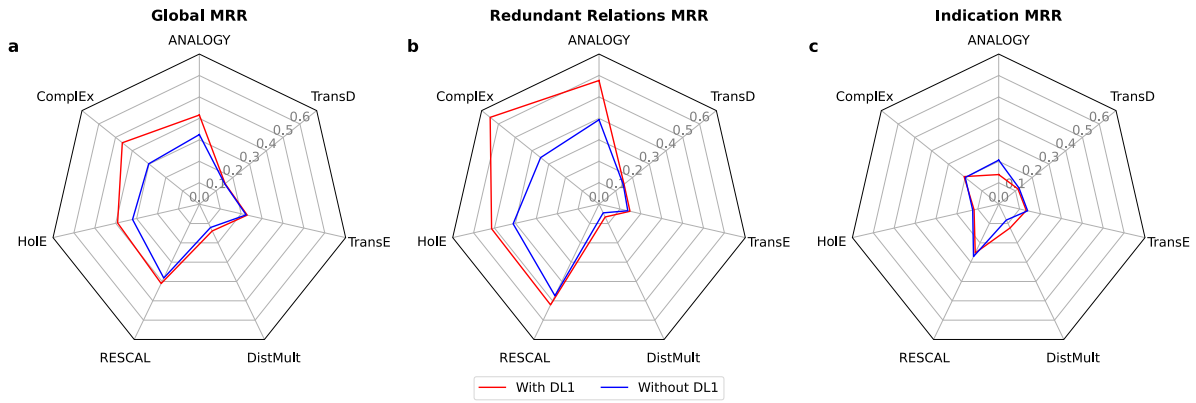

Figure 1. Impact of Training-Test Set Separation on KGE Model Performance (DL1). **(a)** Global MRR: performance of the models across all relations in the test set. **(b)** MRR for Redundant Relations: Model performance on the relations flagged as redundant by the DL1 detection and control procedure. **(c)** MRR for the Indication Relation: Model performance on indication triples. Performance shown with (red) and without (blue) DL1.

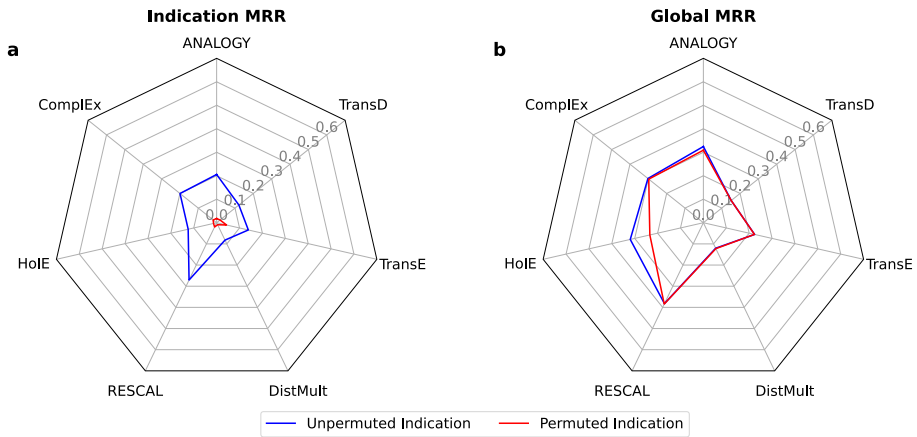

Figure 2. Assessment of node degree exploitation as an illegitimate feature (DL2) in indication prediction. **(a)** MRR for the Indication Relation: Model performance on indication triples. **(b)** Global MRR: Model performance across all relations in the test set. Performance shown with original indications triples (blue) and degree preserving permutation for indication triples (red).

best (RESCAL model in DL1 controlled setting). These results suggest that introducing semantic redundancy through DL1, in addition to compromising the reliability of model evaluation, does not provide any consistent benefits for predicting non-redundant relations like indication.

Experiment 2: Investigating Node Degree as an Illegitimate Predictive Feature (DL2)

A second potential source of data leakage we examined was the use of node degree as an illegitimate feature. To test this data leakage source in the context of indication prediction, we designed an experiment in which the indication triples in the KG were permuted while preserving the original degree of entities (Material and Methods). Our hypothesis was that if the models rely on node degree as an illegitimate feature, the MRR for the indication relation should remain similar between models trained on the original and on the permuted KGs.

Our results, presented in Figure 2-a, indicated that the models did not rely on node degree as a predictive feature. Indeed, models trained on the KG with permuted indication triples exhibited significantly poorer predictive performance for this relation as compared to models trained on the original KG. However, permuting the indication triples did not affect model performance on predicting other types of relations, as the global MRR was similar for most models (Figure 2-b). These findings suggest that the indication relation does not provide information used by the models to predict other types of relations. However, an exception was observed for HolE, which showed a notable decrease in global MRR when trained on the KG permuted for the indication relation (decreasing from 0.32 to 0.23). This suggests that HolE may be more sensitive to disruptions in the internal structure of individual relations, potentially due to its reliance on circular correlation of entity embeddings (23).

Evaluating Sampling Bias in Test-Set Construction and Its Effect on Evaluation Reliability (DL3)

The reliability and utility of the performance metrics obtained on the test set depends on how well the test set correctly represents real-world inference scenarios. In standard practice, test sets are randomly sampled from the same KG that is used for training. As a result, test and training sets often share similar statistical and structural properties. However, at deployment time, the models might encounter entities and relationships with substantially different characteristics compared to those observed during training and evaluation. These differences can potentially lead to overoptimistic estimates of model performance.

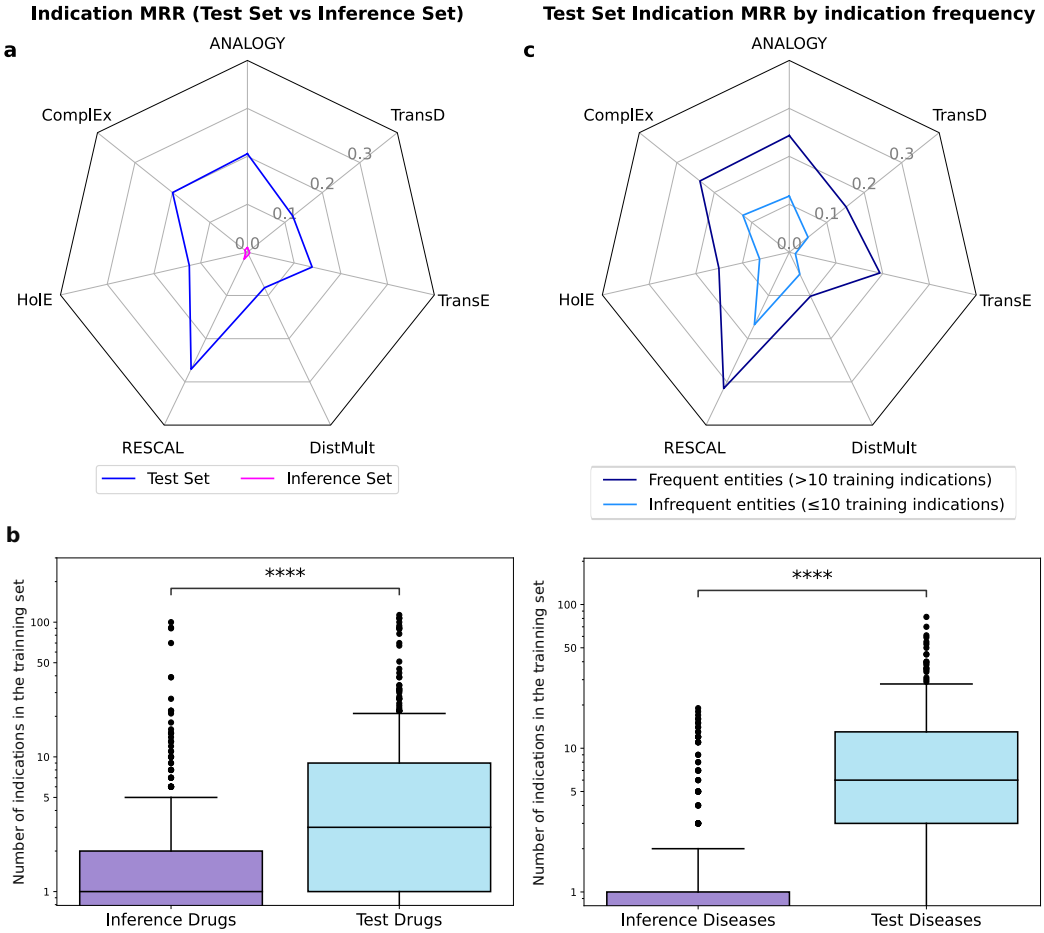

Figure 3. Random Split Test Set and Inference Set Comparisons (DL3) **(a)** MRR comparison between test set of randomly sampled indication triples (blue) and inference set indication triples (pink). **(b)** Distribution of the number of training set indications (shown on a logarithmic scale) per drugs and per diseases across test and inference sets. Statistical significance was assessed using the Mann-Whitney-Wilcoxon test (two-sided). Annotations indicate significance levels: **** ($P \leq 1 \times 10^{-4}$). **(c)** MRR comparison on test *indication* triples involving entities with frequent (> 10) or infrequent (≤ 10) occurrences in training *indication* triples (dark and light blue, respectively).

In this experiment, we investigated whether performance on such randomly sampled test sets reliably predicted performance on an external inference set composed of drug-rare disease indications from Orphanet (13) (Orphanet Drug-Rare Disease Indications). In addition to the random split, we further considered a cold-start splitting strategy, where all known indications for a given drug or disease were assigned to the same training or test split. The resulting cold-start test split is expected to better reflect realistic drug repurposing scenarios, where models are expected to make predictions for understudied drugs or diseases with few or no known indications in the training data.

Random Split

We first examined whether evaluating models on test data randomly sampled from the same KG as the training data (Standard Random Split) provided reliable performance indicators for real-world drug repurposing inference. To do so, we compared the performance of the models obtained on the test set and on our inference set of drug-rare disease indications (Orphanet Drug-Rare Disease Indications).

The results, shown in Figure 3-a, indicated that while the best-performing models achieved MRRs around 0.2 on the test set (RESCAL, ComplEx, ANALOGY), their performance on the inference set was close to 0. The best-performing model on the inference set, RESCAL, achieved an MRR of only 0.016, with a Hit@10 of 0.031, meaning that, on average, the correct entity was ranked among the top 10 predictions in just 3.1% of the cases. In comparison, the same model achieved a Hit@10 of 0.46 on the test set indications, successfully ranking the correct entity in the top 10 predictions in nearly half of the cases.

To understand the reasons behind such poor generalization on the inference dataset, we analyzed the connectivity patterns of drug and disease entities. For each entity belonging to either the test or the inference set, we computed the number of indication triples it participates in in the training set. We found that drugs and diseases from the test set had significantly higher numbers of known indications in the training set as compared to drugs and diseases from the inference set (Figure 3-b). Additionally, we computed the total degree of each entity in the training set—that is, the number of triples it was involved in across all relation types (Supplementary Figure 2). Inference drugs had a lower median total degree (529) compared to test drugs (1196), and inference diseases were also slightly less connected (median degree of 33 vs. 41 for test diseases). These results confirmed that inference entities were globally

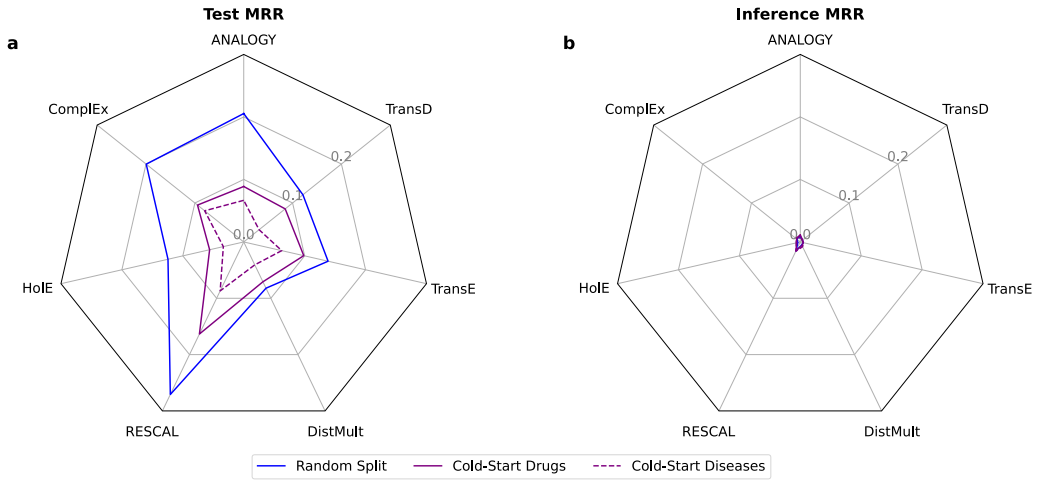

Figure 4. Evaluating Test and Inference Performance under Random vs. Cold-Start Splits (DL3). **(a)** MRR comparison on the test set: randomly sampled indication triples (blue), drug-based cold-start indication triples (solid dark purple) and disease-based cold-start indication triples (dashed dark purple). **(b)** MRR comparison on the inference set: randomly sampled indication triples (blue), drug-based cold-start indication triples (solid dark purple) and disease-based cold-start indication triples (dashed dark purple).

less connected in the training KG, which may partly explain the models' poor generalization ability.

We further investigated whether test triples involving drugs or diseases with more frequent indications in the training set were predicted more accurately than those involving rarer ones. To do so, we selected indication triples from the test set and split them into two groups: triples in which either the drug or the disease appeared in more than 10 indication triples in the training set (frequent group, 590 triples), and triples where both entities had 10 or fewer indication triples (infrequent group, 254 triples). As shown in Figure 3-c, all models performed significantly better on test triples involving entities (drugs or diseases) that appeared in more indication triples in the training set.

These experimental results demonstrate significant limitations of current KGE models for drug repositioning applications. While these models achieve reasonable performance on the test set, their predictive capabilities exhibit a strong dependence on the number of known indications in the training data involving the specific drug or disease under consideration. The models systematically underperform when predicting indications for drugs or diseases with few known indications and show particularly poor generalization to rare disease applications. The reliance on a high number of known indications severely limits the models' practical applicability, as the identification of novel indications for less-studied drugs and diseases represents a critical need.

Cold-start Splits

To assess whether test set construction can better reflect the challenges of real-world drug repurposing inference, we designed a cold-start splitting protocol that we applied on the indication relation, as described in Cold-Start Split. We implemented two variants of this protocol: a drug-based cold-start, where all indication triples involving a given drug were assigned to a single split, and a disease-based cold-start, where the same constraint was applied to diseases. This setup ensured that test entities had no associated indications in the training set, thus mimicking inference conditions for novel or understudied drugs or diseases.

As shown in Figure 4-a, models achieved higher performance on the test set under the random split compared to both cold-start configurations, which was expected since the random setting allows drugs and diseases to be partially known through their training indications. In contrast, the cold-start settings presented a more challenging prediction task.

We observed a substantial and consistent drop in test performance when moving from the random to the cold-start splits. The two best-performing models on the random split, RESCAL and ANALOGY, achieved MRRs of 0.27 and 0.21, respectively. These values dropped to 0.16 and 0.09 in the drug-based cold-start setup, and further down to 0.09 and 0.07 in the disease-based cold-start, where the task proved even more difficult. Similar trends were observed for all other models, confirming that the performance degradation were not model-specific but reflected a general limitation of current KGE approaches. This consistent drop indicates that KGE models rely heavily entity-specific exposure to the indication relation during training to make accurate predictions, and struggle to generalize when drugs or diseases lack such exposure despite appearing in other contexts.

Nevertheless, models still retained some predictive capacity in the cold-start scenario. RESCAL, for instance, reached an MRR of 0.16 (hit@10 = 0.29) in the test set of the drug-based cold-start split. This suggests that even in the absence of indication triples for the considered drugs, structural signals from other parts of the graph remain informative. Still, the persistent performance gap between random and cold-start test sets emphasizes the difficulty of generalizing to new drug or new disease indications.

Figure 4-b shows model performance on our inference set. The inference performance remained low regardless of the test set sampling strategy. The best model in all settings, RESCAL, achieved at most an MRR of 0.016 and hit@10 of 0.03 (in the disease-based cold-start), comparable to the scores observed using random sampling.

These results suggest that even though cold-start split offers a more realistic and challenging evaluation framework, it is not sufficient to bridge the gap between test performance and real-world inference capability.

Taken together, these findings call into question the reliability of standard test set construction strategies, such as random or cold-start sampling, for estimating the real-world inference capabilities of KGE models in drug repurposing scenarios.

Discussion

Data leakage poses a critical challenge in the evaluation of KGE models, as it can lead to inflated performance metrics that do not reflect real-world applicability. In this study, we systematically assessed the impact of three sources of data leakage on the performance of popular KGE models for link prediction in a biomedical KG. Our findings highlight the necessity of systematically identifying and mitigating different sources of data leakage to ensure a realistic assessment of model performance. As KGE models remain largely black boxes, transparent reporting of measures to address data leakage is crucial to improve the reliability and reproducibility of KGE-based link prediction. We advocate for the adoption of standardized practices to evaluate and mitigate data leakage, not only in KGE studies but also in the broader machine learning community.

In our first experiment, we demonstrated that data redundancy between training and test sets (DL1) can result in artificially inflated performance metrics, which fail to reflect the true capabilities of models obtained when such redundancy is controlled. This underscores the importance of adopting consistent and transparent evaluation protocols across studies. We developed and implemented a systematic protocol for detecting and controlling DL1 when performing link prediction tasks. We make this protocol publicly available to the research community and encourage its adoption to ensure robust evaluation of KGE models. Additionally, we identified specific challenges related to undirected relations in KGs. Indeed, the conversion of undirected relations to directed relations through the inclusion of reverse relationship, which is necessary because most KGE models operate exclusively on directed relations, can introduce DL1 issues. While thorough cleaning of the training set as we proposed here can mitigate redundancy caused by reverse relations, optimal strategies for handling undirected relations remain an open question. Further investigations are needed to determine whether treating undirected triples as directed in their original orientation or including reverse relations with controlled redundancy yields superior results.

Our second experiment, dedicated to investigating whether KGE models use node degree as an illegitimate feature (DL2) for their predictions, did not find evidence of node degree being used as a proxy. However, this does not rule out the possibility that other illegitimate features might be exploited by the models and remain to be identified. For instance, one potential area for further investigation is the role of community structure within the KG. Community structure, characterized by densely connected subgraphs, may facilitate prediction shortcuts by encouraging models to rely on local connectivity instead of capturing more general semantic relationships. Exploring to which extent KGE models might be biased by such structural features would be an important next step to better understand their true capabilities.

In our third experiment, we demonstrated that models performing well on the test set obtained via a random split of the initial KG failed to generalize to an external inference dataset corresponding to a real-world drug repurposing inference scenario in rare diseases. This generalization gap was further demonstrated by our analysis of prediction performance stratified by drug and disease indication frequencies in the training data, which revealed significantly better performance for drugs or diseases with more than 10 known indications. This frequency-dependent pattern suggests that the models primarily rely on local structural cues and direct exposure to training data, rather than learning representations that support broader generalization. For drugs (or diseases) with numerous known indications, we speculate that the models tend to predict diseases (or drugs) that are topologically close to those seen during training. Such behavior suggests that the models' generalization capacity is largely confined to the neighborhoods of well-represented entities, rather than enabling inference across distant or disconnected regions of the KG.

To investigate whether test set construction itself contributes to this performance mismatch, we implemented cold-start evaluation protocols that explicitly prevent test drugs or diseases from being observed during training in the context of the target relation. Under these configurations, test performance was lower than in the random split, yet the discrepancy between test and inference performance remained substantial. This persistent mismatch suggests that current evaluation strategies do not adequately reflect the generalization challenges involved in real-world drug repurposing inference.

Altogether, our results raise concerns about the suitability of current KGE methods for link prediction tasks, particularly in the context of drug repurposing. Even when trained and evaluated under stricter, more realistic conditions, these models show poor performance on real-world inference tasks. This calls into question the validity of commonly reported performance metrics and highlights the need for fundamentally different modeling approaches or task formulations in this domain. Nevertheless, KGE models may still serve a valuable role as compact representations of large biomedical knowledge graphs. Rather than using them as standalone predictors, they could be leveraged as input features in more comprehensive learning frameworks. In this perspective, KGE models would not be used for link prediction directly, but to produce representations that could be integrated with other data modalities or inference strategies.

A limitation of our study is that we did not include deep learning-based KGE models, particularly those using Graph Neural Networks (31; 32). However, their reliance on local message passing might inadvertently amplify data leakage by spreading redundant information across the graph (DL1), and may also reinforce reliance on structural biases such as node degree (DL2). Sampling bias and test set representativeness (DL3) remain challenges regardless of the model architecture, and would likely affect GNN-based approaches in similar ways. Moreover, GNN act as encoders and must still be coupled with a decoder (for instance those evaluated in this study) to perform link prediction. It therefore remains unclear whether they would significantly improve generalization in realistic inference settings. Evaluating their behavior under controlled protocols remains an important direction for future work.

Competing interests

BL is employed by Servier and declares no competing interests in relationship with this manuscript. All remaining authors declare no competing interests.

Author contributions statement

G.B. and A.B. conceived the study. G.B. implemented, conducted the experiments, and analyzed the results. T.S. developed the proof of concept for the study. B.L. reviewed and contributed to the implementation. A.B. supervised the project. G.B. and A.B. wrote the manuscript. All authors reviewed and approved the final manuscript.

Acknowledgments

This work was supported by MarMaRa (Marseille Maladies Rares), the French National Research Agency through the CAMUDI project (ANR-21-CE45-0001), and the European Rare Diseases Research Alliance (ERDERA). This project was provided with computer and storage resources by GENCI at IDRIS thanks to the grant 2024-AD010315786 on the supercomputer Jean Zay's V100 and A100 partition. We acknowledge Orphanet and INSERM for providing the dataset of orphan drugs used in this study. We thank Elias Ventre for his thorough manuscript review and valuable scientific input. We used large language models (LLMs) to assist in coding and in polishing the writing of the manuscript.

References

1. The Gene Ontology Consortium *et al.*, "The gene ontology knowledgebase in 2023," *Genetics*, vol. 224, no. 1, 2023.
2. S. Köhler *et al.*, "The human phenotype ontology in 2021," *Nucleic acids research*, vol. 49, no. D1, pp. D1207–D1217, 2021.
3. Z. Chen *et al.*, "Knowledge graph completion: A review," *Ieee Access*, vol. 8, pp. 192 435–192 456, 2020.
4. D. N. Sosa *et al.*, "Contexts and contradictions: a roadmap for computational drug repurposing with knowledge inference," *Briefings in bioinformatics*, vol. 23, no. 4, p. bbac268, 2022.
5. J. Bernett *et al.*, "Guiding questions to avoid data leakage in biological machine learning applications," *Nature Methods*, vol. 21, no. 8, pp. 1444–1453, 2024.
6. S. Kaufman *et al.*, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 4, pp. 1–21, 2012.
7. F. Akrami *et al.*, "Realistic re-evaluation of knowledge graph completion methods: An experimental study," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1995–2010.
8. J. Bernett *et al.*, "Cracking the black box of deep sequence-based protein–protein interaction prediction," *Briefings in Bioinformatics*, vol. 25, no. 2, p. bbae076, 2024.
9. D. Chang *et al.*, "Benchmark and best practices for biomedical knowledge graph embeddings," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2020. NIH Public Access, 2020, p. 167.
10. R. Celebi *et al.*, "Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings," *BMC bioinformatics*, vol. 20, pp. 1–14, 2019.
11. M. Alshahrani *et al.*, "Application and evaluation of knowledge graph embeddings in biomedical data," *PeerJ Computer Science*, vol. 7, p. e341, 2021.
12. S. Zheng *et al.*, "Pharmkg: a dedicated knowledge graph benchmark for bomedical data mining," *Briefings in bioinformatics*, vol. 22, no. 4, p. bbaa344, 2021.
13. S. Pavan *et al.*, "Clinical practice guidelines for rare diseases: the orphanet database," *PloS one*, vol. 12, no. 1, p. e0170365, 2017.
14. E. Alsentzer *et al.*, "Few shot learning for phenotype-driven diagnosis of patients with rare genetic diseases," *medRxiv*, 2024.
15. P. Chandak *et al.*, "Building a knowledge graph to enable precision medicine," *Scientific Data*, vol. 10, no. 1, p. 67, 2023.
16. A. Boschin, "Torchkge: Knowledge graph embedding in python and pytorch," in *International Workshop on Knowledge Graph: Mining Knowledge Graph for Deep Insights*, Aug 2020.
17. A. Bordes *et al.*, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.
18. G. Ji *et al.*, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.
19. M. Nickel *et al.*, "A three-way model for collective learning on multi-relational data." in *Icml*, vol. 11, no. 10.5555, 2011, pp. 3 104 482–3 104 584.
20. B. Yang, Yih *et al.*, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.
21. H. Liu *et al.*, "Analogical inference for multi-relational embeddings," in *International conference on machine learning*. PMLR, 2017, pp. 2168–2178.
22. T. Trouillon *et al.*, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.
23. M. Nickel *et al.*, "Holographic embeddings of knowledge graphs," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
24. T. Shen *et al.*, "A comprehensive overview of knowledge graph completion," *Knowledge-Based Systems*, vol. 255, p. 109597, 2022.
25. Z. Wang *et al.*, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.
26. R. Socher *et al.*, "Reasoning with neural tensor networks for knowledge base completion," *Advances in neural information processing systems*, vol. 26, 2013.
27. A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.
28. V. Fomin *et al.*, "High-level library to help with training neural networks in pytorch," https://github.com/pytorch/ignite, 2020.
29. C. Zhou, C.-P. Cai, X.-T. Huang, S. Wu, J.-L. Yu, J.-W. Wu, J.-S. Fang, and G.-B. Li, "Tarkg: a comprehensive biomedical knowledge graph for target discovery," *Bioinformatics*, vol. 40, no. 10, p. btae598, 2024.
30. D. Krompaß *et al.*, "Non-negative tensor factorization with rescal," in *Tensor Methods for Machine Learning, ECML workshop*, 2013, pp. 1–10.
31. T. N. Kipf *et al.*, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
32. P. Velickovic *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.