Counterfactual Reasoning with Knowledge Graph Embeddings

Anonymous ACL submission

Abstract

Knowledge graph embeddings (KGEs) were originally developed to infer true but missing facts in incomplete knowledge repositories. In this paper, we link knowledge graph completion and counterfactual reasoning via our new task CFKGR. We model the original world state as a knowledge graph, hypothetical scenarios as edges added to the graph, and plausible changes to the graph as inferences from logical rules. We create corresponding benchmark datasets, which contain diverse hypothetical scenarios with plausible changes to the original knowledge graph and facts that should be retained. We develop *COULDD*, a general method for adapting existing knowledge graph embeddings given a hypothetical premise, and evaluate it on our benchmark. Our results indicate that KGEs learn patterns in the graph without explicit training. We further observe that KGEs adapted with COULDD solidly detect plausible counterfactual changes to the graph that follow these patterns. An evaluation on human-annotated data reveals that KGEs adapted with COULDD are mostly unable to recognize changes to the graph that do not follow learned inference rules. In contrast, ChatGPT mostly outperforms KGEs in detecting plausible changes to the graph but has poor knowledge retention. In summary, CFKGR connects two previously distinct areas, namely KG completion and counterfactual reasoning.

1 Introduction

Reasoning about hypothetical situations (*counterfactual reasoning*) and anticipating the effects of a change in the current state of the world is central to human cognition (Rafetseder and Perner, 2014; Van Hoeck et al., 2015), and has been identified as a key concept in game theory (Aumann, 1995; Halpern, 1999) and agent-based systems (Icard et al., 2018; Parvaneh et al., 2020). It has even been argued that the capacity to reason about alternative configurations of the world could be a pre-requisite to the existence of free will and a sense of agency (McCarthy, 2000; Kulakova et al., 2017). Recently, there has been an increased interest in evaluating and improving counterfactual reasoning of AI systems, in particular, large language models (LLMs) (Qin et al., 2019; Frohberg and Binder, 2022; Li et al., 2023).

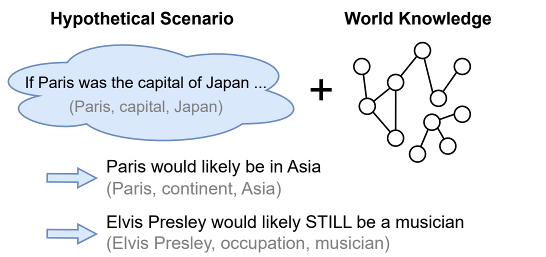


Figure 1: A hypothetical scenario and its implications, expressed in the language of knowledge graph triples

Knowledge graphs (KGs) express rich information about the world as an explicit collection of triples, such as (Paris, capital, France), and knowledge graph embeddings (KGEs) effectively infer true but missing facts from incomplete knowledge repositories (Hogan et al., 2021; Ji et al., 2021). Yet, to the best of our knowledge, KGEs have not been explored for counterfactual reasoning.

In this work, we link counterfactual reasoning to knowledge graph completion (KGC) via our new task *CFKGR*[1] (**C**ounter**F**actual **KG** **R**easoning) which requires models to classify the validity of facts given a hypothetical scenario. CFKGR describes the original world state as a KG and hypothetical scenarios as edges that are added to the graph. The hypothetical scenario leads to the emergence of new facts in the KG while leaving (most) already existing ones intact. Figure 1 illustrates a hypothetical scenario in which Paris is the capital of Japan. To perform well on CFKGR, models must be capable of detecting plausible additions

[1]The data and code is included in this submission and will be made publicly available upon acceptance.

to the graph, e.g., (Paris, continent, Asia), while maintaining knowledge of unaffected facts, e.g., (Elvis Presley, occupation, musician). We create the first benchmark datasets for CFKGR, which are based on the CoDEx KGC benchmark (Safavi and Koutra, 2020) and provide diverse hypothetical scenarios with corresponding plausible additions to the KG derived from *inference rules* (that were mined from the KG (Lajus et al., 2020)). We validate our data-generating process and underlying assumptions via thorough human annotation. Lastly, we introduce *COULDD* (**COU**nterfactual Reasoning with Know**L**edge Graph Embe**DD**ings), a method which updates existing KGEs based on counterfactual information. COULDD follows a standard KGE training scheme using the hypothetical scenario and negative sampling. Training stops once the hypothetical scenario is classified as valid.

In our experiments, COULDD is initialized with five different KGE methods. We observe that it can detect plausible counterfactual changes to the graph that follow prominent inference patterns in the KG while maintaining performance on unaffected triples. We repeat the same experiments with ChatGPT, i.e., gpt-3.5-turbo, provided with similar prompts to human annotators. ChatGPT performs better at detecting plausible additions to the graph than most KGE-based methods but exhibits poor knowledge retention. Qualitative analysis of answers provided by ChatGPT shows that it largely failed to understand the task on retained facts as it tried to infer them from the provided information. Evaluating on human-annotated data leads to a drop in overall performance for KGEs and ChatGPT alike. To summarize, our main contributions are as follows:

- We propose CFKGR, a challenging task for counterfactual reasoning on KGs and create corresponding, partially human-verified, datasets, which we make publicly available.
- We introduce COULDD, a general method for adapting existing KGE methods to make inferences given hypothetical scenarios and show that it improves reasoning on counterfactual graphs over pre-trained embeddings.
- We compare counterfactual reasoning with KGEs to ChatGPT and show that ChatGPT outperforms KGEs in detecting plausible counterfactual inferences but struggles to recall unrelated knowledge, unlike COULDD.

2 CFKGR: Task Description

We introduce *Counterfactual KG Reasoning* (*CFKGR*) a novel task to assess the ability of machine learning systems to reason in hypothetical scenarios. CFKGR describes the originally observed world state as a knowledge graph and introduces hypothetical scenarios by adding previously unseen facts to the graph. To perform well on CFKGR, models need to (1) identify plausible changes to the original world state induced by the hypothetical scenario and (2) understand which facts are unaffected by the hypothetical scenario.

2.1 Definition of Counterfactual Graphs

Formally, CFKGR defines the original world state via a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations represented in the knowledge graph. The fact set $\mathcal{F}$ represents our knowledge about the world as triples $(h, r, t) \in \mathcal{F} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. We denote a hypothetical scenario by a triple $\tau^c := (h, r, t) \notin \mathcal{F}$. The *counterfactual graph*, in which τ^c holds, is then characterized by the fact set $\mathcal{F}^c := \mathcal{F} \setminus \mathcal{F}^- \cup \mathcal{F}^+$, where $\mathcal{F}^+$ denotes the facts that emerge given the hypothetical scenario, and $\mathcal{F}^-$ denotes facts that contradict the scenario and cannot hold any longer. We say τ^c *changes* a triple τ if either $\tau \in \mathcal{F}^+$ or $\tau \in \mathcal{F}^-$.

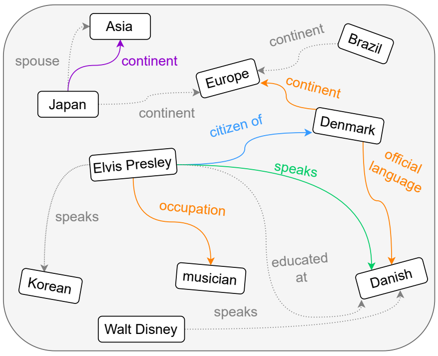
In the following, we formulate the assumptions underlying our task.

Closed-world assumption. We adopt the standard closed-world assumption (Reiter, 1978), which states that facts that are not part of the KG, i.e., $\tau \notin \mathcal{F}$, are false. *Thus, each $\tau \notin \mathcal{F}$ is a possible hypothetical scenario in our setup.*

Logic-world assumption. We assume that plausible changes to the graph largely follow some regularity and can hence be modeled via (potentially very complex) logical rules. While available rule sets have limited coverage and precision, we can leverage them to model a subset of plausible changes to a KG. *By employing the logic-world assumption, we can represent an approximation of $\mathcal{F}^c$ via a set of rules and the original fact set.*

2.2 Evaluation

We formulate CFKGR as a binary classification task. Triples $\tau \in \mathcal{F}^c$ receive label 1, while all other triples are labeled 0. Since scoring all possible triples is infeasible, we consider a smaller set of carefully chosen test cases. Given a counterfactual



Instance	Notation	Original KG	CF KG
Counterfactual	τ^c	$\tau^c \notin \mathcal{F}$	$\tau^c \in \mathcal{F}^c$
Inference	τ^i	$\tau^i \notin \mathcal{F}$	$\tau^i \in \mathcal{F}^c$
Unchanged (near)	τ^n	$\tau^n \in \mathcal{F}$	$\tau^n \in \mathcal{F}^c$
Unchanged (far)	τ^f	$\tau^f \in \mathcal{F}$	$\tau^f \in \mathcal{F}^c$
Corruptions	$\tau_{h'}, \tau_{t'}, \tau_{r'}$	$\tau_{h'}, \tau_{t'}, \tau_{r'} \notin \mathcal{F}$	$\tau_{h'}, \tau_{t'}, \tau_{r'} \notin \mathcal{F}^c$

Figure 2: Overview over the types of facts, given the hypothetical scenario that Elvis Presley is a citizen of Denmark. The green edge (Elvis Presley, speaks, Danish) emerges from adding the blue edge (Elvis Presley, citizen of, Denmark) to the knowledge graph. Purple and orange edges are present in the original KG and unaffected by the scenario. Grey edges are neither present in the original nor the counterfactual knowledge graph.

$\tau^c \notin \mathcal{F}$ and a rule, we define:
(1) a **counterfactual inference** τ^i that follows from a rule and allows us to measure whether the model can correctly predict changes to the graph given τ^c,
(2) **retained facts** which are unaffected by the hypothetical scenario and should still be classified as valid in the counterfactual graph,
(3) random **head, tail, and relation corruptions** of inferences and retained facts, which ensure that the model does not score unsolicited triples as valid additions. We denote the corruptions for a triple τ by $\tau_{h'}$, $\tau_{t'}$ and $\tau_{r'}$.
For (2), we distinguish between **near facts** τ^n, which are in the one-hop neighborhood of τ^c, and **far facts** τ^f, sampled from its complement. Figure 2 illustrates a counterfactual scenario and its associated test cases.

We use the following metrics to evaluate the performance on our benchmark. Concrete formulations of the scores are in Appendix A.
(1) We compute the **F1-score** score over all test cases in the dataset to measure the overall predictive performance on counterfactual graphs.
(2) We measure the **accuracy on changed facts**, i.e., triples that have a different label before and after the hypothetical scenario is introduced.
(3) We compute the **F1-score on unchanged facts**, i.e., triples that have the same label before and after the hypothetical scenario is introduced.

3 CFKGR: Dataset Creation

For our dataset construction, we leverage rules found by rule mining systems, which capture prominent patterns in KGs. Automatically mined rules are compatible with the content of the KG and are known to be a useful tool for KGC (e.g., Meilicke et al., 2019; Sadeghian et al., 2019a). Since there is no trivial way to *reliably* generate $\mathcal{F}^-$, we only consider the addtions $\mathcal{F}^+$. Concretely, we define $\mathcal{F}^+$ via mined *composition rules* of the form

$$(X, r_1, Y) \wedge (Y, r_2, Z) \rightarrow (X, r_3, Z) \qquad (1)$$

where $r_1, r_2, r_3 \in \mathcal{R}$. We refer to $(X, r_1, Y) \wedge (Y, r_2, Z)$ as the *rule body* and (X, r_3, Z) as the *inference*. The triples (X, r_1, Y) and (Y, r_2, Z) are called the first and second body atom, respectively. Replacing X, Y, and Z by concrete entities $x, y, z \in \mathcal{E}$ creates an *instantiation* of the rule. In the following, we will use the short-hand notation (r_1, r_2, r_3) to denote a rule as described in (1). We choose composition rules since they are well studied in standard KG completion benchmarks (Safavi and Koutra, 2020) and inferential benchmarks (Cao et al., 2021; Liu et al., 2023). Moreover, composition rules, as given in (1), infer *local changes*. This is desirable since most relevant changes induced by a hypothetical scenario will occur in its close neighborhood. We consider understanding the implications induced by composition rules as a first step to more general and complex hypothetical reasoning.

3.1 Data Generating Process

In the following, we give a high-level overview of our data generating process and focus on creating hypothetical scenarios for the first body atom of a given rule. Appendix C provides a detailed description and the full algorithm.

Given a knowledge graph and a rule set, we generate several hypothetical scenarios for each

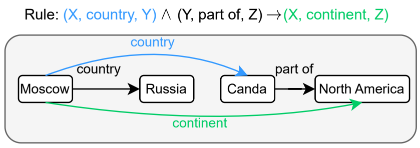


Figure 3: Creation of a hypothetical scenario.

	Valid		Test	
	Rules	Facts	Rules	Facts
CFKGR-CoDEx-S	5	3600	12	8848
CFKGR-CoDEx-M	5	3936	26	19584
CFKGR-CoDEx-L	5	4000	39	30064

Table 1: CFKGR dataset overview. "Rules" denotes the number of rules that were used to create the dataset. "Facts" is the total number of test cases.

rule by altering a fact in the KG such that it triggers the rule, as is illustrated in Figure 3. Concretely, for each rule (r_1, r_2, r_3), we search for existing edges $e_1 := (x, r_1, y) \in \mathcal{F}_{train}$ and $e_2 := (\bar{y}, r_2, z) \in \mathcal{F}_{train}$, ensuring that the resulting hypothetical scenario $\tau^c := (x, r_1, \bar{y})$ and inference $\tau^i := (x, r_3, z)$ are not in the original KG. Sampling e_1 and e_2 without any constraints can however result in nonsensical scenarios and inferences. Hence, we ensure that the entities in τ^c and τ^i are suitable for the given relation by checking whether they were observed as a head or tail of said relation (depending on their position in the triple) in the original KG. Once suitable τ^c and τ^i are found, we randomly sample *two* near facts τ^n from the one-hop neighborhood[2] of τ^c and *one* far fact τ^f from its complement.

When creating head and tail corruptions of a given fact, we restrict the sample space since random corruptions, which tend to result in nonsensical triples, have previously been shown to be easily detectable for KGE methods (Safavi and Koutra, 2020). For head (tail) corruptions, we require that the replacements are also heads (tails) for the relation in the original graph. For relation corruptions, we do not employ additional constraints.

3.2 CFKGR-CoDEx

Based on the procedure described in Section 3.1, we create the first benchmark datasets for CFKGR-based on the CoDEx knowledge graph completion benchmark (Safavi and Koutra, 2020). We choose CoDEx since it covers diverse content, uses easily interpretable relations, and contains rich auxiliary information, such as entity types. CoDEx provides three knowledge graphs of varying sizes (S, M, and L), collected from Wikidata (Vrandečić and Krötzsch, 2014) and corresponding *composition rules obtained by the rule-mining system Amie3* (Lajus et al., 2020). CoDEx-S and CoDEx-M additionally contain verified negative triples. An overview over the resources provided by CoDEx can be found in Appendix B.

We leverage the available Amie3 patterns for each CoDEx dataset as our rule set and create at most 25 unique counterfactual triples per body atom[3] for each rule. We subsequently split them into a validation and test set, ensuring that there are no overlapping rules or counterfactuals between validation and test. Table 1 provides statistics about the created datasets.

In the following section, we will explore how well the resulting test cases align with *human counterfactual reasoning*.

3.3 Human Annotation

We validate our data generating process via human annotation. For each of the 31 rules in CFKGR-M, we verify 10 test instances (5 per atom[4]). We annotate τ^i, τ^f, τ_1^n, τ_2^n and $\tau_{r'}^i$, and omit the remaining corruptions as their construction relies on the commonly-used closed-world assumption (Reiter, 1978). This results in 1530 annotated instances, which were labeled by *four to six independent annotators*[5] as either *likely* (1), *unlikely* (0), or *unsure/too little information* (-1), given verbalizations of the hypothetical scenario and context triggering the respective inference rule. We observe a Krippendorff's alpha of 0.508, which indicates moderate agreement (Landis and Koch, 1977). The annotation guidelines are in Appendix D. Table 2 summarizes the annotation results.

Inferences seem to be the most difficult category to annotate as they show the highest amount of ties and "unsure/too little information" labels. Moreover, we observe the highest number of deviations from our expected label for this test case. This indicates that rules that were mined for *fac-*

[2]Except for the context triggering the rule. This choice was made due to the setup of the human dataset verification.

[3]For some rules, our constraints only allow for fewer unique counterfactuals.

[4]Except for one rule which only produced one unique counterfactual according to our conditions for the second atom.

[5]All annotators have a Bachelor's or Master's degree in a STEM field.

	# Labeled	Expected	As expected	Majority Vote Label 0	1	-1	Tied
Inference	306	1	59.5%	57	182	21	46
Far fact	306	1	99.7%	0	305	0	1
Near fact	612	1	95.4%	18	584	1	9
Relation corr.	306	0	86.3%	264	19	4	19

Table 2: Annotation results. "# Labeled" denotes the number of annotated examples and "Expected" gives the label assigned by our process. "As expected" gives the percentage of samples for which the expected label coincides with the majority vote.

tual knowledge graph completion cannot always be used for *human-like* counterfactual reasoning.

On relation corruptions, we observe a noticeable number of inferences that are not implied by our rules, but are still considered valid by humans or are at least debatable. Possible explanations are the limited coverage of the rule set or unintuitive verbalizations. On the remaining categories, we obtain a label distribution that largely agrees with our assumptions.

4 Counterfactual Reasoning with Knowledge Graph Embeddings

KGE models find low-dimensional vector representations for entities and relations while preserving the information contained in the KG. To judge the plausibility of a given triple, KGE models use a scoring function $\phi(h, r, t) : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$. A triple is typically classified as valid if it satisfies $\phi(h, r, t) \geq \mu_r$, for a relation-specific threshold $\mu_r \in \mathbb{R}$.

Data: $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, data $\mathcal{D}$, params θ_0, #iterations E, #samples N, LR α, thresholds $\mu_1, \mu_2, ..., \mu_{|\mathcal{R}|}$
Result: CFKGR predictions
$\hat{y} \leftarrow \{\}$
foreach $(\tau^c, \mathcal{T}_{\tau^c}) \in \mathcal{D}$ **do**
  $\theta \leftarrow \theta_0$
  for $e \in \{1, ..., E\}$ **do**
    $S \leftarrow$ Sample N from $\mathcal{F}_{train}$
    $B \leftarrow \{\tau^c\} \cup S$
    $\theta \leftarrow$ Optimizer$(\mathcal{L}_\theta(B), \alpha)$
    if $\phi_\theta(\tau^c) \geq \mu_r$ **then**
      break
  $\hat{y} \leftarrow \hat{y} \cup \{\phi_\theta(\mathcal{T}_{\tau^c})\}$
return $\hat{y}$

Algorithm 1: COULDD training and prediction. The short-hand notation $\phi_\theta(\mathcal{T}_{\tau^c})$ denotes scoring all test cases associated with τ^c and $\mathcal{L}_\theta$ denotes the cross-entropy loss.

We propose *COULDD* (**COU**nterfactual Reasoning With Know**L**edge Graph Embe**DD**ings), a general method for adapting existing knowledge graph embeddings given hypothetical scenarios. COULDD is initialized from existing embeddings trained on the original KG. For each hypothetical scenario, these embeddings are updated and subsequently evaluated on the corresponding test cases.

COULDD's update scheme only minimally changes standard KGE training: In each iteration, the existing embeddings are fine-tuned on a batch consisting of the counterfactual triple τ^c and N additional randomly sampled edges from the training graph. Negative training examples are generated by randomly corrupting the head and tail entities of each triple in the batch. Updates are done with the standard cross-entropy loss. Once the counterfactual triple τ^c exceeds the classification threshold, the training is stopped in order to avoid an excessive perturbation of the pre-trained embeddings[6].

Importantly, COULDD only requires access to the counterfactual triple τ^c and the original fact set $\mathcal{F}$ and does not need additional task-specific training data or information about the rules used to generate CFKGR datasets. As a result, *COULDD can also be applied in rule-free evaluation setups*. Algorithm 1 provides a formal description of COULDD.

5 Experiments

In the following, we conduct two types of experiments: First, we evaluate pre-trained KGEs, COULDD, and ChatGPT on our CFKGR datasets with expected labels to assess whether the methods can apply inference rules found by a rule mining system in hypothetical scenarios. In our second set of experiments, we evaluate on human-labeled data to check whether the methods also capture human reasoning, which does not necessarily align with mined inference rules (see Section 3.3).

5.1 General Setups

We use the five pre-trained CoDEx link-prediction models as initializations for COULDD[7]. Further details about the KGE methods are in Appendix E.

For COULDD, we tune the learning rate (α) and number of additional samples per batch (N) on the respective CFKGR validation set, based on

[6]Note that there is no traditional validation set for the individual updates on which we could perform early stopping.

[7]The config files for the models are available at https://github.com/tsafavi/codex

	CFKGR-CoDEx-S			CFKGR-CoDEx-M			CFKGR-CoDEx-L		
	F1	Changed	Unchanged	F1	Changed	Unchanged	F1	Changed	Unchanged
RESCAL	60.82	27.12	63.28	63.05	21.57	66.92	53.84	71.47	$\mathbf{49.64}$
COULDD-RESCAL	$\mathbf{61.68 \pm 0.14}$	$\mathbf{32.48 \pm 0.73}$	$\mathbf{63.48 \pm 0.16}$	$\mathbf{63.85 \pm 0.08}$	$\mathbf{26.23 \pm 0.16}$	$\mathbf{67.16 \pm 0.07}$	$\mathbf{53.94 \pm 0.02}$	$\mathbf{84.56 \pm 0.35}$	48.18 ± 0.06
TransE	58.94	23.15	61.87	53.61	23.61	$\mathbf{55.83}$	49.23	66.31	45.37
COULDD-TransE	$\mathbf{60.49 \pm 0.12}$	$\mathbf{26.8 \pm 0.81}$	$\mathbf{63.16 \pm 0.09}$	$\mathbf{53.91 \pm 0.05}$	$\mathbf{26.06 \pm 0.25}$	55.79 ± 0.06	$\mathbf{52.6 \pm 0.06}$	$\mathbf{76.56 \pm 0.25}$	$\mathbf{47.77 \pm 0.04}$
ComplEx	62.45	29.11	64.90	65.69	11.60	$\underline{\mathbf{71.83}}$	58.44	65.51	$\mathbf{55.26}$
COULDD-ComplEx	$\underline{\mathbf{67.76 \pm 0.3}}$	$\mathbf{37.94 \pm 0.67}$	$\underline{\mathbf{69.95 \pm 0.29}}$	$\underline{\mathbf{66.78 \pm 0.06}}$	$\mathbf{34.67 \pm 0.23}$	69.21 ± 0.07	$\mathbf{59.44 \pm 0.02}$	$\mathbf{82.95 \pm 0.26}$	54.25 ± 0.02
ConvE	61.04	16.64	65.39	$\mathbf{56.83}$	13.15	$\mathbf{61.37}$	55.56	$\mathbf{61.84}$	52.58
COULDD-ConvE	$\mathbf{61.51 \pm 0.11}$	$\mathbf{16.96 \pm 0.72}$	$\mathbf{65.92 \pm 0.12}$	52.69 ± 0.16	$\mathbf{17.04 \pm 0.16}$	56.09 ± 0.16	$\underline{\mathbf{60.6 \pm 0.17}}$	45.53 ± 0.61	$\underline{\mathbf{60.29 \pm 0.14}}$
TuckER	64.25	15.01	$\mathbf{69.40}$	65.21	13.15	$\mathbf{70.98}$	52.87	76.74	$\mathbf{48.05}$
COULDD-TuckER	$\mathbf{66.03 \pm 0.13}$	$\mathbf{35.99 \pm 1.0}$	68.09 ± 0.19	$\mathbf{66.09 \pm 0.17}$	$\mathbf{43.69 \pm 0.38}$	66.95 ± 0.17	$\mathbf{53.53 \pm 0.04}$	$\underline{\mathbf{88.47 \pm 0.34}}$	47.49 ± 0.02
gpt-3.5-turbo	47.81	$\underline{68.9}$	40.20	36.06	$\underline{52.12}$	31.16	45.74	52.10	40.89

Table 3: Test performance of pre-trained embeddings and COULDD on CFKGR. For COULDD, we report the mean and standard deviation across 5 runs. Bold entries denote the best performance between pre-trained KGEs and their counterpart trained with COULDD. The best results on the dataset are underlined. For all scores, higher is better.

the best overall F1, and set the maximum number of update steps (E) to 20. We carry over the remaining hyperparameters from the pre-trained CoDEx models (Safavi and Koutra, 2020). Further details regarding the hyperparameters are in Appendix F.2. Optimization is performed using Adam (Kingma and Ba, 2014), or Adagrad (Duchi et al., 2011), depending on the original model configuration. The relation-specific decision thresholds are carried over from the CoDEx triple classification benchmark. Since no negatives are provided for CoDEx-L, we generate one random tail corruption per validation triple (as in (Safavi and Koutra, 2020)) for threshold tuning. During training, we sample 100 negative examples per triple (50 head and 50 tail corruptions), as this was effective in previous work (Trouillon et al., 2016; Kotnis and Nastase, 2017).
We implement our experiments using LibKGE (Broscheit et al., 2020) and Optuna (Akiba et al., 2019). For experiments with ChatGPT, i.e., *gpt-3.5-turbo*, we use the OpenAI API and temperature 0. Find the used prompts and an example of input and output in Appendix F.3.

5.2 Results

Table 3 contains the results. A detailed evaluation per test case can be found in Appendix G. First, we observe that the KGE performances on CFKGR-CoDEx-L differ noticeably from CFKGR-CoDEx-S and CFKGR-CoDEx-M. This is likely due to lower threshold quality resulting from the absence of hard negative triples for CoDEx-L.
COULDD achieves the best results in terms of overall F1-score on all datasets. In particular, COULDD noticeably improves the performance on changed facts over the pre-trained embeddings, except for ConvE. Importantly, we do not observe a case where applying COULDD leads to catastrophic forgetting. In fact, the additional update steps made by COULDD improve the scores on unchanged facts in many cases.
In terms of overall F1-score, COULDD-ComplEx achieves the best results averaged across the three datasets. On changed facts, COULDD-TuckER is the best-performing KGE method, likely because TuckER is well-suited for modeling compositional relations (Safavi and Koutra, 2020). ChatGPT achieves the best scores on changed facts on two out of three datasets. However, it generally does not perform well on unchanged facts. Possible reasons are that it misses background knowledge present in the KG or does not understand the task on these instances.

In summary, we observe that COULDD consistently improves performance over the pre-trained embeddings, overall and on changed facts in particular, and does not strongly degrade performance on unchanged facts. This indicates that COULDD can be used to infer plausible counterfactual changes to the graph when they follow prominent patterns in the KG.

5.3 Case Study on CoDEx-M

To better understand the results shown in Table 3, we conduct a case study on CoDEx-M for which we have a human-annotated CFKGR subset. In particular, we want to assess how well the pre-trained CoDEx models perform *factual* reasoning with composition rules and how an evaluation on human-assigned labels affects our results. The main results are presented in Table 4. Table 12 in the appendix presents a confusion matrix per test type for COULDD and ChatGPT.

	CFKGR-CoDEx-M*						CoDEx-M (filtered)	
	F1 (E)	F1 (H)	Changed (E)	Changed (H)	Unchanged (E)	Unchanged (H)	Overall	Rule-wise
RESCAL	**89.1**	**87.3**	21.8	14.1	**97.2**	**96.2**	92.7	84.7
COULDD-RESCAL	88.8 ± 0.2	86.8 ± 0.2	**25.2 ± 0.7**	**16.5 ± 0.6**	96.5 ± 0.2	95.3 ± 0.2	–	–
TransE	**81.0**	**79.7**	22.3	17.8	**88.5**	**87.3**	91.3	80.3
COULDD-TransE	80.5 ± 0.1	79.3 ± 0.1	**24.1 ± 0.3**	**20.4 ± 0.4**	87.6 ± 0.1	86.9 ± 0.1	–	–
ComplEx	88.7	87.1	10.6	3.8	**98.3**	**97.4**	96.0	77.8
COULDD-ComplEx	**91.9 ± 0.1**	**90.2 ± 0.2**	**38.0 ± 1.0**	**30.8 ± 1.0**	98.3 ± 0.1	97.3 ± 0.1	–	–
ConvE	**83.7**	**82.3**	15.4	10.3	**92.4**	**91.6**	89.3	79.7
COULDD-ConvE	78.1 ± 0.6	76.9 ± 0.7	**16.8 ± 1.0**	**13.2 ± 0.8**	86.1 ± 0.6	85.4 ± 0.7	–	–
TuckER	89.0	87.7	13.8	8.1	98.2	**97.5**	96.4	90.3
COULDD-TuckER	**92.7 ± 0.1**	**90.7 ± 0.1**	**43.7 ± 0.6**	**34.8 ± 0.7**	**98.4 ± 0.1**	97.2 ± 0.1	–	–
gpt-3.5-turbo	64.0	63.5	53.7	55.1	62.6	62.3	–	–

Table 4: Case study on CFKGR-CoDEx-M* with expected (E) and human-assigned (H) labels and performance on the filtered CoDEx-M test set. "Overall" describes the accuracy across all inferences. "Rule-wise" gives the average accuracy per rule. Bold entries denote the best performance between pre-trained KGEs and their counterpart trained with COULDD. The best results on the dataset are underlined. For all scores, higher is better.

5.3.1 Inference Rules in Factual Contexts

Achieving good performance on changed triples in Table 3 requires (1) a logical adaption to the hypothetical scenario and (2) the application of composition rules that generated the test inferences. We attempt to disentangle these factors by investigating whether the CoDEx models captured regularities during pre-training also expressed in the Amie3 rules and can apply them in *factual* scenarios.

Setup. We filter the original CoDEx-M test set for triples that can be inferred by applying the mined Amie3 rules to the training set. We only keep triples that stem from rules that cover at least five triples in the test set to obtain sensible estimates for the rule-wise performances. This results in a filtered test set of 551 instances inferred from 10 rules.

Results. We notice that the performance on the filtered CoDEx-M test set is consistently high for all pre-trained KGE methods. This indicates that they learned the mined inference patterns during training and lower performances observed in Table 3 are likely due to insufficient adaption to the hypothetical scenario. Appendix H.2 gives a full overview of rule-wise performances.

5.3.2 Evaluation with Manual Labels

Setup. We create a subset of the CFKGR-CoDEx-M test set, denoted by **CFKGR-CoDEx-M***, which solely consists of instances that were manually labeled as described in Section 3.3. We filter out test cases for which less than three annotators agreed or the majority vote label was "unsure/too little information". This results in a dataset composed of 256 hypothetical scenarios and a total of 1171 test cases. The complete label distribution per test case can be found in Appendix H.1.

Results. First, we note that the overall F1-score and performance on unchanged facts greatly differ from the numbers observed for the unfiltered set in Table 3. This is due to the omission of most corruptions as they were not manually labeled. We observe a consistent performance drop for KGE-based methods when evaluating on human-assigned labels instead of expected labels for all metrics. Judging from the confusion matrix in Table 12, COULDD cannot reliably identify *false inferences* that follow from patterns from the KG, but are invalid according to the annotators. Moreover, it classifies most *outdated facts*, which are no longer valid given the hypothetical scenario, as positive.

For ChatGPT, we observe slightly reduced overall performance when evaluating with human-assigned labels. However, ChatGPT's score improves on changed facts for human-assigned labels. A closer look at the confusion matrix reveals that ChatGPT performs better at detecting outdated facts and false inferences than KGEs. However, as observed before, ChatGPT tends to misclassify facts that should be retained. A qualitative inspection reveals that ChatGPT largely misunderstands the task on such triples: instead of answering whether they STILL hold given the hypothetical scenario, it tries to infer them. Table 5 gives an example.

6 Related Work

Inferential KGC Benchmarks. Rule-based *inferential benchmarks* for KGC (Liu et al., 2023; Cao et al., 2021) assess a method's ability to learn implict rule patterns and use them to predict inferences in the test set based on evidence in the training set. Cao et al. (2021) create an inferential

Hypothetical:	Joe Frazier (human) died in Moscow (city with millions of inhabitants, largest city, capital)
Context:	Joe Frazier (human) was cast in Encino Man (film)
Statement:	Is it then still plausible that E-40 (human) has the occupation actor (profession, artist), given that this is the case in the real world?
Answer by ChatGPT:	unlikely
Explanation by ChatGPT:	The hypothetical scenario states that Joe Frazier died in Moscow, which is a city with millions of inhabitants. This fact is not the case in the real world. The context mentions that Joe Frazier was cast in the film Encino Man. **Since the statement is 'Is it then STILL plausible that E-40 has the occupation actor?',** **we need to consider if the hypothetical scenario makes it plausible for E-40 to have the occupation of an actor.** **However, the context does not provide any specific information about E-40 or his occupation.** Therefore, we cannot determine the plausibility of E-40 being an actor based on the given information. Hence, the statement is unlikely.

Table 5: Example prompt and corresponding answer produced by ChatGPT on CFKGR-CoDEx-M*.

test set for CoDEx-M based on a rule set mined by AnyBurl (Meilicke et al., 2019), akin to our experiments in Section 5.3.1, and also find that pre-trained KGEs have strong inferential reasoning capabilities.

Counterfactual Graph Learning. Leveraging counterfactuals in graph learning is an emerging field of research (Guo et al., 2023). Counterfactuals have recently been utilized to ensure the fairness of graph-based systems with respect to sensitive node attributes (Agarwal et al., 2021; Ma et al., 2022; Zhang et al., 2021), improve interpretability by generating counterfactual explanations for predictions (Lucic, 2022; Numeroso and Bacciu, 2021; Prado-Romero et al., 2022; Xu et al., 2022), and enhance link prediction performance on the graph *as-is* (Chang et al., 2023; Lu et al., 2023; Shi et al., 2022; Wang et al., 2021; Zhao et al., 2022).

Our work does not fall into any of the above categories and instead focuses on making predictions in a counterfactual graph.

CF Reasoning Benchmarks for LLMs. Several datasets and evaluation schemes have been proposed for assessing the counterfactual reasoning capabilities of LLMs. Qin et al. (2019) introduce the task of counterfactual story rewriting, in which LLMs have to minimally revise a given story with respect to a counterfactual event. The CRASS benchmark challenges LLMs to select a valid consequence given a questionized counterfactual conditional in a multiple-choice setting (Frohberg and Binder, 2022). Li et al. (2023) present LLMs with a hypothetical premise and two possible completions for a corresponding statement, one of which is valid in the real world while the other holds in the hypothetical scenario.

In contrast, CFKGR poses a binary classification task based on the knowledge contained in a KG.

7 Discussion

Comparison with Human CF Reasoning. Our labeling efforts and experiments show that counterfactual reasoning on KGs is a challenging task. Both KGEs and ChatGPT leave much headroom for improvement on CFKGR. Moreover, even humans find it difficult to judge the plausibility of KG-based counterfactual statements, especially when they involve unfamiliar situations. For instance, "If Meg White was a member of Girls Aloud, would Jack White be part of Girls Aloud?" is a question that most humans likely do not ask themselves. Still, automatic systems can be presented with and evaluated on a wide range of possible scenarios, even if those are implausible or hard to imagine for humans.

Advantage of KG-based Benchmarks. KGs are a powerful tool for defining hypothetical scenarios and their consequences. The rich world knowledge stored in KGs allows to create interesting *case-specific* inferences. In the example question above, would the judgement change if we replace "Girls Aloud" by a band that is not a girl group? This is an aspect missing from current counterfactual reasoning benchmarks for LLMs (Frohberg and Binder, 2022; Li et al., 2023), as they mostly handle generic entities.

8 Conclusion

This work introduces the novel task CFKGR, which requires models to reason on a counterfactual KG. By utilizing the world knowledge stored in KGs, we create datasets consisting of diverse hypothetical scenarios and their implications, as defined by inference rules. Further, we propose COULDD, a general method for counterfactual reasoning on KGs and evaluate its effectiveness on automatically generated and human-annotated data. We extend our experiments to ChatGPT and find that it generally outperforms COULDD at making counterfactual inferences. However, ChatGPT largely does not recognize which facts are invariant to the hypothetical scenario. Both COULDD and ChatGPT leave much headroom on the task, highlighting the difficulty of CFKGR.

9 Limitations

The type of rules that we examine is arguably limited. We consider understanding the implications induced by composition rules as a first step to more general and complex hypothetical reasoning. Moreover, while the set of outdated facts $\mathcal{F}^-$ is a key component for defining the counterfactual KG, there is no trivial way for generating them *reliably* without appropriate rules or extensive human verification. Most rules defined for KGs are however Horn clauses (e.g., Lajus et al., 2020; Meilicke et al., 2019; Sadeghian et al., 2019b), which cannot express negation in the head atom. Hence, we focus on the addditons $\mathcal{F}^+$ in this work.

Verbalizing KG triples, in a way that is intuitive to humans, is not an easy task. We did our best to find suitable verbalizations by consulting Wikidata definitions and ParaRel (Elazar et al., 2021). Still, unintuitve verbalizations and missing context from the KG (with respect to how relations are used) might have influenced our annotation results and ChatGPT experiments.

Lastly, KGs can contain erroneous or outdated facts and automatically constructed CFKGR examples might rely on these facts. It is possible that such instances impacted the performance of ChatGPT on our benchmark. However, we estimate that such cases are rare and, as a result, the effect should be negligible.

10 Ethics Statement

We used well-established and publicly available resources to build our datasets and method. We use the CoDEx knowledge graph and LibKGE, which are both published under the MIT license. The config files for the pre-trained models used in our experiments are all available on the CoDEx github repository[8].

The counterfactual situations included in our datasets are randomly generated and purely hypothetical. They do not convey any implications about the real-world entities referenced in them. Nevertheless, the created instances could be biased towards certain entities due their occurrence in the original knowledge graphs.

We recruited annotators on a voluntary basis. We do not publish any information that could be used to identify the labelers and our data does not contain any personal information regarding the annotators.

[8] https://github.com/tsafavi/codex

References

Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*, pages 2114–2124. PMLR.

T Akiba, S Sano, T Yanase, T Ohta, and M Koyama. 2019. A next-generation hyperparameter optimization framework. In *Proceedings of ACM SIGKDD*, pages 2623–2631.

Robert J Aumann. 1995. Backward induction and common knowledge of rationality. *Games and economic Behavior*, 8(1):6–19.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174.

Yixin Cao, Xiang Ji, Xin Lv, Juanzi Li, Yonggang Wen, and Hanwang Zhang. 2021. Are missing links predictable? an inferential benchmark for knowledge graph completion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6855–6865, Online. Association for Computational Linguistics.

Heng Chang, Jie Cai, and Jia Li. 2023. Knowledge graph completion with counterfactual augmentation. In *Proceedings of the ACM Web Conference 2023*, pages 2611–2620.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Jörg Frohberg and Frank Binder. 2022. CRASS: A novel data set and benchmark to test counterfactual reasoning of large language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2126–2140, Marseille, France. European Language Resources Association.

Zhimeng Guo, Teng Xiao, Charu Aggarwal, Hui Liu, and Suhang Wang. 2023. Counterfactual learning on graphs: A survey. *arXiv preprint arXiv:2304.01391*.

Joseph Y Halpern. 1999. Hypothetical knowledge and counterfactual reasoning. *International Journal of Game Theory*, 28:315–330.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37.

Thomas Icard, Fiery Cushman, and Joshua Knobe. 2018. On the instrumental value of hypothetical and counterfactual thought. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018*. cognitivesciencesociety.org.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.

Eugenia Kulakova, Nima Khalighinejad, and Patrick Haggard. 2017. I could have done otherwise: Availability of counterfactual comparisons informs the sense of agency. *Consciousness and cognition*, 49:237–244.

Jonathan Lajus, Luis Galárraga, and Fabian Suchanek. 2020. Fast and exact rule mining with amie 3. In *The Semantic Web*, pages 36–52, Cham. Springer International Publishing.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. 2023. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 804–815, Toronto, Canada. Association for Computational Linguistics.

Shuwen Liu, Bernardo Cuenca Grau, Ian Horrocks, and Egor V Kostylev. 2023. Revisiting inferential benchmarks for knowledge graph completion. *arXiv preprint arXiv:2306.04814*.

Guoming Lu, Hao Zhang, Ke Qin, and Kai Du. 2023. A causal-based symbolic reasoning framework for uncertain knowledge graphs. *Computers and Electrical Engineering*, 105:108541.

Ana Lucic. 2022. Ter hoeve, gabriele tolomei, maarten de rijke, and fabrizio silvestri. cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 4499–4511.

Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. 2022. Learning fair node representations with graph counterfactual fairness. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 695–703.

John McCarthy. 2000. Free will-even for robots. *Journal of experimental & theoretical artificial intelligence*, 12(3):341–352.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *IJCAI*, pages 3137–3143.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA. Omnipress.

Danilo Numeroso and Davide Bacciu. 2021. Meg: Generating molecular counterfactual explanations for deep graph networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton Van den Hengel. 2020. Counterfactual vision-and-language navigation: Unravelling the unseen. *Advances in Neural Information Processing Systems*, 33:5296–5307.

Mario Alfonso Prado-Romero, Bardh Prenkaj, Giovanni Stilo, Alessandro Celi, Ernesto Estevanell-Valladares, and Daniel Alejandro Valdés-Pérez. 2022. Ensemble approaches for graph counterfactual explanations.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053, Hong Kong, China. Association for Computational Linguistics.

Eva Rafetseder and Josef Perner. 2014. Counterfactual reasoning: Sharpening conceptual distinctions in developmental studies. *Child development perspectives*, 8(1):54–58.

Raymond Reiter. 1978. *On Closed World Data Bases*, pages 55–76. Springer US, Boston, MA.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019a. Drum: End-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019b. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32.

Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8350, Online. Association for Computational Linguistics.

Dan Shi, Anchen Li, and Bo Yang. 2022. Counterfactual-guided and curiosity-driven multi-hop reasoning over knowledge graph. In *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part I*, page 171–179, Berlin, Heidelberg. Springer-Verlag.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR.

Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

Nicole Van Hoeck, Patrick D Watson, and Aron K Barbey. 2015. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in human neuroscience*, 9:420.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Zikang Wang, Linjing Li, Daniel Zeng, and Xiaofei Wu. 2021. Incorporating prior knowledge from counterfactuals into knowledge graph reasoning. *Knowledge-Based Systems*, 223:107035.

Ran Xu, Yue Yu, Chao Zhang, Mohammed K Ali, Joyce C Ho, and Carl Yang. 2022. Counterfactual and factual reasoning over hypergraphs for interpretable clinical predictions on ehr. In *Machine Learning for Health*, pages 259–278. PMLR.

Xu Zhang, Liang Zhang, Bo Jin, and Xinjiang Lu. 2021. A multi-view confidence-calibrated framework for fair and stable graph representation learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1493–1498. IEEE.

Tong Zhao, Gang Liu, Daheng Wang, Wenhao Yu, and Meng Jiang. 2022. Learning from counterfactual links for link prediction. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26911–26926. PMLR.

A Evaluation Metrics

This section gives the concrete formulas for the metrics used in Table 3 and Table 4. We denote the full evaluation dataset by $\mathcal{D} := \{(\tau_1^c, \mathcal{T}_{\tau_1^c}), (\tau_2^c, \mathcal{T}_{\tau_2^c}), ..., (\tau_n^c, \mathcal{T}_{\tau_n^c})\}$, where τ_j^c denote hypothetical scenarios and $\mathcal{T}_{\tau_j^c}$ are the corresponding test cases. For any arbitrary fact τ, we assign two binary labels: The label y_τ indicates whether τ is in the *original fact set* $\mathcal{F}$ ($y_\tau = 1$ if $\tau \in \mathcal{F}$ and 0 otherwise). The label y_τ^c denotes the membership of fact τ to the *fact set of the counterfactual graph* $\mathcal{F}_{\tau_j^c}$ ($y_\tau^c = 1$ if $\tau \in \mathcal{F}_{\tau_j^c}$ and 0 otherwise). Lastly, $\widehat{y}_\tau^c$ denotes the binary prediction made for y_τ^c.

F1: For this metric, we consider all test cases of all hypothetical scenarios without any restrictions. It gives an overall indiction of the predictive performance on the fact set of the counterfactual graph. We choose the F1-score due to the imbalanced label distribution of our constructed test cases.

$$\mathrm{F1} = \frac{2tp}{2tp + fn + fp},$$

where $tp = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}_{\tau_j^c}} \mathbb{I}(y_\tau^c = 1 \wedge \widehat{y}_\tau^c = 1),$

$$fn = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}_{\tau_j^c}} \mathbb{I}(y_\tau^c = 1 \wedge \widehat{y}_\tau^c = 0),$$

$$fp = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}_{\tau_j^c}} \mathbb{I}(y_\tau^c = 0 \wedge \widehat{y}_\tau^c = 1)$$

Changed: We denote the set of *changed facts* in $\mathcal{T}_{\tau^c_j}$ by $\mathcal{T}^*_{\tau^c_j}$. Formally,

$$\mathcal{T}^*_{\tau^c_j} := \{\tau \in \mathcal{T}_{\tau^c_j} : (y_\tau = 0 \wedge y^c_\tau = 1) \vee (y_\tau = 1 \wedge y^c_\tau = 0)\}.$$

Intuitively, $\mathcal{T}^*_{\tau^c_j}$ is comprised of facts that were not present in the original graph but emerge in the counterfactual KG or vice versa. We compute the accuracy on these cases with respect to y^c_τ.

$$\text{Changed} = \frac{\widetilde{tp}}{\widetilde{tp} + \widetilde{fn} + \widetilde{fp} + \widetilde{tn}},$$

where
$$\widetilde{tp} = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}^*_{\tau^c_j}} \mathbb{I}(y^c_\tau = 1 \wedge \widehat{y}^c_\tau = 1),$$
$$\widetilde{fn} = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}^*_{\tau^c_j}} \mathbb{I}(y^c_\tau = 1 \wedge \widehat{y}^c_\tau = 0),$$
$$\widetilde{fp} = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}^*_{\tau^c_j}} \mathbb{I}(y^c_\tau = 0 \wedge \widehat{y}^c_\tau = 1),$$
$$\widetilde{tn} = \sum_{j=1}^{n} \sum_{\tau \in \mathcal{T}^*_{\tau^c_j}} \mathbb{I}(y^c_\tau = 0 \wedge \widehat{y}^c_\tau = 0)$$

Note that in the case of automatically generated labels (Table 3 and CFKGR-CoDEx-M* (Expected) in Table 4), $\mathcal{T}^*_{\tau^c_j}$ only consists of *emerging facts* and hence the ground truth labels y^c_τ are always positive. In these cases, the above accuracy is equivalent to the recall on $\mathcal{T}^*_{\tau^c_j}$.

Unchanged:
$\overline{\mathcal{T}}_{\tau^c_j}$ denotes the set of *unchanged* facts in $\mathcal{T}_{\tau^c_j}$. Formally,

$$\overline{\mathcal{T}}_{\tau^c_j} := \{\tau \in \mathcal{T}_{\tau^c_j} : (y_\tau = 0 \wedge y^c_\tau = 0) \vee (y_\tau = 1 \wedge y^c_\tau = 1)\}.$$

Intuitively, $\mathcal{T}_{\tau^c_j}$ is comprised of facts that do not change their label between $\mathcal{F}$ and $\mathcal{F}_{\tau^c_j}$. We compute the F1-score on such instances due to their imbalanced label distribution in our constructed test cases.

$$\text{Unchanged} = \frac{2\overline{tp}}{2\overline{tp} + \overline{fn} + \overline{fp}},$$

where
$$\overline{tp} = \sum_{j=1}^{n} \sum_{\tau \in \overline{\mathcal{T}}_{\tau^c_j}} \mathbb{I}(y^c_\tau = 1 \wedge \widehat{y}^c_\tau = 1),$$
$$\overline{fn} = \sum_{j=1}^{n} \sum_{\tau \in \overline{\mathcal{T}}_{\tau^c_j}} \mathbb{I}(y^c_\tau = 1 \wedge \widehat{y}^c_\tau = 0),$$
$$\overline{fp} = \sum_{j=1}^{n} \sum_{\tau \in \overline{\mathcal{T}}_{\tau^c_j}} \mathbb{I}(y^c_\tau = 0 \wedge \widehat{y}^c_\tau = 1)$$

	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{F}_{train}\rvert$	$\lvert\mathcal{F}_{val}\rvert$	$\lvert\mathcal{F}_{test}\rvert$	Negatives
S	2034	42	32888	1827	1828	Yes
M	17050	51	185584	10310	10311	Yes
L	77951	69	551193	30622	30622	No

Table 6: Overview of CoDEx datasets (Safavi and Koutra, 2020).

B CoDEx Resources

We use the CoDEx knowledge graph completion benchmark (Safavi and Koutra, 2020), which is comprised of three knowledge graphs (S, M, L) collected from Wikidata based on seed entities and relations for 13 differernt domains (e.g., media and entertainment, politics, science). Table 6 porvides an overview over the resources provided by CoDEx.

C Details of Dataset Creation

This section contains details of the CFKGR dataset creation that were omitted in Section 3 due to space constraints and gives a full algorithmic description of the procedure.

C.1 Formal Description

Section 3 provides a high-level description on how we create CFKGR test instances based on the first body atom of a rule. This section provides a more formal version of the employed constraints and covers the case where the second body atom is selected for creating the hypothetical scenario.

In the following, we define an *atom* variable to distinguish between hypotheticals from the first (atom=1) versus the second atom (atom=2). The general setup stays the same, whether we create hypotheticals from the first or second atom: Given a rule (r_1, r_2, r_3), we search for existing edges $e_1 := (x, r_1, y) \in \mathcal{F}_{train}$ and $e_2 := (\bar{y}, r_2, z) \in \mathcal{F}_{train}$, such that $\tau^c := (x, r_1, \bar{y}) \notin \mathcal{F}$ and $\tau^i := (x, r_3, z) \notin \mathcal{F}$. We employ the following constraints I1, I2, and I3 when sampling e_1 and e_2.

I1: if *atom* = 1: $\exists a \in \mathcal{E} : (a, r_1, \bar{y}) \in \mathcal{F}$,
if *atom* = 2: $\exists b \in \mathcal{E} : (y, r_2, b) \in \mathcal{F}$
I2: $\exists c \in \mathcal{E} : (x, r_3, c) \in \mathcal{F}$
I3: $\exists d \in \mathcal{E} : (d, r_3, z) \in \mathcal{F}$
The above constraints ensure that the constructed triples τ^c and τ^i have sensible entities for the given relation.
When corrupting a given triple (h, r, t), we employ the constraints C1, C2 and C3 for finding corruptions h', r', t'.

C1: $\exists a \in \mathcal{E} : (h', r, a) \in \mathcal{F}$
C2: $\exists b \in \mathcal{E} : (b, r, t') \in \mathcal{F}$
C3: $(h', r, t), (h, r, t'), (h, r', t) \notin \mathcal{F} \cup \mathcal{F}^{+}_{\Delta}$,
where $\mathcal{F}^{+}_{\Delta}$ denotes the set of inferences made by *all rules* in our rule set, given the hypothetical scenario.

C1 and C2 promote challenging head and tail corruptions, which cannot trivially be identified due to the triples being nonsensical. C3 ensures that the generated relation corruptions are not implied by the given hypothetical scenario and our rule set.

```
Data: knowledge graph G = {E, R, F},
      inference rule δ,
      # of CFs to generate per atom M
Result: CFKGR instances for rule δ_j
D_δ ← {}
for atom ∈ {1, 2} do
    for n ∈ {1, ..., M} do
        Randomly sample
        (x, r_1, y) and (ȳ, r_2, z) from F_train,
        according to I1, I2, I3, I4
        if atom = 1 then
            τ^c ← (x, r_1, ȳ)
        else
            τ^c ← (y, r_2, z)
        τ^i ← (x, r_3, z)
        Sample τ^n_1, τ^n_2 from N_1(τ^c)
        Sample τ^f from F \ N_1(τ^c)
        Create corruptions for τ^i, τ^n_1, τ^n_2, τ^f
        according to C1, C2, C3
        T_τ^c ← {τ^c, τ^i, τ^n_1, τ^n_2, τ^f,
                 τ^i_h', τ^i_r', τ^i_t', ..., τ^f_h', τ^f_r', τ^f_t'}
        D_δ ← D_δ ∪ {(τ^c, T_τ^c)}
```
Algorithm 2: Creation of CFKGR instances.

C.2 Additional Constraints for P361 and P463

For counterfactual triples τ^c using the relation P361 ("part of") or P463 ("member of"), we introduce an additional condition when sampling e_1 and e_2 based on *entity types* in order to avoid nonsensical hypothetical scenarios, such as (Iraq, part of, The Quarrymen). Entity types are available for every entity in the CoDEx dataset (Safavi and Koutra, 2020) and provide additional information regarding the entity. For instance, "France" is associated with the entity type "country" (among others) and "7B" is tagged as a "musical group". We denote the set of entity types associated with an entity $e \in \mathcal{E}$ by $type(e)$. We define the following constraint:

I4: $type(\bar{y}) \cap type(y) \neq \{\}$,
if $atom = 1$ and $r_1 \in \{P361, P463\}$ or
if $atom = 2$ and $r_2 \in \{P361, P463\}$

This condition heuristically ensures that the entity that replaces the original head/tail of a triple to create a hypothetical scenario is of a similar type as the original entity. In the example above, (Iraq, part of, The Quarrymen) is no longer a valid generation when the above constraint is enforced, since the "The Quarrymen" shares no entity type with the original tail "Middle East".

C.3 Algorithm

Algorithm 2 describes the dataset creation for CFKGR. $\mathcal{N}_1(\tau^c)$ denotes the one-hop neighborhood of τ^c, excluding the context triggering the rule. The remaining notation follows Sections 2 and 3.

D Human Dataset Verification

This section details the guidelines provided to the annotators and their recruitment.

D.1 Annotator Recruitment and Demographic

We recruited annotators on a voluntary basis (most of which are part of our institution) and did not offer financial compensation. Labelers were made aware that their annotations will be used and published in a scientific paper.

We recruited thirteen annotators in total, twelve of which have at least a Master's degree in STEM, while the remaining has a Bachelor's degree.

D.2 Annotation Guidelines

The main goal of the task is to judge the plausibility of presented statements, given a hypothetical scenario and potentially relevant context.

Each annotation prompt presented to you will consist of the following elements:

1) a hypothetical scenario, which you should assume to be true
2) a context, which gives additional information regarding the entities in the scenario
3) a statement, which should be labeled as "likely", "unlikely", or "unsure/too little information"

Please assign the label "likely" if you think the presented statement is likely to hold given the hypothetical scenario, the context, and your world knowledge. Assign "unlikely" if you do not think so. Assign the label "unsure/too little information" if you cannot confidently judge the plausibility of the statement based on the presented information.

Expressions in parenthess denote entity types, which provide additional information for each entity. They can be helpful when reasoning with lesser-known entities. For instance, the entity '7B' is associated with the entity type 'musical group' to clarify that '7B' refers to a band.

Each statement follows the general structure 'Is it then plausible that ..., given that this IS NOT the case in the real world?' or 'Is it then STILL plausible that ..., given that this IS the case in the real world?'. Please pay attention to this difference when labeling.

Example 1:

Hypothetical scenario: Paris (city with millions of inhabitants, city, big city) is located in Japan (island nation, sovereign state, country)

Context: Japan (island nation, sovereign state, country) is part of the continent Asia (continent, continental area and surrounding islands)

Question: Is it then plausible that Paris (city with millions of inhabitants, city, big city) belongs to the continent Asia (continent, continental area and surrounding islands), given that this is not the case in the real world?

Example 2:

In some cases, the statement you are presented with might not have a strong, obvious connection to the hypothetical scenario (such as shared entities). This is intended and should not affect your annotation. For instance, you might encounter an example similar to the following:

Hypothetical scenario: Paris (city with millions of inhabitants, city, big city) is located in Japan (island nation, sovereign state, country)

Context: Japan (island nation, sovereign state, country) is part of the continent Asia (continent, continental area and surrounding islands)

Question: Is it then still plausible that English (modern language, natural language, language) is the official language of United Kingdom (country, sovereign state, island nation), given that this is the case in the real world?

If you believe that this statement is still plausible in a world where Paris is in Japan, assign 'likely'. If you think otherwise or cannot make a decision based on the presented information, assign 'unlikely' or 'unsure/too little information' respectively.

Example 3:

The statements might not be sensible for all examples. For instance, you could come across a statement like:

Hypothetical scenario: Paris (city with millions of inhabitants, city, big city) is located in Japan (island nation, sovereign state, country)

Context: Japan (island nation, sovereign state, country) is part of the continent Asia (continent, continental area and surrounding islands)

Question: Is it then plausible that Paris (city with millions of inhabitants, city, big city) is the unmarried partner of Asia (continent, continental area and surrounding islands), given that this is not the case in the real world?

These examples are intentional and you should annotate them according to the same scheme as the other examples.

E KGE Methods

TransE (Bordes et al., 2013) treats relations as translations in the embedding space. It finds embedding vectors $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^{d_e}$ such that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ for $(h, r, t) \in \mathcal{F}$, and uses the scoring function $\phi(h, r, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2$. **ComplEx** (Trouillon et al., 2016) maps entities and relations to the complex space and leverages the scoring function $\phi(h, r, t) = \mathrm{Re}(\langle \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} \rangle)$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^{d_e}$ and $\bar{\mathbf{t}}$ denotes the complex conjugate of $\mathbf{t}$. ComplEx is particularly well-suited for modeling antisymmetric relations (e.g., "part of"). **RESCAL**

(Nickel et al., 2011) represents the fact set as a three-dimensional tensor $\mathcal{X}$ with $\mathcal{X}_{i,j,r} = 1$ if (i, r, j) $\in \mathcal{F}$ and $\mathcal{X}_{i,j,r} = 0$ otherwise. Representations for entities and relations are obtained via a low-rank factorization $\mathcal{X}_r \approx ER_rE^T$, $E \in \mathbb{R}^{|\mathcal{E}| \times d_e}$, $R_r \in \mathbb{R}^{d_e \times d_e}$. The score of a given triple is computed as $\phi(h, r, t) = \mathbf{h}^T R_r \mathbf{t}$, where $\mathbf{h}$ and $\mathbf{t}$ are the rows of E corresponding to h and t. Similarly, **TuckER** (Balazevic et al., 2019) leverages Tucker decomposition (Tucker, 1966) to find representations $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{d_e}$, $\mathbf{r} \in \mathbb{R}^{d_r}$, as well as a core tensor $\mathcal{W} \in \mathbb{R}^{d_e \times d_r \times d_e}$ which allows sharing knowledge between all entity and relation embeddings. The scores are defined as $\phi(h, r, t) = \mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$, where $\times_i$ denotes the tensor product along the i-th mode. TuckER was shown to be effective for modeling compositional relations (Safavi and Koutra, 2020). **ConvE** (Dettmers et al., 2018) is a convolutional architecture described by $\phi(h, r, t) = f(\text{vec}(f([\mathbf{M_h}; \mathbf{M_r}] * \omega))\mathbf{W})\mathbf{t}$, where $\mathbf{M_h}$ and $\mathbf{M_r}$ are 2D-reshapings of entity and relation embeddings, ω describes the convolutional filters, and vec denotes vectorization (Ji et al., 2021).

F Experimental Setting

F.1 Implementation and Runtime Details

We run our experiments on a single Tesla V100 GPU with 16GB of memory on a Nvidia DGX1 server. COULDD hyperparameter tuning takes between around 35 minutes and 50 minutes and a run on the test set takes between 3 and 15 minutes, depending on the model and dataset.

For KGE embeddings, we use the pre-trained CoDEx models (Safavi and Koutra, 2020), which were trained using LibKGE (Broscheit et al., 2020). For our experiments with COULDD, we slightly adapt the LibKGE implementation, to allow for our propsed training scheme. For hyperparameter optimization, we use the `GridSampler` implemented in optuna (Akiba et al., 2019) (version 3.3.0). For computing performance metrics (F1, accuracy, confusion matrix), we use scikit-learn (version 1.3.0). All results are reproducible with seed 0.

F.2 Hyperparameters

Table 8 lists the hyperparameters used for our experiments 3. Bold parameters were tuned for COULDD on a validation set via grid search, while the remaining parameters were carried over from the pre-trained models provided by Safavi and Koutra (2020). For further details on the pre-trained models, please refer to Safavi and Koutra (2020). The learning rate (α) was tuned in the range of {0.001, 0.01, 0.1, 0.15, 0.2}. The number of additional samples (N) was chosen in the range of {0, 127, 255, 511, 1023} for all models except ConvE. For ConvE, the range is reduced to {127, 255, 511, 1023} because of its BatchNorm layer.

F.3 GPT Experimental Setup

We used the OpenAI API, and always used the model *gpt-3.5-turbo* and set the temperature to 0. The given system prompt, prompt templates, as well as an input and output example are given in Table 9.

G Evaluation per Test Type

Table 10 provides the perfomance per test case for the results in Table 3.

H Case Study on CoDEx-M

H.1 CFKGR-CoDEx-M* Label Distribution

Table 7 gives the label distribution of expected labels, according to our assumptions, and majority vote labels on CFKGR-CoDEx-M*.

	Expected (E)		Human (H)	
	0	1	0	1
τ^i	0	188	33	155
τ^f	0	255	0	255
τ^n	0	497	16	481
$\tau^i_{r'}$	231	0	217	14

Table 7: Label distribution in the CFKGR-CoDEx-M* test set with expected labels (E) and human-assigned (H) labels.

H.2 Rule-wise performance on filtered CoDEx-M

In Section 5.3.1, we pose the question how well the pre-trained CoDEx models can apply the composition rules mined by Amie3 on regular, non-hypothetical test cases. Table 11 provides information about the peformance on individual rules.

H.3 Confusion matrix on CFKGR-CoDEx-M*

Table 12 gives the confusion matrix for COULDD and ChatGPT on CFKGR-CoDEx-M* with manually assigned labels.

	RESCAL	TransE	ComplEx	ConvE	TuckER
CFKGR-CoDEx-S					
Embedding size	512	512	512	256	512
Reciprocal	No	Yes	Yes	Yes	Yes
Optimizer	Adagrad	Adagrad	Adam	Adagrad	Adagrad
Regularization					
Type	l_3	l_2	None	l_3	l_1
Entity embeddings	2.18×10^{-10}	1.32×10^{-7}	9.58×10^{-13}	3.11×10^{-15}	3.47×10^{-15}
Relation embedding	3.37×10^{-14}	3.72×10^{-18}	0.0229	4.68×10^{-9}	3.43×10^{-14}
Frequency weighting	False	False	True	True	True
Dropout					
Entity embeddings	0.0	0.0	0.0793	0.0	0.1895
Relation embeddings	0.0804	0.0	0.0564	0.0	0.0
Feature map (ConvE)	-	-	-	0.2062	-
Projection (ConvE)	-	-	-	0.1709	-
Additional samples (N)	127	255	127	255	225
Learning rate (α)	0.01	0.01	0.1	0.001	0.01
CFKGR-CoDEx-M					
Embedding size	256	512	512	512	512
Reciprocal	Yes	Yes	Yes	Yes	Yes
Optimizer	Adagrad	Adagrad	Adagrad	Adagrad	Adagrad
Regularization					
Type	l_2	l_2	l_3	l_1	l_1
Entity embeddings	9.56×10^{-7}	1.32×10^{-7}	1.34×10^{-10}	1.37×10^{-10}	3.47×10^{-15}
Relation embedding	2.56×10^{-17}	3.72×10^{-18}	6.38×10^{-16}	4.72×10^{-10}	3.4×10^{-14}
Frequency weighting	False	False	True	True	True
Dropout					
Entity embeddings	0.0	0.0	0.1196	0.0	0.1895
Relation embeddings	0.0	0.0	0.3602	0.0348	0.0
Feature map (ConvE)	-	-	-	0.3042	-
Projection (ConvE)	-	-	-	0.2343	-
Additional samples (N)	255	511	0	511	1023
Learning rate (α)	0.01	0.01	0.1	0.001	0.01
CFKGR-CoDEx-L					
Embedding size	128	128	128	256	256
Reciprocal	No	Yes	Yes	Yes	No
Optimizer	Adagrad	Adam	Adagrad	Adagrad	Adagrad
Regularization					
Type	l_2	l_2	l_2	l_1	l_2
Entity embeddings	2.01×10^{-16}	7.98×10^{-14}	2.01×10^{-16}	6.10×10^{-16}	8.06×10^{-11}
Relation embedding	3.52×10^{-13}	3.42×10^{-9}	3.52×10^{-13}	1.03×10^{-16}	7.19×10^{-19}
Frequency weighting	True	False	True	True	True
Dropout					
Entity embeddings	0.0	0.0	0.0	0.0064	0.1606
Relation embeddings	0.0	0.0	0.0	0.0	0.0857
Feature map (ConvE)	-	-	-	0.1530	-
Projection (ConvE)	-	-	-	0.4192	-
Additional samples (N)	0	1023	0	127	127
Learning rate (α)	0.1	0.2	0.2	0.01	0.01

Table 8: Hyperparameters for COULDD experiments. Bold hyperparameters were tuned by us. The remaining are taken from the original CoDEx paper (Safavi and Koutra, 2020) and kept the same for our experiments.

System Prompt	You are a helpful, honest data labeler who classifies the plausibility of a hypothetical scenario. You will be presented: Hypothetical scenario: This fact is wrong in the real world, but you assume it is true in the current hypothetical world. Context: It is true in the hypothetical world, and gives additional information you can use to reason about the plausibility of the presented statement A statement which is supposed to be labeled as 'likely' or 'unlikely'. Each statement follows the general structure 'Is it then plausible that ..., given that this IS NOT the case in the real world?' or 'Is it then STILL plausible that ..., given that this IS the case in the real world?'. Please pay attention to this difference when labeling.
Statement Template	{{ Statement }} Return 'likely' if you think the presented statement is likely to hold given the hypothetical scenario, the context, and your world knowledge. - If a fact was "the case" in the real world, respond 'likely' if the hypothetical scenario does not make it implausible - If a fact was "not the case" in the real world, respond 'likely' if the hypothetical scenario makes it plausible. Return 'unlikely' only if the presented statement is unlikely to hold given the hypothetical scenario, the context, and your world knowledge. - If a fact was "the case" in the real world, respond 'unlikely' if the hypothetical scenario makes it implausible. - If a fact was "not the case" in the real world, respond 'unlikely' if the hypothetical scenario does not make it plausible. If The hypothetical scenario and the context do not provide any specific information for the statement, respond 'unlikely' given that the statement is "not the case in the real world" and 'likely' if the statement "is the case in the real world". Respond with a JSON. First, include a explanation of your reasoning in the 'explanation' field and give then an answer in the 'answer' field.
Example Statement	Hypothetical scenario: Julio Iglesias (human) is married to Rachel Weisz (human) Context: Rachel Weisz (human) is a citizen of United Kingdom (island nation, Commonwealth realm, sovereign state) Is it then plausible that Julio Iglesias (human) is a citizen of United Kingdom (colonial power, island nation, country), given that this is not the case in the real world?
Original Label	Unlikely
Expected Label (in CF KG)	Likely

Example Output	{
	"explanation": "Based on the hypothetical scenario and the context provided, it is plausible that Julio Iglesias (human) is a citizen of United Kingdom (colonial power, island nation, country). The context states that Rachel Weisz (human) is a citizen of United Kingdom, which implies that she has citizenship in the country. Since Julio Iglesias is married to Rachel Weisz in the hypothetical scenario, it is likely that he would also have acquired citizenship in the United Kingdom through marriage.",
	"answer": "likely"
	}

Table 9: The table shows how ChatGPT was used. It includes templates and an example scenario with the provided output.

Dataset	Method	τ^i	τ^f	$\tau^i_{h'}$	$\tau^f_{h'}$	$\tau^n_{h'}$	τ^n	$\tau^i_{r'}$	$\tau^f_{r'}$	$\tau^n_{r'}$	$\tau^i_{t'}$	$\tau^f_{t'}$	$\tau^n_{t'}$
CFKGR-CoDEx-S	RESCAL	27.12	**99.46**	73.78	39.24	**56.6**	**98.55**	**94.94**	94.21	**94.76**	79.57	51.54	61.66
	COULDD-RESCAL	**32.48 ± 0.73**	99.28 ± 0.2	**73.82 ± 1.04**	**43.15 ± 1.19**	55.5 ± 0.53	98.17 ± 0.16	94.86 ± 0.25	**94.5 ± 0.25**	94.47 ± 0.11	**79.78 ± 0.56**	**54.32 ± 0.41**	**61.95 ± 0.16**
	TransE	23.15	**95.84**	78.84	32.91	55.70	**90.78**	**97.11**	95.48	93.94	86.80	52.80	68.72
	COULDD-TransE	**26.8 ± 0.81**	94.39 ± 0.26	**82.78 ± 0.29**	**36.56 ± 0.31**	**58.16 ± 0.21**	89.84 ± 0.21	97.11 ± 0.16	**95.66 ± 0.11**	**93.96 ± 0.04**	**88.28 ± 0.81**	**58.37 ± 0.27**	**71.57 ± 0.07**
	ComplEx	29.11	**98.73**	72.15	44.12	58.14	**98.82**	**96.2**	**97.11**	96.20	**79.39**	56.06	64.65
	COULDD-ComplEx	**37.94 ± 0.67**	93.31 ± 0.66	**84.27 ± 0.98**	**63.83 ± 0.68**	**71.32 ± 0.85**	94.27 ± 0.19	95.77 ± 0.14	97.0 ± 0.14	**96.42 ± 0.11**	79.06 ± 1.52	**72.12 ± 1.3**	**74.21 ± 0.66**
	ConvE	16.64	**97.65**	81.19	43.76	**65.46**	**93.67**	96.56	91.32	**87.79**	92.95	53.53	73.78
	COULDD-ConvE	**16.96 ± 0.72**	97.22 ± 0.18	**82.21 ± 0.18**	**45.53 ± 0.4**	65.23 ± 0.41	93.49 ± 0.27	**96.6 ± 0.07**	**91.72 ± 0.07**	87.58 ± 0.17	**93.56 ± 0.18**	**55.15 ± 0.49**	**75.8 ± 0.15**
	TuckER	15.01	**98.37**	**83.73**	45.39	**71.34**	98.55	**95.3**	**96.93**	**94.21**	**89.33**	54.79	73.69
	COULDD-TuckER	**35.99 ± 1.0**	97.72 ± 0.54	78.23 ± 0.29	**48.1 ± 1.23**	67.09 ± 0.56	**98.61 ± 0.07**	94.29 ± 0.22	96.93 ± 0.23	93.35 ± 0.15	83.11 ± 0.63	**58.59 ± 0.62**	**74.23 ± 0.3**
CFKGR-CoDEx-M	RESCAL	21.57	**97.96**	79.41	46.90	68.10	**95.75**	91.18	91.42	91.30	87.01	58.50	75.57
	COULDD-RESCAL	**26.23 ± 0.16**	96.44 ± 0.19	**81.44 ± 0.14**	**48.91 ± 0.25**	**70.14 ± 0.27**	94.49 ± 0.19	**91.23 ± 0.08**	**91.47 ± 0.23**	**91.31 ± 0.09**	**87.19 ± 0.16**	**59.54 ± 0.32**	**76.41 ± 0.12**
	TransE	23.61	**88.56**	76.31	36.11	62.50	**75.37**	**92.97**	92.89	89.26	86.19	53.84	70.14
	COULDD-TransE	**26.06 ± 0.25**	85.85 ± 0.18	**76.83 ± 0.27**	**38.94 ± 0.18**	**63.68 ± 0.16**	74.31 ± 0.14	92.78 ± 0.1	**93.17 ± 0.04**	**89.31 ± 0.03**	**86.75 ± 0.14**	**57.63 ± 0.27**	**70.92 ± 0.09**
	ComplEx	11.60	**97.96**	**89.38**	**49.02**	**75.08**	**97.55**	**93.63**	**94.61**	**92.65**	**94.69**	**59.56**	**80.39**
	COULDD-ComplEx	**34.67 ± 0.23**	97.96 ± 0.0	79.17 ± 0.53	48.95 ± 0.08	69.58 ± 0.21	97.21 ± 0.1	93.09 ± 0.11	94.59 ± 0.03	92.34 ± 0.07	90.36 ± 0.31	59.48 ± 0.05	79.01 ± 0.09
	ConvE	13.15	**93.06**	**87.09**	41.91	67.97	**81.78**	**95.1**	**88.32**	**84.76**	**94.12**	53.68	77.33
	COULDD-ConvE	**17.04 ± 0.16**	84.72 ± 0.45	85.38 ± 0.25	**43.94 ± 0.1**	**68.84 ± 0.42**	71.49 ± 0.35	92.4 ± 0.14	86.18 ± 0.61	81.66 ± 0.4	93.5 ± 0.4	**54.31 ± 0.59**	**79.52 ± 0.17**
	TuckER	13.15	97.96	**88.4**	**50.74**	**76.76**	97.14	**92.48**	**91.18**	**88.77**	**95.02**	**58.33**	**80.8**
	COULDD-TuckER	**43.69 ± 0.38**	**98.33 ± 0.11**	73.14 ± 0.54	44.07 ± 0.58	67.06 ± 0.18	**97.99 ± 0.11**	91.99 ± 0.32	90.87 ± 0.46	87.57 ± 0.11	90.1 ± 0.38	58.27 ± 0.68	78.7 ± 0.15
CFKGR-CoDEx-L	RESCAL	71.47	**99.89**	32.09	18.09	**23.39**	**99.63**	68.92	74.08	**72.88**	**51.52**	**53.91**	51.41
	COULDD-RESCAL	**84.56 ± 0.35**	99.89 ± 0.0	**32.37 ± 0.58**	**18.16 ± 0.07**	23.15 ± 0.14	95.58 ± 0.21	**69.2 ± 0.48**	**74.09 ± 0.07**	69.94 ± 0.23	45.71 ± 0.54	53.87 ± 0.07	**51.61 ± 0.21**
	TransE	66.31	**99.41**	30.07	18.31	20.68	99.25	**79.4**	48.00	40.82	48.96	46.89	44.97
	COULDD-TransE	**76.56 ± 0.25**	98.99 ± 0.1	**30.47 ± 0.55**	**27.87 ± 0.66**	**22.47 ± 0.21**	**99.35 ± 0.06**	77.54 ± 0.18	**53.22 ± 0.35**	**43.67 ± 0.19**	**58.12 ± 0.3**	**60.04 ± 0.82**	**55.93 ± 0.37**
	ComplEx	65.51	**99.57**	**36.14**	27.25	**33.02**	99.44	**90.47**	**84.62**	**83.93**	**58.91**	**64.93**	**61.07**
	COULDD-ComplEx	**82.95 ± 0.26**	99.57 ± 0.0	31.73 ± 0.13	**27.29 ± 0.04**	29.44 ± 0.09	**99.53 ± 0.04**	89.03 ± 0.12	84.57 ± 0.03	83.25 ± 0.09	55.5 ± 0.29	64.84 ± 0.07	59.68 ± 0.14
	ConvE	**61.84**	**99.52**	41.35	36.46	40.66	**99.18**	91.06	61.63	53.54	61.58	63.70	60.32
	COULDD-ConvE	45.53 ± 0.61	94.5 ± 0.36	**61.25 ± 0.59**	**53.79 ± 0.3**	**57.2 ± 0.47**	95.32 ± 0.18	**93.18 ± 0.31**	**73.72 ± 0.73**	**67.2 ± 0.54**	**79.45 ± 0.36**	**78.11 ± 0.42**	**75.53 ± 0.51**
	TuckER	76.74	**99.79**	**27.46**	14.74	**22.78**	99.65	**75.36**	63.23	**62.03**	**53.33**	50.56	**49.92**
	COULDD-TuckER	**88.47 ± 0.34**	99.74 ± 0.07	25.92 ± 0.52	**15.94 ± 0.23**	19.58 ± 0.29	**99.68 ± 0.04**	72.59 ± 0.33	**64.14 ± 0.44**	61.29 ± 0.21	50.07 ± 0.29	**52.84 ± 0.14**	48.13 ± 0.54

Table 10: Accuracy by test type of pre-trained embeddings and COULDD on CFKGR. For COULDD, we report the mean and standard deviation across 5 runs.

	Support	PCA	# Test	RESCAL	TransE	ComplEx	ConvE	TuckER
(founded by, citizenship, country)	64	0.826	5	1.000	1.000	0.400	1.000	0.800
(place of death, official language, languages spoken)	836	0.818	36	0.972	0.972	1.000	0.944	1.000
(place of birth, official language, languages spoken)	665	0.790	23	1.000	0.826	1.000	0.826	0.957
(spouse, citizenship, citizenship)	682	0.661	15	0.933	0.933	0.867	0.733	0.933
(citizenship, official language, languages spoken)	9937	0.543	416	0.962	0.918	0.993	0.901	0.978
(country, continent, continent)	100	0.427	5	0.200	0.000	0.200	0.000	0.600
(cast member, citizenship, country of origin)	1464	0.406	87	0.805	0.943	0.920	0.931	0.931
(headquarters location, country, country)	82	0.297	6	1.000	0.833	1.000	0.833	0.833
(located in, country, country)	137	0.346	5	0.600	0.600	0.400	0.800	1.000
(cast member, place of death, narrative location)	87	0.134	6	1.000	1.000	1.000	1.000	1.000

Table 11: Rule-wise performance on the filtered test set of CoDEx-M (Table 4). "Support" denotes the number of instantions of the rule in the full KG. "PCA" is the PCA confidence as computed by Amie3. "# Test" denotes the number of inferences in the test set.

	CFKGR-CoDEx-M (H)															
	τ^i				τ^f				τ^n				$\tau_{r'}$			
	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP
COULDD-RESCAL	13.2	19.8	127.4	27.6	0.0	0.0	8.2	246.8	0.0	16.0	24.0	457.0	199.8	17.2	11.0	3.0
COULDD-TransE	17.0	16.0	125.6	29.4	0.0	0.0	31.6	223.4	6.4	9.6	116.8	364.2	204.8	12.2	12.0	2.0
COULDD-ComplEx	15.4	17.6	101.0	54.0	0.0	0.0	4.0	251.0	2.4	13.6	7.4	473.6	205.4	11.6	13.4	0.6
COULDD-ConvE	18.2	14.8	138.2	16.8	0.0	0.0	35.2	219.8	5.4	10.6	128.4	352.6	200.2	16.8	11.8	2.2
COULDD-TuckER	13.2	19.8	92.6	62.4	0.0	0.0	3.4	251.6	0.0	16.0	8.2	472.8	206.8	10.2	12.0	2.0
gpt-3.5-turbo	21	12	66	89	0	0	188	67	10	6	169	312	127	90	11	3

Table 12: Performance analysis per test type on CFKGR-CoDEx-M with human-assigned labels. For COULDD, the reported values are averaged over 5 model runs.