ExamplE: Explaining Knowledge Graph Embedding Predictions with Influential Examples

Adrianna Janik[1,2,*], Luca Costabello[2]

[1]Nearform, Tankfield, Convent Hill, Tramore, Co. Waterford, X91 PV08, Ireland
[2]Accenture Labs, Dublin, 7 Hanover Quay, Grand Canal Dock, Dublin 2, D02 YN32, Ireland

Abstract

Knowledge Graph Embedding (KGE) models achieve strong link-prediction performance and are increasingly deployed as part of enterprise knowledge graph infrastructure, but offer no direct interpretability; a shortcoming that is increasingly consequential under emerging AI regulation (e.g., EU AI Act). We introduce **ExamplE**, a post-hoc, model-agnostic explanation method grounded in case-based reasoning that attaches to pre-trained KGE models without retraining or architectural change. Given a target link prediction, ExamplE identifies the most influential training triples by sampling the latent-space neighbourhood of the target triple's subject and object embeddings, filtering for pairs that co-occur in the training graph, and ranking them by combined embedding distance. The result is a ranked set of influential example triples from the knowledge graph, that offer human-readable explanation grounded in training cases. We evaluate ExamplE using Remove-and-Retrain (ROAR) and Reversed-ROAR protocols on Fb15k-237 and WN18RR, comparing against a random same-predicate baseline.

Results from case-study trajectories and per-triple cross-sections show that ExamplE-identified examples carry predictive influence in single-triple removal conditions; set-level outcomes are consistent with relational redundancy. KGE models appear to encode relational patterns redundantly across same-predicate triples, so a prediction is not reducible to a single supporting triple. ExamplE is designed for this setting, returning a ranked set rather than a single attribution.

We further introduce **XAI-Fb15k-237**, a human-curated evaluation dataset of 86 interpretable triples from Fb15k-237, designed to support user-study benchmarking of KGE explainability methods. ExamplE generates explanations in ∼5 seconds per triple, compared to ∼6 minutes for GradientRollback (Fb15k-237), with no special training mode or prohibitive memory overhead. Code, XAI-Fb15k-237 are provided as supplemental material.

Keywords

Knowledge Graph Embeddings, Explainability, Example-based Explanations, Case-based Reasoning, Link Prediction

CEUR-WS.org/Vol-4318/paper1.pdf

1. Introduction

Knowledge graph embedding (KGE) models enable powerful link prediction but are fundamentally opaque: a prediction score emerges from learned embedding geometry with no direct pointer back to the training triples that shaped it [1]. This opacity is no longer merely a technical inconvenience; Article 13 of the EU AI Act mandates transparency and instructions for use for high-risk AI systems, and KGE models are actively deployed in biomedical applications such as drug discovery and disease association [2] that may fall within its scope should such a system be classified as high-risk under Annex III. Following Regulation (EU) 2026/1744, these obligations apply to Annex III high-risk systems from 2 December 2027 [3]. Non-compliance with Chapter III obligations such as Article 13 is subject to fines of up to €15 million or 3% of global annual turnover.

We propose **ExamplE**, a post-hoc explanation method for KGE link predictions grounded in case-based reasoning [4], model-agnostic in principle. Given a predicted link, ExamplE retrieves the nearest neighbours of the target triple's subject and object embeddings, filters for pairs that co-occur in the training graph under the same (or similar) predicate, and ranks them by combined embedding distance.

XAI-KG@ISWC2026 - 2nd International Workshop on eXplainable AI and Knowledge Graphs, co-located with ISWC 2026, October 25-29, 2026, Bari, Italy
ada.janik@gmail.com (A. Janik)
https://adrijanik.com (A. Janik)
0000-0002-4387-4563 (A. Janik); 0000-0002-0720-9347 (L. Costabello)
© 2026 Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-07

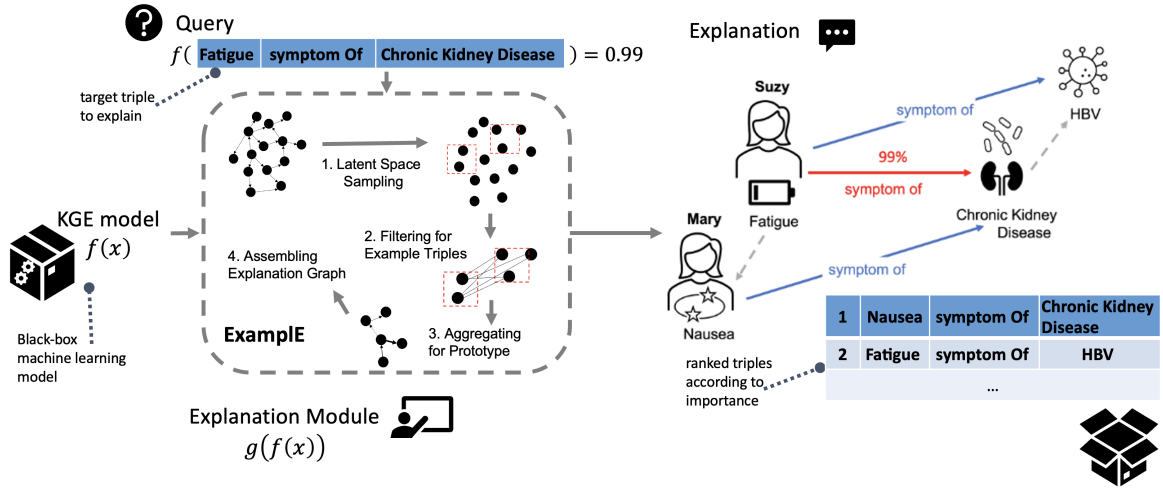


Figure 1: To support prediction of the target statement we identify influential examples by probing the knowledge base constrained w.r.t. the latent-space. This example is drawn from the Fb15k-237 dataset (illustrative). Predicted plausibility score was 99%, and two most influential examples were retrieved as an explanation with the following ranks: 1st: $Nausea \rightarrow symptomOf \rightarrow ChronicKidneyDisease$, 2nd: $Fatigue \rightarrow symptomOf \rightarrow HBV$

The result is a ranked set of *influential examples*: real training triples that the model most likely learned the prediction from (Figure 1).

Explainability for KGE models faces three specific challenges. First, standard benchmarks (Fb15k-237, WN18RR) use encoded identifiers stripped of `rdfs:label` annotations during export, making explanations unreadable without label recovery. Second, there are no shared evaluation datasets or protocols for explainability, preventing fair comparison across methods. Third, KGE scores are not calibrated probabilities, complicating interpretation [5].

A pattern we observe across our evaluation, which we term *relational* redundancy, is that predicted plausibility scores tend to recover after individual training triples are removed and the model is retrained. We read this as consistent with the model being able to reconstruct support for a prediction from same-predicate triples elsewhere in the graph, rather than depending on any one triple in particular.

Contributions:

- **ExamplE**, a CBR heuristic that produces an Explanation Graph for KGE predictions: a ranked set of influential training examples, an aggregated prototype neighbourhood, and the target triple itself, all expressed as a graph compatible with the original. Requires no special training mode; runs in ∼5 s per triple (FB15k-237/TransE).
- **XAI-FB15k-237**, a human-curated evaluation dataset of 86 interpretable triples for benchmarking KGE explainability methods, released alongside a Freebase MID-to-label mapping recovered during construction, usable independently to support explanation work on FB15k-237.
- **rev-ROAR**, a complement to the ROAR evaluation protocol for KGE explainability that tests whether identified examples are sufficient (rather than necessary) to sustain a prediction. Together with ROAR, it surfaces empirical patterns we read as *relational redundancy* in KGE training graphs.
- Open-source implementation of ExamplE as an extendable explanation module integrated into a popular graph ML library, alongside XAI-Fb15k-237.

2. Background and Related Work

2.1. Knowledge Graph Embedding Models

KGE models represent entities and relations as dense vectors, learned by optimising a scoring function over observed triples [6, 7, 8]. Representative models include TransE [6] (translation-based), DistMult [7] and ComplEx [8] (bilinear), and ConvE [9] (convolutional). Despite strong link-prediction performance,

their scoring functions provide no provenance link to individual training triples, motivating post-hoc explanation methods [1].

2.2. Post-Hoc Explanation by Example

A recognised approach to post-hoc interpretability pairs an opaque model with a case-based reasoning (CBR) system over the same training data, so-called ANN-CBR twin-systems [10]: the predictive model is left unmodified, while the CBR system adds interpretability by retrieving similar training instances as explanatory evidence. Keane and Kenny [10] argue that grounding explanations in training cases (rather than synthetic perturbations or feature attributions) parallels reasoning-from-precedent as practised by human experts. A latent-space retrieval variant, introduced by Caruana et al. [11] for MLP explanation in a clinical domain, retrieves cases by distance in the network's learned representation, so retrieval is driven by what the model has learned rather than raw input features. ExamplE adapts this to the KGE setting, using embedding distance as the retrieval signal without modifying trained model at explanation time.

2.3. Explainability Methods for KGE Models

Existing post-hoc methods for KGE explanation pursue alternatives to latent-space retrieval. Exhaustive leave-one-out retraining (removing each candidate triple and measuring the change in target prediction score) is impractical at scale due to its $O(|\mathcal{T}|)$ retraining cost per explained triple, where $|\mathcal{T}|$ denotes the number of training triples. To our knowledge, existing methods instead reduce this cost via gradient attribution, surrogate modelling, or influence estimation.

Gradient-based. **GradientRollback (GR)** [12] stores per-triple gradient updates during a non-standard training mode and subtracts them at explanation time, requiring both a specialised training procedure and gradient access. **ExplaiNE** [13] computes the gradient of link probability with respect to the adjacency matrix and requires gradient access, and has been evaluated on social-network rather than knowledge-graph benchmarks.

Surrogate-based. **KGEx** [14] fits local surrogate KGE models on sampled subgraphs via knowledge distillation, requiring 50–200 Monte Carlo runs per target triple.

Influence-based. **Kelpie** [15] and its extension [16] estimate triple impact by post-hoc retraining on modified triple set.

The methods surveyed above estimate triple influence through model modification, surrogate fitting, or gradient signals over the trained KGE. ExamplE differs from these methods by adopting the latent-space twin-system variant discussed in Section 2.2 to identify candidate training triples, then filtering by predicate and ranking by combined distance. The trained model is left unchanged at explanation time, and each returned example is an observed triple from the training graph.

2.4. Evaluation of KGE Explanations

Evaluating explanations for KGE link prediction is itself an open problem and motivates the dataset contributed in Section 4. **LP-DIXIT** [17] provides an automated protocol using an LLM to measure forward simulatability variation, whether supplying an explanation improves the LLM's ability to reproduce the KGE's prediction. User-study evaluation requires triples whose entities and relations are interpretable to readers, which standard benchmarks such as FB15k-237 do not directly meet: they expose raw identifiers without human-readable labels. Prior work has used ad-hoc label collection [15]; to our knowledge, no shared human-readable resource has been released for this purpose.

2.5. Adjacent Work

GNN explainers [18, 19] target a different model class; adversarial-robustness work [20, 21, 22] identifies influential triples for attack analysis, not explanation.

Algorithm 1 ExamplE: explanation graph construction

Input: model M, target $t{=}(s,p,o)$, graph $\mathcal{G}{=}(\mathcal{E},\mathcal{R},\mathcal{T})$, params m, n, σ
Output: explanation graph EG

1: $S^m \leftarrow \arg\min^m_{s_i \in \mathcal{E}\setminus\{s\}} \operatorname{dist}(f(s), f(s_i))$
2: $O^m \leftarrow \arg\min^m_{o_j \in \mathcal{E}\setminus\{o\}} \operatorname{dist}(f(o), f(o_j))$
3: $eG_t \leftarrow \{(s_i, p, o_j) \in \mathcal{T} : s_i \in S^m,\ o_j \in O^m\}$, sorted by $d_{S,i} + d_{O,j}$
4: $pG_t \leftarrow \bigcap_{t_i \in eG_t} N(t_i, n) \cap N(t, n)$ (strict) **or** $\bigcup_{t_i \in eG_t} N(t_i, n) \cap N(t, n)$ (permissive)
5: **return** $EG \leftarrow pG_t \cup eG_t \cup \{t\}$

3. The ExamplE Method

ExamplE is an instance of the ANN-CBR twin-system pattern (Section 2.2), with the trained KGE as the ANN component and a latent-space retrieval routine as the CBR component. The trained model is left unmodified at explanation time; retrieval is driven by distance in the model's learned entity-embedding space, following the Caruana et al. [11] variant adapted to relational data.

Intuition ExamplE is a post-hoc, local explanation method: given a predicted link $t_{(s,p,o)}$ and a trained KGE model, it asks which training triples most plausibly shaped the model's belief that $t_{(s,p,o)}$ is true. KGE training updates entity embeddings at the triple level, creating a ripple effect: entities appearing in similar relational contexts end up close in latent space. ExamplE exploits this by finding training triples whose subjects and objects are nearest neighbours of s and o in embedding space, filtered to retain only triples with the same predicate p: these are the *influential examples*. The predicate filter grounds the explanation in the same relational context as the target and can be relaxed to retrieve examples from the broader graph neighbourhood when similar predicates are of interest. No model retraining or gradient access is required.

Notation Let $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ be a knowledge graph with entity set $\mathcal{E}$, relation set $\mathcal{R}$, and triple set $\mathcal{T}$. A triple $t_{(s,p,o)} \in \mathcal{T}$ is a directed edge from subject s to object o via predicate p. The n-hop neighbourhood of a triple is defined recursively:

$$N(t_{(s,p,o)}, n) = N(t, n{-}1) \cup \{(s', p', o') \in \mathcal{G} \mid s', o' \in S' \cup O'\} \tag{1}$$

where $S' = \{s : (s,p,o) \in N(t, n{-}1)\}$, $O' = \{o : (s,p,o) \in N(t, n{-}1)\}$, and $N(t,1) = N(s,1) \cup N(o,1) \setminus \{t\}$.

Algorithm The algorithm proceeds in four steps. Let $f(\cdot)$ denote the model's entity embedding function and $dist(\cdot,\cdot)$ a distance measure (e.g. cosine). Pseudo-algorithm available in 3.

Prerequisites. ExamplE requires a calibrated model, one whose scores have been post-processed to lie in $[0,1]$ and approximate probabilities. We follow [5], applying isotonic regression on the validation set.

Formal definitions. Let $S^m \subseteq \mathcal{E} \setminus \{s\}$ be the m entities with smallest $d_{S,i} = \operatorname{dist}(f(s), f(s_i))$, and analogously $O^m \subseteq \mathcal{E} \setminus \{o\}$ with $d_{O,j} = \operatorname{dist}(f(o), f(o_j))$. The filtered example set is:

$$eG_t = \{(s_i, p, o_j) : (s_i, p, o_j) \in \mathcal{T},\ s_i \in S^m,\ o_j \in O^m\} \tag{2}$$

ranked ascending by $\operatorname{score}(s_i, o_j) = d_{S,i} + d_{O,j}$. The strict and permissive prototype graphs are given by Equations 3 and 4:

$$pG_t^{\text{strict}} = \bigcap_i N(t_i, n) \cap N(t, n) \tag{3}$$

$$pG_t^{\text{perm}} = \bigcup_i N(t_i, n) \cap N(t, n) \tag{4}$$

The final Explanation Graph assembles all components: $EG = pG_t \cup eG_t \cup \{t\}$. Figure 2 illustrates the structure of an Explanation Graph; a diagram of the algorithm is also provided in Appendix 11.

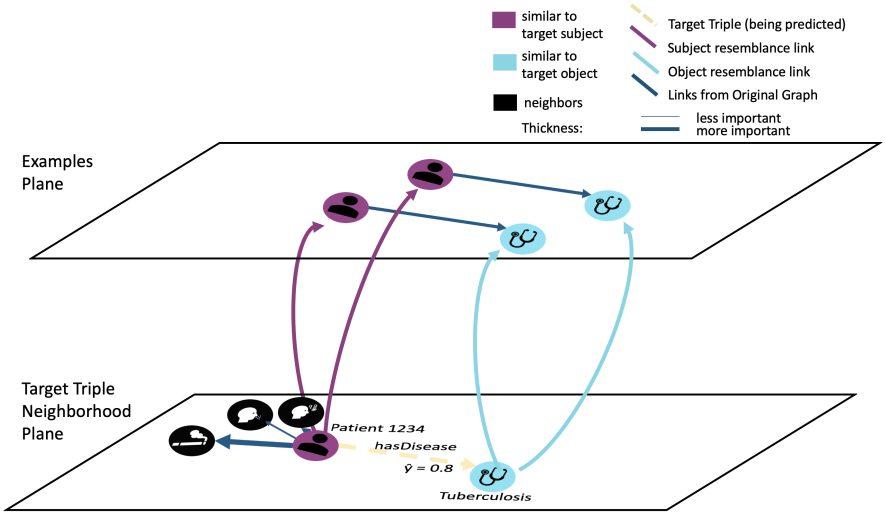


Figure 2: Example Explanation Graph produced by ExamplE. Red: target triple. Blue: influential example triples (eG_t, ranked by latent-space distance). Grey: prototype neighbourhood (pG_t). The structure links the prediction back to supporting evidence in the training graph.

Batch mode. For efficiency, the nearest-neighbour index over $\mathcal{E}$ is built once (KD-Tree: $O(k{\cdot}e{\cdot}\log e)$ time, $O(k{\cdot}e)$ space) and shared across all target triples. This gives the $\sim$5 s per-triple throughput reported in Section 6 for FB15k-237; full complexity analysis is in the Appendix.

4. XAI-Fb15k-237: A Human-Curated Evaluation Dataset

Standard KGE benchmark datasets (Fb15k-237, WN18RR) use encoded identifiers stripped of `rdfs:label` annotations during export, making human evaluation of explanations impractical without label recovery. Each published XAI method follows a different evaluation protocol, preventing fair comparison [23]. Full evaluation of the Fb15k-237 test set is infeasible at scale: assessing 100 triples for readability takes $\approx$10 minutes per annotator, implying $\sim$34 annotator-hours for a single pass over the full set.

Desiderata. An explainability evaluation dataset should be: (1) human-readable to a non-expert; (2) cognitively manageable in size; (3) predicate-diverse; (4) free of self-relations and duplicates; (5) derived from an established benchmark to leverage known model performance; (6) publicly available. Sub-setting an existing benchmark preserves comparability with published results but inherits the original dataset's biases and provenance limitations.

Construction. The construction protocol is shown in Figure 3. Freebase MIDs in the Fb15k-237 test set were mapped to human-readable labels via three complementary sources: the Freebase Browser [24], the QLever Wikidata endpoint [25], and the QLever Freebase endpoint [26]. Triples with unresolvable or ambiguous labels were excluded sequentially, yielding 19,884 triples across 184 predicates. Four KGE models (TransE, DistMult, ComplEx, HolE) were evaluated on this pool; mean subject and object ranks were computed per triple across models, and the three highest-ranked triples per predicate were retained, producing 273 candidates across 91 predicates.

Annotation. Two annotators not involved in dataset construction rated each of the 91 candidate predicate types for human readability (0 = not clearly understandable; 1 = clearly understandable). Raw agreement was 97.8 %; Krippendorff's $\alpha = 0.740$ and Cohen's $\kappa = 0.739$ [27, 28], indicating substantial agreement under conventional thresholds [29]. The high positive rate (≈96 %) is expected given the rank-based selection; the two chance-corrected metrics are consistent and mutually confirming. The published dataset includes a per-predicate readability flag for annotations, allowing users to filter to the most unambiguously interpretable subset from the study or use all preselected triples.

Statistics. XAI-Fb15k-237 contains 86 test triples spanning 86 distinct relation types. We prioritise predicate-level diversity over within-predicate examples to control annotator cognitive load and to maximise relational coverage at fixed evaluation budget. It includes annotated triples with per-predicate readability flags and `mapper.json`, a Freebase MID-to-label mapping recovered during construction. A similar protocol could be applied to other benchmarks with interpretable entities, e.g. CoDEx [30].

5. Experimental Setup

Datasets. We use Fb15k-237 and WN18RR for quantitative evaluation with in-depth analysis for TransE. Models are TransE and ComplEx, trained with open-source KGE library (AmpliGraph) on TensorFlow 2.10 on an Intel Xeon Gold 6142, 64 GB RAM, Tesla V100 16 GB (Ubuntu 18.04). All models use Adam (lr=0.0001), multiclass-NLL loss, L2 regulariser (λ=0.0001), early stopping up to 4000 epochs, seed = 0; TransE hyperparameters: WN18RR k=350, η=30; Fb15k-237 k=400, η=30. Models are calibrated post-training via isotonic regression on the validation set [5]. Link-prediction performance for the trained models is reported in Table 4 (Appendix).

Hypotheses.

- H1 After retraining with the explanation removed, predicted probabilities remain highly correlated with the originals (slope < 1, originals are higher).
- H2 Removing more explanation triples yields a larger probability drop.
- H3 The retrained model scores the target triple as less plausible than the original model.

Baselines. **Random-Predicate (RP):** draw n triples uniformly at random from all triples sharing predicate p with the target: $E = \{(e_i, p, e_j) \in \mathcal{G}\}$, sample $b_1, \ldots, b_n \sim \text{Uniform}(E)$. RP is the demanding choice of baseline: it restricts competitors to the same-predicate pool from which ExamplE selects, so any margin reflects the latent-space ranking rather than predicate filtering alone. A weaker alternative would be Random-Neighbourhood, drawing from the 1-hop neighbourhood $N(t, 1)$ irrespective of predicate; we report against RP throughout.

Metric. Probability Difference (PD%), measuring the relative change in predicted plausibility upon retraining: $PD = \frac{(M(t) - M'(t))}{M(t)} \times 100$, where $M(t)$ is the original model score and $M'(t)$ is the retrained model score. Positive PD means the explanation reduced plausibility as intended.

Evaluation protocols. **ROAR** [31]: remove explanation triples from training and retrain. Two variants: ROAR-1 (remove top-1 example) and ROAR-all (remove full explanation set). **rev-ROAR** (introduced here): remove all same-predicate triples *except* the explanation, then retrain. Where ROAR asks whether the explanation triples were necessary to the prediction, rev-ROAR asks whether they are sufficient to sustain it; the two are complementary and, in redundancy-rich settings (Section 6), rev-ROAR carries more signal. Two variants: rev-ROAR-1 and rev-ROAR-all. Each setup requires training 5 models per dataset per target triple per KGE architecture; predictions are extracted at every 10th epoch to capture temporal dynamics.

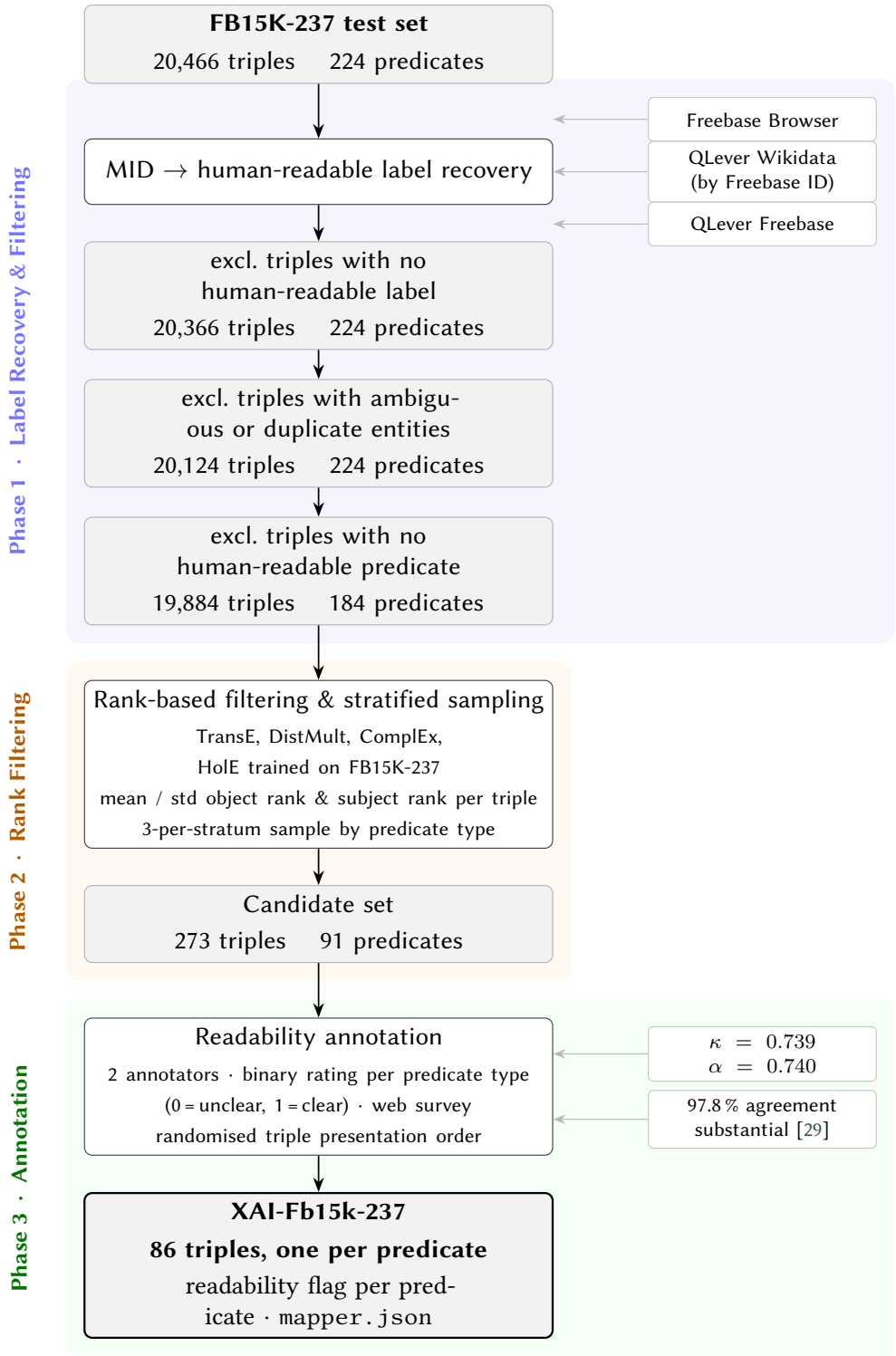


Figure 3: Construction protocol for XAI-Fb15k-237. **Phase 1**: Freebase MIDs are mapped to human-readable labels via three complementary sources; triples with unresolvable labels, ambiguous entities, or unresolvable predicates are excluded sequentially. **Phase 2**: Four KGE models are trained on FB15K-237; triples are rank-filtered and sampled at three per predicate type. **Phase 3**: Two annotators not involved in the dataset construction assess predicate-level readability; Cohen's $\kappa = 0.739$, Krippendorff's $\alpha = 0.740$ (substantial agreement [29]). The published dataset includes a per-predicate readability flag and `mapper.json`.

6. Results

Link-Prediction Baseline We report link-prediction performance to establish the quality of the models used in the explainability evaluation; we make no claim of competitiveness with published

results. For TransE on FB15k-237: H@1 = 0.20, MRR = 0.30. For TransE on WN18RR: H@1 = 0.05, MRR = 0.22. For ComplEx on FB15k-237: H@1 = 0.21, MRR = 0.31. The low H@1 for TransE on WN18RR reflects TransE's known difficulty with symmetric and antisymmetric relations, prevalent in WN18RR; the MRR of 0.22 indicates discriminative ability sufficient for the ROAR protocol. ROAR and rev-ROAR measure the influence of training triples on prediction scores for all evaluated triples, regardless of whether the target triple is correctly ranked first; ExamplE generates explanations for any prediction the model makes. Full results and hyperparameters are in the Appendix Table 4.

epoch	avg PD	rev-ROAR [%] ↓		ROAR [%] ↑	
		1	all	1	all
10 ours	**-0.507**	**69.694**	63.971	**0.106**	5.717
rand.	-0.478	73.757	**62.186**	0.013	**14.647**
20 ours	**-0.066**	**25.457**	23.819	-0.0	**-0.009**
rand.	0.007	27.968	**21.456**	**0.001**	-0.011
30 ours	-0.181	**10.911**	**8.27**	-0.003	**0.013**
rand.	**-0.348**	11.117	8.388	**-0.001**	-0.073
40 ours	**0.009**	**5.312**	4.735	0.0	0.034
rand.	0.096	5.435	**4.154**	**0.002**	**0.048**
50 ours	**-0.049**	**2.707**	2.227	0.03	**0.036**
rand.	0.021	2.863	**2.196**	**0.031**	0.022
60 ours	**-0.112**	**1.635**	1.456	-0.002	-0.016
rand.	-0.057	1.718	**1.332**	**-0.001**	**-0.007**
70 ours	**-0.135**	**0.894**	**0.786**	-0.0	**-0.011**
rand.	-0.072	0.94	0.721	-0.0	-0.015
80 ours	0.053	**0.574**	0.496	-0.001	**-0.001**
rand.	**0.002**	0.599	**0.456**	-0.001	-0.038
90 ours	-0.153	**0.384**	0.325	-0.0	**0.028**
rand.	**-0.421**	0.402	**0.303**	-0.0	-0.0
100 ours	**-0.153**	**0.23**	0.203	-0.007	-0.01
rand.	-0.03	0.244	**0.176**	-0.007	**-0.005**

Table 1
TransE on FB15k-237. PD% across training epochs.
↓ = lower is better; ↑ = higher is better.
Bold = best per cell.

epoch	avg PD	rev-ROAR [%] ↓		ROAR [%] ↑	
		1	all	1	all
10 ours	0.0	13.515	15.018	**1.118**	**1.696**
rand.	**-0.004**	-	-	0.089	-0.306
20 ours	-0.004	26.975	29.598	**0.008**	**3.5**
rand.	**-0.027**	-	**25.585**	-0.021	-0.058
30 ours	0.018	**34.791**	37.518	**0.031**	**1.198**
rand.	**0.005**	34.384	-	-0.046	0.035
40 ours	**-0.004**	38.778	42.265	**0.048**	**0.041**
rand.	0.02	-	-	0.021	0.036
50 ours	**0.006**	**40.661**	44.588	**0.038**	**0.095**
rand.	0.025	42.159	**41.134**	-0.01	0.004
60 ours	**0.015**	**42.071**	45.949	**0.054**	**0.018**
rand.	0.042	43.586	**42.388**	0.011	0.028
70 ours	-0.032	**43.149**	46.509	**0.071**	**0.021**
rand.	**-0.309**	44.506	-	0.005	-0.002
80 ours	**-0.017**	**42.835**	46.048	**0.025**	**0.017**
rand.	-0.252	44.057	-	0.001	-0.02
90 ours	**0.047**	**42.512**	45.655	0.01	0.012
rand.	0.058	43.58	-	**0.011**	**0.013**
100 ours	**-0.008**	**41.778**	44.503	**0.023**	0.002
rand.	0.01	42.673	**41.57**	0.021	**0.014**

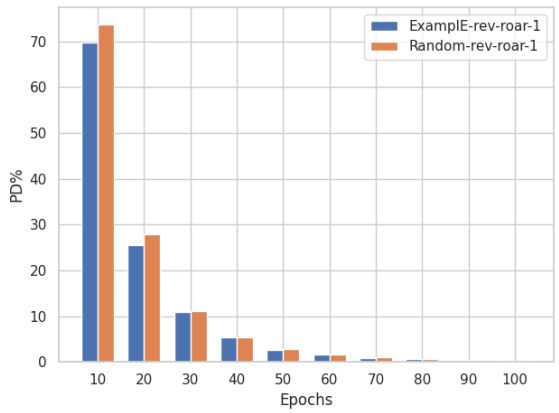
Table 2
TransE on WN18RR. PD% across training epochs.
Conventions as in Table 1.
"—" = random baseline found no valid same-predicate triple.

Quantitative Evaluation: TransE on FB15k-237 and WN18RR. Tables 1 and 2 report per-epoch probability difference (PD%) for ExamplE and the Random-Predicate baseline (RP) on TransE with FB15k-237 and WN18RR. In rev-ROAR, the training set is reduced to only the identified explanation and the model is retrained from scratch: *lower PD% is better*, indicating the explanation is sufficient to sustain the prediction. In ROAR, the explanation is removed and the model is retrained: *higher PD% is better*, indicating the removed example carried influence. Each condition is evaluated at the top-1 example (1) and at the full returned set (all). Predictions are extracted every 10 epochs through to epoch 100 to capture training dynamics, with each checkpoint reported in Tables 1 and 2. The cost of the ROAR/rev-ROAR retraining loop (five models per target triple per condition per architecture) made denser sampling infeasible. We organise observations by metric column.

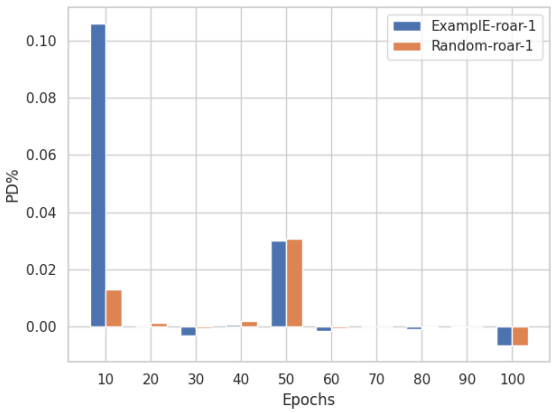
Average PD. Across both datasets, average-PD values fall within a narrow range around zero ($[-0.5, 0.1]$ on FB15k-237, $[-0.31, 0.06]$ on WN18RR), with no consistent sign or ordering between methods. We do not draw conclusions from this column; the magnitudes lie within per-triple variation across the test set.

rev-ROAR-1 (lower is better). ExamplE produces lower PD% than RP at every reported epoch on both datasets, wherever RP returns a valid value. This is the most consistent separation in the tables. PD% trajectories differ: FB15k-237 contracts monotonically from 70% at epoch 10 to below 1% by epoch 70; WN18RR grows from 13% to 42% by epoch 70. RP is undefined in several early-epoch WN18RR rows due to the dataset's small predicate set.

rev-ROAR-all (lower is better). The pattern inverts. RP achieves lower PD% than ExamplE at most



(a) rev-ROAR-1 (lower is better), ExamplE shows higher impact consistently.



(b) ROAR-1 (higher is better), ExamplE inconsistent beyond epoch 10.

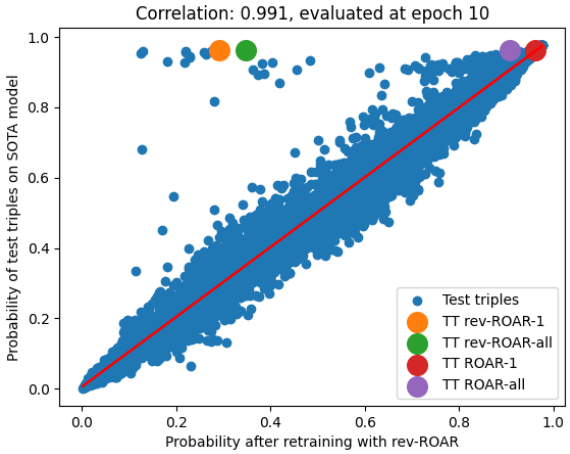
Figure 4: Per-triple PD% for ExamplE vs RP, TransE on FB15k-237, checked every 10 epochs for 100 epochs.

epochs on both datasets, with margins typically within 1–3 percentage points. ExamplE's selection criterion does not transfer from the top-1 case to the full-set case in this condition.

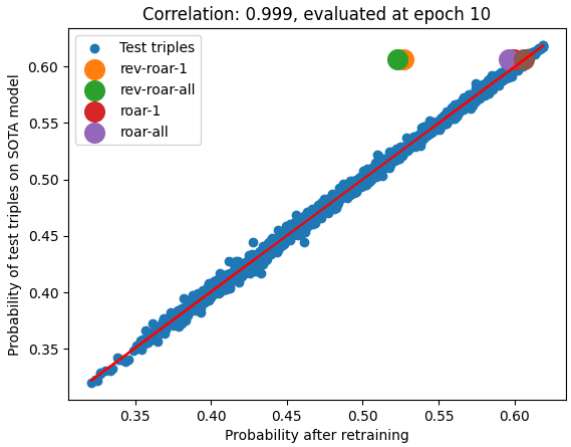
ROAR-1 (higher is better). The two datasets diverge. On FB15k-237, ExamplE separates from RP at epoch 10 (0.106 vs 0.013, an order-of-magnitude difference), but from epoch 20 onward both methods report values within $[-0.01, 0.03]$ with no consistent ordering. On WN18RR, ExamplE remains positive and above RP at almost every reported epoch through 100, with the largest separation at epoch 10 (1.118 vs 0.089).

ROAR-all (higher is better). The two datasets diverge more sharply. On FB15k-237 at epoch 10, RP exceeds ExamplE substantially (14.647 vs 5.717), and from epoch 20 onward both methods are within $[-0.07, 0.05]$ and indistinguishable from noise. On WN18RR, ExamplE exceeds RP at every reported epoch from 10 through 80.

Per-Triple Visual Comparison Figure 4 shows per-triple PD% for ExamplE versus RP across epochs on FB15k-237. In the rev-ROAR-1 condition (left), ExamplE points cluster below the baseline; in ROAR-1 (right), they are inconsistent with large difference in favor of ExamplE at epoch 10. The remaining conditions are in the Appendix 8.



(a) TransE on FB15k-237 (rev-ROAR).



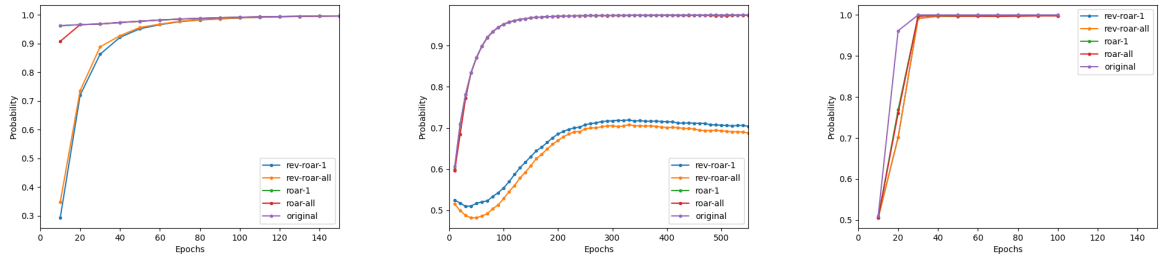
(b) TransE on WN18RR (ROAR).

Figure 5: Snapshot at epoch 10: Pearson correlation between original and post-retraining plausibility scores across triples. Correlation saturates at 1 above epoch 20.

Hypothesis Evaluation **H1** (calibration preservation): supported. Pearson correlation between original and post-retraining scores reaches 1 above epoch 20 in all settings (Figure 5); slope below 1 throughout. **H2** (size–impact monotonicity): supported. Removing the full explanation set produces larger PD% than removing only the top-1 example in the ROAR condition across both datasets and all epochs where the signal is detectable.

H3 (ExamplE examples carry predictive influence): supported in single-triple conditions on both datasets and in WN18RR ROAR-all through epoch 80; not supported in rev-ROAR-all on either dataset or in FB15k-237 ROAR-all at epoch 10 (RP 14.647% vs ExamplE 5.717%). Findings suggest that rev-ROAR-all in high-redundancy settings is sensitive to the breadth of the perturbation set, not only to per-triple targeting. The top-1/full-set asymmetry is discussed below.

Relational Redundancy and Recoverability Across both datasets, post-retraining plausibility scores recover toward original values as training progresses (Figure 6). We read this as a property of how the model relates to the data: a KGE model encodes regularities spanning multiple triples, several of which can independently support the same prediction. We refer to this property as **relational redundancy**. Two patterns support the reading. First, the asymmetry between top-1 and full-set conditions: ExamplE separates from RP at the single-triple level on both datasets but not at the set level, consistent with ranked-by-distance selection helping when one triple is chosen and offering little advantage when many are. Second, the dataset contrast: signal compresses faster on FB15k-237 (denser, more relations) than on WN18RR (sparser, 11 predicates) and also on a stronger model (ComplEx), and the ROAR-all FB15k-237 anomaly fits this frame. The reading is consistent with the data but inferential; we have not measured redundant-triple counts directly.



(a) TransE / FB15k-237. (b) TransE / WN18RR. (c) ComplEx / FB15k-237.

Figure 6: Target-triple plausibility across epochs under ROAR and rev-ROAR for representative single triples. On FB15k-237 (left, right) scores recover with training; on WN18RR (centre) the gap widens. Aggregate behaviour is in Tables 1,2.

Qualitative Examples Figure 7 shows ExamplE outputs for two target triples: *Artie Lange* $\xrightarrow{influenced_by}$ *Jackie Gleason* (FB15k-237, within the quantitative evaluation scope) and *Billy Idol* $\xrightarrow{languages_spoken}$ *English* (CoDEx-M, run for efficiency evaluation only; included for its interpretable labels). The full numeric explanations are in the Appendix. Two examples do not constitute a systematic assessment of explanation quality; a broader evaluation requires a user study, for which XAI-Fb15k-237 was designed.

Dataset	Total [s]	/triple [s]
FB15k-237	104,069	5.09
CoDEx-M	60,364	5.85
WN18RR	36,049	12.33

Table 3: ExamplE explanation generation time, TransE. ComplEx times are approximately double (∼9.6 s/triple on FB15k-237; ∼13.1 s/triple on WN18RR).

Efficiency Comparison ExamplE generates an explanation in ∼5 s per triple on FB15k-237 (Table 3). Reported and measured per-triple times for prior methods on FB15k-237 are ∼32 s for Kelpie's necessary-

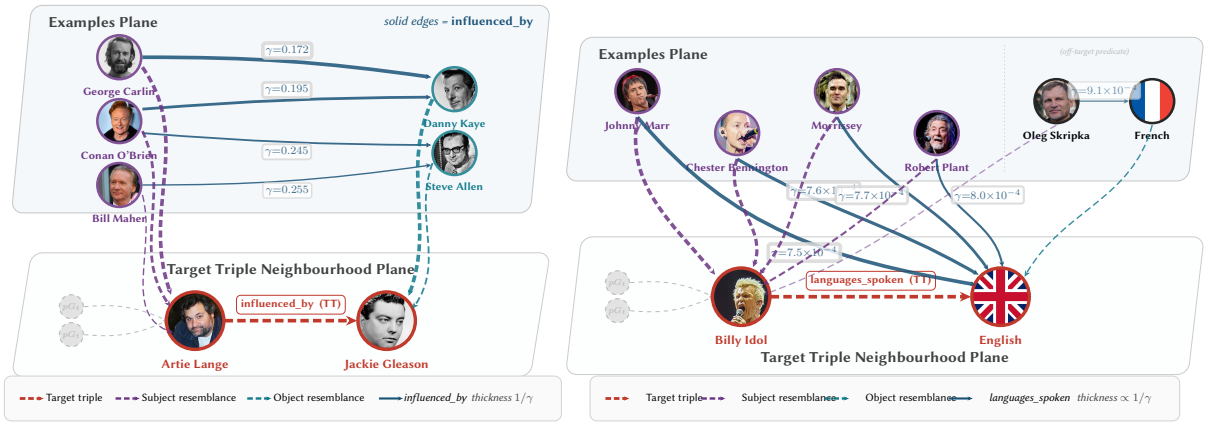


Figure 7: ExamplE explanation graphs (dashed red = target triple). Violet/teal dashed arcs link similar subjects/objects across planes; solid navy edges in the Examples Plane are the original graph's relation. Edge thickness $1/\gamma$.

explanation mode [15] (measured locally on a 100-triple sample from the article), ∼1 min 45 s for KGEx [14] (measured locally on a 20-triple sample using a private implementation provided by the authors), and ∼6 min for GradientRollback [12] (as reported). These figures span different hardware and configurations and are indicative rather than controlled. Memory overhead for ExamplE is $O(k \cdot |\mathcal{E}|)$, ≈22 MB for TransE on FB15k-237 (k=400). GradientRollback's influence matrix requires $|\mathcal{T}| \times 3k$ additional parameters at training time; at k=100 on FB15k-237 this exceeds 2.4 GB (over 100× the training set's 22.6 MB), derived analytically from the published method design [12] and corroborated by a small local run with reduced parameters (Figure 8). Full-scale runs were infeasible in our setup. We compare on efficiency only; relative explanation quality across methods is not assessed here.

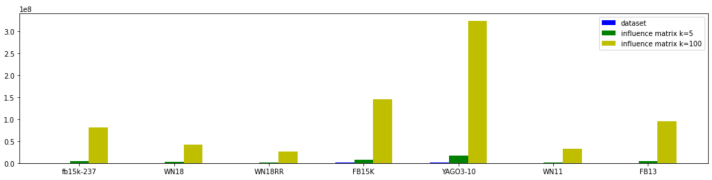


Figure 8: GradientRollback memory footprint vs. embedding dimension k on FB15k-237, derived analytically from [12]. At k=100, the influence matrix exceeds 2.4 GB; ExamplE uses ∼22 MB at k=400.

7. Discussion

Summary. The pre-specified hypotheses are supported in the conditions where the evaluation has signal: ExamplE examples carry predictive influence at the single-triple level (H3, with caveats discussed in Section 6), the effect is monotone in explanation size in ROAR (H2), and post-retraining scores correlate with originals at r=1 above epoch 20 (H1). The dataset contrast (FB15k-237 signal compresses by epoch 70, WN18RR persists) is consistent with relational redundancy varying with graph density and motivates ExamplE's set-based output.

Limitations. ExamplE is a heuristic without a faithfulness guarantee; we cannot certify optimality of the returned set. XAI-Fb15k-237 was curated for user studies but no study with external annotators has been conducted. The empirical comparison with prior work is limited and bounded to efficiency: GradientRollback was run locally, but its training-time memory overhead made full-scale experiments at our embedding sizes infeasible; KGEx has no public implementation, and our measurement is limited to a small-sample local run using a private implementation provided by the authors; Kelpie was run on a 100-triple sample of FB15k-237 in necessary-explanation mode. Hardware and configurations differ across methods, so the timing comparison is indicative rather than controlled, and explanation quality

is not assessed. The methods also produce different explanation types (counterfactual perturbation sets for Kelpie, similarity-based ranked examples for ExamplE), so efficiency comparison should be read with that distinction in mind.

Scope of explanation. ExamplE describes how the model arrives at its scores by proximity in embedding space, not the real-world mechanism by which the fact comes to hold. KGE models represent regularities through geometric proximity rather than variables supporting interventions or counterfactuals [32]; a post-hoc explanation importing causal vocabulary would not be faithful to what the model computes [33, 34]. Example-based explanations suit similarity-driven predictions [35]; where causal accounts are required, the limitation lies in the model class, and architectures combining learned representations with explicit symbolic structure [36] may be more appropriate.

Future work. The most direct next step is a user-study comparison of explanation quality across methods on XAI-FB15k-237, alongside evaluation of the prototype aggregation step (Section 3). A complementary direction is establishing faithfulness guarantees for the returned set, since the present method is a heuristic. Applying ExamplE to GNN-based models is a further extension, as is testing the ranked examples as few-shot prompts on KG-grounded QA, with relevance to retrieval-augmented generation and grounding LLM reasoning in knowledge graph databases.

8. Conclusion

We presented ExamplE, a case-based heuristic that explains KGE link predictions by retrieving training examples from the latent-space neighbourhood of the target triple. The method requires no special training mode and, in our experiments, generated explanations in ∼5 s per triple on FB15k-237; GradientRollback reports ∼6 min under a different setup, a difference indicative rather than directly comparable. The ROAR and rev-ROAR evaluation supports a relational-redundancy reading of post-retraining recoverability, which in turn motivates the set-based output of the method. We release XAI-Fb15k-237, a curated set of 86 interpretable triples for user-study evaluation, alongside the open-source implementation.

The design is intentionally simple, which enables ExamplE to attach to deployed KGE pipelines without retraining or architectural change. As emerging transparency requirements (e.g., Article 13 of the EU AI Act [37]) create pressure on production systems, this positions ExamplE as a lightweight way to support such transparency obligations for already-deployed models in the knowledge graph infrastructure. Explanation alone, however, does not establish regulatory compliance: Article 13 compliance also depends on documentation, instructions for use, and the broader risk-management and quality-management obligations of Chapter III, which fall outside the scope of this work.

References

[1] F. Bianchi, G. Rossiello, L. Costabello, M. Palmonari, P. Minervini, Knowledge graph embeddings and explainable AI (2020). URL: http://arxiv.org/abs/2004.14843. doi:10.3233/SSW200011. arXiv:2004.14843.

[2] L. Costabello, S. Pai, N. McCarthy, A. Janik, Knowledge graph embeddings tutorial: From theory to practice, 2020. URL: https://doi.org/10.5281/zenodo.4268208. doi:10.5281/zenodo.4268208, https://kge-tutorial-ecai2020.github.io/.

[3] Regulation (eu) 2026/1744 of the european parliament and of the council of 8 july 2026 amending regulations (eu) 2024/1689, (eu) 2018/1139 and (eu) 2023/1230 as regards the simplification of the implementation of harmonised rules on artificial intelligence (digital omnibus on ai), 2026. URL: https://eur-lex.europa.eu/eli/reg/2026/1744/oj/eng, cELEX:32026R1744.

[4] P. Pradeep, M. Caro-Martínez, A. Wijekoon, Empowering explainable artificial intelligence through case-based reasoning: A comprehensive exploration, IEEE Transactions on Knowledge and Data Engineering (2025).

[5] P. Tabacof, L. Costabello, Probability calibration for knowledge graph embedding models (2020). URL: http://arxiv.org/abs/1912.10000. arXiv:1912.10000.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013.

[7] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[8] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, New York, NY, USA, 2016, pp. 2071–2080.

[9] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D Knowledge Graph Embeddings, arXiv:1707.01476 [cs] (2018). URL: http://arxiv.org/abs/1707.01476, arXiv: 1707.01476.

[10] M. T. Keane, E. M. Kenny, How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems, in: K. Bach, C. Marling (Eds.), Case-Based Reasoning Research and Development, Lecture Notes in Computer Science, Springer International Publishing, 2019, pp. 155–171. doi:10.1007/978-3-030-29249-2_11.

[11] R. Caruana, H. Kangarloo, J. D. Dionisio, U. Sinha, D. Johnson, Case-based explanation of non-case-based learning methods, Proceedings. AMIA Symposium (1999) 212–215.

[12] C. Lawrence, T. Sztyler, M. Niepert, Explaining neural matrix factorization with gradient rollback (2020). URL: http://arxiv.org/abs/2010.05516. arXiv:2010.05516.

[13] B. Kang, J. Lijffijt, T. De Bie, Explanations for Network Embedding-Based Link Predictions, in: M. Kamp, I. Koprinska, A. Bibal, T. Bouadi, B. Frénay, L. Galárraga, J. Oramas, L. Adilova, Y. Krishnamurthy, B. Kang, C. Largeron, J. Lijffijt, T. Viard, P. Welke, M. Ruocco, E. Aune, C. Gallicchio, G. Schiele, F. Pernkopf, M. Blott, H. Fröning, G. Schindler, R. Guidotti, A. Monreale, S. Rinzivillo, P. Biecek, E. Ntoutsi, M. Pechenizkiy, B. Rosenhahn, C. Buckley, D. Cialfi, P. Lanillos, M. Ramstead, T. Verbelen, P. M. Ferreira, G. Andresini, D. Malerba, I. Medeiros, P. Fournier-Viger, M. S. Nawaz, S. Ventura, M. Sun, M. Zhou, V. Bitetta, I. Bordino, A. Ferretti, F. Gullo, G. Ponti, L. Severini, R. Ribeiro, J. Gama, R. Gavaldà, L. Cooper, N. Ghazaleh, J. Richiardi, D. Roqueiro, D. Saldana Miranda, K. Sechidis, G. Graça (Eds.), Machine Learning and Principles and Practice of Knowledge Discovery in Databases, volume 1524, Springer International Publishing, Cham, 2021, pp. 473–488. URL: https://link.springer.com/10.1007/978-3-030-93736-2_36. doi:10.1007/978-3-030-93736-2_36.

[14] V. Baltatzis, L. Costabello, Kgex: Explaining knowledge graph embeddings via subgraph sampling and knowledge distillation, in: S. Villar, B. Chamberlain (Eds.), Proceedings of the Second Learning on Graphs Conference, volume 231 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 27:1–27:13. URL: https://proceedings.mlr.press/v231/baltatzis24a.html.

[15] A. Rossi, D. Firmani, P. Merialdo, T. Teofili, Explaining Link Prediction Systems based on Knowledge Graph Embeddings, in: Proceedings of the 2022 International Conference on Management of Data, ACM, Philadelphia PA USA, 2022, pp. 2062–2075. URL: https://dl.acm.org/doi/10.1145/3514221.3517887. doi:10.1145/3514221.3517887.

[16] R. Barile, C. d'Amato, N. Fanizzi, Explanation of Link Predictions on Knowledge Graphs via Levelwise Filtering and Graph Summarization, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web, volume 14664, Springer Nature Switzerland, Cham, 2024, pp. 180–198. URL: https://link.springer.com/10.1007/978-3-031-60626-7_10. doi:10.1007/978-3-031-60626-7_10.

[17] R. Barile, C. d'Amato, N. Fanizzi, LP-DIXIT: Evaluating Explanations for Link Predictions on

Knowledge Graphs using Large Language Models, in: Proceedings of the ACM on Web Conference 2025, ACM, Sydney NSW Australia, 2025, pp. 4034–4042. URL: https://dl.acm.org/doi/10.1145/3696410.3714667. doi:10.1145/3696410.3714667.

[18] R. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, GNNExplainer: Generating explanations for graph neural networks 32 (2019) 9240–9251. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7138248/.

[19] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, Y. Chang, GraphLIME: Local interpretable model explanations for graph neural networks (2020). URL: http://arxiv.org/abs/2001.06216. arXiv:2001.06216.

[20] P. Bhardwaj, J. Kelleher, L. Costabello, D. O'Sullivan, Poisoning Knowledge Graph Embeddings via Relation Inference Patterns, 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), Association for Computational Linguistics, Online, 2021, pp. 1875–1888. URL: https://aclanthology.org/2021.acl-long.147. doi:10.18653/v1/2021.acl-long.147.

[21] P. Betz, C. Meilicke, H. Stuckenschmidt, Adversarial Explanations for Knowledge Graph Embeddings, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 2022, pp. 2820–2826. URL: https://www.ijcai.org/proceedings/2022/391. doi:10.24963/ijcai.2022/391.

[22] P. Pezeshkpour, Y. Tian, S. Singh, Investigating robustness and interpretability of link prediction via adversarial modifications, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 3336–3347. URL: https://aclanthology.org/N19-1337. doi:10.18653/v1/N19-1337.

[23] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning (2017).

[24] University of Freiburg, Freebase easy browser, https://freebase-easy.cs.uni-freiburg.de/browse/, 2014. Accessed: 2025-05.

[25] Qlever wikidata endpoint, https://qlever.dev/wikidata, 2016. Accessed: 2025-05.

[26] Qlever freebase endpoint, https://qlever.dev/freebase, 2016. Accessed: 2025-05.

[27] K. Krippendorff, Computing Krippendorff's Alpha-Reliability, Technical Report, University of Pennsylvania, Annenberg School for Communication, 2011. URL: https://repository.upenn.edu/entities/publication/034a6030-c584-4d14-9d3d-7b7e8d16df20.

[28] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement 20 (1960) 37–46. URL: https://journals.sagepub.com/doi/10.1177/001316446002000104. doi:10.1177/001316446002000104.

[29] J. R. Landis, G. G. Koch, The Measurement of Observer Agreement for Categorical Data, Biometrics 33 (1977) 159. URL: https://www.jstor.org/stable/2529310?origin=crossref. doi:10.2307/2529310.

[30] T. Safavi, D. Koutra, CoDEx: A comprehensive knowledge graph completion benchmark, 2020. URL: http://arxiv.org/abs/2009.07810. arXiv:2009.07810 [cs].

[31] S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A benchmark for interpretability methods in deep neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[32] J. Pearl, Causality: models, reasoning, and inference, second edition, reprinted with corrections ed., Cambridge University Press, Cambridge New York, NY Port Melbourne New Delhi Singapore, 2022.

[33] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215. URL: https://www.nature.com/articles/s42256-019-0048-x. doi:10.1038/s42256-019-0048-x.

[34] A. Jacovi, Y. Goldberg, Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp.

4198–4205. URL: https://www.aclweb.org/anthology/2020.acl-main.386. doi:10.18653/v1/2020.acl-main.386.
[35] B. Kim, R. Khanna, O. O. Koyejo, Examples are not enough, learn to criticize! Criticism for Interpretability, in: Advances in Neural Information Processing Systems, volume 29, Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/hash/5680522b8e2bb01943234bce7bf84534-Abstract.html.
[36] A. d. Garcez, L. C. Lamb, Neurosymbolic AI: the 3rd wave, Artif. Intell. Rev. 56 (2023) 12387–12406. URL: https://doi.org/10.1007/s10462-023-10448-w. doi:10.1007/s10462-023-10448-w.
[37] Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act), https://eur-lex.europa.eu/eli/reg/2024/1689, 2024. Official Journal L, 12 July 2024.

Supplemental Material Statement

Code, dataset, and experimental outputs supporting this paper are provided as supplemental material under this link: https://tinyurl.com/4m2bnf3r. The archive is organised into four top-level folders:

- `kge-xai-ampligraph-main/` contains an open-source KGE library AmpliGraph extended with a case-based explanation module. The explanation package is located at `ampligraph/explanations/` and provides an abstract explainer interface (`abstract_explainer.py`), the ExamplE method (`example.py`), and a random-baseline explainer (`random_explainer.py`), alongside utility code for graph navigation and explanation visualisation. The host library provides KGE model implementations (TransE, ComplEx, DistMult, HolE, ConvE), calibration, evaluation, and dataset loaders. Installation instructions, licence, unit tests, and a tutorial notebook (`notebooks/AmpliGraph-Tutorials/Explanations 1 - Example-based Explanations.ipynb`) are included.
- `kge-xai-ampligraph-main-py312/` contains the same AmpliGraph explanation library upgraded to Python 3.12 / TensorFlow 2.16+ (Keras 3). This version is provided for forward compatibility; the experimental results in the paper were produced with `kge-xai-ampligraph-main/`.
- `ExamplE-experiments/` contains the experiment scripts used to produce the results in Section 6 and the Appendix: explanation generation (`get_explanations.py`), the ROAR/rev-ROAR retraining loop (`retraining_experiment.py`, `run_experiment.py`), efficiency measurement (`training_time_experiment.py`), probability-difference aggregation (`check_proba_differences_of_experiments.py`, `measure_diff.py`), and plotting (`Plots-ExamplE.ipynb`, `Training time-probability experiment ExamplE.ipynb`). The `User-Study/` subfolder contains the inter-annotator agreement script (`iaa.py`), the rating spreadsheet (`user-eval.xlsx`), and the agreement report (`iaa-report.txt`) underlying the Krippendorff's α and Cohen's κ values reported in Section 4.
- `fb15k-237/` contains the XAI-Fb15k-237 dataset contributed by this paper. `triples_agreed.csv` provides the 91 curated test triples; `interpretable_triples.csv` contains the subset of 86 on which both annotators agreed; `mapper.json` provides the Freebase-MID-to-label mapping recovered during construction; `test_human.txt` and `test_human_ids.txt` provide human-readable and identifier-form versions of the test split. The folder also includes the original Fb15k-237 train/valid/test splits for convenience and the original dataset licence document (Microsoft Research).

A README at the library root provides installation instructions.

Declaration of use of Generative AI

The authors used Anthropic's Claude (Opus 4.7 model) for editorial and language-level support during manuscript preparation, including identifying cross-section inconsistencies and drafting prose revisions

specified by the authors. All scientific content and final wording were authored and reviewed by the human authors, who retain full responsibility for the manuscript.

Appendix

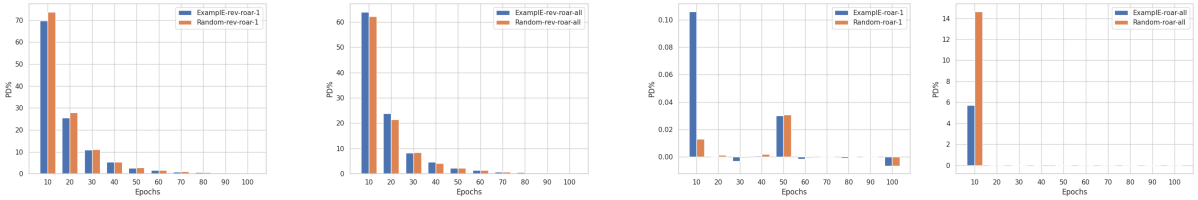


Figure 9: Probability differences between ExamplE and random baseline, TransE trained on Fb15k-237.

Approach	TransE		ComplEx		DistMult		ConvE	
	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR
Fb15k-237	0.20	0.30	0.21	0.31	0.21	0.30	0.21	0.30
WN18RR	0.05	0.22	0.47	0.50	0.43	0.47	0.44	0.47

Table 4
MRR and Hits@1 on the Fb15k-237 and WN18RR benchmark datasets.

A. Computational Complexity Analysis

We have analyzed computational complexity of the ExamplE in the scenario of batch explanations. ExamplE requires access to the training dataset so space complexity starts from $O(t)$. Given: e - number of entities (e.g.: 14,541 in Fb15k-237). k - embedding vector dimension (e.g. k=400, TransE on Fb15k-237). m - number of nearest neighbours considered (parameter of ExamplE default m=25). t - number of triples in the train set. x - number of examples, as explanation $x << t$. we can split computational complexity into steps: 1) Sampling - this step is entirely dependent on the nearest neighbour algorithm implementation, in the experiments we used implementation provided in sklearn, which by default tries to adjust parameters for best efficiency. In the worst case scenario it uses a brute force approach which complexity of the prediction time is $O(e \times k \times m)$ with negligible complexity of initialization of the algorithm and negligible space complexity too. In the best case scenario kNN algorithm tries to adjust the inner data structure for optimized inference time with the cost of initialization and space e.g. in the case of KD-Tree it is $O(k \times e \times log(e))$ of extra initialization time and $O(k \times e)$ space with a benefit of inference time being $O(m \times log(e))$. ExamplE needs to find m nearest neighbours for both subject and object entity (in the default case) in this step. 2) Filtering for example triples - in this step we need to take a cartesian product of obtained sets of neighbours in step 1: which leaves us with $O(m^2)$ (default case, in full case it is $O(m^3)$ if we are considering predicates embedding as well) and forces us to filter examples according to the dataset. First we are mapping it into a tuples ($O(t)$, where t is a number of train triples), than we utilize sets intersections implementation in Python with complexity of $O(min(t, m^2))$. In the post-processing step we compute the score per each example obtained. The computational complexity in batch explain is always dependent on the number of target triples to obtain explanations for.

step	initialization time	space
$sampling_{KD-Tree}$	$O(k \times e \times log(e))$	$O(k \times e)$
$sampling_{brute-force}$	$O(1)$	$O(1)$

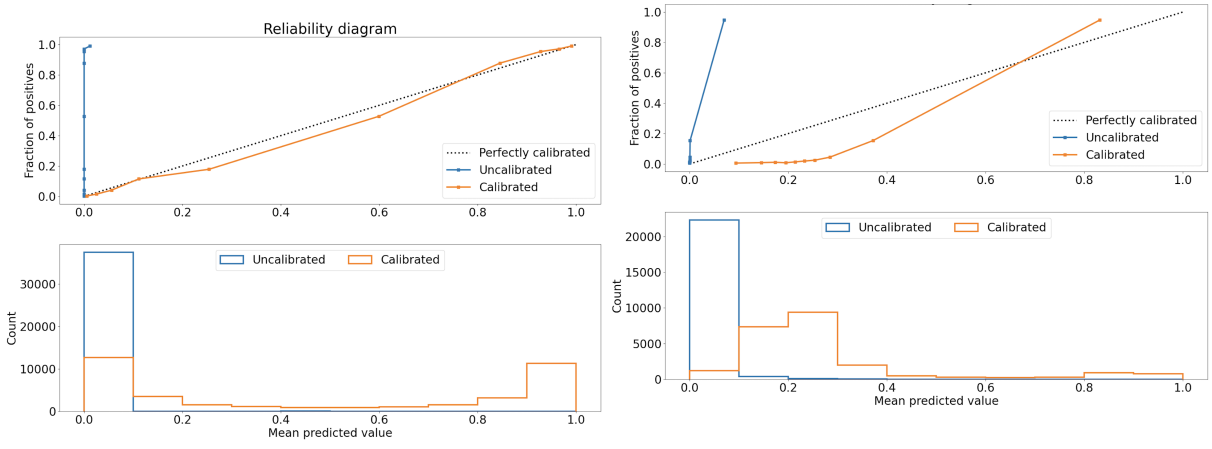
Table 5
Computational time and space complexity of initialization phase.

step	prediction time	space
$sampling_{KD-Tree}$	$O(m \times log(e))$	$O(1)$
$sampling_{brute-force}$	$O(e \times k \times m)$	$O(1)$
product	$O(m^2)$	$O(1)$
mapping	$O(t)$	$O(t)$
filtering	$O(min(t, m^2))$	$O(1)$
post-processing	$O(x)$	$O(1)$

Table 6
Computational time and space complexity of prediction phase.

B. Models Calibration

Knowledge Graph Embedding models predictions are uncalibrated - meaning they do not represent probability of the triple but rather a plausibility score. Explainability approaches presented in this work requires models to be calibrated. Achieving such property with knowledge graph embedding models requires additional post-processing to make sure the returned predictions can be interpreted as probabilities. We calibrate the trained model using procedure described in [5] on the validation test. Figure B shows the reliability diagram for uncalibrated and calibrated scores compared with perfectly calibrated reference for two datasets. We can see that uncalibrated scores (in blue) are unevenly distributed and concentrated around the lower part of predicted values, whereas calibrated scores (in orange) although not perfect are distributed much more evenly across the mean predicted values space. By calibrating pre-trained knowledge graph embedding models to return values between 0 and 1 that are more evenly distributed, we return more reliable and interpretable probabilities of triples.



(a) (b)

Figure 10: (a) Reliability diagram of the TransE model's calibration for FB15k-237 dataset. (b) Reliability diagram of the TransE model's calibration for WN18RR dataset.

C. Method Diagram

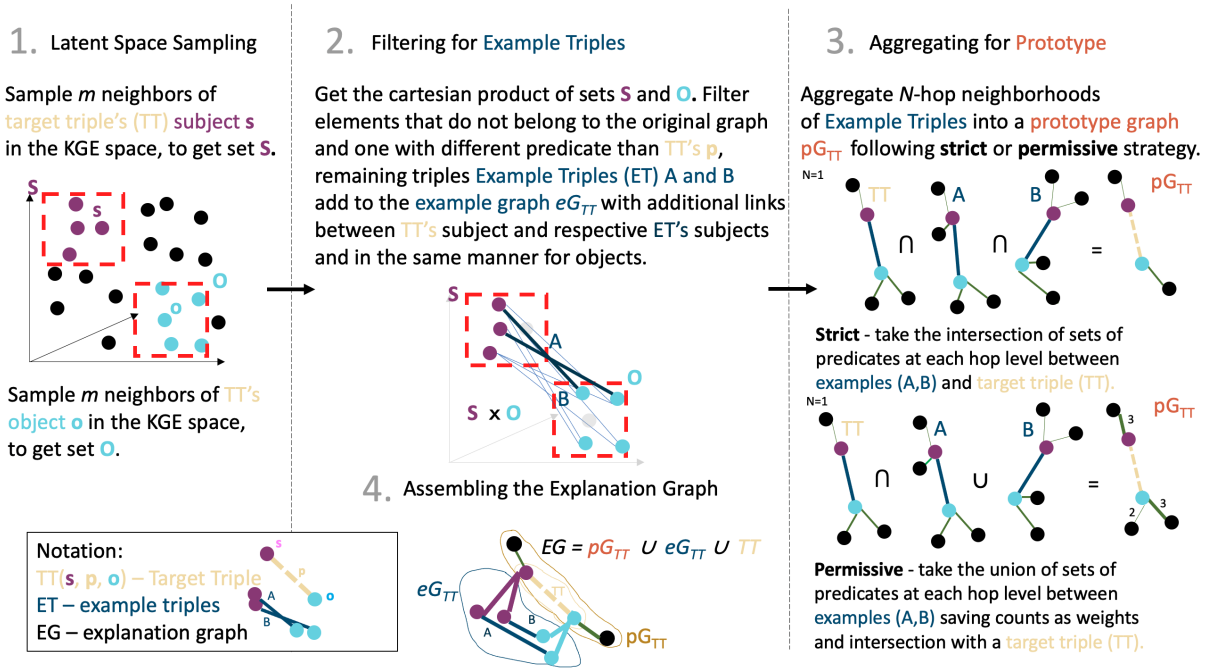


Figure 11: Step-by-step illustration of the ExamplE algorithm: (1) latent-space sampling of nearest neighbours of the target subject and object; (2) filtering for Cartesian-product pairs co-occurring in $\mathcal{T}$ under predicate p; (3) optional prototype aggregation; (4) assembly into the Explanation Graph.

D. Examples

S	P	O	Score
Artie Lange	/influence/influence_node/influenced_by	Jackie Gleason	TT
George Carlin	/influence/influence_node/influenced_by	Danny Kaye	0.17232
Conan O'Brien (aka Big Red)	/influence/influence_node/influenced_by	Danny Kaye	0.19481
Conan O'Brien (aka Big Red)	/influence/influence_node/influenced_by	Steve Allen	0.24502
Bill Maher	/influence/influence_node/influenced_by	Steve Allen	0.25498

Table 7
Example explanation for a FB15k-237 test triple. First row: target triple (TT); lower score = closer to TT.

S	P	O	Score
Billy Idol	languages spoken, written, or signed	English	TT
Johnny Marr	languages spoken, written, or signed	English	0.00075
Chester Bennington	languages spoken, written, or signed	English	0.00076
Morrissey	languages spoken, written, or signed	English	0.00077
Loreena McKennitt	languages spoken, written, or signed	English	0.00077
Gordon Lightfoot	languages spoken, written, or signed	English	0.00080
Alan Stivell	languages spoken, written, or signed	English	0.00080
Robert Plant	languages spoken, written, or signed	English	0.00080
Oleg Skripka	languages spoken, written, or signed	French	0.00091
Alan Stivell	languages spoken, written, or signed	French	0.00091
Oleg Skripka	languages spoken, written, or signed	Russian	0.00097
Oleg Skripka	languages spoken, written, or signed	Ukrainian	0.00117

Table 8
Example explanation for a test triple in CODEX-M dataset - first row represents Target Triple (TT). The lower the score, the closer is example to the Target Triple.

S	P	O	Score
02314321	_hypernym	08102555	TT
02314001	_hypernym	08102555	0.02788
02313495	_hypernym	08102555	0.04839
01928360	_hypernym	08102555	0.05155
02314717	_hypernym	08102555	0.06077
01928737	_hypernym	08102555	0.06294
02321759	_hypernym	02316038	0.09046

Table 9
Example explanation for a test triple in WN18RR dataset - first row represents Target Triple (TT). The lower the score, the closer is example to the Target Triple. Example reported in raw identifier form. WN18RR exposes WordNet synset offsets without rdfs:label annotations; the same method produces directly readable output on FB15k-237 (Table D) and CoDEx-M (Table D). The contrast illustrates the readability gap addressed by XAI-Fb15k-237 (Section 4).