Machine Learning (2022) 111:1799–1820
https://doi.org/10.1007/s10994-021-06048-w

Generating contrastive explanations for inductive logic programming based on a near miss approach

Johannes Rabold[1] · **Michael Siebers**[1] · **Ute Schmid**[1]

Received: 23 April 2020 / Revised: 27 July 2021 / Accepted: 27 August 2021 /
Published online: 24 September 2021

Abstract
In recent research, human-understandable explanations of machine learning models have received a lot of attention. Often explanations are given in form of model simplifications or visualizations. However, as shown in cognitive science as well as in early AI research, concept understanding can also be improved by the alignment of a given instance for a concept with a similar counterexample. Contrasting a given instance with a structurally similar example which does not belong to the concept highlights what characteristics are necessary for concept membership. Such near misses have been proposed by Winston (Learning structural descriptions from examples, 1970) as efficient guidance for learning in relational domains. We introduce an explanation generation algorithm for relational concepts learned with Inductive Logic Programming (GENME). The algorithm identifies near miss examples from a given set of instances and ranks these examples by their degree of closeness to a specific positive instance. A modified rule which covers the near miss but not the original instance is given as an explanation. We illustrate GENME with the well-known family domain consisting of kinship relations, the visual relational Winston arches domain, and a real-world domain dealing with file management. We also present a psychological experiment comparing human preferences of rule-based, example-based, and near miss explanations in the family and the arches domains.

Keywords Explainable AI · Relational concepts · Contrastive explanations · Inductive logic programming · Near miss examples

Editors: Nikos Katzouris, Alexander Artikis, Luc De Raedt, Artur d'Avila Garcez, Sebastijan Dumančić, Jay Pujara.

✉ Ute Schmid
ute.schmid@uni-bamberg.de

Johannes Rabold
johannes.rabold@uni-bamberg.de

Michael Siebers
michael.siebers@uni-bamberg.de

[1] Cognitive Systems, University of Bamberg, Bamberg, Germany

1 Introduction

Explaining classifier decisions has gained much attention in current research. If explanations are intended for the end-user, their main function is to make the human comprehend how the system reached a decision (Miller 2019). In the last years, a variety of approaches to explainability has been proposed (Adadi and Berrada 2018; Molnar 2019): Explanations can be local—focusing on the current class decision—or global—covering the learned model (Ribeiro et al. 2016; Adadi and Berrada 2018). A major branch of research addresses explanations by visualizations for end-to-end image classification (Samek et al. 2017; Ribeiro et al. 2016). Alternatively, explanations can be in form of symbolic rules (Lakkaraju et al. 2016; Muggleton et al. 2018) or in natural language (Stickel 1991; Ehsan et al. 2018; Siebers and Schmid 2019). A further approach to explanations is to offer prototypical examples to illustrate a model (Bien and Tibshirani 2011; Gurumoorthy et al. 2019). Finally, counterexamples can be used as counterfactuals or contrastive explanations. Counterfactuals typically are minimal changes in feature values which would have resulted in a different decision, such as *You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan.* (Wachter et al. 2017). In philosophy, counterfactuals have been characterized by the concept of a 'closest possible world', that is, the smallest change required to obtain a different (and more desirable) outcome (Pollock 1976). Contrastive explanations have been proposed mainly for image classification. For instance, the contrastive explanation method CEM (Dhurandhar et al. 2018) highlights what is minimally but critically absent in an image to belong to a given class. The MMD-critic (Kim et al. 2016) can identify nearest prototypes and nearest miss instances in image data such as handwritten digits and Imagenet datasets. Furthermore, an algorithm ProtoDash has been proposed to identify prototypes and criticisms for arbitrary symmetric positive definite kernels which has been applied to both tabular as well as image data.

An approach related to counterexamples has been proposed in early AI research by Winston in the context of learning relational concepts such as arch (Winston 1970). He demonstrated that presenting near miss examples where only a small number of relational aspects is missing to make an object a member of a class results in faster learning. Similarly, in cognitive science research, it has been shown that alignment of structured representations helps humans to understand and explain concepts (Gentner and Markman 1994). Gentner and Markman found that it is easier for humans to find differences between pairs of similar items than between pairs of dissimilar items. For example, it is easier to explain the concept of a light bulb by contrasting it with a candle than with a cat (Gentner and Markman 1994).

Induction of relational concepts has been investigated in Inductive Logic Programming (ILP) (Muggleton and De Raedt 1994), statistical relational learning (Koller et al. 2007), and recently also in the context of deep learning with approaches such as RelNN (Kazemi and Poole 2018) and Differentiable Neural Computers (DNCs) (Graves et al. 2016). DNCs have been demonstrated to be able to learn symbolic relational concepts such as family relations or travel routes in the London underground system. These domains are typical examples for domains where ILP approaches have been demonstrated to be highly successful (Muggleton et al. 2018). For DNCs, questions and answers are represented as Prolog clauses. However, in contrast to ILP, the learned models are black-box. For the family domain as well as for an isomorphic fictitious chemistry domain it has been shown, that rules learned with ILP fulfill Donald Michie's

criterion of ultra-strong machine learning (Muggleton et al. 2018). Ultra-strong machine learning according to Michie requires a machine learning approach to teach the learned model to a human, whose performance is consequently increased to a level beyond that of the human studying the training data alone. This characteristics has been related to the comprehensibility of learned rules or explanations (Muggleton et al. 2018): Comprehensibility has been defined such that a human who is presented with this information is able to classify new instances of the given domain correctly.

For ILP as well as for other relational learners such as DNCs, verbal explanations can be helpful to make a system decision transparent and comprehensible. For example, it can be explained why *grandfather(ian,kate)* holds by presenting the relations on the path from *ian* to *kate* in the family tree given in Fig. 1: *Ian is a grandfather of Kate because Ian is male and Ian is the father of Tom and Tom is the father of Kate.* Alternatively, it might be helpful for understanding the concept *grandfather* to present a contrastive example in form of a near miss explanation. For instance, *Jodie is NOT the grandfather of Kate because she is NOT male* or *Mat is NOT the grandfather of Ian because he is in a child-of-child relation to Ian and NOT in a parent-of-parent relation.* The first near miss corresponds to the concept of a *grandmother*, emphasizing the importance of the attribute *male* for *grandfather*. The second near miss corresponds to the concept of a *grandson*, emphasizing the importance of the relation *parent*. To our knowledge, generating such near miss examples to explain learned *relational* concepts has not been investigated yet—neither in the context of ILP nor for other machine learning approaches.

In the following, we discuss the function of near miss examples. Afterwards, we present an algorithmic approach to generate near miss examples in the context of ILP and demonstrate the approach for a generic family domain, a visual domain and a real world domain dealing with file management (Siebers and Schmid 2019). Finally, we present an empirical evaluation with human participants where we compare human preferences of different types of explanations for the family and the arch domain, namely rule-based global explanations, example-based explanations, as well as near miss and far miss contrastive explanations.

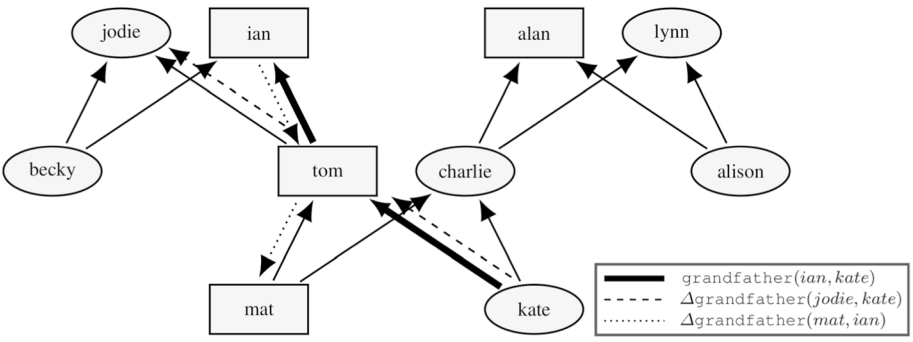

Fig. 1 An example family tree. Rectangles denote male persons, ellipses denote female persons, and solid arrows denote the parent relation. The bold solid arrows indicate a trace for a positive example. Non-solid arrows indicate near miss explanations

2 The function of near miss examples

Near miss examples have been introduced by Winston as a human-like strategy to machine learning (Winston 1970): A near miss example for a concept is an example which does not belong to the concept but has a strong overlap to positive examples. Such near miss examples are helpful to guide the model construction of a machine learning algorithm (Telle et al. 2019). Winston illustrated learning with near misses in the context of relational visual domains. Concepts are represented as compounds of primitive blocks such as cubes. For instance, positive examples for arches must consist of at least two objects playing the role of supporters (pillars) and another object on top of them (roof). Negative examples for an arch might be a tower of several cubes—a far miss—or two pillars with no space between them covered by a roof—a near miss.

In the context of machine learning of relational concepts such as the Winston arches, molecules (King et al. 1996), or Turing-complete languages (Telle et al. 2019), carefully constructed near misses as negative examples can speed up learning (see Fig. 2a). In this case, the machine learning expert has a similar role to that of a school teacher who identifies helpful examples (Schmid et al. 2003). We propose that what is effective for learning is also effective for explaining a learned model (see Fig. 2b): for some concept that an AI system has learned, it can explain its model by constructing a near miss example. While machine learning typically is unidirectional—the human provides training examples and the system generalizes a model—explanations can support interactive machine learning scenarios based on a bidirectional partnership between human and AI system (Nguyen et al. 2018).

While there are some considerations about what constitutes helpful examples in educational psychology (Gentner et al. 2003) and the insights given in Winston's seminar work (Winston 1970), there exist no general principles to construct helpful near miss explanations. We base our algorithm presented in the next section on some general observations which we will illustrate with the family domain example of Fig. 1.

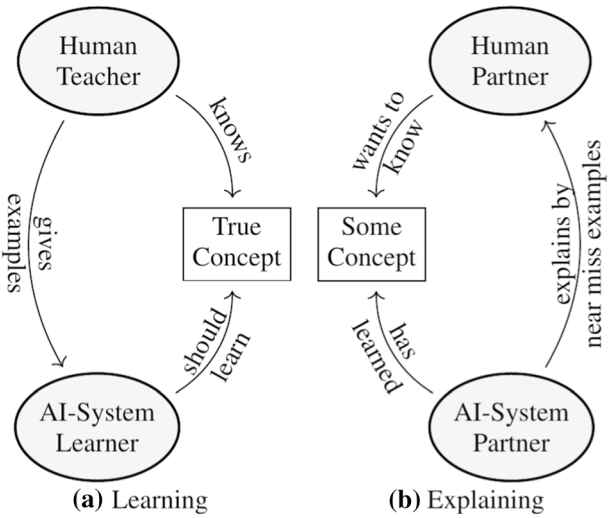

Fig. 2 Duality of learning and explaining

3 Near miss explanations

In the following, we will introduce the GENME algorithm for generating near miss explanations. Our approach complements the interpretable machine learning approach ILP (Muggleton et al. 2018) with a contrastive explanation component. First, we will introduce notation and basic concepts of ILP. Then, the concept of a near miss explanation is introduced formally and the generation algorithm is presented.

3.1 Notation

ILP algorithms learn models for subsets of the logic programming language Prolog (Sterling and Shapiro 1994). In the following, we consider function-free Horn clausal theories.

A *term* is either a variable or a constant symbol. A *predicate symbol* followed by a bracketed n-tuple of terms is called *atom*, or *literal*. The function $\text{sym}(A)$ returns the predicate symbol of atom A. The number of arguments a predicate symbol takes is its *arity*. A predicate symbol is called *attribute* if it has arity one and *relation* if its arity is larger than one. A literal is called ground when it has only constants as arguments. By convention, constants and predicate symbols are represented by lowercase strings and variables by uppercase strings.

A *clause* is an implication where the antecedent is a finite set of literals and the consequent is an atom. We write the implication reversed, as $H \Longleftarrow \{L_1, \ldots, L_m\}$. The consequent of a clause H is called its *head* and its antecedent is called the *body*. We define $\text{head}(C) = H$ and $\text{body}(C) = \{L_1, \ldots, L_m\}$. For convenience, braces surrounding the body can be omitted. If the body of clause C is the empty set, $H \Longleftarrow \{\}$, we call C a *fact*, omit the antecedents, and simply write H. Clauses with non-empty body can also be called *rules*. A clause is called ground when all its literals are ground. A finite set of clauses is called a *finite clausal theory* T, or simply a *theory*. The set of constant symbols occurring in T is $\mathscr{C}_T$. If it is clear from the context, we will omit the index.

The function $\text{vars}(L)$ returns the set of variables occurring in literal L. This is extended to sets of literals, $\text{vars}(\{L_1, L_2, \ldots, L_n\}) = \bigcup_{i=1}^{n} \text{vars}(L_i)$, and clauses, $\text{vars}(C) = \text{vars}(\text{head}(C)) \cup \text{vars}(\text{body}(C))$. A *substitution* is a mapping from variables to terms. We denote a substitution θ by $\{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ where $x_1, \ldots, x_k$ are variables and $t_1, \ldots, t_k$ are terms. A substitution is applied to a term by simultaneously replacing all x_i in the term by the corresponding t_i's. A substitution is applied to a literal by applying it to all terms in the literal. A substitution is applied to a clause by applying it to all literals in the clause. We denote the application of the substitution θ to a term, literal, or clause X by $X\theta$. A substitution θ is called *minimal* w.r.t. clause C, if $C\theta$ is ground and there exists no θ' such that $C\theta'$ is ground and $|\sigma| < |\theta|$.

If a literal or a set of literals K is true given a clausal theory T, we say that T models K and write $T \models K$. Theory T models an atom A if there exists a clause $C \in T$ and a substitution θ such that $A = \text{head}(C\theta)$ and $T \models \text{body}(C\theta)$. A theory T models a set of literals $\{L_1, \ldots, L_n\}$ if there is a substitution θ such that T models $L_i\theta$ for $1 \leq i \leq n$. By definition, the empty set $\{\}$ is modeled by any theory.

3.2 Basic concepts of ILP

ILP is a sub-field of symbolic machine learning which deals with learning clausal theories from examples (Muggleton and De Raedt 1994). Such clausal theories allow to represent

relational concepts where the target is a relation or an attribute defined over relational structures. For instance, clauses for the binary relation *grandfather* can express family relations between persons or the attribute *arch* can express whether a construction is an arch. ILP learns such clauses from positive examples, like *grandfather(ian,kate)*, and negative examples, such as *grandfather(alan,tom)* (see Fig. 1). Positive and negative examples for the target concept are represented as ground atoms. Additionally, a background theory must be provided. In the family domain, the facts *parent(tom,kate)* and *male(ian)* can be part of the background theory (Fig. 3). If a background theory consists only of facts, it is often called background knowledge. The learned theory together with the background theory must model all positive examples and no negative example.

Assume that the learned theory for *grandfather* consists of a single clause:

$$\texttt{grandfather}(A, B) \Longleftarrow \texttt{male}(A), \texttt{parent}(A, C), \texttt{parent}(C, B). \tag{1}$$

In general, a learned theory can include several clauses characterizing the target concept. For example, the target *grandparent* can be described by a set of clauses taking into account the genders of the respective parents. It can also be the case that target clauses are not exclusive. That is, a positive example P might follow from multiple clauses. With the learned theory, new instances given as ground atoms can be classified. For example, *grandfather(alan,kate)* will be classified as positive; *grandfather(becky,tom)* will be classified as negative.

3.3 Near miss examples and explanations

Positive classified instances are modeled by the learned theory as introduced in Sect. 3.1. As mentioned above, theory T consists of predefined background clauses and clauses learned for the target concept. For example, the *grandfather* relation holds for *ian* and *kate* given the theory in Equation 1 together with background knowledge about parent relations and gender of persons in a given family domain as the one in Fig. 3. An explanation for this fact has to make explicit how this can be derived from the theory. The reason why

Background Knowledge:

```
female(jodie)      parent(jodie, becky)
female(lynn)       parent(jodie, tom)
female(becky)      parent(ian, becky)
female(charlie)    parent(ian, tom)
female(alison)     parent(alan, charlie)
female(kate)       parent(alan, alison)
male(ian)          parent(lynn, charlie)
male(alan)         parent(lynn, alison)
male(alan)         parent(tom, mat)
male(tom)          parent(tom, kate)
male(mat)          parent(charlie, mat)
                   parent(charlie, kate)
```

Selection of positive examples:	**Selection of negative examples:**
`grandfather(ian, kate)`	`grandfather(alan, tom)`
`grandfather(alan, mat)`	`grandfather(jodie, mat)`

Fig. 3 Background knowledge for the family domain together with a selection of positive and negative examples for the *grandfather* concept

grandfather(ian,kate) holds is that *ian* is male and *ian* is a parent of *tom* and *tom* is a parent of *kate*. In general, we call an explanation for a positive example *local explanation*:

Definition 1 (*Local Explanation*) A local explanation for a positive example P is a ground clause $C\theta$ where $C \in T$ such that $P = \text{head}(C\theta)$ and $T \models \text{body}(C\theta)$.

To emphasize which information is crucial for making someone a grandfather of someone else, a near miss explanation might be helpful. As introduced in Sect. 2, a near miss example is not a positive instance for the target concept, but illustrates a semantically similar concept. For instance, given the positive example *grandfather(ian,kate)*, possible near miss examples could be the female parent of a parent (that is the grandmother) of *kate* or a male child of a child (that is a grandson) of *ian*. Formally, we define near miss explanations and near miss examples as follows:

Definition 2 (*Near Miss Explanation*) Let $C\theta$ be a local explanation, C' a minimally changed clause, and θ' a minimal substitution. We call $C'\theta'$ a near miss explanation if $T \models \text{body}(C'\theta')$ and $T \not\models \text{head}(C'\theta')$. Using the operator Δ to mark near miss examples, ΔL is a near miss example if $L = \text{head}(C'\theta')$.

What constitutes a minimally changed clause is domain dependent. In general, the most basic change is replacing one literal in the body by its negation. For example, the attribute *male(alan)* could be changed to ¬*male(alan)*; an attribute *large(x)* to ¬*large(x)*; a relation *above(top,bottom)* to ¬*above(top,bottom)*. However, negations are too unspecific for many domains and are not part of our definition of theories. Therefore, we propose to explicitly define *semantically opposing* predicate symbols when modeling a particular domain. In natural language semantics, such relational opposites are one of the basic relations between lexical units (Palmer 1981). In the family domain, pairs *male* ↔ *female* and *parent* ↔ *child* are semantic opponents. To explain *grandfather(ian,kate)*, the near miss example *Δgrandfather(jodie,kate)* (which is actually the grandmother) can be derived by replacing *male(A)* with *female(A)* in Equation 1. An alternative near miss can be constructed by inverting the *parent* relation to *child*. Because *grandfather* relies on the transitive sequence of two *parent* relations, both occurrences should be replaced, resulting in *Δgrandfather(mat,ian)* (which should actually read *grandson(mat,ian)*).

Depending on the domain, a minimal change of a clause C might therefore consist of either replacing a single literal or multiple literals. Which literals are to be replaced may also depend on the semantic opponents involved. Thus, we introduce domain dependent functions called rewriting filters $\mathscr{V}_{p \mapsto q}$ to formalize this connection. $\mathscr{V}_{p \mapsto q}(C)$ selects subsets of the literals in body(C) such that replacing the predicate symbol p with the predicate symbol q in the selected literals in the body of C constitutes a minimal change to C.

Consider the clause D, *grandfather(X,Y)* ⟸ *male(X)*, *parent(X,Z)*, *parent(Z,Y)*, *male(Z)*. This is an overly specific definition of *grandfather* where not only the person denoted as grandfather (e.g., the constant *ian* bound to variable *X*) is constrained to male, but also the intermediate parent (e.g., the constant *tom* bound to variable *Z*). Given the semantics of the family domain, rewriting of *male* by its semantic opponent *female* is only necessary for *male(A)*. Applying $\mathscr{V}_{male \mapsto female}$ to D thus should yield $\{\{male(X)\}\}$. Likewise, for the transitive relation *parent* it makes sense to constrain replacement by its semantic opponent such that *parent* is replaced by *child* if and only if the occurring variables in these literals form a chain. Applying $\mathscr{V}_{parent \mapsto child}$ to D thus should yield $\big\{\{parent(X,Z), parent(Z,Y)\}\big\}$.

The specific constraints depend on the given domain. Therefore, we assume that the person modelling the domain provides this information explicitly by the rewriting filters. There may be more than one applicable rewriting filter and each filter may yield more than one subset of body literals to change. Changing the literals from exactly one subset constitutes a minimal change. In general, there may be multiple minimally changed clauses C' for a given clause C. Applying the substitution θ of a local explanation $C\theta$ to a minimally changed clause C' does not result in a near miss explanation. Since the head of a clause is not changed, $\text{head}(C'\theta') = \text{head}(C\theta)$, the head of the near miss explanation would be a positive instance. The substitution θ' for a near miss explanation may share more or less elements with θ. For a fixed C' different substitutions θ' constitute near miss explanations with different degrees of similarity to the local explanation $C\theta$ for the positive instance P:

Definition 3 (*Degree of Near Miss Explanation*) Given a near miss explanation $C'\theta'$ w.r.t. local explanation $C\theta$, the degree d of the near miss explanation is the number of changed replacements, $d = |\theta \setminus \theta'|$.

3.4 Algorithm

The GENME algorithm (Algorithm 1) identifies near miss explanations for a positive example. Given a theory T, a finite set of rewriting filters O, and a positive example P, it returns a family of sets, $(\mathscr{E}_d)_{d\in\mathbb{N}}$, where each $\mathscr{E}_d$ contains near miss explanations of degree d, inducing a partial ordering over contrastive explanations in relation to instance P. GENME follows a generate-and-test approach. First, it generates the set of all *near miss candidates*.

Definition 4 (*Near Miss Candidate*) A *near miss candidate* for a positive example P is a ground atom N which has the same predicate symbol as P, $\text{sym}(N) = \text{sym}(P)$, but is not modeled by the theory T, $T \nvDash N$.

Algorithm 1 GENME: The Near Miss Explanation Generation Algorithm

Require: Theory T, Finite set of rewriting filters O, Positive Example P
1: Initialize family of result sets $(\mathscr{E}_i)_{i\in\mathbb{N}}$ to empty sets
2: Initialize the set of all near miss candidates $\mathscr{N}$
3: **for all** local explanations $C\theta$ for P where $C \in T$ **do**
4: **for all** $\mathscr{V}_{\mathrm{p}\mapsto\mathrm{q}} \in O$ **do**
5: **for all** $\mathscr{L} \in \mathscr{V}_{\mathrm{p}\mapsto\mathrm{q}}(C)$ **do**
6: create C' from C by replacing p with q in every literal in the body which is in $\mathscr{L}$
7: **for all** $N \in \mathscr{N}$ **do**
8: $d \leftarrow 0$
9: $\mathscr{E} \leftarrow \{\}$
10: **while** $\mathscr{E} = \{\}$ and $d < |\theta|$ **do**
11: $d \leftarrow d + 1$
12: **for all** partitions of θ into θ_1 and θ_2 such that $|\theta_2| = d$ **do**
13: **for all** $\theta_2' = \{x_i \mapsto t_i' \mid x_i \mapsto t_i \in \theta_2\}$ such that no $t_i' = t_i$ **do**
14: $E \leftarrow C'(\theta_1 \cup \theta_2')$
15: $\mathscr{E} \leftarrow \mathscr{E} \cup \{E \mid T \models \mathrm{body}(E) \text{ and } \mathrm{head}(E) = N\}$
16: **end for**
17: **end for**
18: **end while**
19: $\mathscr{E}_d \leftarrow \mathscr{E}_d \cup \mathscr{E}$
20: **end for**
21: **end for**
22: **end for**
23: **end for**
24: **return** $(\mathscr{E}_i)_{i\in\mathbb{N}}$

As second step, GENME checks for all near miss candidates whether there is a fitting near miss explanation, iterating over all local explanations in the process. For each local explanation, GENME iterates over all rewriting filters (line 4) and all selected subsets of body literals to generate a minimally changed clause C'. For each such minimally changed clause, GENME iterates over all near miss candidates (line 7) and all possible substitutions θ' with constants in $\mathscr{C}$ in increasing degree from θ (lines 10–18). If there are substitutions θ' such that $\mathrm{head}(C'\theta')$ equals the near miss candidate and the theory models $\mathrm{body}(C'\theta')$ for a given degree (line 15), then all near miss explanations for this degree are added to $\mathscr{E}_d$ (lines 15 and 19) and GENME continues with the next candidate.

3.5 Complexity, termination and correctness

The core building blocks of GENME are *consequence tests*, that is, whether some set of literals $\mathbf{L}$ is modeled by the theory T. Thus, we will assess the complexity of our algorithm by the number of *consequence tests* ($T \models \mathbf{L}$) required. Let the p-theory T_p be the subset of theory T that contains all clauses of T whose head has the predicate symbol p, $T_p = \{C \in T \mid \mathrm{head}(C) = p\}$. For the runtime complexity of GENME the following theorem holds:

Theorem 1 (Complexity) *Given a theory T, set of rewriting filters O, and positive example $P = p(t_1, \ldots, t_a)$, the runtime complexity of GENME is polynomial in the size of the p-theory $|T_p|$, the number of rewriting filters $|O|$, and the number of constants in the theory $|\mathscr{C}|$ and exponential in the arity a of p, the maximal number of variables in any clause in the*

p-theory v_{max}, and the maximal number of literals in the body of any clause in the p-theory l_{max}.

Proof In its first step, GENME creates the set of all near miss candidates $\mathcal{N}$. Therefore, it must iterate over all clauses C in the p-theory and all minimal substitutions σ for C. For each combination of C and σ, one consequence test must be performed (and fail). Since σ is a minimal substitution, $|\sigma| = \text{vars}(C)$. Thus, creating $\mathcal{N}$ has a complexity of $\mathcal{O}(|\mathscr{C}|^{v_{max}}|T_p|)$ where v_{max} is the maximal number of variables in any clause in T_p.

For the second step, we consider the algorithm loops from inside out. For fixed C', θ_1, θ_2, and N, the inner-most loop (lines 13–16) iterates over all possible altered substitutions θ_2' and tests whether $C'(\theta_1 \cup \theta_2')$ is a near miss explanation. Since every term in θ_2' must be different than the corresponding one from θ_2, there are $(|\mathscr{C}| - 1)^d$ possible θ_2''s. Thus, the inner-most loop has a complexity of $\mathcal{O}(|\mathscr{C}|^d)$. This loop is repeated for every possible partitioning such that θ_2 has cardinality d (line 12) where d increases up to $|\theta|$ unless some near miss explanation is found (line 10). Thus, the complexity of the complete while loop (lines 8–19) is $\sum_{d=1}^{|\theta|} \binom{|\theta|}{d} (|\mathscr{C}| - 1)^d = \mathcal{O}(2^{|\theta|}|\mathscr{C}|^{|\theta|})$.

The while loop is repeated for every near miss candidate. Since every near miss candidate must have the same predicate symbol as P which has arity a, there may be up to $|\mathscr{C}|^a - 1$ near miss candidates. For any given clause C, the current selected rewriting filter $\mathscr{V}_{p \mapsto q}$ (line 4) may select an arbitrary subset of C's body literals except the empty set (this would imply $C = C'$). Consequently, there are up to $2^{|\text{body}(C)|} - 1$ literal sets $\mathscr{L}$ (line 5). The complexity of finding near miss explanations for given local explanation $C\theta$ is $\mathcal{O}(|O||\mathscr{C}|^{|\theta|+a}2^{|\text{body}(C)|+|\theta|})$.

Similar to near miss candidates, local explanations are constructed from some clause C in the p-theory and a minimal substitution θ for C. There are up to $|T_p||\mathscr{C}|^{v_{max}}$ local explanations. Consequently, the complexity of the second part of the algorithm (lines 3–23) is $\mathcal{O}(|T_p||O||\mathscr{C}|^{2v_{max}+a}2^{l_{max}+v_{max}})$ where l_{max} is the maximal number of literals in the body of any clause in T_p. Finally, since the complexity of the first step and the complexity to generate all local explanations is lower than the complexity of the second part, the complexity of the algorithm is identical to the complexity of the second part. □

Theorem 2 *GENME will perform a finite number of consequence tests.*

Proof Assume, some run of GENME will perform exactly $f(T, O, P)$ consequence tests. As shown above, $f(T, O, P) \in \mathcal{O}(|T_p||O||\mathscr{C}|^{2v_{max}+a}2^{l_{max}+v_{max}})$. Since T is a finite clausal theory and T_p is a subset of T, T_p is finite. Since every clause in T is finite, $\mathscr{C}$, v_{max}, and l_{max} are finite. The set of rewriting filters O is finite. Consequently, $f(T, O, P)$ is finite. □

Theorem 3 (Termination) *GENME will terminate in finite time.*

Proof As stated in Sect. 3.1, our definition of clausal theories is identical to function-free Horn clauses. Function-free Horn clausal theories are decidable (Tamaki and Sato 1986). Thus, consequence tests can be evaluated in finite time. The theorem directly follows from theorem 2 and this fact. □

Theorem 4 (Correctness) *GENME returns a family of sets of local explanations $(\mathscr{E}_d)_{d \in \mathbb{N}}$. Each element of any set $\mathscr{E}_d$ is a near miss explanation w.r.t a local explanations as given in Definition 2.*

Proof Each $E \in \mathscr{E}$ with $E = C'(\theta_1 \cup \theta_2')$ (line 14) is a near miss explanation if and only if (i) C' is a minimally changed clause, (ii) $\theta_1 \cup \theta_2'$ is a minimal substitution, (iii) $T \models \text{body}(E)$, and (iv) $T \not\models \text{head}(E)$.

(i) holds because C' is constructed from C by applying a single rewriting filter as defined in Sect. 3.3 (line 6). (ii) holds since θ_1, θ_2 is a partition of the minimal substitution θ, θ_2' replaces variables only with constants and has the same number of elements as θ_2 (line 13). (iii) is explicitly tested in line 15. (iv) follows from $\text{head}(E) = N$ which is explicitly tested (line 15). As N is a near miss candidate (lines 2 and 7), by Definition 4, $T \not\models N$ holds. □

4 Application to example domains

We realized the GENME algorithm in Prolog. In the following, we demonstrate the generation of near miss explanations applying GENME for the family domain, a relational visual domain of blocksworld arches, and a real world domain dealing with file management.

4.1 Family domain

In Fig. 3 the family domain has been introduced with the background clauses *male*, *female*, and *parent* for a small number of constants (first names). In addition, *child* has been introduced as semantic opponent in Sect. 3.3. As an example for a target concept *grandfather* has been defined in Equation 1. We apply GENME to this target concept and additionally to the concept of a *daughter*:

$$\texttt{daughter}(A, B) \Longleftarrow \texttt{female}(A), \texttt{child}(A, B). \tag{2}$$

For both cases, the same family tree is used (see Fig. 1). The following rewriting filters, as introduced in Sect. 3.3, are provided: $\mathscr{V}_{male \mapsto female}$, $\mathscr{V}_{female \mapsto male}$, $\mathscr{V}_{parent \mapsto child}$, and $\mathscr{V}_{child \mapsto parent}$. GENME is applied to explain the positive examples $P_1 =$ *grandfather(ian,kate)* and $P_2 =$ *daughter(becky,jodie)*.

For *grandfather*, there are four positive examples in the given domain (see Fig. 1) and given the 10 persons additionally 96 pairs for which the *grandfather* relation does not hold. Out of these 96 near miss candidates for P_1, GENME identifies 8 as near miss examples (8.3 %, see Table 1). The near miss explanation with the lowest degree of 1 is *grandfather(jodie,kate)* $\Longleftarrow$ *female(jodie), parent(jodie,tom), parent(tom,kate)*. Indeed, this is intuitively a very close near miss (the grandmother of *kate*). For the given family tree there are 8 pairs of persons for which the *daughter* relation holds. GENME identifies 10 out of 92 candidates for P_2 as near miss examples (10.9 %). For P_2, there is a single near miss explanation with the lowest degree of one, namely *jodie's* son (*daughter(tom,jodie)* $\Longleftarrow$ *male(tom), child(tom,jodie)*).

Both minimal near miss explanations were constructed by exchanging the unary predicates *male* resp. *female* by their semantic opponent. Rewriting *parent* or *child* also results in plausible near miss explanations, but with degree 2: For P_1, *mat* is the grandson of *ian* and for P_2, *jodie* is the mother of *becky*.

Table 1 Number of found near miss explanations by degree in the family domain

	gf(ian,kate) $\lvert\mathscr{N}\rvert = 96$	*dt(becky,jodie)* $\lvert\mathscr{N}\rvert = 92$
male $\leftrightarrow$ *female*		
$\lvert\mathscr{E}_1\rvert$	1	1
$\lvert\mathscr{E}_2\rvert$	2	3
$\lvert\mathscr{E}_3\rvert$	1	0
parent $\leftrightarrow$ *child*		
$\lvert\mathscr{E}_1\rvert$	0	0
$\lvert\mathscr{E}_2\rvert$	2	6
$\lvert\mathscr{E}_3\rvert$	2	0

$\mathscr{N}$ denotes the set of all near miss candidates, $\mathscr{E}_d$ the set of near miss explanations of degree d, *gf* the *grandfather* relation, and *dt* the *daughter* relation. $x \leftrightarrow y$ denotes the use of $\mathscr{V}_{x\mapsto y}$ or $\mathscr{V}_{y\mapsto x}$, respectively

4.2 Winston arches domain

Winston introduced the concept of an *arch* in the context of a blocksworld domain by characteristic relations between blocks: *contains* (a structure contains a block), *supports* (a block supports another block), *(not_)meets* (two blocks do (not) meet horizontally), and *is_a* (a block has a certain shape; the shape can either be *wedge* or *brick*). The target concept of *arch* is defined as:

$$\begin{aligned}\texttt{arch}(A) \Longleftarrow\ & \texttt{contains}(A,X), \texttt{contains}(A,Y), \texttt{contains}(A,Z),\\ & \texttt{is\_a}(X,T), \texttt{is\_a}(Y,\texttt{brick}), \texttt{is\_a}(Z,\texttt{brick}),\\ & \texttt{supports}(Y,X,A), \texttt{supports}(Z,X,A),\\ & \texttt{not\_meets}(Y,Z,A).\end{aligned} \tag{3}$$

Figure 4 shows some positive and negative examples for a restricted concept of an *arch* where all positive examples have a block of type *wedge* as the top. In Fig. 5, the background

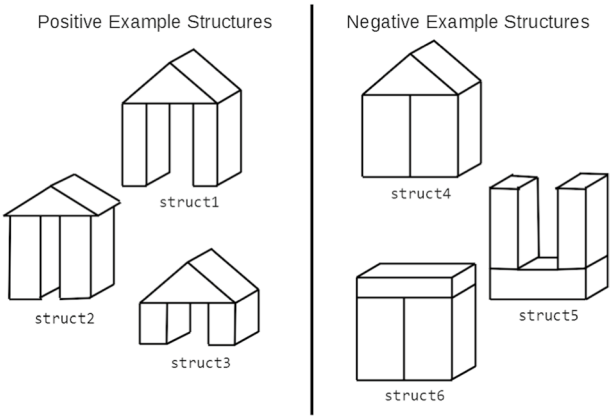

Fig. 4 Some positive and negative example structures for the Winston arches domain

Background Knowledge:

```
contains(struct1, a1)      contains(struct4, a1)
contains(struct1, b)       contains(struct4, b)
contains(struct1, c)       contains(struct4, c)
supports(b, a1, struct1)   supports(b, a1, struct4)
supports(c, a1, struct1)   supports(c, a1, struct4)
not_meets(b, c, struct1)   meets(b, c, struct4)
contains(struct2, a1)      contains(struct5, a2)
contains(struct2, b)       contains(struct5, b)
contains(struct2, c)       contains(struct5, c)
supports(b, a1, struct2)   supports(a2, b, struct5)
supports(c, a1, struct2)   supports(a2, c, struct5)
not_meets(b, c, struct2)   not_meets(b, c, struct5)
contains(struct3, a1)      contains(struct6, a2)
contains(struct3, b)       contains(struct6, b)
contains(struct3, c)       contains(struct6, c)
supports(b, a1, struct3)   supports(b, a2, struct6)
supports(c, a1, struct3)   supports(c, a2, struct6)
not_meets(b, c, struct3)   meets(b, c, struct6)
is_a(a1, wedge)            is_a(a2, brick)
is_a(b, brick)             is_a(c, brick)
```

Background Theory:

```
supported_by(X, Y, A)  ⟸ supports(Y, X, A)
```

Fig. 5 Background knowledge and background theory for the Winston arches domain

knowledge for these examples is given. In addition, *supported_by* is defined as the inverse of *supports*.

With the positive example *arch(struct1)* and the rewriting filters $\mathscr{V}_{supports \mapsto supported\_by}$, $\mathscr{V}_{supported\_by \mapsto supports}$, $\mathscr{V}_{meets \mapsto not\_meets}$, and $\mathscr{V}_{not\_meets \mapsto meets}$ GENME yields the following near miss explanation with degree $d = 1$:

$$\begin{aligned} \texttt{arch(struct4)} \Longleftarrow\ & \texttt{contains(struct4, a1)}, \texttt{contains(struct4, b)}, \\ & \texttt{contains(struct4, c)}, \texttt{is\_a(a1, wedge)}, \texttt{is\_a(b, brick)}, \\ & \texttt{is\_a(c, brick)}, \texttt{supports(b, a1, struct4)}, \\ & \texttt{supports(c, a1, struct4)}, \texttt{meets(b, c, struct4)} \end{aligned}$$

Δarch(struct4) is a plausible near miss example since the only change is that the two supporting pillars meet. Further explanations are found for $d = 3$:

$$\begin{aligned} \texttt{arch(struct6)} \Longleftarrow\ & \texttt{contains(struct6, a2)}, \texttt{contains(struct6, b)}, \\ & \texttt{contains(struct6, c)}, \texttt{is\_a(a2, brick)}, \texttt{is\_a(b, brick)}, \\ & \texttt{is\_a(c, brick)}, \texttt{supports(b, a2, struct6)}, \\ & \texttt{supports(c, a2, struct6)}, \texttt{meets(b, c, struct6)} \end{aligned}$$

$$\begin{aligned} \texttt{arch(struct5)} \Longleftarrow\ & \texttt{contains(struct5, a2)}, \texttt{contains(struct5, b)}, \\ & \texttt{contains(struct5, c)}, \texttt{is\_a(a2, brick)}, \texttt{is\_a(b, brick)}, \\ & \texttt{is\_a(c, brick)}, \texttt{supported\_by(b, a2, struct5)}, \\ & \texttt{supported\_by(c, a2, struct5)}, \texttt{not\_meets(b, c, struct5)}. \end{aligned}$$

Table 2 Number of found near miss explanations by degree in the Winston arches domain

	arch(struct1) $\lvert\mathcal{M}\rvert = 3$
meets ↔ *not_meets*	
$\lvert\mathscr{E}_1\rvert$	1
$\lvert\mathscr{E}_2\rvert$	0
$\lvert\mathscr{E}_3\rvert$	1
supports ↔ *supported_by*	
$\lvert\mathscr{E}_1\rvert$	0
$\lvert\mathscr{E}_2\rvert$	0
$\lvert\mathscr{E}_3\rvert$	1

$\mathscr{N}$ denotes the set of all near miss candidates, $\mathscr{E}_d$ the set of near miss explanations of degree d. $x \leftrightarrow y$ denotes the use of $\mathscr{V}_{x\mapsto y}$ or $\mathscr{V}_{y\mapsto x}$, respectively

Background Knowledge:

```
file(file10)                                  file(file11)
file_name('1fTmw4WN.PNG', file10)             file_name('Sv4Xy5n6.PNG', file11)
media_type(png, file10)                       media_type(png, file11)
file_size(6902, file10)                       file_size(12287, file11)
creation_time('1984-12-18', file10)           creation_time('1996-12-20', file11)
```

Selection of positive examples:

```
irrelevant(file10)
irrelevant(file11)
```

Selection of negative examples:

```
irrelevant(file121)
irrelevant(file168)
```

Fig. 6 Excerpt of background knowledge for the file management domain together with a selection of positive and negative examples for the *irrelevant* concept

These two explanations have a larger distance to the to be explained *struct1* than *Δarch(struct4)*. That is, they are "not that near misses" which we will call far misses when we want to discriminate between misses with lower and higher degrees. Table 2 shows the number of near misses GENME found for the different rewriting filters.

4.3 File management domain

In the file management domain, we aim at identifying irrelevant files which could be deleted by the user (Siebers and Schmid 2019). The domain is represented by relations such as *creation_time*, *file_size*, *file_name*, and *media_type*. Figure 6 shows an excerpt of the background knowledge for some file system. Additionally, the theory contains clauses for auxiliary relations, such as *older*, *larger*, and *in_same_folder*. A typical rule for irrelevancy is:

$$\begin{aligned}\texttt{irrelevant}(F) \Longleftarrow & \texttt{in\_same\_folder}(F,G), \texttt{older}(F,G) \\ & \texttt{media\_type}(M,F), \texttt{media\_type}(M,G).\end{aligned} \tag{4}$$

We apply GENME to explain two arbitrary positive examples, *irrelevant(file10)* and *irrelevant(file11)*. We provide the rewriting filters $\mathscr{V}_{older\mapsto newer}$ and $\mathscr{V}_{newer\mapsto older}$ which both allow changing a single literal. As shown in Table 3, GENME identifies 68 near miss

Table 3 Number of found near miss explanations by degree in the file management domain

	irr(file10) $\lvert\mathcal{M}\rvert = 80$	*irr(file11)* $\lvert\mathcal{M}\rvert = 80$
older ↔ *newer*		
$\lvert\mathscr{E}_1\rvert$	1	1
$\lvert\mathscr{E}_2\rvert$	19	8
$\lvert\mathscr{E}_3\rvert$	48	59

$\mathcal{N}$ denotes the set of all near miss candidates, $\mathscr{E}_d$ the set of near miss explanations of degree d and *irr* the *irrelevant* concept. $x \leftrightarrow y$ denotes the use of $\mathscr{V}_{x\mapsto y}$ or $\mathscr{V}_{y\mapsto x}$, respectively

examples from the set of 80 near miss candidates for both examples (85.0 %). For both examples, only a single near miss explanation of degree 1 is found: a file of the same media type located in the same folder which is newer than the file under consideration.

5 Empirical study of human preferences of explanation types

To investigate whether near miss explanations are considered helpful by humans, we conducted an empirical study on preferences of explanation modalities for the abstract relational family domain and the visual relational arches domain. For both domains, a cover story introducing a need for explanation to a specific recipient has been presented. Participants had to evaluate the helpfulness of explanations for the given setting by selecting their preferences in a pairwise comparison. In addition, an explicit rating of the helpfulness of the different modalities for different explanatory goals has been assessed. Details about the material, the method and results are described in the following subsections.

5.1 Rule-based and example-based explanations

Overall, four different types of explanations have been considered:

- **General rule (R)** a global explanation of the concept a specific instance belongs to,
- **Example (E)** an example-based explanation in form of a specific instance belonging to the concept,
- **Near Miss (N)** a contrastive (negative) example which has a high degree of structural similarity to the specific instance under consideration but does not belong to the class,
- **Far Miss (F)** a negative example for the considered concept which has a low degree of structural similarity to the specific instance under consideration.

Explanations were presented in form of natural language sentences—or, in the case of the arches domain, partially by visual illustrations. Natural language explanations can be generated from ILP learned rules in a straight-forward manner (Siebers and Schmid 2019; Schmid 2021).

The four types of explanations address different information needs (Miller 2019): To understand the general concept, a global rule can be assumed to be especially helpful. However, it might be the case that the helpfulness is different for abstract in contrast to visual domains. In the second case, a visual prototype might be more effective (Gurumoorthy

et al. 2019). In cognitive psychology, visual prototypes have been shown to be an effective means of concept representation for basic categories (Rosch 1979). For simple domains, an arbitrary instance might convey information similar to a prototype. For instance, in medical textbooks, example images are given to illustrate what a specific skin disease looks like. A near miss example should be especially helpful to highlight what (missing) information would be necessary to make an object belong to a class (Gentner and Markman 1994). This is often helpful if feature values or relations are hard to grasp. For instance, mushroom pickers use images to distinguish an edible mushroom from the visually most similar toadstool. Arbitrary negative examples, especially far misses, can be assumed to be less helpful to understand a concept or why a specific instance belongs to a concept. This type of explanation has been introduced as a baseline. We assume that combining different types of explanations can be more effective than each of these explanations alone. Especially, a combination of a global rule with a near miss might be most efficient to explain relational concepts.

As cover story for the **family domain** (see Sects. 3.2 and 4.1) the family tree of Kate as given in Fig. 1 but without extra arrows for miss examples has been presented. Participants were asked to imagine a conversation with their friend Kate who is originating from a native American tribe. She is curious about the different definitions of family relations in western culture since she is not familiar with them and the definitions that she grew up with are very different from the participant's. In particular, Kate wants to understand the **grandfather** relation between herself (Kate) and Ian.

The four explanations to choose from are:

- (R)ule: A grandfather is a male parent of one of your parents.
- (E)xample: One of your parents, Tom, has a male parent called Ian. Ian is your grandfather.
- (N)ear Miss: Jodie, the female parent of your parent Tom is NOT your grandfather; it is your grandmother.
- (F)ar Miss: Mat, the male child of Tom, who is the child of Ian is NOT the grandfather of Ian; it is his grandson.

The **Winston arches domain** has been introduced to the participants as shown in Fig. 4 without the object labels. Participants were given the context of playing with building blocks with their five-year-old son introducing a new type of building called **arch** given the examples and counterexamples in Fig. 4 with focus on the arch labeled *struct1*.

The explanations given for the arch domain are:

- (R)ule: An arch consists of two rectangle blocks that do not touch. They support a triangle block.
- (E)xample: *given by presenting the structure labeled struct1.*
- (N)ear Miss: *given by presenting the structure labeled struct4.*
- (F)ar Miss: *given by presenting the structure labeled struct6.*

5.2 Method

Given the cover story, the helpfulness of the different types of explanations has been assessed with a complete pairwise comparisons (Thurstone 1927). Choosing one

alternative over another is considered as less prone to subjective biases than a direct rating of each item.

In a first part of the study, all pairings of the four explanation types have been presented in a randomized sequence and participants had to always choose that explanation of the pair which they found more helpful given the cover story. In a second part of the study, pairs of pairs of explanations have been presented in a random order. In a final part, the helpfulness of the four explanation types for different information needs as been assessed explicitly. Participants rated how helpful an explanation is to understand

- the **general** concept,
- a particular **example** instance for the concept,
- what is *not* in the concept (**exclusion**).

on a scale from 1 to 5 with labels from *not at all* to *absolutely*.

The study was conducted as an online experiment with 73 valid participants (42 females, 31 males) with average age 35.72 (min 18, max 64). From an initial 151, we excluded 78 participants that either took 10 minutes or less for the experiment or that gave an incorrect answer to questions testing the participants' attention. 43 participants were employed, 27 were students, 2 were self-employed and 1 person was retired. About 50% of the participants received first the family domain followed by the arches domain and the other half of participants started with the arches followed by the family domain.

Although this is an exploratory study, given the considerations above, we can formulate the following hypotheses: (1) Near miss examples should be preferred over far miss examples in the pairwise comparisons; (2) Near miss examples should be rated as the most helpful to understand the boundaries of a given concept; (3) For the visual domain example-based explanations should be preferred over the rule-based explanation, while—in contrast—for the abstract domain, the verbal, rule-based explanation is preferred.

5.3 Results and discussion

For the family domain, preference choices for the six pairings of the four explanations resulted in the following frequency ranking (rounded relative frequencies in brackets): R (0.43) > E (0.37) > N (0.17) > F (0.03). The rule for the relational concept was preferred over all other explanation modalities followed by example, near miss and far miss. Preferences between pairs of explanations were assessed by 15 pairwise comparisons. Preference choices resulted in the frequency ranking RE (0.32) > RN (0.21) > EN (0.19) > EF (0.13) > RF (0.13) > NF (0.02).

For the visual arches domain, single preferences favoured the example as explanation, near miss was preferred over far miss: E (0.45) > R (0.30) > N (0.18) > F (0.08). For the 15 pairwise comparisons, preference choices resulted in the frequency ranking EN (0.27) > RE (0.25) > EF (0.21) > RN (0.14) > RF (0.08) > NF (0.04).

Exact binomial tests comparing preferences for near miss and far miss examples show significant preferences for near miss examples for both the family (empirical mean = 0.890, $p < 0.001$) and arches domain (empirical mean = 0.795, $p < 0.001$).

The first hypothesis, that near miss explanations are generally preferred over far miss explanations was tested by comparing the frequencies of the single choices as well as by comparing pairs containing near miss explanations with such containing far miss explanations (see Table 4).

Table 4 Frequencies of preferences of near miss over far miss explanations (significance tested with exact binomial test, Holm correction has been used to adapt the p-values for multiple testing, *** denotes a $p < 0.001$)

	Single choice	Paired choice
Family	0.890***	0.753***
Arches	0.795***	0.695***

The helpfulness ratings of the different types of explanations for different explanatory goals are summarized in Table 5. As expected for the abstract relational family domain, if the goal is to understand the general concept of a grandparent, the rule-based explanation is preferred over example-based explanations. For the goal to understand why a particular instance belongs to a concept, the example is the preferred explanation. For the purpose of understanding the boundaries of the concept, the near miss explanation is rated as most suitable and is significantly higher than the rating for the next preferred far miss explanation (t–test for dependent samples, df = 72, $t = -4.6382$, $p < 0.001$). Likewise, for the visual relational arches domain the near miss example was rated to be most helpful when the goal is to highlight what makes an instance to be outside the concept, but the difference is rather small and not significant (t–test for dependent samples, df = 72, $t = -1.0691$, p–value = 0.2886). Interestingly, for the goals of understanding the general concept and why some instance is an example for a concept, the ratings are interchanged: The rule was rated as better suited to understand why a particular example belongs to the concept and the example was rated most suited to explain the general concept.

The second hypothesis—that near misses are rated most helpful for the goal to understand the boundaries of a concept—has been confirmed although the difference to the next-best choice is only significant for the family domain.

The results for the helpfulness ratings in Table 5 already show an interesting difference between the family and the arches domain with respect to rule-based versus example-based explanations. The frequencies of preferences for rule versus example in the single pairwise comparisons is summarized in Table 6. There is a significant interaction of the preferred explanation type and the domain: 38 choices are for the rule in the family domain and the example in the arches domain. In the family-domain, in general the rule has been preferred

Table 5 Results of the questions on which explanations fulfilled which purpose in (a) the family domain and (b) the arches domain

	(R)ule	(E)xample	(N)ear Miss	(F)ar Miss
(a) Family				
general	**4.97**	4.52	2.93	2.19
example	4.14	**4.70**	2.49	2.37
exclusion	2.95	2.62	**4.30**	3.67
(b) Arches				
general	4.45	**4.70**	2.70	2.36
example	**4.56**	4.27	2.74	2.38
exclusion	3.25	2.73	**3.95**	3.82

For each purpose (rows) the mean rating value over all participants is given for each explanation. Bold numbers highlight the highest value for each purpose

Table 6 Frequencies of preferences of rule and of example for the Family and the Arches domain (combined values for the single choice for the pair rule—example and for the pairwise choices with rule in one pair and example in the other; McNemar's $\chi^2 = 30.625$, df = 1, $p < 0.001$)

		Arches		
		Rule	Example	Σ
Family	Rule	13	38	51
	Example	2	20	22
	Σ	15	58	73

over the example (51 vs. 22) while in the arches domain, the examples has been preferred over the rule (58 vs. 15).

The empirical results confirm that near misses are considered helpful by humans to understand the boundaries of a concept and thereby to avoid false positives. Our findings furthermore indicate that (1) different types of explanations are rated as most helpful for different explanatory goals, and (2) that what type of explanation is helpful also depends on the domain. Instead of the use of a cover story, a more realistic setting should be investigated as a next step. Similar to an experiment in the context of explaining the choice of moves in a strategy game (Ai et al. 2021), an explanation interface can be added to the learned models. Then, it can be assessed whether participants getting the explanation considered most helpful by the system show better performance than participants getting another explanation. In addition to performance, the effect of explanations on trust in the machine learned model can give interesting insights for the design of helpful explanations (Thaler and Schmid 2021).

6 Conclusions and further work

We introduced near miss explanations for relational concepts learned with ILP. As cognitive science research suggests (Gentner and Markman 1994), near miss explanations can play an important role to highlight what aspects are necessary for an instance to belong to a given class. The GENME algorithm has been presented which generates near miss explanations with different degrees of nearness to a specific positive instance for a given concept.

The current version of GENME is realized in Prolog and relies on a generate-and-test strategy, first generating and then checking all near miss candidates. Checking the candidates also follows a generate-and-test strategy, first minimally changing a clause and constructing a minimal substitution, and then checking whether these constitute a near miss explanation. There are different possibilities to improve this approach. For instance, near miss candidates might be used to restrict the choice of minimal substitutions. As the complexity of GENME is exponential in the size of the substitution, substantial improvement may be gained. Alternatively, the set of near miss candidates might not be generated explicitly beforehand but could be constructed step by step. This strategy might reduce the number of consequence tests. In the worst case, the current algorithm as well as the proposed improvements have the same complexity. To determine which algorithmic strategy is more promising regarding average complexity, extensive empirical tests with different domains will be required.

In an empirical study, we investigated the helpfulness of near miss explanations in contrast to other example-based explanations and rule-based explanations. The abstract relational family domain and the perceptual relational arches domain were presented. Results showed that humans rated near miss explanations as helpful. Interestingly, for the abstract relational domain, rule-based explanations were favored over example-based explanations while for the perceptual relational domain example-based explanations were preferred. In general, there was a significant preference for near miss explanations over far misses. Our empirical findings together with the proposed GENME algorithm introduce near miss explanations as a new type of explanations in relational domains.

Acknowledgements We thank Sebastian Seufert and Klaus Stein for support with the generation of the domains and for helpful discussions of the near miss algorithm. We thank Johannes Langer for his support with the statistical data analyses.

Author Contributions All authors contributed equally to this work.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data and code availibility Participation in the study was voluntary. No individual's data or image is published. Data, material, and code used for this work can be obtained by writing the authors an e-mail.

Declarations

Conflict of interest Part of the work reported in this paper is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project 427404493 (Dare2Del). The authors declare that there is no conflict of interest.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access, 6*, 52138–52160.

Ai, L., Muggleton, S. H., Hocquette, C., Gromowski, M., & Schmid, U. (2021). Beneficial and harmful explanatory machine learning. *Machine Learning, 110*(4), 695–721.

Bien, J., Tibshirani, R., et al. (2011). Prototype selection for interpretable classification. *The Annals of Applied Statistics, 5*(4), 2403–2424.

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., & Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems* (pp. 592–603).

Ehsan, U., Harrison, B., Chan, L., & Riedl, M. O. (2018). Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 81–87). ACM.

Gentner, D., & Markman, A. B. (1994). Structural alignment in comparison: No difference without similarity. *Psychological Science, 5*(3), 152–158.

Gentner, D., Loewenstein, J., & Thompson, L. (2003). Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology, 95*(2), 393.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature, 538*(7626), 471.

Gurumoorthy, K. S., Dhurandhar, A., Cecchi, G., & Aggarwal, C. (2019). Efficient data representation by selecting prototypes with importance weights. In *IEEE International Conference on Data Mining (ICDM 2019)* (pp. 260–269). IEEE.

Kazemi, S. M., & Poole, D. (2018). RelNN: A deep neural model for relational learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kim, B., Koyejo, O., & Khanna, R. et al. (2016). Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems (NeurIPS 2016), Barcelona, Spain* (pp. 2280–2288).

King, R. D., Muggleton, S. H., Srinivasan, A., & Sternberg, M. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences, 93*(1), 438–442.

Koller, D., Friedman, N., Džeroski, S., Sutton, C., McCallum, A., Pfeffer, A., Abbeel, P., Wong, M.-F., Heckerman, D., Meek, C., et al. (2007). *Introduction to statistical relational learning*. MIT Press

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1675–1684). ACM.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence, 267*, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

Molnar, C. (2019). Interpretable Machine Learning. Lulu.com.

Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming, Special Issue on 10 Years of Logic Programming, 19–20*, 629–679.

Muggleton, S. H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., & Besold, T. (2018). Ultra-strong machine learning: Comprehensibility of programs learned with ILP. *Machine Learning, 107,* 1119–1140.

Nguyen, A. T., Kharosekar, A., Krishnan, S., Krishnan, S., Tate, E., Wallace, B. C., & Lease, M. (2018). Believe it or not: Designing a human-AI partnership for mixed-initiative fact-checking. In *The 31st Annual ACM Symposium on User Interface Software and Technology* (pp. 189–199). ACM.

Palmer, F. R. (1981). *Semantics: A New Outline*. Cambridge University Press.

Pollock, J. L. (1976). The 'possible worlds' analysis of counterfactuals. *Philosophical Studies, 29*(6), 469–476.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM.

Rosch, E. (1979). Principles of categorization. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and Categorization* (pp. 27–48). L. Erlbaum.

Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

Schmid, U. (2021). Interactive learning with mutual explanations in relational domains. In S. Muggleton & N. Charter (Eds.), *Human-like Machine Intelligence* (pp. 337–353). Oxford University Press.

Schmid, U., Wirth, J., & Polkehn, K. (2003). A closer look at structural similarity in analogical transfer. *Cognitive Science Quarterly, 3*(1), 57–89.

Siebers, M., & Schmid, U. (2019). Please delete that! Why should I?—Explaining learned irrelevance classifications of digital objects. *KI, 33*(1), 35–44. https://doi.org/10.1007/s13218-018-0565-5.

Sterling, L., & Shapiro, E. Y. (1994). The art of Prolog: advanced programming techniques. MIT Press

Stickel, M. E. (1991). A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. *Annals of Mathematics and Artificial Intelligence, 4*(1–2), 89–105.

Tamaki, H., & Sato, T. (1986). OLD resolution with tabulation. ppIn E. Shapiro (Ed.), *Third International Conference on Logic Programming* (pp. 84–98). Heidelberg: Springer.

Telle, J. A., Hernández-Orallo, J., & Ferri, C. (2019). The teaching size: Computable teachers and learners for universal languages. *Machine Learning, 108*(8–9), 1653–1675.

Thaler, A., & Schmid, U. (2021). Explaining machine learned relational concepts in visual domains—Effects of perceived accuracy on joint performance and trust. In *Proceedings of the 43rd Annual Conference of the Cognitive Science Society (CogSci'2021)*. Cognitive Science Society.

Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review, 34*(4), 273.

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology, 31*(2), 2018.

Winston, P. H. (1970). Learning structural descriptions from examples. *Technical Report MIT/LCS/TR-76, MIT*