[Title Page]

A Novel Data-Driven Framework Based on BIM and Knowledge Graph for Automatic Model Auditing and Quantity Take-off

Hao Liu[a], Jack C.P. Cheng[a,]*, Vincent J.L. Gan[b,]*, Shanjing Zhou[c]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, 999077, Hong Kong.

[b] Department of the Built Environment, National University of Singapore, 117566, Singapore.

[c] Centre for Systems Engineering and Innovation, Department of Civil and Environmental Engineering, Imperial College London, London, SW7 2AZ, UK.

*Corresponding authors.

Email: cejcheng@ust.hk (Jack Cheng); vincent.gan@nus.edu.sg (Vincent Gan)

ABSTRACT

Model auditing is a critical step before conducting Building Information Modeling (BIM)-based Quantity Take-off (QTO) because these models may contain various human errors and mistakes, leading to insufficient semantic information and inconsistent modeling style in BIM models. The traditional object-oriented approach has difficulties in representing unstructured BIM data (e.g., interrelationships), while rule-based methods involve tremendous human efforts to develop rule sets, lacking flexibility for different requirements. Therefore, this study aims to establish a novel data-driven framework based on BIM and knowledge graph (KG) to represent unstructured BIM data for automatic inferences of auditing results of BIM model mistakes. It starts by establishing a BIM-KG data model via identifying required information for auditing purposes. Subsequently, BIM data is automatically transformed into the BIM-KG representations, the embeddings of which are trained using a knowledge graph embedding model. Automatic mechanisms are then developed to utilize the computable embeddings to effectively identify mistake BIM elements. The framework is validated using illustrative examples and the results show that 100% mistake elements can be identified successfully without human intervention.

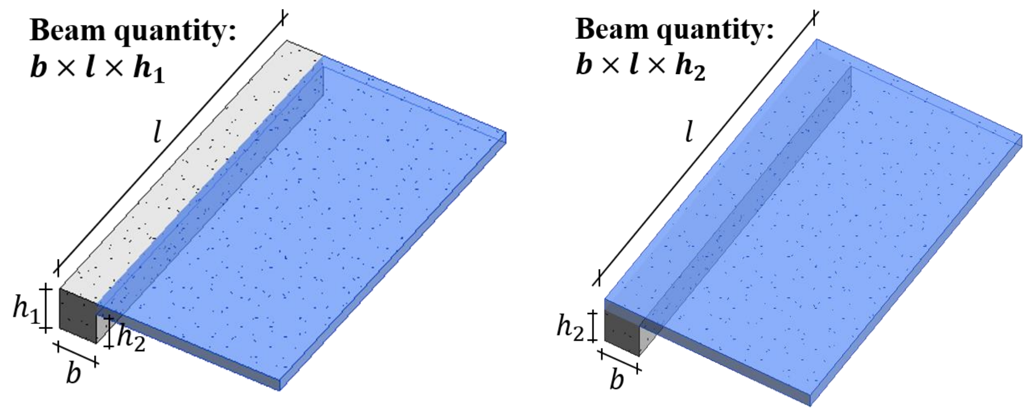
Keywords:

Building information modeling, Quantity take-off, BIM information quality, BIM model auditing, Knowledge graph embedding

1. INTRODUCTION

Quantity take-off (QTO) is a process of recognizing measurement items, obtaining dimensional information, and calculating the items in units such as areas and volumes from construction documents [1]. It plays a significant role in a construction project since it affects multiple important tasks throughout the project lifecycle, such as the estimation of preliminary cost in early design, preparation of the bill of quantities for project tendering, and material procurement in construction [2,3]. Traditionally, QTO is time-consuming and error-prone as it requires professional quantity surveyors to manually interpret 2D design drawings and calculate the results based on predefined rules in the measurement standards [4–6]. With the development of Building Information Modeling (BIM) techniques, this process has been revolutionized because quantities can be automatically extracted from 3D models together with the geometric attributes [5]. Therefore, the BIM-based QTO can provide more automatic and accurate estimation of material quantities, greatly reducing human effort and errors in estimations [5,7,8].

In order to obtain accurate quantities that are compliant with measurement standards under the BIM-based method, BIM models need to be created in a consistent way according to specifications on modeling styles and semantic information [9–11]. Fig 1 illustrates how the inconsistent modeling styles can impact quantities from BIM models: because the geometric representations are different, the output quantities are different for the beam (and the slab) in the two modeling styles shown in Fig 1 (a) and (b). According to the Hong Kong Standard Method of Measurement [12] (HKSMM) where the major measurement logic is similar to that in commonwealth countries (the UK, Singapore, etc.), either may be correct, depending on the concrete grade information. As shown in Fig 2, if the beam has a different concrete grade than the slab, it is measured through the slab (i.e., $b \times l \times h_1$); otherwise, it is measured to the soffit of the slab (i.e., $b \times l \times h_2$). In this case, all the beam-suspended slab joints should be created in either of the ways but consistently to enable easy adjustments for the output quantities to achieve accurate QTO. For instance, if all the beam-suspended slab joints in the BIM model are created in the way shown in Fig 1 (a), accurate beam quantities can be obtained by simply making 0 or $b \times l \times (h_1 - h_2)$ adjustments for all the beams according to the concrete grade information after extracting the quantities (i.e., $b \times l \times h_1$) from the BIM model. Otherwise, if there are multiple modeling styles (e.g., the styles shown in Fig 1 (a) and (b)) for such beam-suspended slab joints, it would be time-consuming to make adjustments for the beams created in different ways. In addition, as Fig 2 shows, BIM models should contain sufficient semantic information such as concrete grade so that the calculation logic can be determined successfully in the BIM-based QTO process.



(a) The beam takes precedence over the slab

(b) The slab takes precedence over the beam

Fig 1. Inconsistent modeling styles for a beam-suspended slab joint (adopted from [13])

CLASSIFICATION TABLE					MEASUREMENT RULES
18. Suspended slabs 19. Coffered and troughed slabs			m³	1. Horizontal 2. Sloping ≤ 15° 3. Sloping > 15°	M.12 The measurement of suspended slabs is taken across columns and beams, except where the columns or beams are of a different mix.

Fig 2. Part of the HKSMM descriptions for measuring slab quantities [12]

Nevertheless, it is not uncommon to see different BIM modelers using different methods of modeling in practice [3,14], resulting in different modeling styles for the same thing, regarding the precedencies between elements in different parts of the building and making the output quantities troublesome to adjust. In addition, they are not as aware of the importance of the required information in BIM models for QTO as quantity surveyors are [15]. Thus, the BIM models may not contain all the necessary information for determining the measurement logic in the standards [16]. Therefore, it is necessary to audit BIM models against agreed specifications to ensure a consistent modeling style and sufficient semantic information in the BIM model before conducting BIM-based QTO.

However, the proper representation of unstructured BIM data for model auditing is still a matter of concern. BIM information is stored in an object-oriented and parametric manner, indicating that a BIM model is assembled by different elements with different sets of properties [17,18]. Such object-oriented representations are implicit when expressing the unstructured interrelationships between elements [19]. An example of different precedencies and topological relationships between building elements regarding their modeling styles is shown in Fig 1. The second challenge concerns the utilization of the representation of unstructured BIM data for automatic BIM model auditing. The specified requirements are buried in various texts. Heavy human intervention is required to align them with the BIM data so as to identify elements that are not modeled appropriately. Thus, an automatic way that utilizes BIM data to identify the mistake patterns is needed to reduce the considerable human effort in this process. As a semantic graph representation with heterogeneous features, Knowledge Graph (KG) [20] provides new insights to express and process object properties and relationships explicitly and automatically. Previous studies have leveraged such a representation for BIM model information management [21–24], demonstrating the potential of KG to analyze the rich semantics in BIM models. However, the application of KG for BIM model auditing is still unmature.

Therefore, this study aims to develop a novel data-driven framework, with the aid of BIM and KG, to automatically audit BIM models for the purpose of QTO. This includes the design of KG-based representation and transformation mechanisms to express unstructured BIM data, as well as the development of BIM-KG data manipulation and inference mechanisms to determine auditing results efficiently for mitigating human intervention. Since concrete structures are one of the most common structures and their quantities from BIM are vulnerable to the aforementioned modeling issues, this study includes them as the scope of QTO-oriented BIM model auditing. Meanwhile, this study focuses on auditing mistakes about semantic information (i.e., absent or inappropriate semantic attributes) and modeling style (i.e., inconsistent topological arrangements between elements) considering that they are typical modeling concerns in BIM models [9,15,25]. The proposed framework begins with the establishment of a BIM-KG data model through identifying the required information to conduct BIM model auditing for QTO. Based on the BIM-KG data model, BIM data is transformed to the KG representations automatically. Following this, manipulation mechanisms utilizing the transformed representations are designed to efficiently identify the elements not in compliance with the requirements. The performance of this proposed framework is validated using a set of BIM models with different kinds of mistakes about semantic information and modeling style.

The rest of the paper is organized as follows: Section 2 introduces related studies about BIM modeling specifications, attempts on BIM model auditing and using knowledge graph techniques for BIM. Section 3 describes the methodology part, including the BIM-KG data model, transformation, and utilization mechanisms to conduct BIM model auditing automatically, followed by the illustrative examples to validate its performance in Section 4. Finally, Section 5 presents the conclusion and future work.

2. RELATED WORK

As shown in Fig 3, previous studies are reviewed from two aspects in this section, namely BIM model auditing and knowledge graph techniques for BIM. Section 2.1 introduces BIM modeling specifications about semantic information and modeling styles to state the necessity and contents of auditing requirements. Section 2.2 focuses on existing methods to audit BIM models against the auditing requirements as well as their limitations. Section 2.3 aims to review the uses of knowledge graph techniques in BIM to show the potential benefits of such techniques to the auditing. Finally, research gaps are identified in a summary.

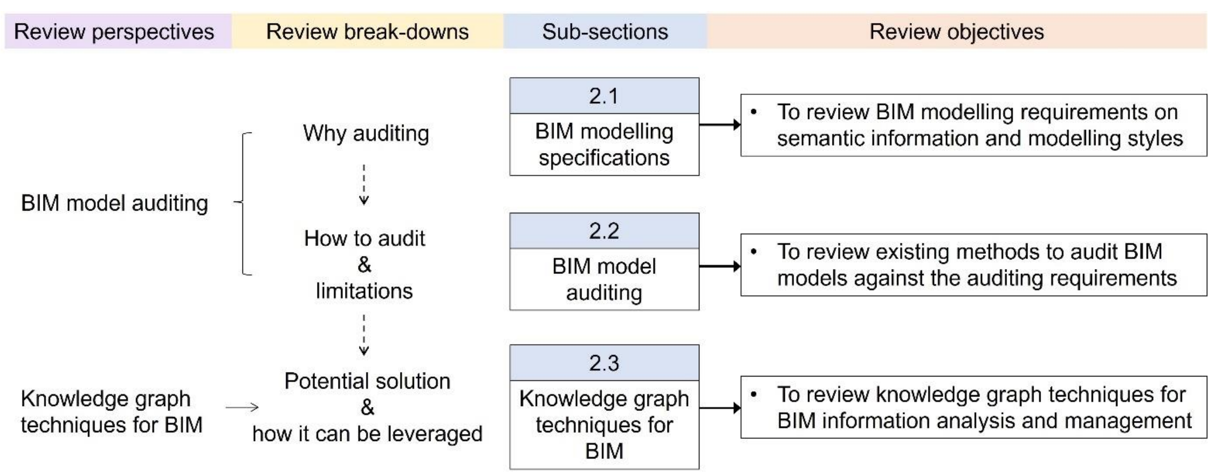


Fig 3. Overview of related work

2.1. BIM Modeling Specifications

The construction industry is dominated by official and/or recommended specifications to guide its activities and procedures [9]. Likewise, BIM models should be created having uniformity so as to prevent conflicts in downstream applications. To achieve this, administration organization/regulatory bodies in different areas issue various BIM modeling specifications to regulate BIM modeling practice, some of which emphasize the requirements on semantic information and modeling style in BIM models. The Standard for Graphic Expression of Building Information Modeling [26] in China includes general descriptions about the types of required semantic information and detailed guidance about joining precedencies when modeling joints to regulate BIM model expressions for different disciplines. Similarly, the Hong Kong Housing Authority issued the Standard Approach of Modeling (SAM) [27] for creating structural BIM models with emphasis on some semantic information in particular and explicit descriptions about consistent BIM modeling methods. As for QTO, two BIM requirements [28,29] in Finland specify the required semantic information (e.g., construction type, material type) and how BIM models should be created consistently, especially for joining precedencies between different building elements (e.g., "*The joining of slabs and walls must be modeled such that the slab ends to the surface of the load-bearing wall structure without extending inside it.*"), in detail to ensure that BIM models meet necessary conditions for QTO purposes. In addition, the National BIM Standard-United States Version 3 (NBIMS-US V3) [30] published by National Institute of Building Sciences in the US describes detailed exchange requirements from design to QTO (e.g., classification code, construction type) so that the quantities can be generated from BIM models successfully. The BIM Model Information Requirements for Quantity Take-off (BIM MIR for QTO) [31] from the Hong Kong Institute of Surveyors also emphasizes the importance of semantic information in BIM models and specifies the minimum information requirements (e.g., concrete grade) as well as basic modeling styles to ensure consistency for QTO (e.g., "*Beam is defined as horizontal element. The beams will not cut the vertical element. The geometry of beams is joined with the slabs where the slabs take precedence.*"), supplemented with guidance on extracting quantities in compliance with measurement rules. These specifications emphasis and specify the requirements of semantic information and modeling styles for BIM models to support downstream tasks (e.g., QTO) and produce consistent results [26–31]. However, the requirements are buried in texts. How to align them with BIM data to make sure BIM models follow these specifications is unclear and needs further investigations.

2.2. BIM Model Auditing

To tackle this problem, some software tools have been developed to audit BIM models before the model delivery between different disciplines to ensure that the specified requirements in the specifications are satisfied. For example, Solibri Model Checker [32] is widely adopted to check the required semantic data and geometry constraints (e.g., spacing limit, clashing elements) in Industry Foundation Classes (IFC) models through rigorous rule patterns set by domain experts. Similarly, through customizing rule sets, Autodesk Model Checker for Revit [33] can check Revit models against the requirements and generate compliance reports. On the other hand, various studies have explored ways to formulate systematic frameworks utilizing different software tools so that BIM model auditing can be performed comprehensively. Zadeh et al. [34] proposed a framework to assess the information conformance with owner requirements when using BIM models for facility management (FM), and conducted testing by integrating different tools such as Solibri Model Checker and Revit Schedule. Cavka et al. [35] defined levels of owner's operational requirements in the FM stage, including model structure, model content, and design compliance, and proposed ways to audit BIM models before handover mainly based on Solibri Model Checker. Similarly, with the help of the Solibri Model Checker, Gholami et al. [36] performed the quality checking of BIM models against the energy analysis requirements such as architectural layout and general space check (e.g., space boundary). Making use of customized checklists and queries on database systems such as Microsoft Access, Donato et al. [37] assessed the information quality (e.g., adequacy, readability) of BIM models during the architectural design process. In short, these studies attempted to audit BIM models with respect to the requirements on semantic data and geometry constraints, with predefined rule patterns that involve extensive human effort and external software tools outside the BIM authoring software.

2.3. Knowledge Graph for BIM

A knowledge graph is a representation of facts using a graph-structured data model, which includes: 1) entities that are real objects or abstract concepts; 2) relationships between entities; 3) semantic descriptions denoting the meanings of the entities and relationships (e.g., type, property) [38]. It is referred to a graph due to the graph data structure [39] and a knowledge base for manipulation and inference based on the stored facts [40]. The graph representation provides explicit expressions of entities and relationships with rich semantics [41–43], and thus is more amenable to semantic pattern recognition [44]. Such a representation technique has been explored to represent and analyze BIM information that inherently contains heterogeneous entities and relationships for different purposes. For example, the graph representation can be used to express BIM models for BIM data management. The semantic information and relationships (e.g., connectivity, containment) in BIM models can be represented using graph data models, based on which flexible and efficient information queries can be achieved [21,22]. The flexibility of graph representation can also support generative building design. Through representing spaces as entities and adjacencies as relationships, spatial layout designs can be generated using graph transformations [45,46] and evolutionary algorithms [47]. Moreover, some studies have explored the use of graph representations to validate BIM data against certain requirements. Pauwels et al [48] described a typical semantic rule checking process by extracting related IFC information and making use of semantic web rules or Resource Description Framework (RDF) graphs to express rules. Zhong et al [49] developed an ontology-based framework to support building environmental compliance checking under BIM environment, where knowledge about BIM, environmental information and building regulations is expressed in ontologies and SPARQL rules are utilized to conduct the rule-reasoning process. Jiang et al [50] presented BIM models and building codes in ontologies and developed mapping and checking rules to automatically validate BIM data against ambiguous regulatory information. In short, these studies tried to automate the BIM data validation process using ontological representations and rule-based reasoning techniques, which rely heavily on human experts to develop both graph representations and rules.

In summary, BIM models need to have sufficient semantic information and consistent modeling style to enable a smooth BIM-based QTO process. There are some studies [34–37] trying to check such prerequisites from specifications and/or owner's requirements for BIM models. But their representations of BIM data are limited to an object-oriented approach and have difficulties in expressing and auditing the unstructured complex constraints (e.g., appropriate joining precedencies between different building elements) explicitly. Moreover, their ways of relying on external software tools and predefined rule patterns to compare BIM data against the requirements are labor-consuming. A few studies have demonstrated the strength of graphs to represent and validate BIM information. However, a proper representation for auditing both semantic information and modeling style is still lacking. On the other hand, existing studies are still limited to a top-down approach that requires considerable effort from domain experts to design both graph representations and validation rules. To address these problems, this study proposes a semi-bottom-up framework that is driven by BIM data in compliance with the requirements and proper BIM-KG representations to automatically identify problematic BIM models. Proper BIM-KG representations and the transformation mechanisms are designed to explicitly express unstructured BIM data in terms of semantic information and modeling style. A knowledge graph embedding model is utilized to transform the BIM-KG data into embeddings/vectors, based on which corresponding mechanisms are developed for automatic inferences of auditing results.

3. METHODOLOGY

Fig 4 presents an overview of the proposed methodology, with explanatory pictures attached as examples for some key concepts and steps. Knowledge from QTO-related BIM modeling specifications (e.g., BIM MIR for QTO [31]) is leveraged to identify the requirements (e.g., requirements on semantics and topology aspects) for QTO-oriented BIM model auditing. Following this, the BIM-KG data model is established to represent relevant entities, attributes and relationships and the standard BIM model is defined as a BIM model that has sufficient semantic information and consistent modeling style according to the requirements from relevant specifications. Based on the BIM-KG data model, BIM model information of interest (e.g., semantic information, topological arrangement relationship) that is needed to audit the identified requirements is automatically transformed from a standard BIM model to BIM-KG triples in the form of <*head*, *relation*, *tail*> through attribute extraction and geometry manipulation. An improved knowledge graph embedding model (i.e., improved TransR) is then used to get the embeddings of the standard BIM-KG triples. For the new BIM model to be audited, the corresponding new BIM-KG triples are generated and their embeddings are obtained through contextual comparison between the standard BIM-KG triples and the new ones.

Furthermore, the embeddings are utilized to score those new BIM-KG triples and automatically decide on the auditing results with respect to a self-determined proper threshold (i.e., a value that can clearly distinguish scores of correct and mistake BIM-KG triples / elements given that correct and mistake triples / elements have high and low scores, respectively). The rejected elements are returned to the BIM modelers for modification over several rounds until acceptance. Details of the methodology are described in following subsections.

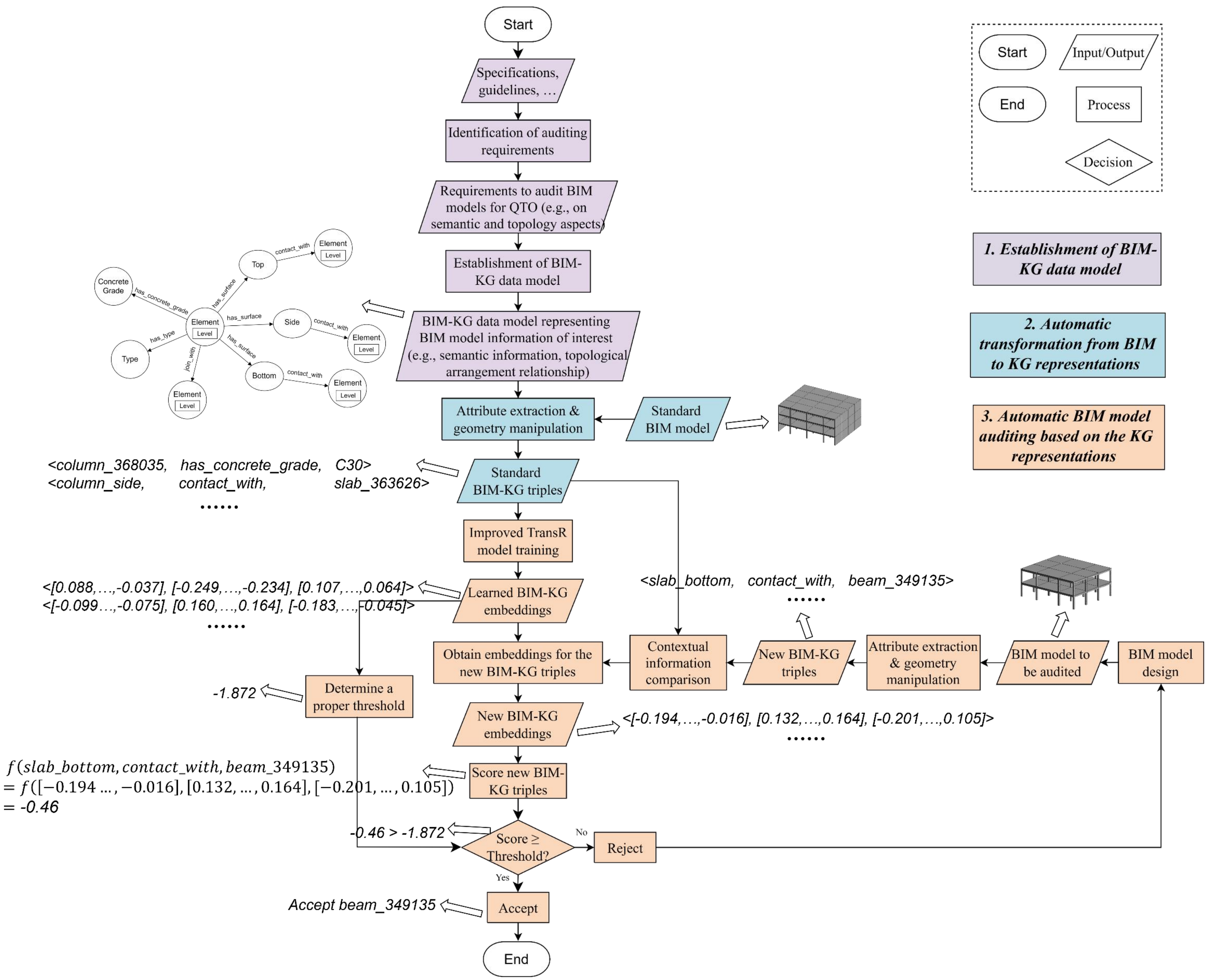


Fig 4. Overview of the proposed methodology framework

3.1. BIM-KG data model

3.1.1. BIM-KG information requirements

In this study, BIM Model Information Requirements for Quantity Take-off (BIM MIR for QTO) [31] by The Hong Kong Institute of Surveyors is selected as an example BIM modeling specification for illustration. The requirements on semantic information and consistent modeling style are common to most BIM modeling specifications. Regarding the auditing scope, typical building elements (i.e., slab, beam, column, wall) are selected for illustrative purposes. The requirements mainly cover two aspects: 1) consistent modeling style. Fig 5 shows the requirements on the modeling style of beams as an example. Vertical elements and slabs take precedence over beams. In essence, it shows the requirements on the proper topological arrangement relationships between elements. Different topological arrangements result in different contact situations between elements. For instance, the slab is in contact with the side and top of the beam when the beam and slab take precedence, respectively (shown in Fig 1). Therefore, the topological contact relationships between elements are required to identify different modeling styles. In addition, connectivity information is needed since elements are connected with each other at the joints. 2) sufficient semantic information. To perform the QTO logic specified in the measurement rules for these common concrete elements, elements should carry adequate object information including concrete grade and construction type (e.g., horizontal/slope etc. in the beam example shown in Fig 5).

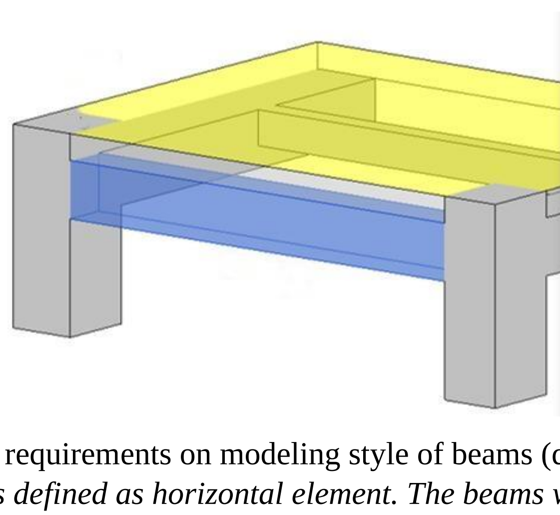


Fig 5. Example requirements on modeling style of beams (description in the specification: *Beam is defined as horizontal element. The beams will not cut the vertical element. The geometry of beams is joined with the slabs where the slabs take precedence.*) (adopted from [31])

3.1.2. BIM-KG representation

Based on the identified required information for QTO-oriented BIM model auditing, a BIM-KG data model is established to transform BIM data into KG triples with reference to the IFC data model under IFC4_ADD2_TC1 [51]. Fig 6 shows how the nodes, relations, and auxiliary information in the BIM-KG data model are mapped and established based on the identified information requirements for auditing in Section 3.1.1. For example, as concrete grade information is needed for auditing, the nodes *element* and *concrete grade* and the relation *has_concrete_grade* are mapped from the entities *IfcBuildingElement*, *IfcPropertySet*, and *IfcRelDefinedByProperties* specified in the IFC data model. Although level information is not included in the required information for auditing, it can be utilized to locate elements. Thus, level information is defined as auxiliary information in the *element* nodes. In addition to those nodes and relations that can be established in reference to the IFC data model, the relation *contact_with* is introduced to associate face nodes (i.e., *top*, *side*, *bottom*) with *element* nodes so that the information about topological arrangement relationships between elements mentioned in Section 3.1.1 can be captured for auditing modeling styles. Of note is that the *element* entity can be a *real element* or *empty* since elements and their faces may not join/contact with anything. To construct this BIM-KG, triples in the form of <*head*, *relation*, *tail*> with auxiliary information (i.e., levels of the elements) are generated from the BIM models, which will be introduced in next section. Table 1 lists some examples of triples under the defined BIM-KG data model.

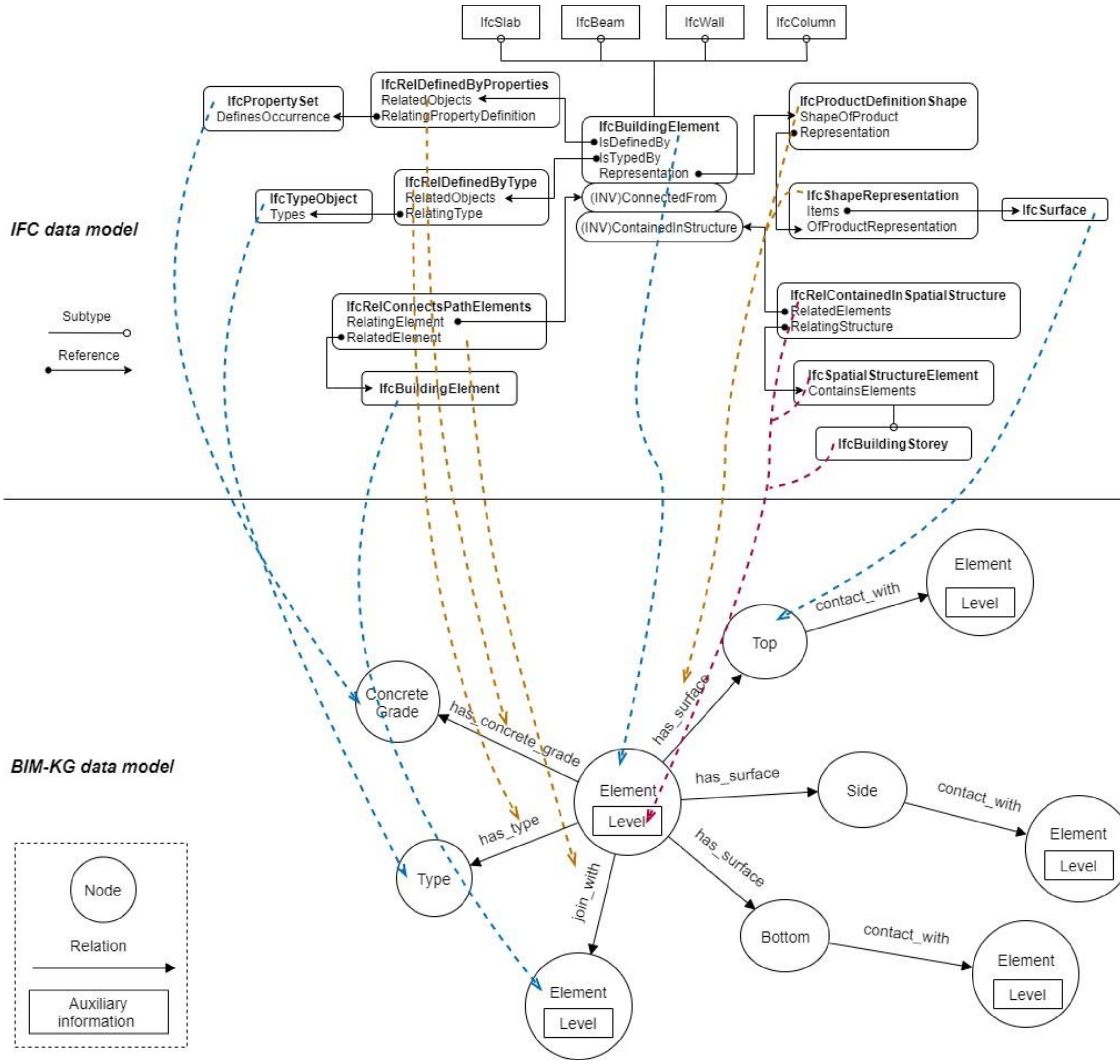


Fig 6. BIM-KG data model for QTO-oriented BIM model auditing and its relationship with IFC data model

Table 1. Examples of triples to construct the defined BIM-KG

BIM-KG triple definition	BIM-KG triple instance
<element, has_concrete_grade, element>	*< beam_123[1], has_concrete_grade, C30>*
<element, join_with, element>	*<beam_123[1], join_with, slab_234[1]>*
<element, has_surface, top>	*<beam_123[1], has_surface, beam_top>*
<top, contact_with, element>	*<beam_top, contact_with, slab_234[1]>*

[1]: 123 and 234 are the unique identifiers of the elements and serve as suffixes to distinguish different element instance entities

3.2. Automatic Transformation to KG

With respect to the BIM-KG data model in Fig 6, a transformation mechanism is developed to automatically convert BIM data into BIM-KG triples to construct the BIM-KG. The details are illustrated in Fig 7 with *beam_123* as an example. First, the category and ID information of the element are extracted to form a unique entity name. The semantic attributes and joining elements are obtained to generate triples on semantic information and element connectivity such as <*beam_123, has_concrete_grade*, *empty*> and <*beam_123*, *join_with*, *slab_234*>. The generation of triples related to the topological contact information between elements is based on [13]. The faces of the element are extracted and thickened on both sides (i.e., the extracted faces are extruded into solids, as shown in Fig 7). Intersection checking is performed between the corresponding generated solids and other elements to detect the elements in contact with the faces. As a result, triples such as <*beam_side*, *contact_with, slab_234*> and <*beam_side*, *contact_with*, *empty*> are obtained. Finally, these triples form the base of the BIM-KG for model auditing.

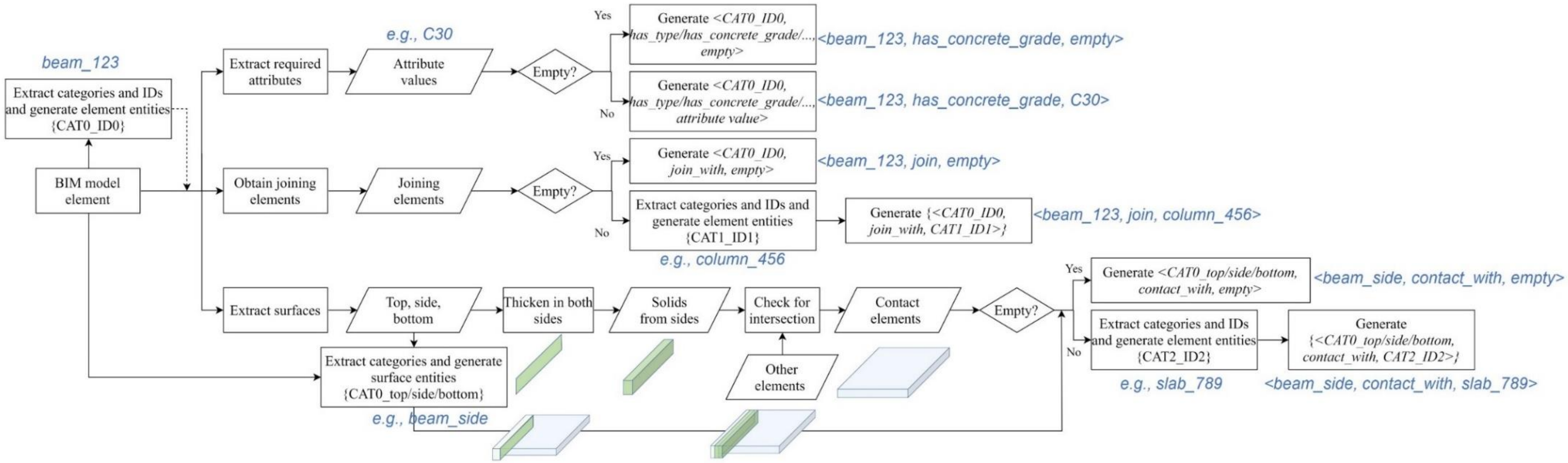


Fig 7. Mechanism to transform BIM data to KG triples

3.3. Automatic BIM Model Auditing Based on KG

3.3.1. Improved TransR model to obtain KG embeddings

Based on the transformation mechanism described in Section 3.2, standard BIM-KG representations are generated from the standard BIM model. Then, auditing mechanisms are developed to automatically determine the mistake elements in BIM models based on the manipulation of the transformed BIM-KG representations. First, a knowledge graph embedding model is utilized to train the embeddings for the entities and relations. Given a KG with a collection of triples $S = \{(h, r, t)\}$, where h is a head entity, t is a tail entity, and r is the relation between them, knowledge graph embedding is a technique that converts the entities and relations into vectors (i.e., embeddings). By doing so, the manipulation of KG entities and relations for downstream applications can be simplified to numerical computations while the inherent structure is preserved [52].

There are many methods to obtain knowledge graph embeddings, among which TransE and its variants are simple yet effective with good performance [53,54]. TransE [55] is a pioneering and representative model for obtaining knowledge graph embeddings. As shown in Fig 8, TransE iteratively optimize embeddings of entities and relations in the KG triples. More specifically, embeddings of entities and relations in the KG triples (positive triples) are first initialized randomly. Then, entities and relations in the positive triples are shuffled to get negative triples (i.e., triples that are unobserved in the KG). The assumption in TransE is that the sum of embeddings of the head entity and relation (i.e., $\vec{h} + \vec{r}$) should be close to the embedding of the tail entity ($\vec{t}$) if the triple is positive, as shown in Fig 9. Thus, a Euclidean distance-based score function is utilized to score the positive and negative triples, as defined in Eq. (1), where $\vec{h}, \vec{r}, \vec{t}$ are the embeddings of the head entity, relation, and tail entity, respectively, and $f(h, r, t)$ is the score of the triple (h, r, t). The embeddings are then updated to maximize and minimize the scores of positive and negative triples, respectively. After updating in iterations, the learned embeddings make $\vec{h} + \vec{r} \approx \vec{t}$ (i.e., the assumption) hold for every (h, r, t). Details of TransE can be found in [55].

$$f(h, r, t) = -\left\|\vec{h} + \vec{r} - \vec{t}\right\|_{L_1/L_2} \tag{1}$$

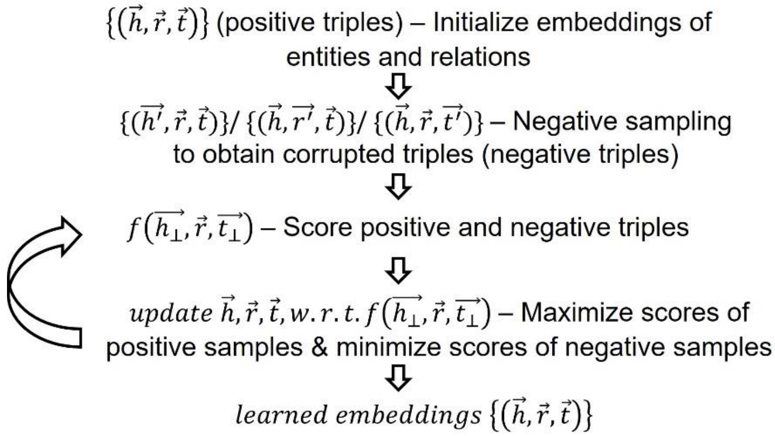


Fig 8. Key idea of TransE model [55]

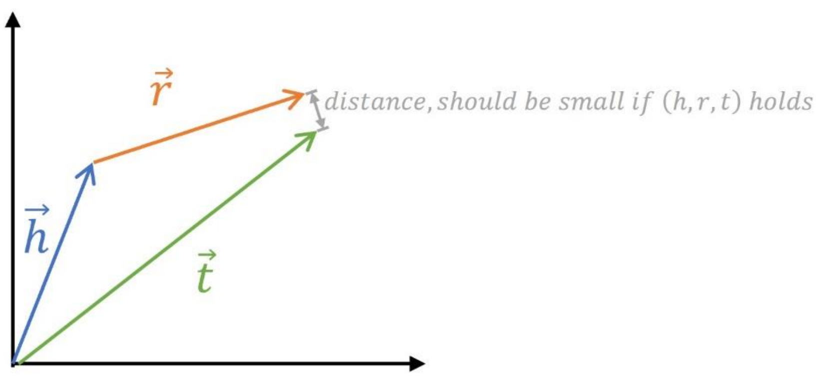


Fig 9. Illustration of the TransE assumption (i.e., the sum of embeddings of the head entity and relation, $\vec{h}+\vec{r}$, should be close to the embedding of the tail entity, $\vec{t}$, if the triple is positive)

However, TransE suffers from complex relations such as 1-to-N, N-to-1, and N-to-N [53,56]. To address these limitations, TransR [53] is proposed, where the entities are embedded in the same vector space while different relations are represented in different relation spaces, and projection matrixes for different relations are introduced to project the entity embeddings into the relation-specific spaces, as follows:

$$\overrightarrow{h_\perp} = M_r\vec{h} \tag{2}$$

$$\overrightarrow{t_\perp} = M_r\vec{t} \tag{3}$$

$$f(h,r,t) = -\left\|\overrightarrow{h_\perp} + \vec{r} - \overrightarrow{t_\perp}\right\|_{L_1/L_2} \tag{4}$$

in which M_r refers to the projection matrix of relation r, $\overrightarrow{h_\perp}$ and $\overrightarrow{t_\perp}$ stand for the head and tail entity embeddings respectively that are projected from the entity space into the space of relation r. Consequently, the scoring function is formulated in Eq. (4). Other parts (e.g., embedding initialization, negative triple sampling, embedding updating with respect to triple scores) are similar to TransE (Fig 8, Fig 9). Details of TransR can be found in [53].

TransR regards different projection matrixes as dense ones with the same dimension. However, such assumption ignores the imbalance of relations, which means the numbers of triples/entities connected by different relations are different. For example, in the BIM-KG data model for the QTO-oriented BIM model auditing (shown in Fig 6), the relation *contact_with* connects at least three times as many triples/entities as *has_type* does. Considering the projection matrixes for different relations in the same way cannot distinguish them. To address this limitation, dynamic sparse matrixes are proposed instead of static dense ones for the projection operation in the original TransR so as to overcome the relation imbalance issue, as follows:

$$\delta_r = 1 - n_r/n_{max} \tag{5}$$

$$\overrightarrow{h_\perp} = M_{\delta_r}\vec{h} \tag{6}$$

$$\overrightarrow{t_\perp} = M_{\delta_r}\vec{t} \tag{7}$$

where a sparse factor δ_r is introduced to dynamically adjust the sparsity of the projection matrixes for different relations, n_r is the number of triples connected by relation r and n_{max} is the maximum value, δ_r indicates how many elements in the projection matrix are 0 values. Correspondingly, M_{δ_r} is the dynamic sparse projection matrix of relation r, where a ratio of δ_r elements in the matrix are set to 0 randomly. The scoring function is defined as the same as that in TransR, as shown in Eq. (4).

Through the training of this improved TransR model, embeddings of the entities and relations in the standard BIM-KG from the standard BIM model are obtained. The embeddings are then utilized in subsequent steps to determine mistake elements. The standard BIM model/standard BIM-KG is the single source of truth, based on which wrong patterns in different BIM models are identified using the embeddings. Details are provided in the following sections 3.3.2 and 3.3.3.

3.3.2. Contextual comparison to obtain embeddings of new BIM-KG entities

Following the transformation mechanism described in Section 3.2, new BIM-KG entities are generated from the new BIM model to be audited. Note that the embeddings are trained over the standard BIM-KG. In other words, only the entities and relations in the standard BIM-KG have corresponding embeddings. For example, an entity *slab_234* in the standard BIM-KG has an embedding of *[-0.201,0. 089,…,0.105].* However, the new BIM-KG entities may not appear in the standard BIM-KG, which causes problems in the auditing. For instance, a new entity *slab_567* does not exist in the standard BIM-KG and thus has no corresponding embedding. Consequently, it cannot be scored (i.e., take the embeddings of entities and relations in a BIM-KG triple into a predefined score function such as Eq. (4) to calculate a triple score) and compared with a threshold to decide whether it should be rejected or accepted according to the auditing procedure in Fig 4. To tackle this problem that new unseen entities from new BIM models may not have embeddings, a contextual comparison mechanism is developed as follows.

Proxy entities are selected for the new unseen entities from the new BIM model to be audited. For a new unseen entity, a proxy entity is an entity with high semantic similarities in the standard BIM-KG where each entity has an embedding trained by the improved TransR. The embedding of the proxy entity then serves as the embedding of the new unseen one for scoring the new triple in subsequent steps. The semantic similarities between the proxy entity and the new unseen entity are different for different kinds of entities. For new element entities (e.g., *slab_567*), the proxies should share similar semantic properties and spatial positions. For new attribute entities, the proxies should belong to similar elements. For example, the proxy for the new entity *slope* in a new triple *<beam_123, has_type, slope>* should be one of the beam types such as *horizontal* rather than *vertical* which is a column type. To achieve this, a mechanism based on contextual information comparison is developed to obtain appropriate proxy entities. As shown in Fig 10, if the unseen new entity is an element entity (e.g., *slab_567*), the entities from the standard BIM-KG that share the same relation, category, and level act as candidate proxy entities. Otherwise, if it is an attribute entity (e.g., *slope*), the candidate proxy entities are those with the same relation from the standard BIM-KG. Following this, one-hop neighbors (i.e., the directly connected entities in the KG) of the obtained candidate proxy entities are compared with those of the new entity respectively to derive common neighbor information (e.g., category, property) between this new entity and each of its candidate proxies. Based on such neighbor information, the candidate proxy entity with the most similar common neighbors is selected as the proxy entity. Finally, the embedding of the proxy entity serves as that of the new entity for the subsequent scoring process.

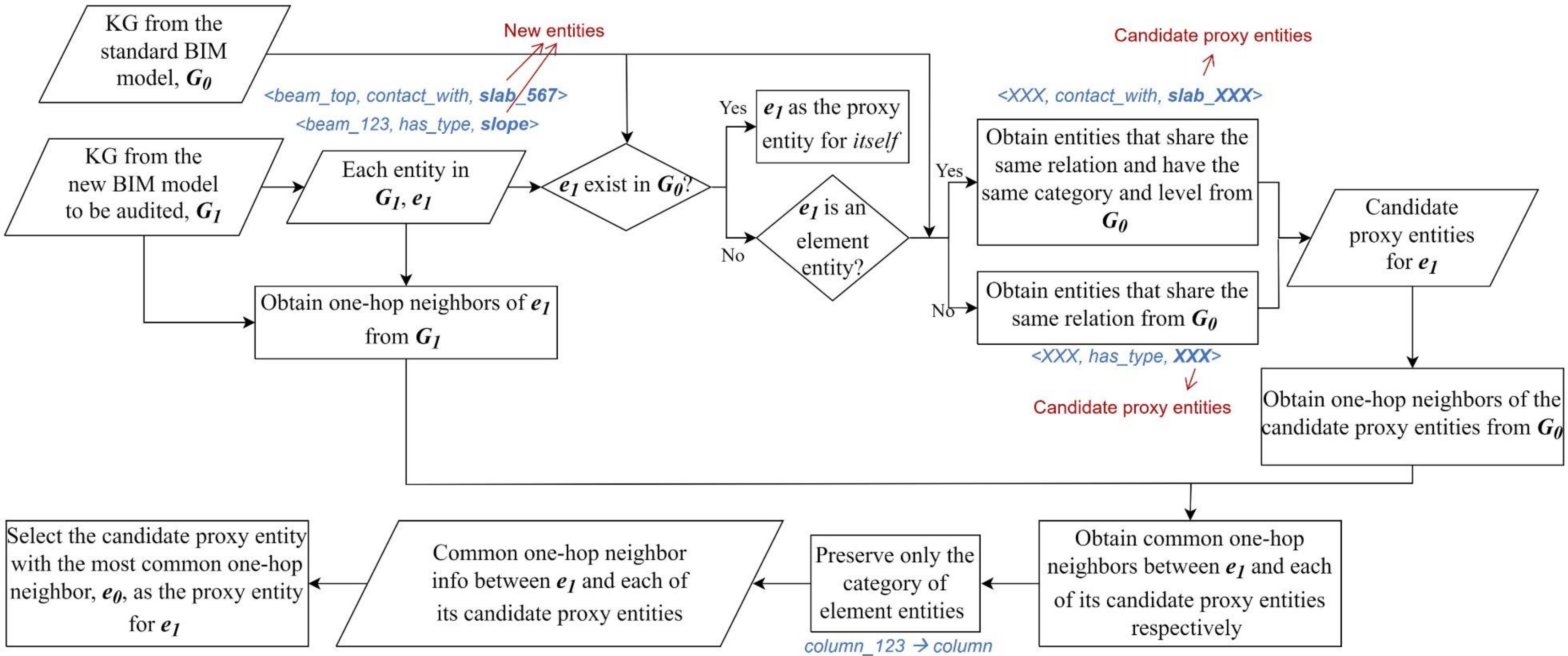


Fig 10. Mechanism to obtain embeddings of new BIM-KG entities

3.3.3. Self-evolving mechanism to determine a proper threshold

Once the embeddings of the unseen entities from the new BIM model to be audited are obtained, the scores of relevant triples can be calculated and compared with the threshold to determine the auditing results (i.e., whether the involved elements in the triples should be accepted or rejected). For example, the score of a new BIM-KG triple *<beam_top, contact_with, slab_234>* from a new BIM model is calculated using the obtained embeddings and then compared with a threshold to decide whether *slab_234* should be rejected or accepted. However, the threshold the scores need to be compared with is unknown. To obtain a proper threshold automatically, a self-evolving mechanism is proposed. It is a process of updating the threshold from a randomly initialized one to a proper one according to the auditing results in iterations over a set of BIM models without human intervention.

As shown in Fig 13, a set of new BIM models to be audited with different mistakes about semantic information and modeling style is used to develop the threshold iteratively. In each epoch (i.e., a complete pass through the entire set of new BIM models), a set of triples from new BIM models are scored using the scoring function shown in Eq. (4). A triple is accepted if its score is greater than the current threshold, meaning that the involving elements in this triple are classified as correct (i.e., the element is modeled in accordance with the specification). Otherwise, the elements are deemed as mistakes and rejected for modifications. Based on the classification results, the confusion matrix shown in Fig 11 is derived.

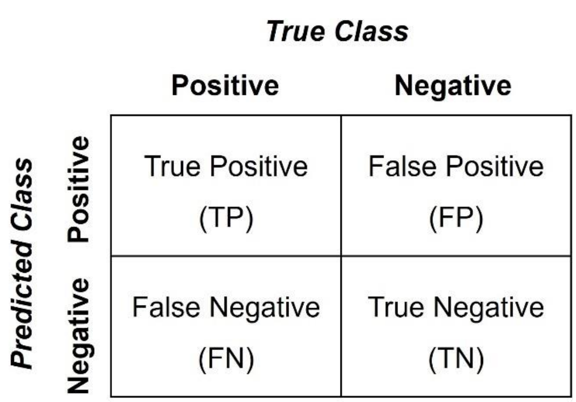


Fig 11. Confusion matrix

Then, the sensitivity that measures how many truly mistake elements are classified as mistakes and the specificity that measures how many truly correct elements are classified as correct are calculated according to Eqs. (8) - (9). Subsequently, the threshold is iteratively updated with respect to Eq. (10), as follows:

$$Sensitivity = TP/(TP + FN) \tag{8}$$

$$Specificity = TN/(TN + FP) \tag{9}$$

$$\theta = \theta_0 + \lambda(f(h, r, t)_{max} - f(h, r, t)_{min}) - \gamma(f(h, r, t)_{max} - f(h, r, t)_{min}) \tag{10}$$

$$\lambda = \lambda + \lambda_0(1 - sensitivity) \tag{11}$$

$$\gamma = \gamma + \gamma_0(1 - specificity) \tag{12}$$

where θ is the learned threshold and θ_0 is the initial one, λ is the introduced adjusting factor to increase the threshold to filter mistake elements as much as possible, γ is the introduced penalty factor to decrease the threshold to avoid reporting correct elements as mistake ones, and $f(h, r, t)_{max}$ and $f(h, r, t)_{min}$ are the maximum and minimum scores of the triples from the standard BIM model, respectively. λ and γ are also obtained iteratively and dynamically according to Eqs. (11) – (12), where λ_0 and γ_0 are the initial values, respectively.

Finally, if the average sensitivity and specificity over the BIM models in the development set meet certain criterions, the average threshold is regarded as the final one. Otherwise, another epoch is conducted to further optimize the threshold until the performance criterions (i.e., sensitivity and specificity) are satisfied or the number of epochs reaches the limit. In this mechanism, the criteria for sensitivity (i.e., 0.95) is stricter than that for specificity (i.e., 0.9) because it is more important to find as many mistake elements as possible for model auditing problems.

Through this self-evolving mechanism, a threshold to filter out mistake elements is obtained automatically. For the new BIM model to be audited, corresponding new BIM-KG triples are generated according to the transformation mechanism provided in Section 3.2 and their embeddings are obtained through the contextual comparison described in Section 3.3.2. Then, the Euclidean distance-based score function shown in Eq. (1) takes the embeddings of the new BIM-KG triples to calculate their scores. As described in Section 3.3.1, given that the embeddings will favor high scores for positive triples (i.e., BIM-KG triples with correct elements) and low scores for negative triples (i.e., BIM-KG triples with mistake elements), the scores are compared with the obtained threshold to decide the auditing results. More specifically, if the score of a new BIM-KG triple is lower than the threshold, the element involved in the triple is deemed as a mistake element and thus is rejected. Otherwise, the element is regarded as a correct one and is accepted. Fig 12 shows an example of how the BIM-KG embeddings help to identify mistake elements with the threshold as the criterion. Three embeddings $\overrightarrow{slab\_side}$ ($\vec{h}$), $\overrightarrow{contact\_with}$ ($\vec{r}$), and $\overrightarrow{beam\_350477}$ ($\vec{t}$) are obtained for the triple <*slab_side*, *contact_with*, *beam_350477*>. The score function takes the embeddings to calculate a score for the triple. In this example, *beam_350477* is a mistake element modeled inappropriately according to the BIM OIR for QTO [31] (i.e., it wrongly takes the precedence over slabs). As shown in Fig 12, the triple score is lower than the threshold. Given that BIM-KG triples with mistake patterns are supposed to obtain low scores (i.e., lower than the threshold), the element, *beam_350477*, in the triple is deemed as a mistake one and is rejected for further modifications.

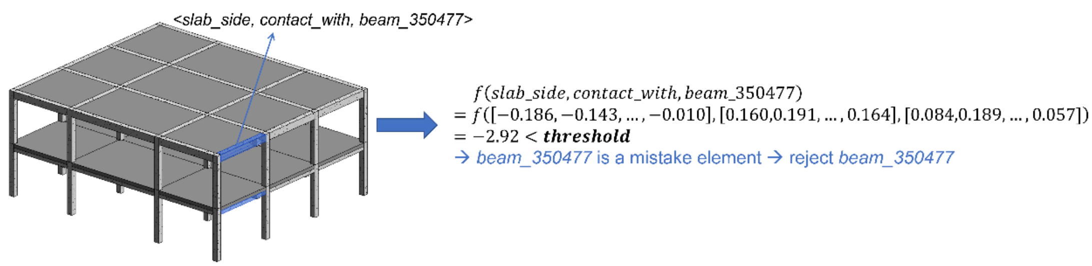


Fig 12. Examples of scoring BIM-KG triples with the threshold as the criterion to identify mistake elements

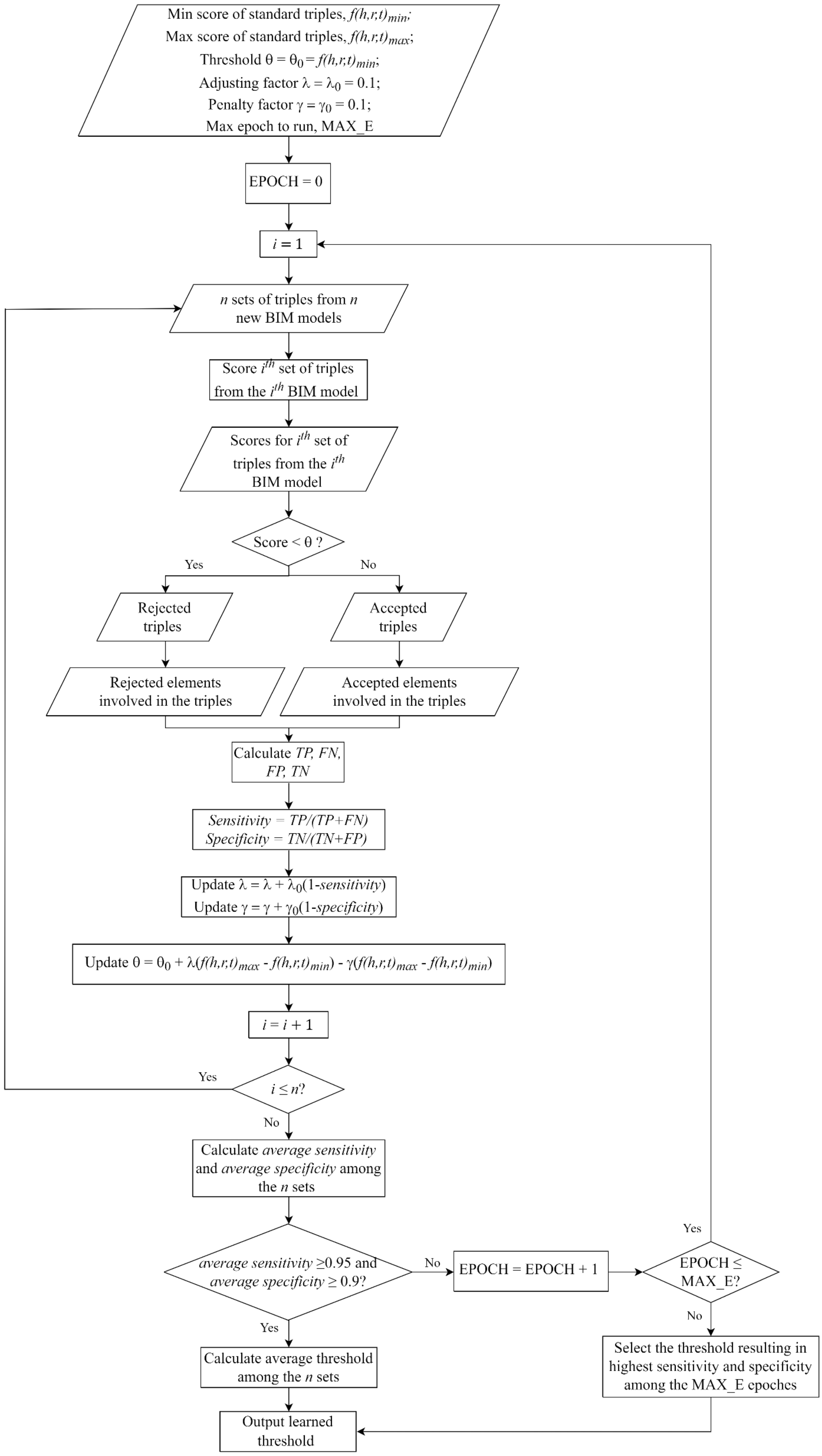


Fig 13. Mechanism to determine a proper threshold

4. ILLUSTRATIVE EXAMPLES

The proposed framework has been validated through identifying different kinds of mistake elements about semantic information and modeling style based on real BIM projects. The BIM models presented in this paper were developed based on real-world BIM projects with similar characteristics accordingly, for illustration. The BIM models are created by Autodesk Revit 2021 [57]. Dynamo 2.10 [58] is utilized to develop the prototype program for the mechanism in Section 3.2 in order to perform the transformation from BIM data into BIM-KG triples. The BIM-KG entities and relations are stored in a graph database, namely Neo4j Community Edition 4.3.2 [59]. The improved TransR model in Section 3.3.1 is implemented with Python 3.7.10 and PyKEEN 1.5.1.dev0 [60]. Other BIM modeling auditing mechanisms described in Section 3.3.2 and 3.3.3 are also built with Python 3.7.10. Py2neo 2021.1.5 [61] is used to conduct the interaction (e.g., retrieval of one-hop neighbors) between the Neo4j database and the Python scripts for the auditing mechanisms.

4.1. Configuration of BIM Models

Fig 14 shows 13 BIM models prepared using Autodesk Revit 2021 for illustration purposes. Model A (Fig 14 (a)) is the standard BIM model in accordance with the BIM MIR for QTO [31], meaning that it is created according to the following requirements:

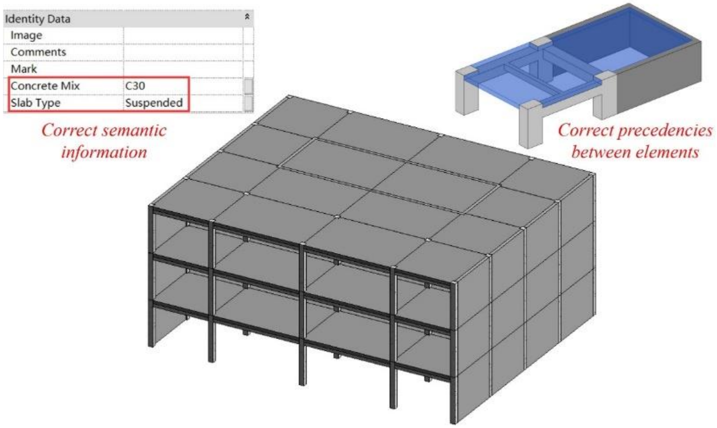
(1) Sufficient semantic information

- The model contains all the necessary semantic attributes such as type and concrete grade for QTO.

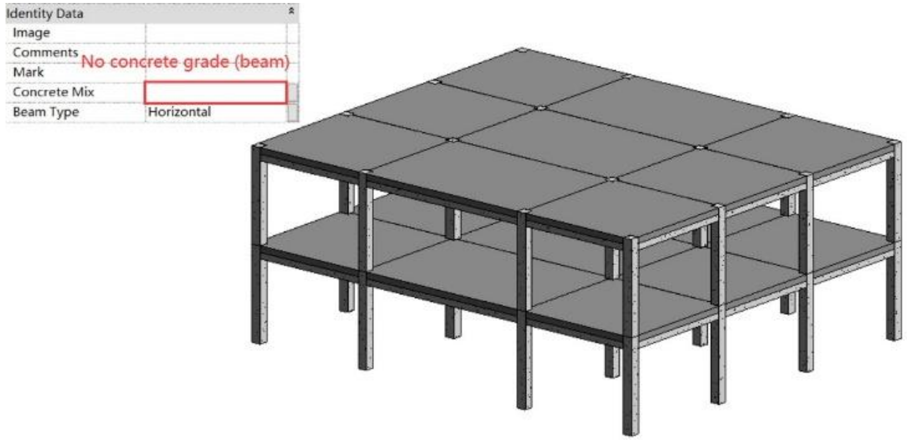
(2) Consistent modeling style

- Vertical elements (i.e., columns and walls) take precedence over horizontal elements (i.e., slabs and beams).
- Slabs take precedence over beams.

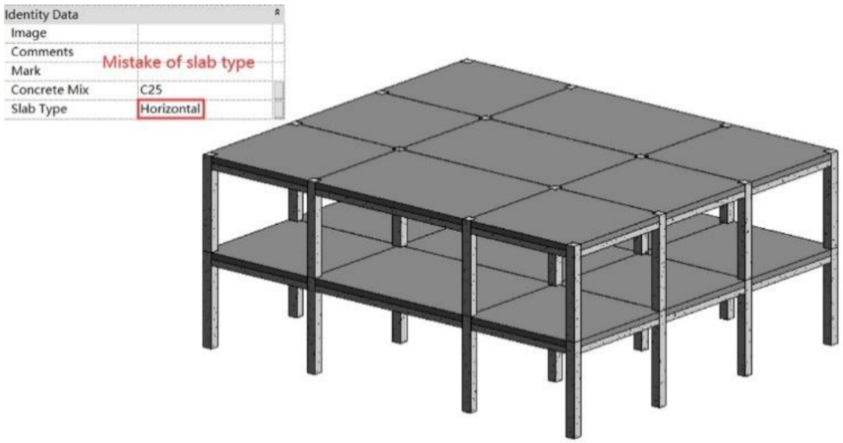
Models B – M are regarded as the new BIM models to be audited. As shown in Fig 14 (b) – (m), they cover common types of building structures (i.e., frame structure, shear wall structure, shear wall-frame structure) and different mistakes (i.e., insufficient semantic information, different inconsistent precedencies between elements). Models B – I are used for the development of a proper threshold, while models J – M are for the testing purpose. Table 2 shows the quantities that are taken off from the illustrative BIM models with different mistakes. They deviate from the baselines (i.e., correct quantities calculated according to the measurement rules). Besides, they are inconsistent and thus are difficult to apply uniform adjustments to correct them. This indicates that it is significant to audit the BIM models to ensure they are prepared consistently according to the requirements so that the quantities can be taken off correctly.



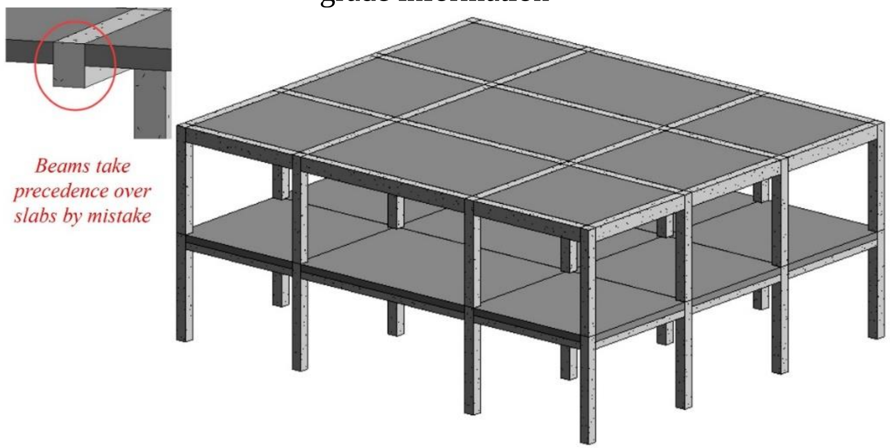
(a) Standard BIM model



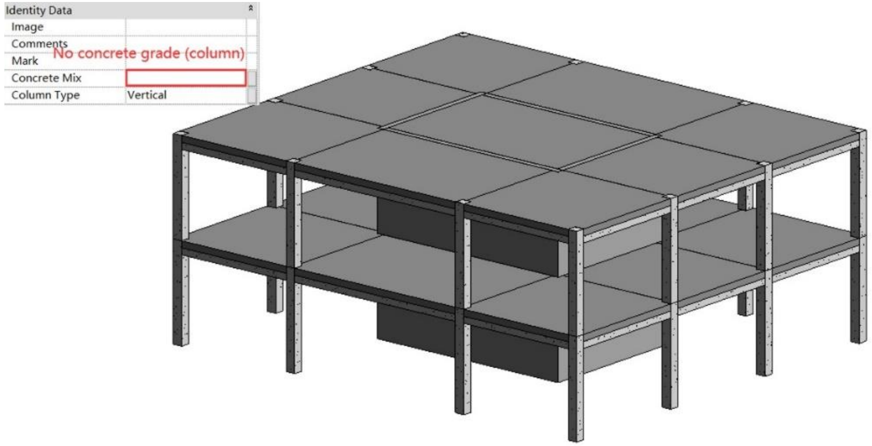
(b) Frame structure type – beams on the second floor lack concrete grade information



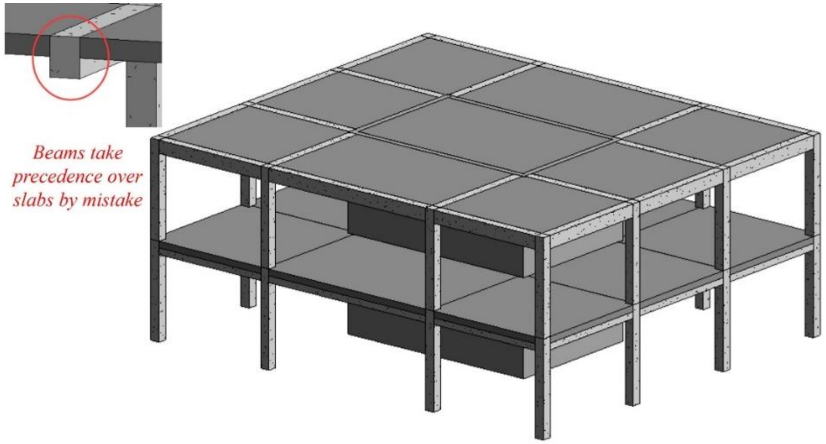
(c) Frame structure type – slabs on the second floor have wrong type information



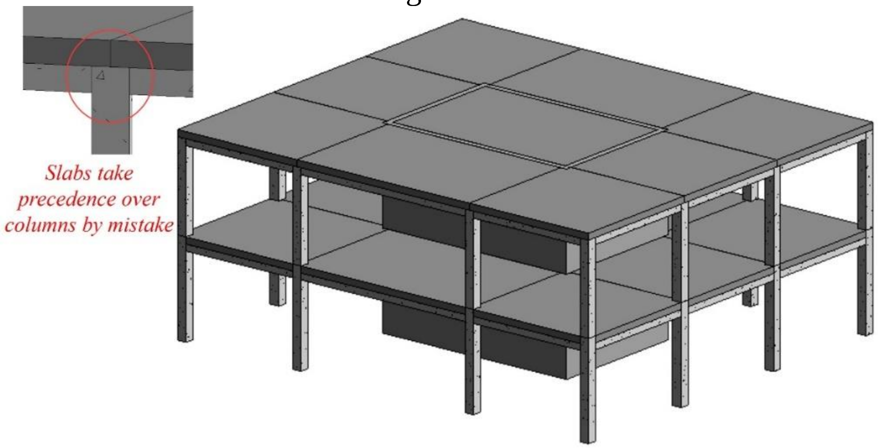
(d) Frame structure type – beams and slabs on the second floor have wrong modeling style



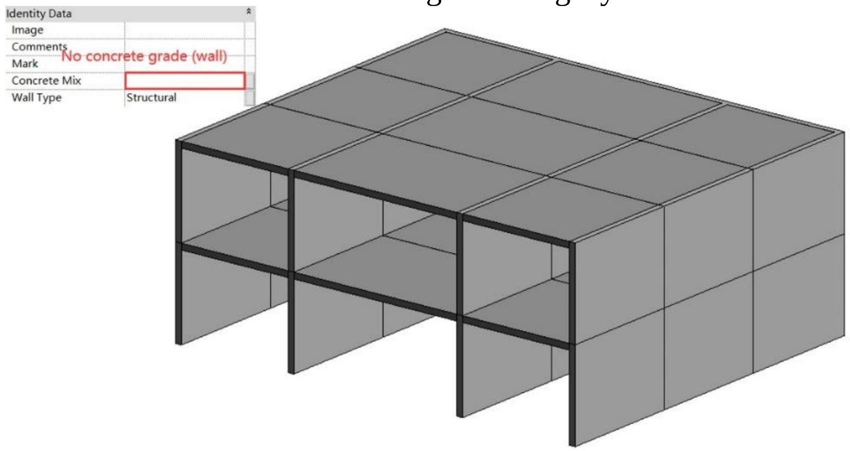
(e) Shear wall-frame structure type – columns on the second floor lack concrete grade information



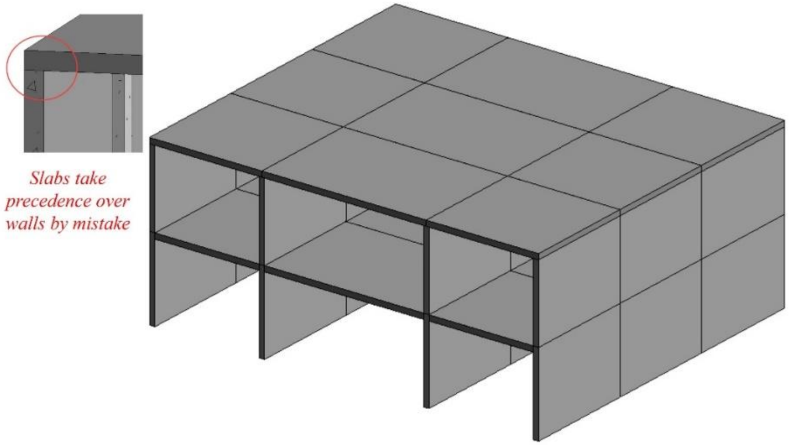
(f) Shear wall-frame structure type – beams and slabs on the second floor have wrong modeling style



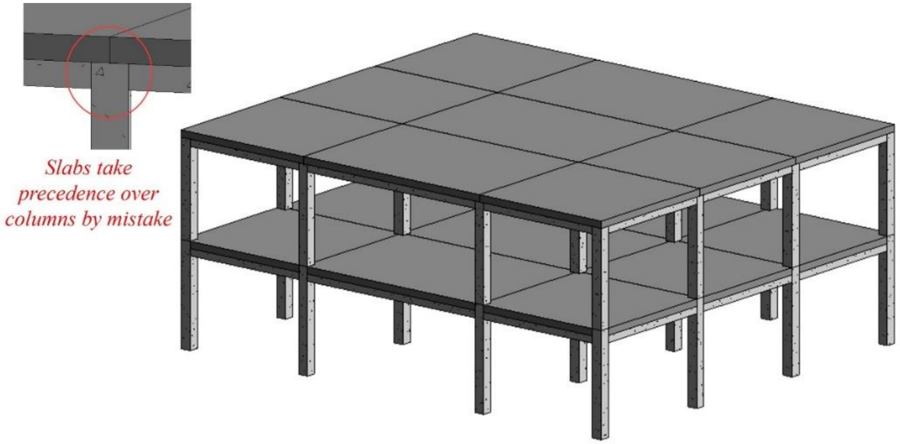
(g) Shear wall-frame structure type – slabs and columns on the second floor have wrong modeling style



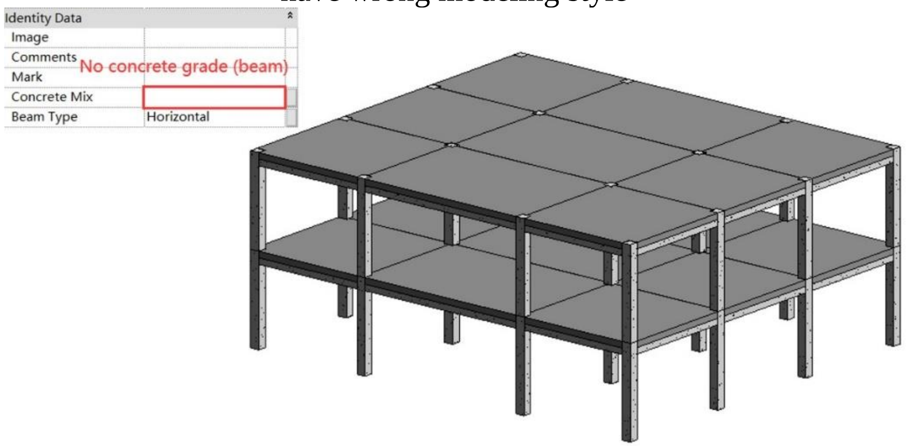
(h) Shear wall structure type – walls on the second floor lack concrete grade information



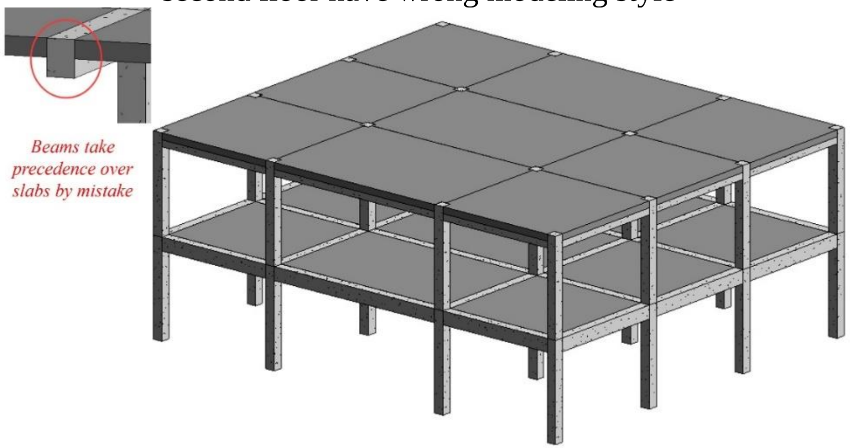
(i) Shear wall structure type – slabs and walls on the second floor have wrong modeling style



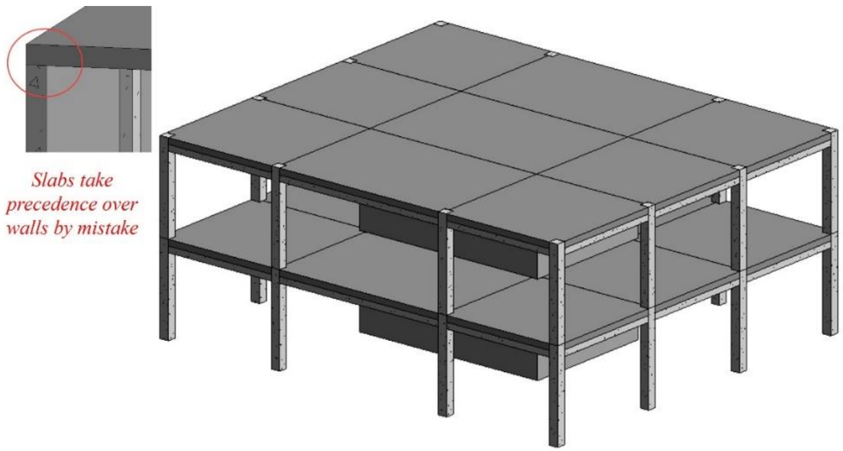
(j) Frame structure type (for testing) – Slabs and columns on the second floor have wrong modeling style



(k) Frame structure type (for testing) – Beams on the first floor lack concrete grade information



(l) Frame structure type (for testing) – Beams and slabs on the first floor have wrong modeling style



(m) Shear wall-frame structure type (for testing) – Slabs and walls on the second floor have wrong modeling style

Fig 14. Configurations of BIM models for the illustration: (a) Model A; (b) Model B; (c) Model C; (d) Model D; (e) Model E; (f) Model F; (g) Model G; (h) Model H; (i) Model I; (j) Model J; (k) Model K; (l) Model L; (m) Model M

Table 2. Quantities of concrete elements in the illustrative BIM models

		Frame structure						Shear-wall frame structure				Shear wall structure	
		Model B	Model C	Model D	Model J	Model K	Model L	Model E	Model F	Model G	Model M	Model H	Model I
Slab	Take-off quantities (m^3)	-*	70.36	64.55	70.65	-*	64.55	-*	65.70	70.58	71.17	-*	71.22
	Baseline quantities (m^3)	70.36						70.36				68.74	
Beam	Take-off quantities (m^3)	-*	11.62	17.42	11.62	-*	17.42	9.31	13.97	9.31	9.31	NA	
	Baseline quantities (m^3)	11.62						9.31					
Column	Take-off quantities (m^3)	8.06	8.06	8.06	7.77	8.06	8.06	-*	6.05	5.83	6.05		
	Baseline quantities (m^3)	8.06						6.05					
Wall	Take-off quantities (m^3)	NA						23.30	23.30	23.30	22.47	-*	66.73
	Baseline quantities (m^3)							23.30				69.20	

Note: assume all the elements have the same concrete grade when calculating the quantities for the models

*: *Cannot take off quantities due to the lack of necessary semantic information (e.g., concrete grade)*

4.2. Automatic Transformation to KG Representations

A program was developed to examine the BIM-KG transformation mechanism in Section 3.2 in Dynamo [61], which enables customized BIM data extraction and processing. As shown in Fig 15, BIM-KG fact triples in the form of <*head*, *relation*, *tail*> are obtained from the standard BIM model automatically. The entities and relations are then stored in the Neo4j [64] database, which is a native and flexible graph data platform. Through this process, BIM models can be automatically transformed into a set of fact triples whose embeddings are trained for the computation of mistake elements. The stored KG representations can be used for data (e.g., one-hop neighbors) retrieval in subsequent steps.

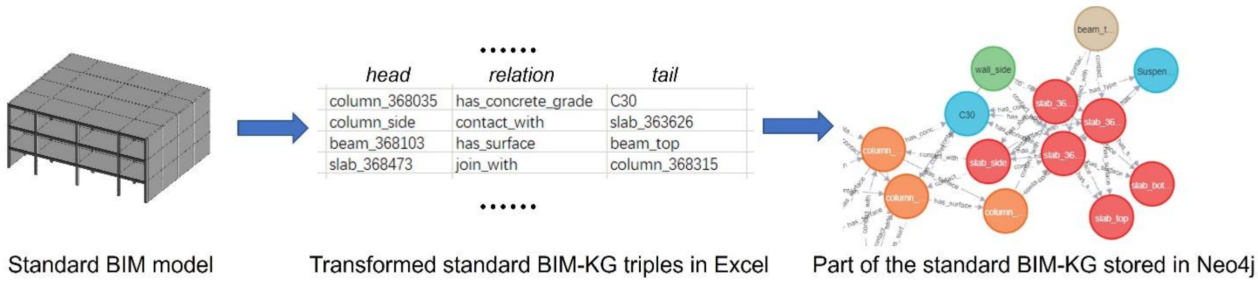


Fig 15. Examples of BIM-KG transformation

4.3. Automatic BIM Model Auditing Based on the KG Representations

4.3.1. Improved TransR model to obtain KG embeddings

The improved TransR model in Section 3.3.1 is utilized to train the standard BIM-KG fact triples to obtain the embeddings of the entities and relations. Fig 16 presents examples of the obtained embeddings.

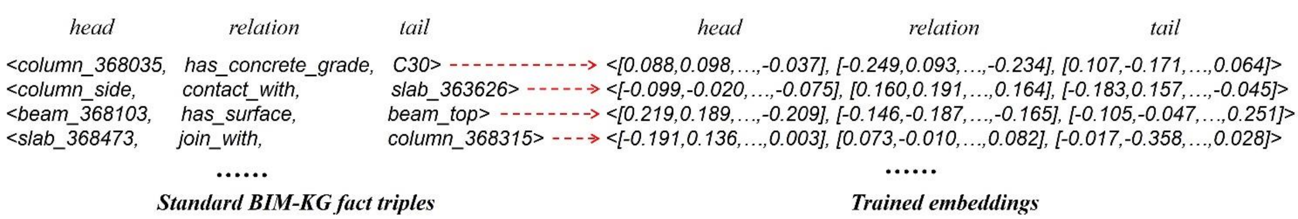


Fig 16. Examples of trained embeddings

For evaluation of knowledge graph embeddings, a common practice is to perform the link prediction task and calculate two metrics, namely Mean Rank and Hits@10. For each fact triple (h, r, t) existing in the KG, h is replaced by every other entity h' in the entity set. The scores (i.e., plausibility) of all the corrupted triples (h', r, t) as well as the original correct one (h, r, t) is calculated using the embeddings and scoring function and ranked in descending order. Such a ranking process is also applicable to the situation where t is replaced. The average rank of the correct fact triples (h, r, t) is the Mean Rank. The proportion of the correct fact triples ranked in the top 10 over all the correct ones is the Hits@10. Eqs. (13) – (14) show the calculations, where R is the set of ranks of all the correct fact triples, r is a rank in R, $|R|$ means the number of the ranks in R, and $[r \leq 10]$ denotes the Boolean calculation (i.e., $[r \leq 10]$ equals 0 if r is greater than 10, otherwise, it is 1). A lower Mean Rank and higher Hits@10 indicate better quality of the obtained embeddings.

Table 3 lists the evaluation metrics of the TransR and proposed modified TransR. The results show that the proposed modified TransR outperforms the baseline model, TransR, consistently, clearly indicating that it has better expressivity and can improve the quality of the trained embeddings.

$$Mean\ Rank = \frac{1}{|R|}\sum_{r \in R} r \tag{13}$$

$$Hits@10 = \frac{1}{|R|}\sum_{r \in R}[r \leq 10] \tag{14}$$

Table 3. Comparison between the TransR and proposed modified TransR

Model	Mean Rank	Hits@10 (%)
TransR	14	68.97
Proposed modified TransR	13	74.37

4.3.2. Contextual comparison to obtain embeddings of new BIM-KG entities

As mentioned in Section 3.3.1, it is necessary to find embeddings for the new entities from BIM models to be audited. In this section, we use Model B shown in Fig 14 (b) to illustrate the process of obtaining proper embeddings for new entities. Similar to Section 4.2, BIM-KG triples, including entities and relations, are obtained from Model B according to the mechanism in Section 3.2. Then, these new entities and relations are also stored in the Neo4j graph database. As described in Section 3.3.2 (Fig 10), candidate proxy entities are obtained first for each new entity. For instance, the new entity *column_350358*

(Fig 17) from the second floor in the triple <*slab_side*, *contact_with*, *column_350358*> considers the column entities that are in the same level and in contact with other entities as well (i.e., connected by the *contact_with* relation) as its candidate proxy entities. Fig 18 shows five examples. Following this, one-hop neighbors of these candidate proxy entities and the new entity are retrieved from the BIM-KGs of Model B and the standard BIM model, respectively. Then, they are compared to find the common one-hop neighbor information, as illustrated in Fig 18. The common one-hop neighbor information indicates the semantic similarities described in Section 3.3.2 between the candidate proxy entities and the new entity. For example, as shown in Fig 6 and Fig 18, such information on an element entity reveals the common semantic properties (e.g., concrete grade such as *C40*, type such as *vertical*) and common categories of joining and contacting elements nearby that describe the spatial position (e.g., corner, edge, middle). Therefore, in Fig 18, the new entity *column_350358* selects *column_36065* or *column_358067* as its proxy from the candidates since they share the most common semantic properties (i.e., *C40*, *vertical*) and spatial positions (i.e., edge) that are derived from the number of common categories of joining and contacting elements. Through such a contextual comparison based on one-hop neighbor information between two BIM-KGs, the proxy entity that has the most semantic similarity with the new one is derived from the standard BIM model. Its embedding is then used as the embedding of the new entity for calculating the scores of relevant triples. Afterwards, the calculated scores are utilized to identify mistake elements in the next step.

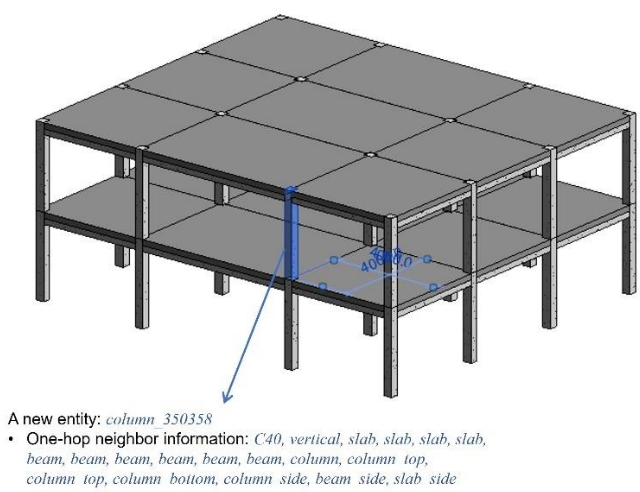


Fig 17. A new entity in Model B and its one-hop neighbor information retrieved from the BIM-KG of Model B

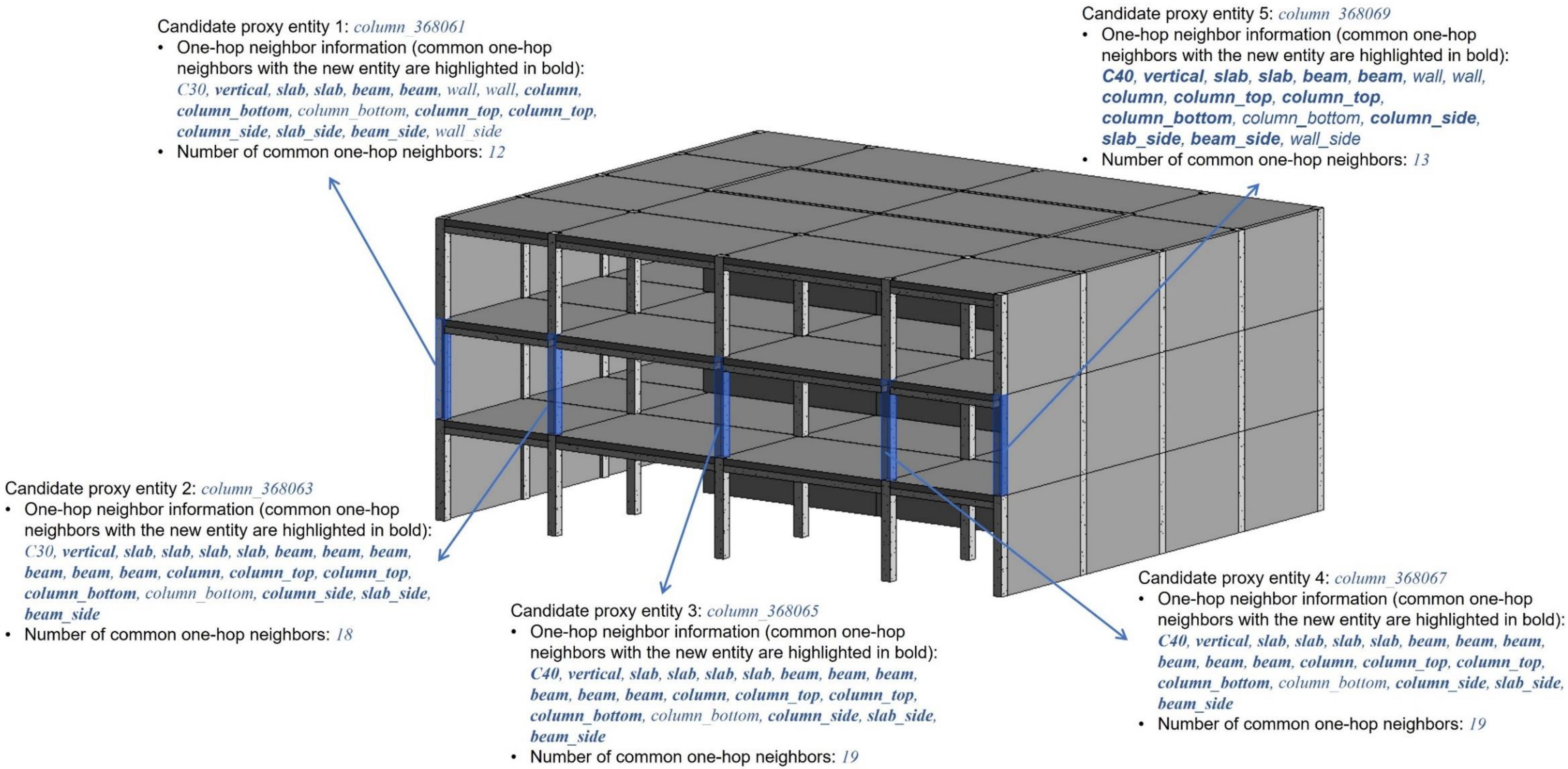


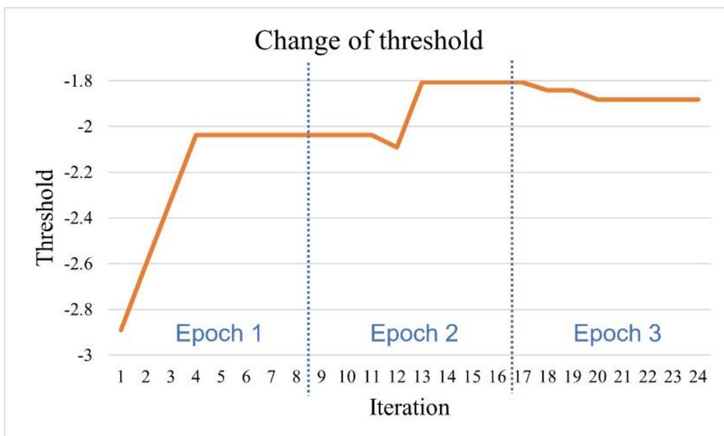
Fig 18. Examples of one-hop neighbor comparison between the new entity in the new BIM model and its candidate proxy entities in the standard BIM model

4.3.3. Self-evolving mechanism to determine a proper threshold

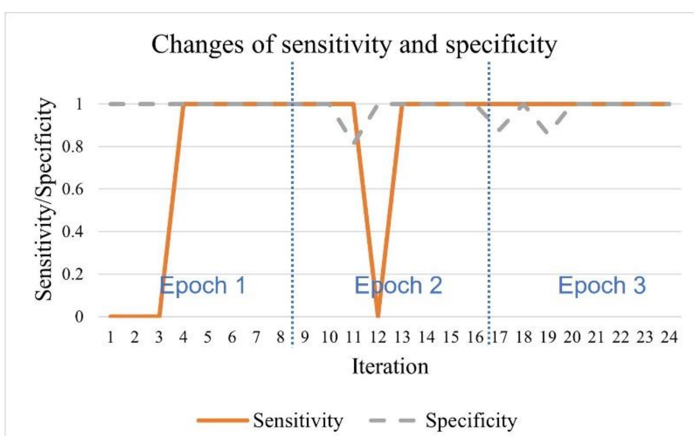
Once the embeddings of the entities and relations from the new BIM models to be audited are obtained, the scores of relevant triples are calculated using the scoring function in Eq. (1). They are then compared with a threshold to derive auditing results, as shown in Fig 12. To facilitate such a scoring and comparing process, Models B – I are utilized to learn a proper threshold iteratively according to the mechanism described in Section 3.3.3. Note that the threshold is a constant learned from a set of BIM models (i.e., Models B – I) with different mistakes about semantic information (i.e., absent or inappropriate semantic attributes) and modeling style (i.e., inconsistent topological arrangements between elements). New different BIM models with similar mistakes on the two aspects are compared with the constant threshold to determine mistake elements. Fig 19 presents the self-evolving processes of the threshold as well as other parameters in Fig 13. Each epoch iterates all the models B – I with a gradually changing threshold to filter mistake elements. At the beginning of the first epoch, the low initial threshold results in 0% sensitivity and 100% specificity, meaning that all the elements are classified as correct (i.e., the threshold is too low so that no mistake elements are identified). Consequently, the adjusting factor is increased to raise the threshold, which gradually increases the sensitivity. Once the threshold is so high that the specificity drops, the penalty factor is raised to lower the threshold so as to avoid the case where the correct elements are misclassified as mistake ones. Finally, if the sensitivity and specificity reach the plateau and satisfy the criterions in Fig 13, the adjusting and penalty factors are stabilized, leading to a converged threshold, which is -1.872. Then, the comparison between the triple score and the threshold shown in Fig 12 can be undertaken to identify mistake elements.

Four unseen models J – M (Fig 14 (j) – (m)) with different kinds of mistake elements are used to for evaluation. Similarly, BIM-KG triples are first obtained from these testing models. The embeddings for the new entities and relations are derived and then utilized to compute triple scores, which are compared with the learned threshold (i.e., 1.872) to classify the mistake and correct elements. As shown in Table 4, all the mistake elements in the four testing models are identified successfully with the reasons aligned with the mistakes described in Fig 14 (j) – (m). Table 5 shows the performance metrics of the classifications. 100% sensitivity is achieved in all the testing BIM models, indicating that all the mistake elements are identified successfully. In addition, the proposed method consistently provides high specificities, meaning that only a few correct elements are erroneously recognized as mistake elements. This suggests that BIM modelers can be effectively informed of all the mistake elements that they are most concerned about. The few correct elements that are identified as mistake elements require little additional effort to be excluded during the model modification process.

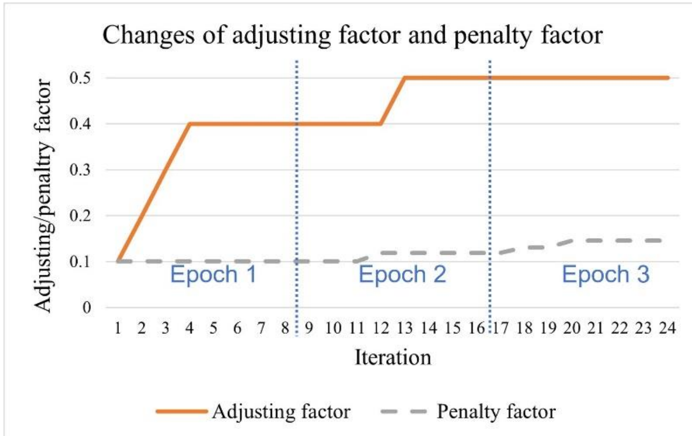
After the mistake elements are identified with respect to the proper threshold, they are rejected. The BIM model is then sent back to the BIM modeler with the list of reject elements for them to edit until no element is identified as a mistake. As a result, it can be ensured that all the elements are modeled consistently according to the modelling requirements in the specification.



(a)



(b)



(c)

Fig 19. Changes of parameters in the self-evolving mechanism to determine the threshold: (a) Changes of threshold; (b) Changes of sensitivity and specificity; (c) Changes of adjusting factor and penalty factor

Table 4. Overall auditing results of the testing models

	Mistake elements	All are identified successfully?	Reason
Model J	9 slabs; 16 columns	Yes	9 slabs take precedence over 16 columns by mistake
Model K	24 beams	Yes	24 beams lack concrete grade information
Model L	9 slabs; 24 beams	Yes	24 beams take precedence over 9 slabs by mistake
Model M	9 slabs; 4 walls	Yes	9 slabs take precedence over 4 walls by mistake

Table 5. Performance metrics of the testing models

	Model J	Model K	Model L	Model M	Average
Sensitivity	100%	100%	100%	100%	100%
Specificity	100%	100%	88%	90%	94%

5. CONCLUSIONS

In this paper, the information requirements of BIM model auditing for QTO purposes are identified from QTO-oriented BIM modeling specifications in order to establish a BIM-KG data model to represent unstructured BIM data (including properties and interrelationships) explicitly, based on which BIM-KG triples are transformed from BIM models automatically. An improved knowledge graph embedding model is proposed to translate the BIM-KG representations into computable embeddings. Then, auditing mechanisms, including deriving embeddings for new BIM-KG entities and obtaining an appropriate threshold iteratively, are developed to utilize these embeddings for automatic mapping and inferences of auditing results without human intervention. The framework is applied to 13 BIM models for illustration. The results validate the effectiveness of the approaches through automatically and successfully identifying mistake elements in BIM models with different kinds of errors regarding semantic information and modeling style. In addition to the presented mistakes, the proposed framework is also applicable to other BIM modeling issues which originate from the absence of semantic information or inappropriate topological arrangements between elements. For example, inconsistent installation sequences of interior material elements in BIM models (e.g., a gypsum board may be modeled after the floor heating system or the expanded polystyrene in a BIM model, which causes quantity deviations in the areas of the gypsum board) [14] can also be detected since such issues may arise from inconsistent topological arrangements between elements. Overall, this study contributes to the following:

- The proposed framework utilizes BIM models as training sources to obtain computable embeddings so that the underlying patterns among BIM data can be captured. Such a BIM-based data-driven manner enables automatic and efficient identification of mistake elements without human intervention. To the best knowledge of the authors, this is pioneering research in BIM-based data-driven model auditing for QTO that can greatly reduce the required human efforts on manual inspection or development of rule patterns.
- This research brings insights on how to improve the efficiency of auditing BIM models for QTO in a fundamental way, through BIM data representation (i.e., the design of the BIM-KG representation and transformation mechanisms that can preserve both object properties and interrelationships explicitly) and manipulation (i.e., the development of the BIM-KG utilization mechanisms that manipulate the transformed BIM data to achieve the QTO-oriented model auditing purposes). The basic principles are generalizable to future studies on this problem and thus this research provides a reliable foundation.

- The modified TransR model provides better expressivity in terms of the imbalance of relations in BIM-KG representations, and outperforms the baseline model (i.e., TransR). The modified TransR can hence output better embeddings to support downstream manipulations such as the BIM-KG triple scoring.
- The proposed framework for BIM model auditing is generic and can be applied for not only QTO purposes but also other applications or projects where BIM models need to satisfy different but consistent requirements on semantic information and modeling style, e.g., standard modeling approaches incorporating structural design concepts [27].

However, there are certain limitations as follows. This study mainly focuses on typical building elements of concrete structures and the mistakes about semantic information (i.e., absent or inappropriate semantic attributes) and modeling style (i.e., inconsistent topological arrangements between elements) in BIM models. Besides, the framework is developed across different platforms and scripting languages, which may make it difficult for domain engineers to grasp. Therefore, future works include: (1) considering more types of building elements and structures, as well as BIM modeling mistakes, to make the framework more comprehensive; (2) developing a more user-friendly one-stop interface integrating different components in the framework to facilitate the usage of it in domain engineers.

REFERENCES

[1] Z. Shen, R.R.A. Issa, Quantitative evaluation of the BIM-assisted construction detailed cost estimates, Journal of Information Technology in Construction (ITcon). 15 (2010) pp. 234–257. https://doi.org/10/18.

[2] S. Aram, C. Eastman, R. Sacks, A knowledge-based framework for quantity takeoff and cost estimation in the AEC industry using BIM, in: The 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC), Sydney, Australia, 2014: pp. 434–442. https://doi.org/10.22260/ISARC2014/0058.

[3] C. Khosakitchalert, N. Yabuki, T. Fukuda, Improving the accuracy of BIM-based quantity takeoff for compound elements, Automation in Construction. 106 (2019) pp. 102891. https://doi.org/10.1016/j.autcon.2019.102891.

[4] L. Holm, J. E. Schaufelberger, D. Griffin, T. Cole, Construction cost estimating: process and practices, 1st Edition, Pearson, 978-0130496652, 2004.

[5] R. Sacks, C. Eastman, G. Lee, P. Teicholz, BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, John Wiley & Sons, 978-1-119-28753-7, 2018.

[6] M. Juszczyk, R. Kozik, A. Leśniak, E. Plebankiewicz, K. Zima, Errors in the Preparation of Design Documentation in Public Procurement in Poland, Procedia Engineering. 85 (2014) pp. 283–292. https://doi.org/10.1016/j.proeng.2014.10.553.

[7] A. Nadeem, A.K.D. Wong, F.K.W. Wong, Bill of Quantities with 3D Views Using Building Information Modeling, Arabian Journal for Science and Engineering. 40 (2015) pp. 2465–2477. https://doi.org/10.1007/s13369-015-1657-2.

[8] T. Akanbi, J. Zhang, Y.-C. Lee, Data-Driven Reverse Engineering Algorithm Development Method for Developing Interoperable Quantity Takeoff Algorithms Using IFC-Based BIM, Journal of Computing in Civil Engineering. 34 (2020) pp. 04020036. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000909.

[9] A. Monteiro, J.P. Martins, A survey on modeling guidelines for quantity takeoff-oriented BIM-based design, Automation in Construction. 35 (2013) pp. 238–253. https://doi.org/10.1016/j.autcon.2013.05.005.

[10] C.E. Firat, D. Arditi, J.-P. Hämäläinen, J. Stenstrand, J. Kiiras, Quantity take-off in model-based systems, in: Proceedings of the 27th CIB W78 International Conference, Cairo, Egypt, 2010: pp. 16–18. https://itc.scix.net/pdfs/w78-2010-112.pdf (accessed April 5, 2021).

[11] K. Zima, Impact of information included in the BIM on preparation of Bill of Quantities, Procedia Engineering. 208 (2017) pp. 203–210. https://doi.org/10.1016/j.proeng.2017.11.039.

[12] Hong Kong Institute of Surveyors, Hong Kong Standard Method of Measurement of Building Works, Fourth Edition, Pace Publishing Limited, 988-98402-4-3, 2005.

[13] H. Liu, J.C.P. Cheng, V.J.L. Gan, S. Zhou, A knowledge model-based BIM framework for automatic code-compliant quantity take-off, Automation in Construction. 133 (2022) pp. 104024. https://doi.org/10.1016/j.autcon.2021.104024.

[14] S. Kim, S. Chin, S. Kwon, A Discrepancy Analysis of BIM-Based Quantity Take-Off for Building Interior Components, Journal of Management in Engineering. 35 (2019) pp. 05019001. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000684.

[15] Construction Industry Council (CIC), CIC BIM Standards - General (Version 2.1 - 2021), (2021). https://www.bim.cic.hk/zh-hant/resources/publications_detail/100 (accessed December 30, 2021).

[16] D. Olsen, J.M. Taylor, Quantity Take-Off Using Building Information Modeling (BIM), and Its Limiting Factors, Procedia Engineering. 196 (2017) pp. 1098–1105. https://doi.org/10.1016/j.proeng.2017.08.067.

[17] G. Lee, R. Sacks, C.M. Eastman, Specifying parametric building object behavior (BOB) for a building information modeling system, Automation in Construction. 15 (2006) pp. 758–776. https://doi.org/10.1016/j.autcon.2005.09.009.

[18] B. Succar, Building information modelling framework: A research and delivery foundation for industry stakeholders, Automation in Construction. 18 (2009) pp. 357–375. https://doi.org/10.1016/j.autcon.2008.10.003.

[19] R. Sacks, M. Girolami, I. Brilakis, Building Information Modelling, Artificial Intelligence and Construction Tech, Developments in the Built Environment. (2020) pp. 100011. https://doi.org/10.1016/j.dibe.2020.100011..

[20] A. Sheth, S. Padhee, A. Gyrard, Knowledge Graphs and Knowledge Networks: The Story in Brief, IEEE Internet Computing. 23 (2019) pp. 67–75. https://doi.org/10.1109/MIC.2019.2928449.

[21] A. Khalili, D.K.H. Chua, IFC-Based Graph Data Model for Topological Queries on Building Elements, Journal of Computing in Civil Engineering. 29 (2015) pp. 04014046. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000331.

[22] A. Ismail, A. Nahar, R. Scherer, Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard, (2017). https://ifcwebserver.org/doc/ifc2gdb_eg-ice2017_ismail.pdf (accessed July 5, 2021).

[23] N. Skandhakumar, F. Salim, J. Reid, R. Drogemuller, E. Dawson, Graph theory based representation of building information models for access control applications, Automation in Construction. 68 (2016) pp. 44–51. https://doi.org/10.1016/j.autcon.2016.04.001.

[24] Y. Hu, D. Castro-Lacouture, C.M. Eastman, S.B. Navathe, Component Change List Prediction for BIM-Based Clash Resolution from a Graph Perspective, Journal of Construction Engineering and Management. 147 (2021) pp. 04021085. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002092.

[25] International Organization for Standardizatio (ISO), ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles, 2018. https://www.iso.org/standard/68078.html (accessed June 20, 2021).

[26] Ministry of Housing and Urban-Rural Development of the People's Republic of China, Standard for Graphic Expression of Building Information Modeling, (2018). http://download.mohurd.gov.cn/bzgg/hybz/JGJT%20448-2018%20%E5%BB%BA%E7%AD%91%E5%B7%A5%E7%A8%8B%E8%AE%BE%E8%AE%A1%E4%BF%A1%E6%81%AF%E6%A8%A1%E5%9E%8B%E5%88%B6%E5%9B%BE%E6%A0%87%E5%87%86.pdf (accessed July 3, 2021).

[27] Hong Kong Housing Authority, Standard Approach of Modelling (SAM) For Creating Building Information Structural model for Development and Construction Division of Hong Kong Housing Authority, (2014). https://www.housingauthority.gov.hk/en/common/pdf/business-partnerships/resources/building-information-modelling/modelling_guidelines.pdf (accessed June 29, 2021).

[28] A. Kiviniemi, M. Rekola, K. Belloni, J. Kojima, T.K. ja T. Mäkeläinen, H. Kulusjärvi, J. Hietanen, Senate Properties: BIM Requirements 2007 Volume 7: Quantity take-off, (2007). https://web.archive.org/web/20120519151455/http://www.senaatti.fi/tiedostot/BIM_2007_Vol_7_Quantity_take-off_R1_0.pdf (accessed June 30, 2021).

[29] A. Kiviniemi, M. Rekola, K. Belloni, J. Kojima, T.K. ja T. Mäkeläinen, H. Kulusjärvi, J. Hietanen, Senate Properties: BIM Requirements 2007 Volume 3: Architectural Design, (2007). https://web.archive.org/web/20120519151206/http://www.senaatti.fi/tiedostot/BIM_2007_Vol3_Architectural_Design.pdf (accessed June 30, 2021).

[30] National Institute of Building Sciences, National BIM Standard - United States® Version 3, (2015). https://www.nationalbimstandard.org/nbims-us (accessed June 25, 2021).

[31] The Hong Kong Institute of Surveyors, BIM Model Information Requirements for Quantity Take-off, Pre-publishing, (2021).

[32] Solibri, Solibri Model Checker, 2021. https://www.solibri.com/ (accessed June 30, 2021).

[33] Autodesk, Autodesk Model Checker for Revit, 2021. https://interoperability.autodesk.com/modelchecker.php (accessed June 29, 2021).

[34] P.A. Zadeh, G. Wang, H.B. Cavka, S. Staub-French, R. Pottinger, Information Quality Assessment for Facility Management, Advanced Engineering Informatics. 33 (2017) pp. 181–205. https://doi.org/10.1016/j.aei.2017.06.003.

[35] H.B. Cavka, S. Staub-French, E.A. Poirier, Levels of BIM compliance for model handover, Journal of Information Technology in Construction (ITcon). 23 (2018) pp. 243–258. http://www.itcon.org/paper/2018/12.

[36] E. Gholami, A. Kiviniemi, S. Sharples, Implementing Building Information Modelling (BIM) in Energy-Efficient Domestic Retrofit: Quality Checking of BIM Model, in: Eindhoven, The Netherlands, 2015: pp. 235–244. https://itc.scix.net/pdfs/w78-2015-paper-024.pdf (accessed April 28, 2021).

[37] V. Donato, M.L. Turco, M.M. Bocconcino, BIM-QA/QC in the architectural design process, Architectural Engineering and Design Management. 14 (2018) pp. 239–254. https://doi.org/10.1080/17452007.2017.1370995.

[38] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, IEEE Transactions on Neural Networks and Learning Systems. (2021) pp. 1–21. https://doi.org/10.1109/TNNLS.2021.3070843.

[39] F.N. Stokman, P.H. de Vries, Structuring Knowledge in a Graph, in: G.C. van der Veer, G. Mulder (Eds.), Human-Computer Interaction, Springer, Berlin, Heidelberg, 1988: pp. 186–206. https://doi.org/10.1007/978-3-642-73402-1_12.

[40] A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge bases, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI Press, San Francisco, California, 2011: pp. 301–306. https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/viewFile/3659/3898 (accessed July 10, 2021).

[41] A. Nahar, Applying graph theory concepts for analyzing BIM models based on IFC standards, Master Thesis, Technische Universität Dresden, 2017. https://tu-dresden.de/bu/bauingenieurwesen/cib/ressourcen/dateien/publikationen/Projekt-_Diplomarbeiten/Masterarbeit_Ahmed_Nahar_2017.pdf?lang=en (accessed May 26, 2021).

[42] Z. Pan, C. Su, Y. Deng, J. Cheng, Video2Entities: A computer vision-based entity extraction framework for updating the architecture, engineering and construction industry knowledge graphs, Automation in Construction. 125 (2021) pp. 103617. https://doi.org/10.1016/j.autcon.2021.103617.

[43] C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, X. Wang, Developing a hybrid approach to extract constraints related information for constraint management, Automation in Construction. 124 (2021) 103563. https://doi.org/10.1016/j.autcon.2021. pp. 103563.

[44] Z. Wang, R. Sacks, T. Yeung, Exploring graph neural networks for semantic enrichment: Room type classification, Automation in Construction. (2021) pp. 104039. https://doi.org/10.1016/j.autcon.2021.104039.

[45] X.-Y. Wang, Y. Yang, K. Zhang, Customization and generation of floor plans based on graph transformations, Automation in Construction. 94 (2018) pp. 405–416. https://doi.org/10.1016/j.autcon.2018.07.017.

[46] V.J.L. Gan, BIM-based graph data model for automatic generative design of modular buildings, Automation in Construction. 134 (2022) pp. 104062. https://doi.org/10.1016/j.autcon.2021.104062.

[47] B. Strug, E. Grabska, G. Ślusarczyk, Supporting the design process with hypergraph genetic operators, Advanced Engineering Informatics. 28 (2014) pp. 11–27. https://doi.org/10.1016/j.aei.2013.10.002.

[48] P. Pauwels, T.M. de Farias, C. Zhang, A. Roxin, J. Beetz, J. De Roo, C. Nicolle, A performance benchmark over semantic rule checking approaches in construction industry, Advanced Engineering Informatics. 33 (2017) pp. 68–88. https://doi.org/10.1016/j.aei.2017.05.001.

[49] B. Zhong, C. Gan, H. Luo, X. Xing, Ontology-based framework for building environmental monitoring and compliance checking under BIM environment, Building and Environment. 141 (2018) pp. 127–142. https://doi.org/10.1016/j.buildenv.2018.05.046.

[50] L. Jiang, J. Shi, C. Wang, Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning, Advanced Engineering Informatics. 51 (2022) pp. 101449. https://doi.org/10.1016/j.aei.2021.101449.

[51] buildingSMART, IFC4_ADD2_TC1 - 4.0.2.1 [Official], (2020). https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ (accessed August 17, 2021).

[52] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, IEEE Transactions on Knowledge and Data Engineering. 29 (2017) pp. 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499.

[53] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, Austin, Texas, 2015: pp. 2181–2187.

[54] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework, IEEE Transactions on Pattern Analysis and Machine Intelligence. PP (Accepted/In press). https://doi.org/10.1109/TPAMI.2021.3124805.

[55] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th

International Conference on Neural Information Processing Systems - Volume 2, Curran Associates Inc., Red Hook, NY, USA, 2013: pp. 2787–2795.
[56] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec, Canada, 2014: pp. 1112–1119. https://ojs.aaai.org/index.php/AAAI/article/view/8870 (accessed September 25, 2021).
[57] Autodesk Revit 2021, Multidisciplinary BIM software for higher-quality, coordinated designs, 2021. https://www.autodesk.com.hk/products/revit/overview (accessed June 10, 2021).
[58] Dynamo, Open source graphical programming for design, 2020. https://dynamobim.org/ (accessed January 10, 2021).
[59] Neo4j, Neo4j: Graph Database Platform, 2021. https://neo4j.com/ (accessed June 29, 2021).
[60] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, J. Lehmann, PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings, Journal of Machine Learning Research. 22 (2021) pp. 1–6. http://jmlr.org/papers/v22/20-825.html.
[61] Py2neo, The Py2neo Handbook, 2021. https://py2neo.org/2021.1/ (accessed July 16, 2021).