Design knowledge graph-aided conceptual product design approach based on joint entity and relation extraction

Huang, Yuexin; Yu, Suihuai; Chu, Jianjie; Su, Zhaojing; Zhu, Yaokang; Wang, Hanyu; Wang, Mengcheng; Fan, Hao

DOI
10.3233/JIFS-223100

Publication date
2023

Document Version
Final published version

Published in
Journal of Intelligent and Fuzzy Systems

Journal of Intelligent & Fuzzy Systems 44 (2023) 5333–5355
DOI:10.3233/JIFS-223100
IOS Press

Design knowledge graph-aided conceptual product design approach based on joint entity and relation extraction

Yuexin Huang[a,b,*], Suihuai Yu[a], Jianjie Chu[a], Zhaojing Su[c], Yaokang Zhu[d], Hanyu Wang[a], Mengcheng Wang[a] and Hao Fan[a]
[a]*Key Laboratory of Industrial Design and Ergonomics, Ministry of Industry and Information Technology, Northwestern Polytechnical University, Xi'an, China*
[b]*School of Industrial Design Engineering, Delft University of Technology, Delft, The Netherlands*
[c]*Department of Industrial Design, College of Arts, Shandong University of Science and Technology, Tsingtao, China*
[d]*School of Computer Science and Technology, East China Normal University, Shanghai, China*

Abstract. Design knowledge is critical to creating ideas in the conceptual design stage of product development for innovation. Fragmentary design data, massive multidisciplinary knowledge call for the development of a novel knowledge acquisition approach for conceptual product design. This study proposes a Design Knowledge Graph-aided (DKG-aided) conceptual product design approach for knowledge acquisition and design process improvement. The DKG framework uses a deep-learning algorithm to discover design-related knowledge from massive fragmentary data and constructs a knowledge graph for conceptual product design. The joint entity and relation extraction model is proposed to automatically extract design knowledge from massive unstructured data. The feasibility and high accuracy of the proposed design knowledge extraction model were demonstrated with experimental comparisons and the validation of the DKG in the case study of conceptual product design inspired by massive real data of porcelain.

Keywords: Conceptual product design, design knowledge graph, deep learning, knowledge acquisition, joint entity and relation extraction

Abbreviations

DKG	Design Knowledge Graph
XML	Extensible Markup Language
OWL	Web Ontology Language
BiLSTM	Bidirectional LSTM
CRF	conditional random fields
MLM	masked language modelling
Seq2Seq	sequence-to-sequence

*Corresponding author. Yuexin Huang, E-mail: yuexin.huang@outlook.com, ORCID: http://orcid.org/0000-0002-2253-9141.

1. Introduction

Concept product design plays an essential role in innovative product development, and existing studies have shown that concept product design influences about 70–80% of product lifecycle costs [1]. Due to the design process requires a significant amount of design-related knowledge, such as shapes, patterns, and materials, acquiring valuable design information is critical for conceptual product design. However, retrieving data directly from websites is inefficient for professional designers, because information on the Internet is always massive and fragmented.

ISSN 1064-1246/$35.00

To get the required design knowledge from massive fragmentary data, most existing methods focus on designing a framework to manage massive design-related data. For example, some researchers proposed an Extensible Markup Language (XML)-based model to streamline design and manufacturing activities [2]. Besides, ontology [3, 4] and Web Ontology Language (OWL) [5] methods were used to acquire and manage design knowledge from massive online data. The design platforms are constructed manually by experts, and previous design knowledge management methods have fragmentation problems, limited reasoning and insufficient visualisation of data. As a result, these limitations severely restrict the application.

Knowledge graph is a semantic network that connects different data using entities, relations and attributes [6]. It has been used widely in digital industrial products and services, intelligent education, digital clinical database [7–9] and other areas. In the knowledge graph-based knowledge acquisition methods for product development, Hao et al. [10] proposed a knowledge graph-based integration and navigation method for engineering design decision knowledge. Zhang et al. [11] presented a framework for the design knowledge representation based on the knowledge graph and CBR. Huet et al. [12] built a knowledge graph-based design assistant platform for design knowledge of complex engineering systems. Guo et al. [13, 14] proposed an automatic process decision-making system based on the knowledge graph. Wang et al. [15] presented a knowledge graph-based design requirement elicitation framework in the intelligent product-service system. Li et al. [16] proposed a knowledge graph-aided concept–knowledge approach for product-service system development. The above knowledge graph-based methods solve the fragmentation, limited reasoning and insufficient visualisation of knowledge acquisition for product development. As for the knowledge graph for conceptual product design, some specific tasks, including product attributes acquisition [17], modelling of design workflow [18] and modelling of process execution sequence [19], have been applied knowledge graph to conceptual product design. However, there are few studies on acquiring massive fragmentary design knowledge, such as design aesthetics and creative cultural products. Compared with conventional knowledge acquisition methods for conceptual product design, such as XML [2], ontology [3, 4] and OWL [5], knowledge graph has significant advantages in automation, reasoning and visualisation. Therefore, this study proposes a knowledge graph-aided design knowledge acquisition method for conceptual product design.

According to existing research [20], the key technology of knowledge graph construction is to extract entities and relations from unstructured data. However, there are three challenges for applying knowledge graph to conceptual product design as follows:

(1) Analyse conceptual product design elements and construct a relevant knowledge graph framework. Most knowledge graph research concentrated on the manufacturing area, such as manufacturing documents [21], product attributes acquisition [17], manufacturing-related semantic relations identification [22], and intelligent query system for equipment faults [23]. Other knowledge graph frameworks in different domains cannot be used directly, and there is inadequate research on the application of knowledge graph in concept product design to apply knowledge graph to conceptual product design. Therefore, it is challenging to propose a generic framework to construct knowledge graph to facilitate the conceptual product design process.

(2) Create entity and relationship extraction models compatible with cultural design data. Overlapping entity redundancy scenarios, computational costs (see Section 4 for details) and other design knowledge extraction problems make extracting design knowledge from unstructured data challenging. Entity and relation extraction models that fit the design data characteristics are essential in knowledge graph construction.

(3) The performance of the DKG is limited by specific design tasks. Compared with other fields applying knowledge graph technology, there are unique processes in concept product design, such as design inspiration acquisition, design knowledge comparison, design knowledge management and design implementation. Applying the knowledge graph to the conceptual product design process should solve the existing design problem.

To fulfil the research gap and challenges above, the objective of this study is to propose a Design Knowledge Graph-aided (DKG-aided) conceptual product design approach for acquiring interrelated design knowledge and improving the design process. The specific study objectives are as follows:

(1) A generic and compatible DKG framework is proposed for the conceptual product design process and design knowledge characteristics.
(2) The design knowledge extraction model is designed to extract design-related entities and relations from unstructured data, solve the problem of overlapping entity redundancy scenarios and computational costs, and build the DKG automatically.
(3) Applying the DKG framework to a case study by verifying the real online data to promote the conceptual product design process. Validate that the DKG facilitates the conceptual product design process, including design inspiration acquisition, knowledge comparison and knowledge management.

The remainder of the study is organised as follows: In Section 2, the related works of design knowledge, knowledge graph, and knowledge extraction methods are analysed. In Section 3, the DKG framework for the conceptual product design process is proposed. In Section 4, a joint entity and relation extraction model is designed for the design knowledge extraction mission. In Section 5, results and discussion are presented. Section 6 contains the conclusion and future work.

2. Related works

This section reviews the contributions and limitations of previous methods of design knowledge-aided conceptual product design. The state-of-art knowledge extraction methods are also presented.

2.1. Design knowledge-aided conceptual product design

Design knowledge includes explicit and implicit information to support the entire conceptual product design process. As shown in Fig. 1, to analyse the design knowledge in the conceptual product design, some researchers [24, 25] conducted research on the design knowledge flow in conceptual product design, with data hierarchy, function hierarchy and conceptual design hierarchy. The data hierarchy stores explicit and implicit design knowledge such as modelling, pattern, function and colour. Design knowledge is acquired, represented and stored in the management system in the function hierarchy. Design knowledge evolves from the existing design knowledge, expressed in the next conceptual product design hierarchy to support design practice, collaborative design, idea stimulation and design analysis.

However, design-related data is massive and fragmentary, and the search results of simple keywords

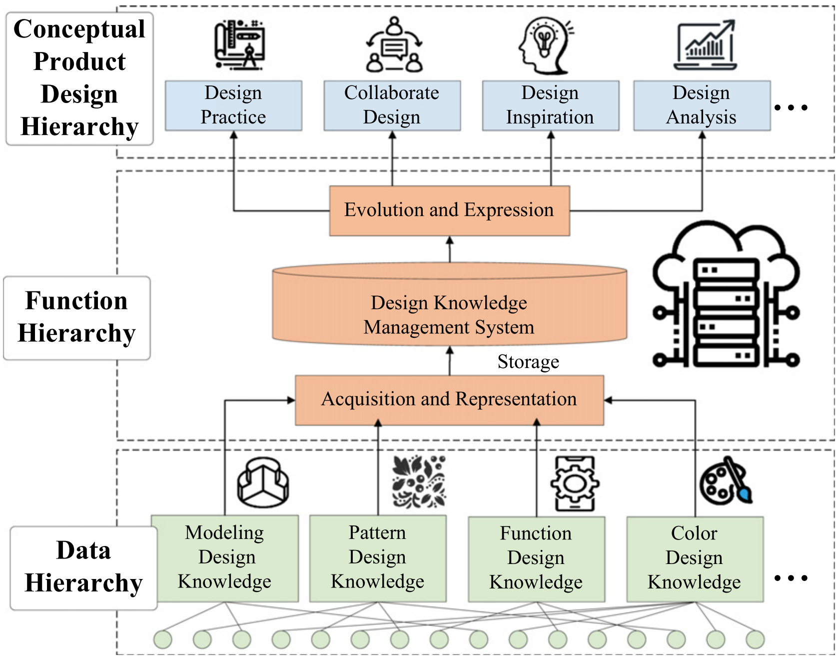

Fig. 1. Design knowledge flow of conceptual product design (summarised from references [24, 25]).

Table 1
Comparison among previous design knowledge-aided conceptual product design methods

Method	Contributions	Disadvantages
XML based-method	proposed a CMSDIM model based on XML to streamline design and manufacturing activities [2]	has no semantic description
Ontology based-method	constructed a semantic modelling system based on ontology and XML topic map that could effectively model the dynamic design knowledge [3]	has limited modularity, reasoning, automation, understandability
	proposed an ontology-based method for knowledge modelling of manufacturing process planning [4]	has limited reasoning, automation, understandability, and visualisation
OWL-based method	proposed an OWL-based knowledge representation model, including functional and aesthetic aspects of design information representation, for supporting conceptual product design [5]	the acquisition of user perceptions and designer intentions relied on carrying out surveys, and the information annotation was a manual process
Knowledge graph-based method	proposed a knowledge graph-based requirement elicitation framework in the intelligent product-service system [15]	only supports the requirement elicitation stage
	proposed a knowledge graph-aided concept–knowledge approach for product-service system development [16]	needs enough amount of knowledge resources, and plenty of time and space is required
	proposed a node2vec-based parameterised representation of geometric elements and assembly constraints approach [20]	the number of nodes in common mechanical products is small, which will affect parameterised representation

are always incorrect. Acquiring valuable design knowledge from such information is inefficient. Moreover, the search results of different web pages are disparate silos of data, and do not know the underlying relationships between other data and how they are interconnected to understand the design knowledge. Researchers have done in-depth work in previous methods of design knowledge-aided conceptual product design, which is summarised in Table 1, including XML ontology, OWL and knowledge graph. For example, Bloomfield et al. [2] proposed an XML-based model to streamline design and manufacturing activities. He et al. [4] presented ontology-based methods for knowledge modelling of process planning. Hu et al. [5] constructed an OWL-based representation model for conceptual product design knowledge, including functional and aesthetic aspects of design information representation.

Although classic design knowledge-aided conceptual product design methods have made great contributions, most of these methods have many obvious disadvantages. For example, the XML method has no semantic description. The ontology and OWL methods have limitations in semantic description, reasoning, automation and visualisation. It is worth mentioning that the knowledge graph provides a novel approach to the semantic representation of design knowledge, while the existing knowledge graph-aided methods only support specific stages in the design process, such as requirement elicitation and digital model representation. Therefore, it is necessary to propose a novel framework with massive high-quality and quantity knowledge data to support the whole conceptual product design process.

The knowledge graph is a semantic network that links different data with entities, relationships and attributes [26]. It has certain irreplaceability in areas such as visualisation of design knowledge [27], modelling of design workflow [18], and modelling of process execution sequence [19]. Nowadays, specific tasks, such as product attributes acquisition [17], design knowledge management [16] and unified process information modelling [20], have been applied knowledge graph to the conceptual product design. However, there are few studies on acquiring multi-discipline design knowledge, such as design aesthetics and creative cultural products, on helping designers conduct design tasks. Hence, the most urgent demand for conceptual product design is constructing a knowledge graph to realise intelligent design knowledge service platforms for designers.

2.2. *State-of-art knowledge extraction methods*

Knowledge graph construction mainly includes domain ontology construction, knowledge extraction, knowledge fusion, knowledge reasoning, and graph application [28]. Knowledge extraction (entity extraction and relation extraction) is the key step of knowledge graph construction. Extracting design-related knowledge and then constructing a knowledge graph to combine disparate silos of data is helpful for designers in conducting conceptual product design.

Entity extraction is the first step of knowledge extraction. In the early stage, statistical algorithms and machine learning, such as support vector machines [29], hidden Markov [30], and Conditional Random Fields (CRF) [31], are used to extract entities from unstructured text data. After that, deep learning has been widely used in entity extraction by using sequence labelling tasks to complete entity extraction tasks, and the accuracy has reached over 70%. For example, Dou et al. [32] proposed the ID-CNN-CRF algorithm to extract the entities of intangible cultural heritage automatically. Li et al. [33] extracted medical entities from electronic medical records by using Bidirectional LSTM-CRF (BiLSTM LSTM-CRF). After that, large-scale pre-training language models, such as BERT [34] and XLNet [35], have increased the accuracy of various tasks in natural language processing to more than 93%. Despite the advantage of high accuracy, the above models incur a large amount of computing cost.

Relation extraction is the second step of knowledge extraction. In the beginning, semantic rules and templates were used to extract relations. Afterwards, researchers have done in-depth work on deep learning methods in the past few years. For example, Samuel et al. [36] presented the model of BiLSTM to extract relations based on the word vector attention mechanism. Li et al. [37] proposed the Lattice LSTM model of word vectors to extract Chinese character relations. The above research has achieved good performance in relation extraction independently. The relation extraction is dependent on entity extraction, which makes it hard to extract overlapping entity-relation triples. In the case of overlapping entities and relations, the sequence tagging method in the traditional relationship extraction method can only extract one relationship of an entity pair. It cannot solve the relationship extraction in the overlapping phenomenon [38, 39].

To solve the extraction problem of overlapping scenes, some research processes entity extraction and relation extraction together by the joint entity and relation extraction methods. For example, a sequence-to-sequence (Seq2Seq) model with a copy mechanism was proposed to extract overlapping triples [40]. Based on the Seq2Seq model, they improved considerably with reinforcement learning [41–43]. Moreover, Graph Convolutional Networks were introduced to model text as relational graphs [44]. Li et al. [45] presented a Multi-Turn QA model to extract triples by answering templated questions. Most previous methods treat relations as discrete labels assigned to entity pairs, which have poor effects if multiple relationships exist between entities. Wei et al. proposed [46] CASREL to learn entities and relations through an end-to-end cascade binary tagging framework to address the overlapping problem. The training objective of the cascade framework is designed at the triple level, which naturally solves the overlapping problem.

2.3. Research gaps

The literature above has explored design knowledge-aided conceptual product design methods and possible applications of knowledge graphs in conceptual product design. The state-of-art knowledge extraction methods were examined. But there are still some critical challenges:

(1) The lack of a knowledge graph-aided conceptual product design framework to support knowledge acquisition.
(2) There are limitations in the design knowledge extraction, which is hard to solve the overlapping triple problem of the knowledge extraction task.
(3) The current conceptual product design approaches focus on specific tasks, which is insufficient compared with the overall design process.

Based on the previous research, it is evident that there are still some deficiencies in the knowledge graph-aided conceptual product design. Motivated by research gaps and challenges, this study proposes the DKG framework for the conceptual product process and discusses the definition, hierarchies, modules, and application of the DKG. A design knowledge extraction model is proposed to construct the DKG. The construction and application of the DKG are demonstrated to accomplish knowledge acquisition, management and sharing for conceptual product design. Advantages, limitations, conclusions and future work are also discussed.

3. DKG framework for conceptual product design

3.1. DKG framework

The process of conceptual product design can be described as a flow of knowledge transformation. It

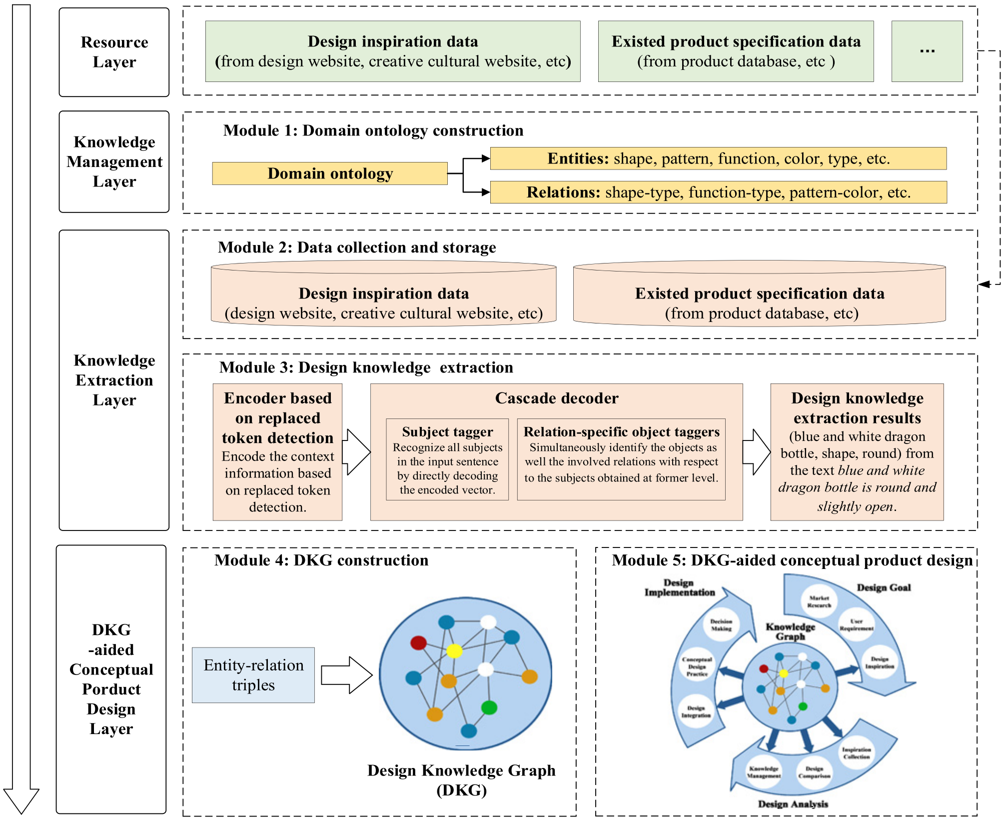

Fig. 2. The proposed DKG framework.

has the design goal stage (market research, design requirement, design inspiration), design analysis stage (knowledge collection, design comparison, knowledge management), and design implementation stage (design integration, conceptual design practice, decision-making) [22].

Compared with the conventional design knowledge management methods, such as XML and OWL, knowledge graph has advantages for automatic construction, information visualisation and semantic association. Owing to the application innovation, this section proposes a DKG framework to support the conceptual product design process, such as design inspiration, knowledge collection, design comparison, knowledge management, design integration, and conceptual design practice. (Fig. 2).

The proposed DKG framework includes resource layer, knowledge management layer, knowledge extraction layer, and the DKG-aided conceptual product design layer. In the resources layer, design inspiration data, existing product specification data, and other design knowledge constitute the data resources. It is the foundation of the proposed framework with massive data. The knowledge management layer aims to provide an ontology for the following knowledge extraction layer. Many design-related concepts and relationships, such as shape, pattern and function, become the entities and relations of the domain ontology. The knowledge extraction layer includes the data collection, storage, and design knowledge extraction module. The design-related knowledge is collected from the resource layer, and then extract the design knowledge from unstructured data using the proposed design knowledge extraction model. The DKG-aided conceptual product design layer involves integrating entities and relations, constructing the DKG and supporting the conceptual product design process.

Five core modules are further elaborated to construct the proposed four-layered framework, including the domain ontology construction module, data collection and storage module, design knowledge extraction module, DKG construction module, and DKG-aided conceptual product design module.

3.2. Module 1: domain ontology construction

The main objective of this section is to clarify the top-level schemas in design knowledge acquisition and define the domain concepts in ontology as a reference for the later modules. The DKG-aided conceptual design is a knowledge-intensive process that highly relies on the domain knowledge of the specific project. The domain ontology is used due to its good representation of knowledge structure.

Based on the previous research [47], the process of domain ontology construction is as follows: Firstly, The concepts are classified from data according to some rules, such as the requirements of designers, the CIDOC CRM domain ontology model [48], the creative design ontology model [49, 50], and other design ontologies. The concepts should satisfy the principles of independence, sharing and minimisation. Then, the top-down approach connects the concepts and builds a hierarchical framework. Finally, the relational framework is constructed to describe the relationships between different concepts.

It is worth mentioning that the entities and relationships are usually pre-defined by designers or experts in the phase of domain ontology construction, and there are significant differences between different design cases.

3.3. Module 2: data collection and storage

The DKG data mainly includes design inspiration data and existing product specification data. Design inspiration data is a powerful resource for conceptual product design, including design websites, creative cultural websites, intangible cultural heritage websites and Wiki websites. Existing product specification data refer to the relevant products designed in the previous projects, collected from the database as a reference for future design projects.

Different types of design-related data are stored in the relational database and the NoSQL database. The relational database stores integrated data due to the structural format. The NoSQL database stores the extracted knowledge in the graph-based form, where the nodes depict entities, and the edges represent their relationships. Neo4 j, as a typical NoSQL database, has the advantage of high performance for retrieval and application. Hence, Neo4 j is used for the DKG-aided conceptual product design in the following case study.

3.4. Module 3: design knowledge extraction

A joint entity and relation extraction method is proposed to discover implicit design knowledge from massive and fragmented online data. An encoder based on replaced token detection is presented to contextualise the input sentences. A cascade decoder inspired by the CASREL model [46] is constructed for converting triple extraction to mapping two entities. For example, the entity-relation triple (blue and white dragon pattern bottle, shape, round) can be extracted from the text ‘blue and white dragon pattern bottle is round’. The proposed design knowledge extraction model is further discussed in Section 4.

3.5. Module 4: DKG construction

Design Knowledge Graph $DKG = <V, E>$ can be constructed where V stands for entities of the graph and E means the relations between entities. Each entity represents a design-related knowledge node, and relationships define the edges between them. Considering all the possibilities of the relationships between entities, all types of entities and relations will appear in the DKG. With the help of the design knowledge extraction model, the increasing amount of design knowledge can be collected automatically and periodically from online resources and stored in the dataset, which dynamically constitutes the entire DKG.

3.6. Module 5: DKG-aided conceptual product design

The function of the website is developed according to the conceptual product design process with the extracted design knowledge. The Neo4 j is used to store large numbers of entities and relations. The Flask web framework written in Python is used to deploy the knowledge extraction model. The website provides designers with better design knowledge services such as idea stimulation, inspiration management, and decision-making. The DKG-aided conceptual product design is discussed further in Section 5.4. Meanwhile, the design knowledge extraction model is demonstrated in the next section.

4. The joint entity and relation extraction model for design knowledge extraction

4.1. Problem description

It is essential to extract entity-relation triples $G = head, relation, tail$ from the unstructured design-related text for constructing the DKG. For example, extracting entity-relation triple (blue and white dragon pattern bottle, is shape of, round) from the text *The blue and white dragon pattern bottle is round and slightly open.* However, there are two obstacles hinder the design knowledge extraction. Using pre-training language models to realise the first step of knowledge extraction (entity extraction) incurs a large amount of computing cost, as discussed in Section 2.2. Besides, there are always overlapping scenarios in design knowledge extraction tasks, as shown in Fig. 3(a).

4.2. *The framework of the design knowledge extraction model*

The first problem of design knowledge extraction is the computing cost during using pre-training language models for encoding the context information. Despite the numerous successes of pre-training models such as BERT [34] and XLNet [35], which generally uses masked language modelling (MLM), reducing the computing cost remains a challenge. MLM method corrupts the input by replacing tokens with [MASK] before training a model to recreate the original tokens, which typically require lots of amount of computing to be effective for good performance. In the literature [51], the replaced token detection is a more sample-efficient pre-training task. The replaced token detection method corrupts the input instead of masking the input by replacing some tokens with plausible alternatives sampled from a generator network. Then it trains a discriminative model that forecasts whether each token was replaced by a generator sample or not instead of training the model that predicts the original tokens of the corrupted tokens. Several experiments in the previous research [51] demonstrate that the replaced token detection is more sample-efficient than MLM pre-training methods like BERT [34] and XLNet [35] because it is defined over all tokens instead of the masked subset. Hence, assuming the same model data, size and computation, the contextual representation of the replaced token detection method outperforms the MLM method. It also works well at a large scale and performs similar to XLNet and RoBERTa [52] while using less than 1/4 of the compute for training.

Therefore, we believe that introducing the replaced token detection for design knowledge extraction can help encode the context information and reduce the computing cost in the pre-training task.

Furthermore, the second obstacle in design knowledge extraction is overlapping triples. The essence of the overlapping problem is overlapping entity redundancy (Fig. 3(a)). Some research [41–45] treat relations as discrete labels assigned to entity pairs to solve the overlapping entity redundancy, which has poor effects if there are multiple relationships exist between entities. The cascade framework inspired from CASREL [46] learns both entities and relations through an end-to-end cascade binary tagging framework. The design knowledge extraction is to identify all potential (subject, relation, object) triples in a sentence, in which some triples may share subjects or objects. To achieve this purpose, we directly model triples and create training objectives at the triple level. In contrast to earlier approaches such as [41], the training goal is defined separately for entities and relations without directly modelling their integration at the triple level. Detailly, given annotated input sentence x_j from the design-related training set D and some possible overlapping triples $T_j = \{(s, r, o)\}$ in x_j, maximise the data likelihood of the training set D:

$$\prod_{j=1}^{|D|}\left[\prod_{(s,r,o)\in T_j} p\left((s,r,o)\,\middle|\,x_j\right)\right] \tag{1}$$

$$=\prod_{j=1}^{|D|}\left[\prod_{s\in T_j} p\left(s\,\middle|\,x_j\right)\prod_{(r,o)\in T_j|s} p\left((r,o)\,\middle|\,s,x_j\right)\right] \tag{2}$$

$$=\prod_{j=1}^{|D|}\left[\prod_{s\in T_j} p\left(s\,\middle|\,x_j\right)\prod_{r\in T_j|s} p_r\left(o\,\middle|\,s,x_j\right)\prod_{r\in R\backslash T_j|s} p_r\left(o_\emptyset\,\middle|\,s,x_j\right)\right]. \tag{3}$$

where $s \in T_j$. is a subject in T_j. $T_j\,|s$ is the set of triples whose subject is s in T_j. $(r, o) \in T_j\,|s$ denotes a pair of (r, o) led by subject s in T_j. R is t set of all possible relations. $R\backslash T_j|s$ denotes all relations except those led by s in T_j. $o_\emptyset$ is a 'null' object. Equation (2) exploits the chain rule of probability. Equation (3) applies the crucial fact that for a given subject s, any relation relevant to s would have corresponding objects in the sentence and other relations would have no object in the sentence.

This formulation has several advantages. First, since the data likelihood begins at the triple level, maximising this likelihood equates to optimising the final evaluation criteria directly at the triple level. Second, by making no assumptions about how several triples may share entities inside a sentence, it is designed to solve the issue of overlapping triples. Thirdly, the decomposition in Equation (3) motivates

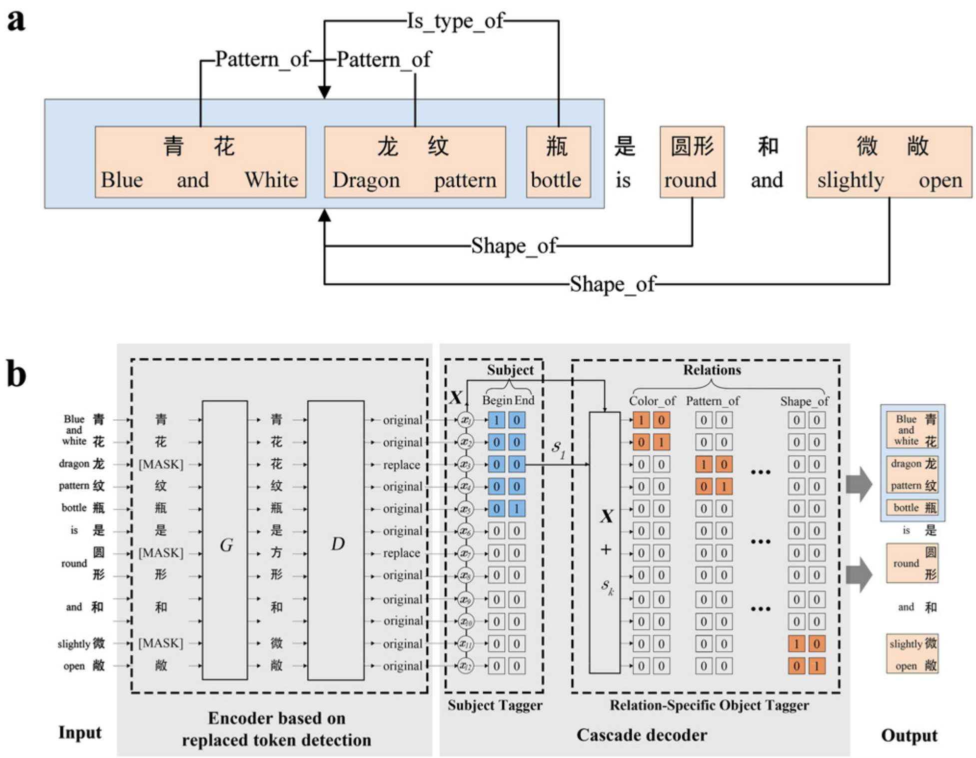

Fig. 3. An overview of the proposed design knowledge extraction model. a, An example of the overlapping scenarios. Several triples share the same subject' Blue and white dragon pattern bottle'. b, The proposed design knowledge extraction model. One subject 'Blue and white dragon pattern bottle' is detected at the subject tagger level. The Relation-specific object taggers are employed to find corresponding objects of the subject on different relations.

a novel tagging scheme for triple extraction. As opposed to classifying relations for (subject, object) pairs, each relation is described as a function that maps subjects to objects in this way. Hence, the cascade framework finds all possible subjects in the sentence, and then it applies relation-specific object taggers to find all relevant relations to the corresponding objects. Since the formulation is optimised at the entity-relation triple layer, optimising the data likelihood optimises the final evaluation at the triple level. It can essentially solve the overlapping entity redundancy.

As a result, to achieve the design knowledge extraction and solve the problem of computing cost and overlapping entity redundancy, this study instantiates the design knowledge extraction model as an encoder based on replaced token detection and a cascade decoder. The encoder module based on replaced token detection extracts feature information from the input sentences. The cascade decoder consists of the subject tagger and relation-specific objects tagger. The subject tagger is used to find all the subjects in the input sentence, and the relation-specific object taggers are employed to find corresponding objects of the subject in turn. Finally, the novel model can extract multiple overlapping entity-relation triples from the unstructured text at the same time.

4.3. The encoder based on replaced token detection

The encoder module based on replaced token detection extracts feature information $\boldsymbol{x}_j$ from the input design-related sentence x_j, which will feed into subsequent tagging modules. It trains a generator G and a discriminator D, as shown in Fig. 3(b). Each one consists of an encoder that maps a sequence of input tokens $\boldsymbol{x} = [x_1, \ldots, x_n]$ into contextualised vector representations $h(\boldsymbol{x}) = [h_1, ..., h_n]$. For a given position t, the generator outputs a probability for generating a token x_t with a softmax layer:

$$p_G(x_t|\boldsymbol{x}) = \exp(e(x_t)^T h_G(\boldsymbol{x})_t) / \sum_{x'} \exp(e(x')^T h_G(\boldsymbol{x})_t). \quad (4)$$

where e denotes token embeddings. The discriminator D predicts whether the token x_t comes from the data rather than the generator distribution, with a sigmoid output layer:

$$D(\boldsymbol{x}, t) = \mathrm{sigmoid}(w^T h_D(\boldsymbol{x})_t). \tag{5}$$

Then training the generator using the MLM pre-training task. First, MLM selects a random set of positions to mask out $\boldsymbol{m} = [m_1, \cdots, m_k]$. The selected tokens are replaced with a [MASK] token and denote this as $\boldsymbol{x}^{\text{masked}} = \text{REPLACE}(\boldsymbol{x}, \boldsymbol{m}, [\text{MASK}])$. The generator predicts the original identities of the masked-out tokens. The discriminator distinguishes the corrupted examples $x^{corrupt}$ from the data that the generator has replaced. The loss functions are

$$\mathcal{L}_{\text{MLM}}(\boldsymbol{x}, \theta_G) = \mathbb{E}\left[\sum_{i \in \boldsymbol{m}} -\log p_G\left(x_i \middle| \boldsymbol{x}^{\text{masked}}\right)\right]. \tag{6}$$

$$\mathcal{L}_{\text{Disc}}(\boldsymbol{x}, \theta_D) = \mathbb{E}\sum_{t=1}^{n}\left(x_t^{corrupt} = x_t\right)\log D\left(\boldsymbol{x}^{corrupt}, t\right)$$

$$+\left(x_t^{corrupt} \neq x_t\right)\log(1 - D\left(x^{corrupt}, t\right)). \tag{7}$$

Finally, minimising the combined loss over a large design corpus χ of raw text. We approximate the expectations in the losses with a single sample and do not back-propagate the discriminator loss through the generator. After pre-training, fine-tune the discriminator on downstream tasks.

$$\min_{\theta_G, \theta_D} \sum_{x \in \chi} \mathcal{L}_{\text{MLM}}(\boldsymbol{x}, \theta_G) + \lambda \mathcal{L}_{\text{Disc}}(\boldsymbol{x}, \theta_D). \tag{8}$$

4.4. The cascade decoder

After having the feature information $\boldsymbol{x}_j$ from the input design-related sentence x_j using the encoder based on replaced token detection, the cascade decoder is introduced to extract design-related entities and their relations from the overlapping entity redundancy scenes. The cascade decoder inspired by CASREL [46] is modelled according to the formulation (1)(2)(3), which consists of a subject tagger and relation-specific object taggers (Fig. 3(b)).

4.4.1. Subject tagger

The subject tagger module is designed to find all design-related subjects of the sentence by decoding the encoded vector $\boldsymbol{h}_N$ produced by the N-layer encoder. Specifically, it adopts two binary classifiers to detect the start and end positions of subjects, respectively. The subject tagger assigns each token a binary tag (0/1) to indicate whether the token is a start/end position of a subject or not.

The subject tagger on each token is operated as follows:

$$p_i^{starts} = \sigma(\boldsymbol{W}_{start}\boldsymbol{x}_i + \boldsymbol{b}_{start}). \tag{9}$$

$$p_i^{ends} = \sigma(\boldsymbol{W}_{end}\boldsymbol{x}_i + \boldsymbol{b}_{end}). \tag{10}$$

where $p_i^{start\_s}$ and $p_i^{end\_s}$ respectively denote the probability of identifying the $i-th$ token as the start and end position of the subject. If the probability exceeds the threshold, the token will be assigned with a tag 1. Otherwise, the token will be assigned with a tag 0. $\boldsymbol{x}_i$ represents the encoded representation of the $i-th$ token. σ is the sigmoid activation function, $\boldsymbol{W}_{(\cdot)}$ is the trainable weight, $\boldsymbol{b}_{(\cdot)}$ represent the bias.

The subject tagger identifies the span of the design-related subject s by optimising the following likelihood function:

$$p_\theta(s|\boldsymbol{x}) = \prod_{t \in starts_s, end_s} \prod_{i=1}^{L} \left(p_i^t\right)^{\boldsymbol{I}y_i^t=1}$$

$$\left(1 - p_i^t\right)^{\boldsymbol{I}y_i^t=0}. \tag{11}$$

where L represents the length of the sentence. If z is true, $\boldsymbol{I}\{z\} = 1$, $\boldsymbol{I}\{z\} = 0$ otherwise. $y_i^{starts_s}$ indicates the binary tag of the subject start position for the $i-th$ token in $\boldsymbol{x}$, and y_i^{ends} indicates the binary tag of the subject end position for the $i-th$ token. The parameters $\theta = \{\boldsymbol{W}_{start}, \boldsymbol{b}_{start}, \boldsymbol{W}_{end}, \boldsymbol{b}_{end}\}$. If there are several subjects, the span of the subject is decided according to the nearest start and end pair match principle.

4.4.2. Relation-specific object taggers

The relation-specific object tagger module is constructed to find the design-related objects and the involved relations simultaneously. It consists of relation-specific object taggers with the same structure as the subject tagger for all possible relations.

The relation-specific object tagger on each token is operated as follows:

$$p_i^{starto} = \sigma\left(\boldsymbol{W}_{start}^r(\boldsymbol{x}_i + \boldsymbol{v}_{sub}^k\right) + \boldsymbol{b}_{start}^r). \tag{12}$$

$$p_i^{endo} = \sigma\left(\boldsymbol{W}_{end}^r(\boldsymbol{x}_i + \boldsymbol{v}_{sub}^k\right) + \boldsymbol{b}_{end}^r). \tag{13}$$

where $p_i^{start_o}$ and $p_i^{end_o}$ respectively denote the probability of identifying the $i-th$ token as an object's start and the end position. $\boldsymbol{v}_{sub}^k$ represents the encoded representation of the $k-th$ subject. The same decoding process is iteratively applied to each subject.

Given an encoded input sentence x and an extracted subject s, the object tagger identifies the span of a design-related object o for potential relation r by optimising the following likelihood function:

$$p_{\phi r}(o|\boldsymbol{s},\boldsymbol{x}) = \prod_{t\in\{start_o, end_o\}} \prod_{i=1}^{L} \left(p_i^t\right)^{I\left\{y_i^t=1\right\}} \left(1-p_i^t\right)^{I\left\{y_i^t=0\right\}} \quad (14)$$

where indicates the binary tag of object start position $y_i^{start_o}$ for the $i-th$ token in $\boldsymbol{x}$, and $y_i^{end_o}$ shows the binary tag of object end position for the $i-th$ token. If there is a null object $o_\emptyset$, $y_i^{start\_o_\emptyset}$ and $y_i^{end\_o_\emptyset}$ indicates 0. The Parameter $\varphi_r = \left\{\boldsymbol{W}_{start}^r, \boldsymbol{b}_{start}^r, \boldsymbol{W}_{end}^r, \boldsymbol{b}_{end}^r\right\}$.

4.4.3. Data log-likelihood objective

Taking the log of Equation (3), the objective $J(\Theta)$ is:

$$J(\Theta) = \sum_{j=1}^{|D|} \left[\sum_{s\in T_j} \log p_\theta\left(s\,|\boldsymbol{x}_j\right) + \sum_{r\in T_j|s} \log p_{\phi_r}\left(o\,|s\,,\boldsymbol{x}_j\right) + \sum_{r\in R\backslash T_j|s} \log p_{\phi_r}\left(o_\emptyset\,|s\,,\boldsymbol{x}_j\right) \right]. \quad (15)$$

where $p_\theta(s\,|\boldsymbol{x})$ represents the likelihood function of subject taggers, $p_{\phi r}(o\,|s\,,\boldsymbol{x})$ stands for the likelihood function of the relation-specific object taggers, $p_{\phi_r}\left(o_\emptyset\,|s\,,\boldsymbol{x}_j\right)$ means the likelihood function of null object, parameters $\Theta = \{\theta, \{\phi_r\}_{r\in R}\}$. The model is trained by maximising $J(\Theta)$ through Adam stochastic gradient descent [53] over shuffled mini batches.

5. Case study

To verify the effectiveness of the proposed DKG framework and the extraction model, a case study of conceptual product design inspired by intangible cultural heritage was discussed. Innovative cultural products can benefit local manufacturing enterprises, so we constructed the DKG based on the proposed method to assist designers in conceptual product design. The core steps of the case study are shown in Fig. 4. The case study was implemented based on Python 3.6, CUDA 10.1, PyTorch 2.7, Neo4 j, and SQL Server. The hardware system is Windows 10 with an i7 7700K CPU and 1080Ti 11 G GPU.

5.1. Domain ontology construction and dataset preparation

Design data refer to massive data with many design-related characteristics in the websites. Product designers can get inspiration from data of intangible cultural heritage, such as colour, pattern and material processing. Thus, we apply a large amount of text data collected from the Chinese Palace Museum website (https://www.dpm.org.cn/collection/ceramics.html) to validate the proposed model and framework. A case study was conducted to show how the knowledge graph organises intangible cultural heritage data and supports conceptual product design. This study collected 1151 pieces of porcelain, 4658 images, and 639,676 words of description from the website as the source of the design-related dataset.

Domain ontology describes the design-related concepts with entities and explains the relations between entities. This study defines the domain ontology of the DKG by the multidisciplinary team. The multidisciplinary team for this study comprises product designers, intangible cultural heritage (ICH) researchers and computer programmers. Multidisciplinary teams are selected on the basis that they must have worked for at least five years to ensure the validation of this study. The function of product designers is to define the conceptual product design knowledge in the data, including shape, colour, material, structure, and function. The ICH researchers help product designers to classify creative culture design, such as chronology, processing techniques, shape, function and excavation sites. Computer programmers are responsible for the computer technology for the DKG. In this study, the domain ontology was analysed with the help of five experienced product designers, two ICH researchers and two programmers, who worked together to develop the domain ontology. The process of domain ontology construction was as follows: (1) Opinions on the ontology classification were collected from a multidisciplinary team, and the following topics were discussed: 'What are the components of knowledge of an object for conceptual product design? What framework can be developed?' (2) Implementing focus groups. The

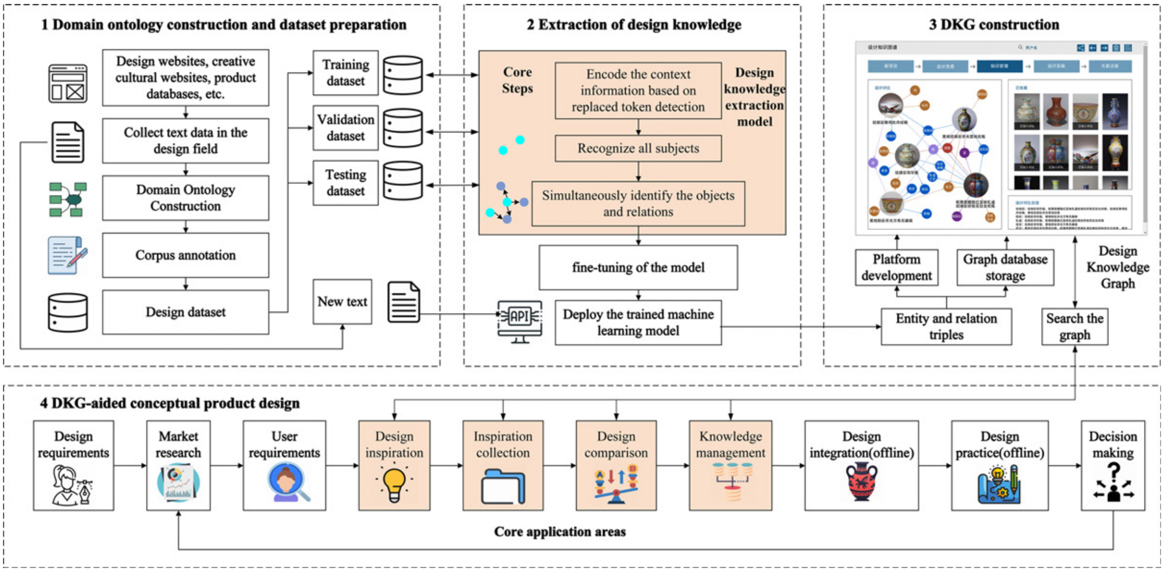

Fig. 4. The core step of the case study.

moderator introduces the topics. The multidisciplinary team discussed the topics, and the researcher recorded the verbal and non-verbal information of the participants using video. (3) The concept classification is derived by analysing and organising the information using content analysis and framework structure methods. Finally, domain ontology concepts include name, dynasty, pattern, shape, colour, glaze, and function. Name mainly refers to the name of objects, people, and organisations. For example, 'Blue and white bowl with pine, bamboo and plum pattern' is the name of porcelain. Dynasty, pattern, shape, and colour have design-related knowledge, such as 'Qing dynasty', 'Bamboo pattern', 'round shape', and 'blue and white'. The glaze is a kind of impervious layer of a vitreous substance, which can serve to colour, decorate, underlying design or texture. The function, such as tableware, decoration and sacrifice, is the reference for designers to conduct conceptual product design. Then, the relations between the entities were defined. A part of the entities and relations are shown in Table 2 and Fig. 5. For example, 'P4 has time span' is the relation between the entity 'Dynasty' and the entity 'Time-span', and 'P2 has colour of' is the relation between the entity 'Glaze' and the entity 'Colour'.

A dataset was constructed to train and test the effectiveness of the proposed design knowledge extraction model. The data were tagged manually using the Brat annotation software based on the domain ontology to build the corpus, as shown in Fig. 6(a). The manual annotation method was adopted using Brat annotation software. The annotated corpus was converted to dataset formats using Python, as shown in Fig. 6(b). The dataset was divided into the training set, validation set and testing set according to the ratio of 8:1:1 and contained 8689 triples and 97 types of relations. In addition, the number of entities and relations has a long-tail effect, which may impact the results.

5.2. *Extraction of design knowledge*

To verify the effectiveness of the proposed design knowledge extraction model in Section 3, the design knowledge extraction model was evaluated on the constructed dataset in Section 5.1. The model was tested on the public dataset DuIE [54] as well, which has 194,747 sentences and 49 relations. The statistics of the datasets as shown in Table 3.

The ten-fold cross-validation method was adopted to train the model. The thresholds of the two layers and the hyperparameters were determined on the validation set and shown in Table 4. The model was trained by using the stochastic gradient descent algorithm with an optimiser as Adam. The early stopping mechanism was adopted to prevent the over-fitting situation, which means stopping the training process when the model does not gain improvement for at least five consecutive epochs. The number of the bidirectional Transformer blocks of the pre-training model is 12, the hidden state size is 256, and the number of attention heads is 4. The pre-training model

Table 2
Main relations of the dataset

ID	Property name	ID	Property name
P1	has function of (is used for)	P87	at (was a place of)
P2	has colour of	P88	at someplace within
P3	has pattern of	P91	falls within (contains)
P4	has time-span (is time-span of)	P92	has created (was created by)
P5	consists of (forms part of)	P93	was motivated by (motivated)
P6	took place at (witnessed)	P94	is identified by (identifies)
P7	has current location (currently holds)	P95	has identified (was identified by)
P8	has type (is a type of)	P96	incorporates (is incorporated in)
...	......	P97	has association with (is associated with)

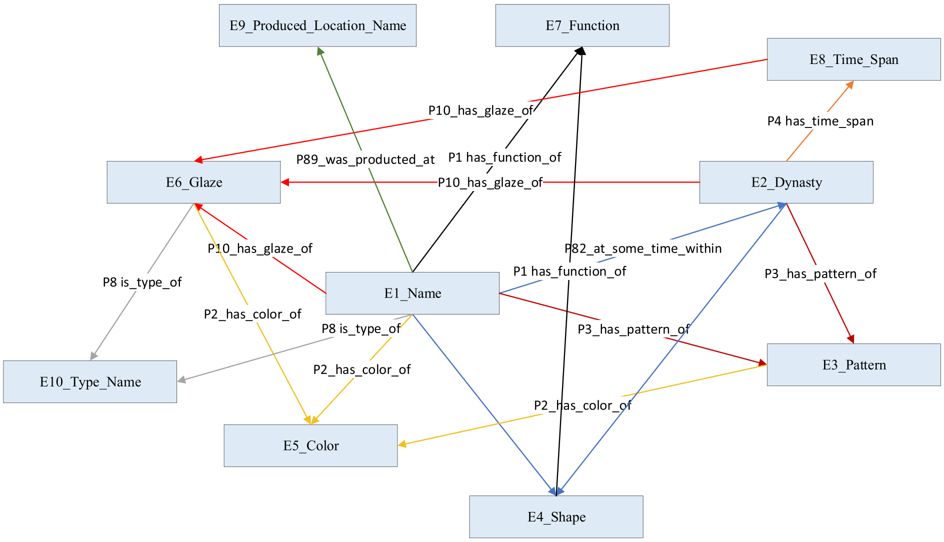

Fig. 5. Part of the relation diagram.

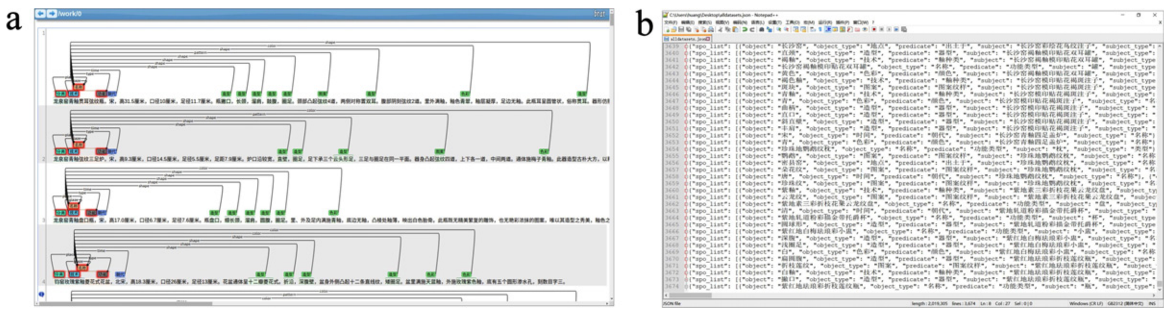

Fig. 6. The design dataset construction. **a**, Manual annotation of design data using Brat software. **b**, The design dataset construction.

contains 12M parameters. The max length of the input sentences is 100 words for a fair comparison, as the previous research [38, 42] suggested.

The training process was stopped when the model did not improve for 5 consecutive epochs, and the iterations were 13 epochs. The F1-score, precision, and recall results were 87.5%, 87.7% and 87.2%, respectively. The loss value was considerable at the beginning. While after six epochs, loss value approached zero, and the precision results tended to be stable, proving the robust performance of the model.

To demonstrate the feasibility and high accuracy of the proposed design knowledge extraction model, the model was compared with some state-of-art joint entity-relation extraction models: (1) Novel Tagging

Table 3
Statistics of the datasets

Category	Our dataset	DuIE
Train	7,017	364,218
Valid	838	20,027
Test	834	25,520
Overall	8,689	409,795

[38], a kind of sequence tagging entity-relation triple extraction model, the structure is BiLSTM encoding layer and LSTM decoding layer. (2) GraphRel [44], the structure is the BiLSTM encoding layer and Bi-GCN encoding layer. (3) MultiHead [55], multi-head selection model, the structure is embedding layer, BiLSTM encoding layer, CRF layer, label embedding layer and output layer. (4) Since the embedding layer of the above models adopts BiLSTM [56], an ablation experiment was conducted using the same embedding layer for comparison. So the encoding layer of our model was modified to the same BiLSTM in the fourth model. The results for the baselines were obtained from the constructed dataset and DuIE public dataset. The four comparison models trained the Chinese Wikipedia corpus using the skim-gram of word2vec [57] to get 300-dimensional word vectors as the input.

The results of different models for knowledge extraction on two datasets are shown in Table 5. The results of our model are much higher than NovelTagging, GraphRel, MultiHead and our model with the BiLSTM encoding layer. Our model overwhelmingly outperforms the baselines even without using the pre-training model.

5.3. DKG construction

The trained model was deployed using Flask web framework to automatically implement the design knowledge extraction function and develop a DKG for inspiring designers. Firstly, the classes of Flask were imported, and the instances of the classes were created as the application of the WSIG service. Then, the instance was used as a decorator, a function to take another function as its argument and return another function to the browser. Finally, Gunicorn and Nginx were configured to achieve the model deployment.

After deploying the design knowledge extraction model, the function of design knowledge extraction was realised. The test result is shown in Fig. 7(a), which shows excellent performance in the design knowledge extraction, especially extracting triples from sentences with overlapping scenarios. More than 10,000 entity relationship triples were extracted using the design knowledge extraction model, and the accuracy was manually adjusted to meet the requirement of the knowledge graph. The accuracy reaches 88.2% through manual sampling evaluation, ensuring the accuracy and reliability of design knowledge extraction. The extracted entity relationships were stored in the Neo4 j database (Fig. 7(b)).

Based on the DKG framework and the design knowledge extraction model, the DKG platform was built using a huge number of triples.

The functional architecture of the DKG is shown in Fig. 8(a), which achieves knowledge graph construction, knowledge graph update, knowledge graph management and application. It provides the man-

Table 4
Hyperparameters

Parameter	Best rate	Experimental scope
Leaning rate	1e-5	1e-2, 1e-3, 1e-4, 1e-5, 1e-6
Batch size	32	8, 16, 32, 64
Epoch	20	10, 15, 20, 25, 30
Early stopping	5	3, 5, 7
Threshold of the subject tagger	0.5	0.4, 0.45, 0.5, 0.55, 0.6
Threshold of the relation-specific object tagger	0.4	0.35, 0.4, 0.45, 0.5, 0.55

Table 5
Comparison of experiment results for our dataset and DuIE dataset

Methods	Our dataset			DuIE		
	Pre	Rec	F1	Pre	Rec	F1
NovelTagging [38]	0.567	0.312	0.403	0.589	0.367	0.452
GraphRel [44]	0.594	0.515	0.552	0.603	0.521	0.559
MultiHead [55]	0.713	0.624	0.666	0.644	0.637	0.640
Our_model (BiLSTM [56])	0.826	0.818	0.822	0.796	0.786	0.791
Our model	**0.877**	**0.872**	**0.875**	**0.845**	**0.824**	**0.834**

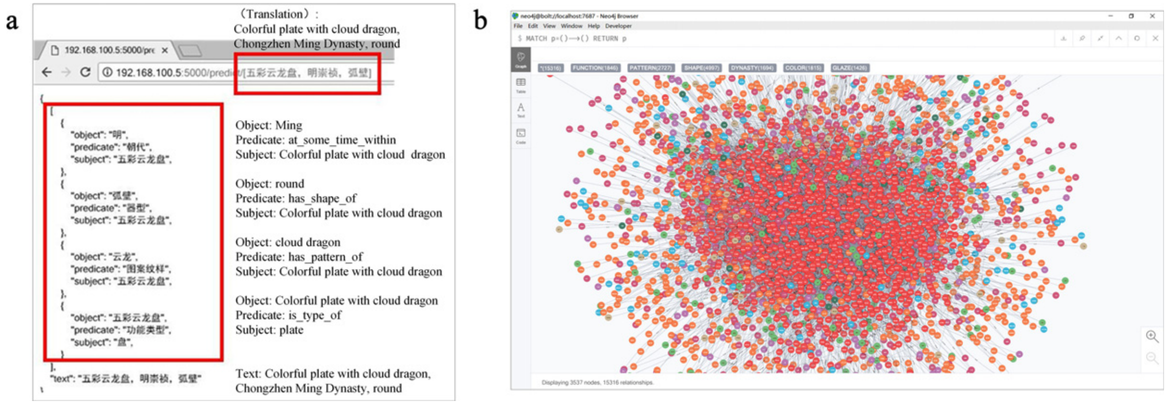

Fig. 7. Design knowledge deployment. **a**, The testing results of design knowledge extraction model. **b**, The extracted entity-relation triples were stored in the Neo4j graph database.

agement for models, tools, users and data, which is comprehensive and extendable for conceptual product design.

The technical architecture of the DKG is shown in Fig. 8(b), which is divided into the infrastructure layer, data layer, functional layer, application layer proxy and user interface layer from the bottom up. The infrastructure layer uses Git, Docker, Kubernetes and Jenkins, and the data layer uses MySQL, Redis, Neo4 j, MongoDB and Elasticsearch to complete data storage and search services. The functional layer is the core layer for DKG, providing the functions of knowledge graph construction, knowledge graph update and application, and the corresponding public and management components. The application layer proxy achieves Nginx reverse proxying, load balancing and static proxy. The user interface layer provides designers with access to the DKG platform.

5.4. DKG-aided conceptual product design

The conceptual product design process mainly includes the market goal, design requirements, design inspiration, knowledge collection, design comparison, knowledge management, design implementation, design prototyping and decision-making [24]. In the case study, six product designers, three engineers and three managers conducted a conceptual product design inspired by porcelain design knowledge on the DKG. All the people had full permission for the DKG with the help of the collaborative design module.

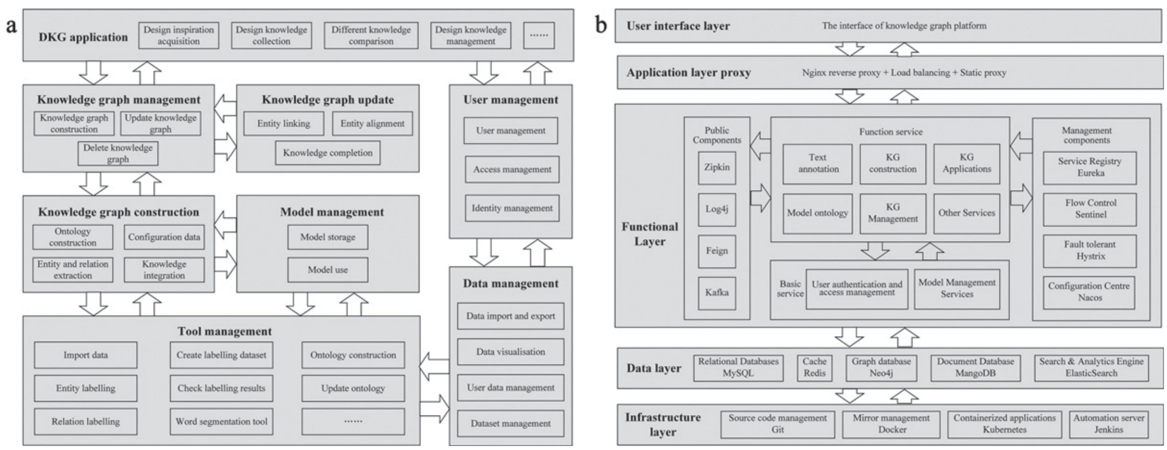

Fig. 8. The architecture of Design Knowledge Graph. **a**, The functional architecture of the knowledge graph. **b**, The technical architecture of the knowledge graph.

Figure 9(a) to Fig. 9(e) show the conceptual product design process of ‘the porcelain patterns’ theme based on the DKG. The process includes creating a new project, design inspiration acquisition, having detailed information, design knowledge comparison and management, and collaborative decision-making. The details are as follows:

(1) Creating a new project. As shown in Fig. 9(a), designers create a new project on the DKG and fill in market goals, design requirements and project description. For example, designers plan to design a series of products with traditional dragon patterns. They fill in the project name ‘Series of products in traditional dragon style’, the market goal ‘modern style products with dragon pattern’, and the design requirement ‘dragon pattern’. The project module enables everyone to create a project and easily collaborate on the DKG.
(2) Design inspiration acquisition. As shown in Fig. 9(b), the DKG identifies design-related entities and relations from input sentences using the design knowledge extraction model. Then, the DKG defines the search scope and matches the related design knowledge with the text-matching model. Finally, the DKG shows relevant knowledge and inspires designers. For instance, designers entered ‘design of bottles which have dragon pattern’. The DKG identified the design-related entities ‘dragon’ and ‘bottle’ in the input text using the design knowledge extraction model. The search scope is defined further and matches relevant data using the text-matching model. As a result, the DKG visualises images and text of associated nodes, showing information such as ‘Blue and white Bottle with dragon pattern’ and ‘Cloisonne and Faience bottle with branches and dragon pattern’. Therefore, designers can obtain visualised and correlated design cases on the DKG to acquire valuable design knowledge more conveniently. Compared with conventional design knowledge acquisition methods, such as XML and OWL, the innovation of the DKG is that the design inspiration is obtained in a more relational, visual and reasonable way.
(3) Detailed design knowledge. Figure 9(c) shows that the DKG can search, edit, and save detailed information on design cases. For example, the designer clicks on the node ‘Yaozhou celadon glaze carved lotus pattern amphora’. In this case, the search and the inference operations are performed on the node. Specifically, the node is mapped to the DKG, and the DKG returns relevant nodes and presents design details such as ‘blue and white glaze’ and ‘lotus petal pattern’. Editing and saving functions are performed by clicking on the ‘Edit’ and ‘Save’ buttons. The design knowledge obtained using search engineers is relatively fragmented, imprecise, weak in reasoning and less relevant. The search results of the DKG are more precise and have relevance and inference characteristics. Unlike conventional text presentation, the DKG is visually presented and has graphical interaction.
(4) Design knowledge comparison and management. As shown in Fig. 9(d), the DKG can further explore similarities and differences to facilitate the design scheme. For example, a designer selects five porcelain bottles for comparison, and the DKG returns interrelated design knowledge, shown in a graph. Four bottles have the ‘enamel’ glaze type, and two have the ‘rolling process’ to help the designer understand similarities and differences between cases. Compared with the previous design knowledge comparison and management methods, the DKG can present comparative information in a correlated and visualised format.
(5) Uploading the conceptual design scheme and collaborative decision-making. As shown in Fig. 9(e) and Fig. 9(f), designers use modelling software and graphic design software to create a product concept design scheme. Designers complete the conceptual design and upload several prototypes on the DKG by using the collaborative decision-making function. The decision module allows different designers to present their decisions on different schemes. After providing comments on the design schemes’ aesthetic, functional, feasibility and commercial aspects of the design scheme, the DKG presents an overall score to help the decision maker decide on the final design solution.

The entire conceptual product design communication is in progress through the DKG prototype platform. The DKG has been demonstrated to solve the problems of fragmented design knowledge, poor

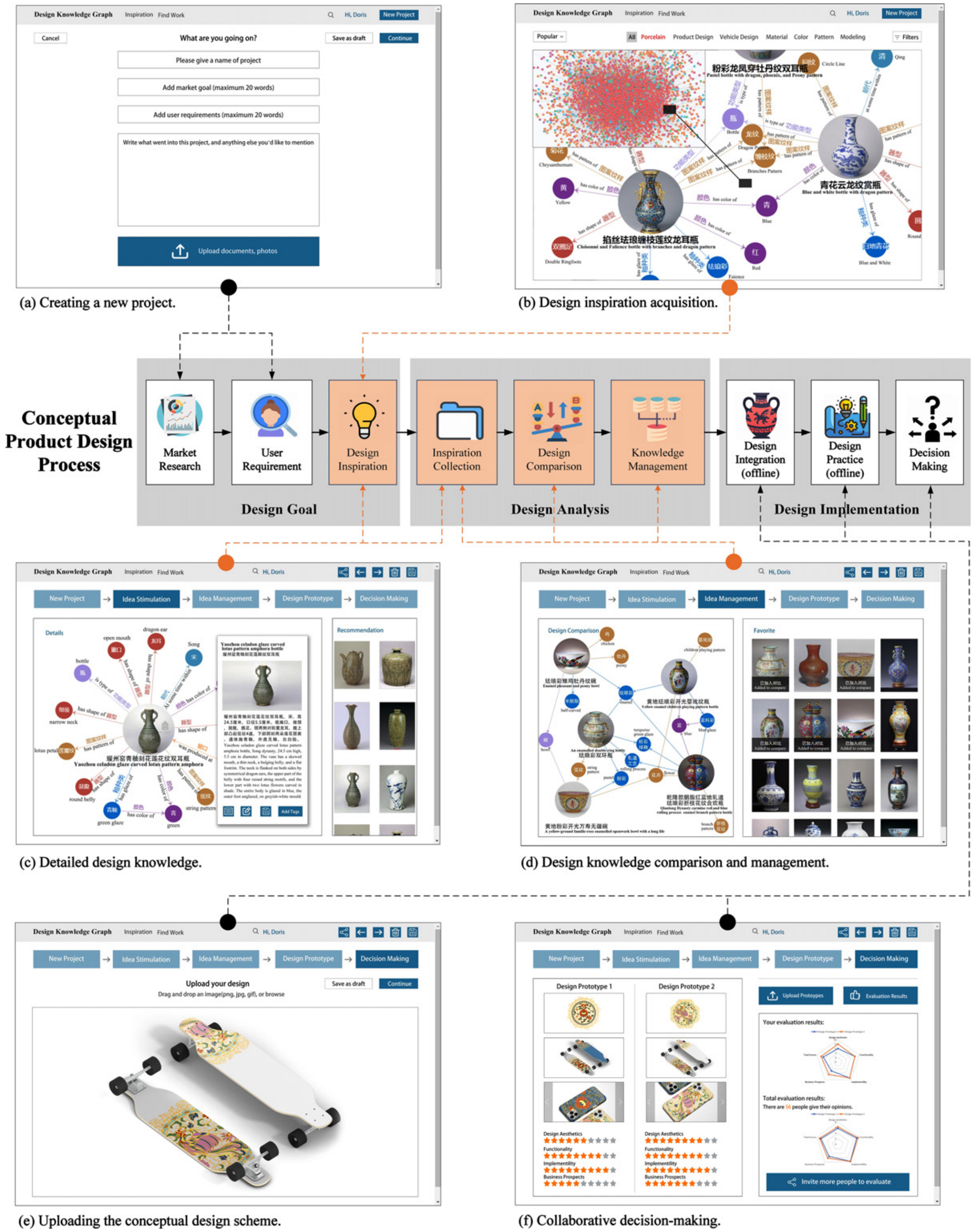

Fig. 9. The illustration of the DKG-aided conceptual product design (translation).

reasoning and poor visualisation through the conceptual product design inspired by porcelain culture. DKG improves the efficiency of design knowledge acquisition during the process of conceptual product design, providing designers with relevant, reasonable and visual design knowledge. The test results show an excellent effect from all system participants, who believe that the widespread application of this study will increase conceptual product design efficiency on a large scale. Similarly, other design cases can be implied in this study, serving as a further reference for designers.

6. Discussion

This section discusses the knowledge extraction model, DKG framework, application and limitation to verify the innovation of the study.

6.1. Comparison of the knowledge extraction model

Regarding the knowledge extraction model, the proposed design knowledge extraction model has two advantages: The end-to-end cascade framework solves the overlapping entity redundancy problem in design knowledge extraction. The model performs better after using a pre-training encoder based on replaced token detection, showing the importance of prior knowledge. A comparative experiment was conducted in Section 5.2, proving the model has higher quality in the joint entity and relation extraction, as is shown in Fig. 10, the X-axis represents the name of the datasets, and Y-axis stands for the results.

The detailed comparison is as follows: (1) The decoding layer of NovelTagging adopts the sequence labelling method, which cannot solve the problem of overlapping triples. The results of the NovelTagging are the worst of the 5 models. (2) The decoding layer of GraphRel has a positive effect on the extraction, and the F1-score of the two datasets (55.2% and 55.9%) is slightly higher than NovelTagging (40.3% and 45.2%). (3) MultiHead adopts the multi-head selection to assess the relations of the entities and solves the problem of overlapping entity redundancy. Therefore, the results of MultiHead on our dataset (71.3%, 62.4% and 66.6%) are about 50% higher than NovelTagging (56.7%. 31.2% and 40.3%) and GraphRel (58.9%, 36.7% and 45.2%). (4) Even without using the pre-training model, the F1 scores of our model (Our_Model_BiLSTM) are 82.2% and 79.1% on two datasets, which improves about 23.4% and 23.6% than Multi-Head. (5) Our model outperforms the baselines. The F1 scores are 87.5% and 83.4% on our dataset and DuIE public dataset, achieving 31.4% and 30.3% improvements over the MultiHead model respectively. Our model using pre-training based on replaced token detection is about 7% higher than our model using BiLSTM. In conclusion, these experiments verify the validation of our model compared to existing methods.

We also explore the extraction capability of overlapping entities. The confusion matrix allows the detailed investigation of the model's ability to extract each entity type. The confusion matrix for the various models on the design dataset is shown in Fig. 11. The x-axis displays the predicted labels, and the y-axis indicates the real labels.

As illustrated in Fig. 11, our model performs better than NovelTagging, GraphRel, MultiHead, and our model (BiLSTM). The accuracy of our model on the design dataset is higher than 80%, except for the ' name' category. The 'name' category has the worst accuracy (about 71%), and other models also perform worst in the 'name' category (17%, 34%, 50%, and 64%). We found that the poor extraction of the category 'Name' may be due to most of the 'name' containing 'pattern' and 'shape'. In addition, although 'pattern and shape' have large data samples, they have a severe long-tail effect and weak extraction ability. Therefore, data augmentation should be taken on the data 'pattern' and 'shape'. 'Dynasty' and 'colour' have a better extraction effect since these two types of entities have a low overlapping situation with other entities.

6.2. DKG framework

As for the framework, we compared the knowledge graph-based framework with the XML-based [2] method and the OWL method [5]: (1) The conceptual product design method based on XML [2] can describe the structural information among each knowledge. Nevertheless, the XML-based method does not have semantic description, reasoning, understandability, and visualisation. (2) Compared to an XML-based approach, the OWL-based approach [5] has improved machine understandability, coding, reasoning, comprehension, and visualisation. (3) The knowledge graph-based method is more effective for information logicality, knowledge coding, automatic reasoning, machine-understandable and data visualising than XML and OWL methods.

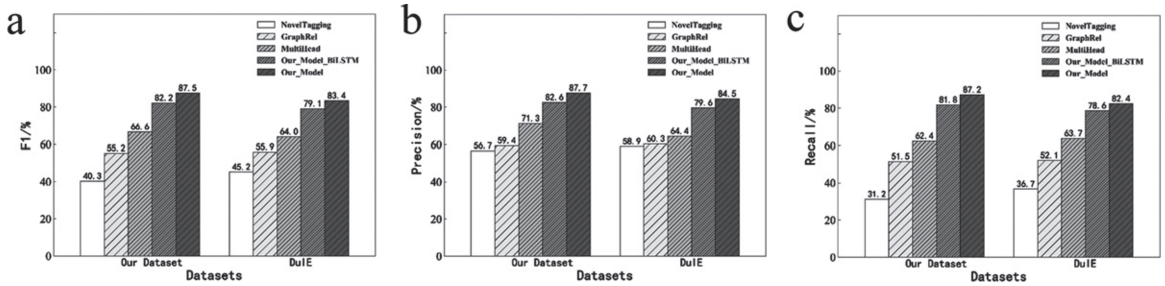

Fig. 10. Comparisons of different methods on our dataset and DuIE dataset. **a**, F1-score of different methods on two datasets. **b**, Precision results of different methods on two datasets. **c**, Recall results of different methods on two datasets.

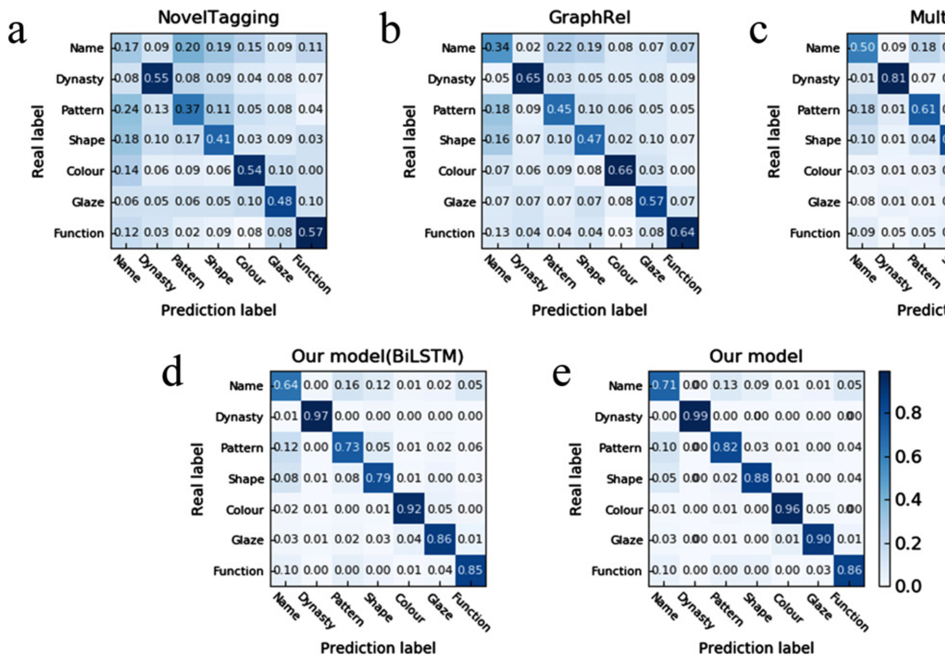

Fig. 11. Normalised confusion matrix of different methods on our dataset. **a**, Normalised confusion matrix of NovelTagging. **b**, Normalised confusion matrix of GraphRel. **c**, Normalised confusion matrix of MultiHead. **d**, Normalised confusion matrix of our model (BiLSTM). **e**, Normalised confusion matrix of our model.

The case study shows that the proposed DKG framework supports the entire conceptual product design process, especially in the idea stimulation and creative management stages. It also supports knowledge collection and comparison, which benefits knowledge representation and knowledge management for product development. Secondly, the proposed design knowledge extraction model can effectively extract design knowledge from unstructured text, which gathers various design resources for the conceptual product design process. The design knowledge extraction model can extract design-related knowledge automatically if data source websites have new data. Thirdly, the DKG builds a bridge between siloed data sources, historical cases, and new projects. It solves the problem of fragmentation, describes the relationship between knowledge resources, and improves the efficiency of the conceptual product design process.

6.3. Applications and limitations

The application of the proposed DKG framework and the design knowledge extraction model is highly compatible and wildly useful. Some design-related websites, such as Behance, Dribble, and Muzli, can apply this framework and extraction model to facilitate design knowledge acquisition.

In addition, the collaborative platform allows it to be the enterprise-level solution in the product development industry and has high extensibility. Furthermore, given the advantages of visualization and interconnected, consumer users can also use the DKG to acquire valuable information, understand the relationships between different knowledge. As a result, all users can indirectly inspired by DKG.

This study still has some limitations. The DKG construction approach requires a large amount of historical data. A substantial quantity of tagged data is also necessary for the design knowledge extraction model. Collating and labelling large amounts of data is laborious and makes it challenging to construct the DKG. Moreover, the DKG lacks certain functionalities, such as requirement analysis and knowledge recommendation. Hence, designers must manually search the DKG for design knowledge. These limitations should be taken into account in future research.

7. Conclusion

Despite the importance of design knowledge in conceptual product design, fragmentary data and large amounts of multidisciplinary knowledge make valuable design knowledge difficult to acquire for designers. This study proposes a Design Knowledge Graph-aided (DKG-aided) conceptual product design approach for knowledge acquisition and design process improvement. Specifically, the DKG comprises five modules based on design requirements and data characteristics to obtain the required valuable design knowledge and facilitate conceptual product design. Secondly, the design-related entities and relations are extracted automatically based on a joint entity and relation extraction model to accumulate design knowledge and build the DKG automatically. The joint entity and relation extraction model contextualises the input sentences using an encoder based on replaced token detection, extracts entities and relations simultaneously by introducting a cascade decoder. It solves the problems of computing cost and overlapping triple. Experimental results illustrate that the proposed model significantly outperforms NovelTagging, Graphrel and Multi-Head. The F1-scores on our dataset and the DuIE public dataset are 87.5% and 83.4% respectively, representing 31.4% and 30.3% improvements over the Multi-Head model. Finally, the feasibility of the proposed approach is validated in the case study of conceptual product design inspired by massive real data of porcelain. The DKG-aided approach improves the data acquisition process of fragmentation and non-visualisation by constructing the DKG with the joint entity and relation extraction model, giving designers more intuitive access to valuable design knowledge. The DKG can be applied to collaborative design platforms and inspiration websites to help designers easier to obtain the valuable design knowledge and thus improve design efficiency.

In the future, it could be interesting to explore automatic DKG construction methods for small samples and include a broader range of functions such as design requirements analysis, design knowledge recommendation and design implementation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

CRediT authorship contribution statement

Yuexin Huang: Conceptualisation, Methodology, Software, Validation, Formal analysis, Supervision, Writing – original draft, Writing – review & editing. **Suihuai Yu**: Project administration, Writing – review & editing. **Jianjie Chu**: Conceptualisation, Resources, Supervision, Writing – review & editing. **Zhaojing Su**: Validation, Software. **Yaokang Zhu**: Software, Writing – original draft, Visualization. **Hanyu Wang**: Investigation. **Mengcheng Wang**: Investigation, Validation. **Hao Fan**: Conceptualisation, Supervision, Writing – review & editing.

Acknowledgments

The authors would like to express sincere gratitude to the anonymous reviewers for the invaluable comments that have improved the quality of the study. This study is supported by the Chinese Special Projects of the National Key Research and Development Plan (2019YFB1405701).

References

[1] S. Kota and C.L. Lee, General framework for configuration de-sign: part 1-methodology, *Journal of Engineering Design* **4** (1993), 277–289. https://doi.org/10.1080/09544829308914787.

[2] R. Bloomfield, E. Mazhari, J. Hawkins and Y.J. Son, Interopera-bility of manufacturing applications using the core manufactur-ing simulation data (CMSD) standard information model, *Computers and Industrial Engineering* **62** (2012), 1065–1079. https://doi.org/10.1016/j.cie.2011.12.034.

[3] Z.Y. Wu, J.H. Liao, W.Y. Song, H.L. Mao, Z.F. Huang, X.X. Li and H.Y. Mao, Semantic hyper-graph-based knowledge representation architecture for complex product development, *Computers in Industry* **100** (2018), 43–56. https://doi.org/10.1016/j.compind.2018.04.008.

[4] Y. He, C.P. Hao, Y.L. Wang, Y.F. Li, Y. Wang, L.Y. Huang and X.C. Tian, An ontology-based method of knowledge model-ling for remanufacturing process planning, *Journal of Cleaner Production* **258** (2020). https://doi.org/10.1016/j.jclepro.2020.120952.

[5] H. Huicong, L. Ying, F.L. Wen and G. Xin, A knowledge-based approach toward representation and archiving of aesthetic in-formation for product conceptual design, *Journal of Compu-ting and Information Science in Engineering* **22** (2022). https://doi.org/10.1115/1.4053674.

[6] J. Jia, Y. Zhang and M. Saad, An approach to capturing and reus-ing tacit design knowledge using relational learning for knowledge graphs, *Advanced Engineering Informatics* **51** (2022), 101505. https://doi.org/10.1016/j.aei.2021.101505.

[7] X. Li, M. Lyu, Z. Wang, C.-H. Chen and P. Zheng, Exploiting knowledge graphs in industrial products and services: A sur-vey of key aspects, challenges, and future perspectives, *Computers in Industry* **129** (2021), 103449. https://doi.org/10.1016/j.compind.2021.103449.

[8] P. Sun and L. Gu, Fuzzy knowledge graph system for artificial intelligence-based smart education, *Journal of Intelligent & Fuzzy Systems* **40** (2021), 2929–2940. .

[9] L. Diao, W. Yang, P. Zhu, G. Cao, S. Song and Y. Kong, The research of clinical temporal knowledge graph based on deep learning, *Journal of Intelligent & Fuzzy Systems* **41** (2021) 4265–4274. https://doi.org/10.3233/JIFS-189687.

[10] J. Hao, L. Zhao, J. Milisavljevic-Syed, Z. Ming, Integrating and navigating engineering design decision-related knowledge using decision knowledge graph, *Advanced Engineering In-formatics* **50** (2021), 1474–0346. https://doi.org/10.1016/j.aei.2021.101366.

[11] Y. Zhang, X. Liu, J. Jia and X. Luo, Knowledge representation framework combining case-based reasoning with knowledge graphs for product design, *Comput-Aided Des Appl* **17** (2019) 763–782. https://doi.org/10.14733/cadaps.2020.763-782.

[12] A. Huet, R. Pinquié, P. Véron, A. Mallet and F. Segonds, CAC-DA: A knowledge graph for a context-aware cognitive design assistant, *Computers in Industry* **125** (2021). https://doi.org/10.1016/j.compind.2020.103377.

[13] L. Guo, F. Yan, T. Li, et al., An automatic method for con-structing machining process knowledge base from knowledge graph, *Robotics and Computer-Integrated Manufacturing* **73** (2022). https://doi.org/10.1016/j.rcim.2021.102222.

[14] L. Guo, F. Yan, Y. Lu, et al., An automatic machining process decision-making system based on knowledge graph, *Interna-tional Journal of Computer Integrated Manufacturing* **12** (2021), 1348–1369. https://doi.org/10.1080/0951192X.2021.1972461.

[15] Z.X. Wang, C.H. Chen, P. Zheng, X.Y. Li and L.P. Khoo, A novel data-driven graph-based requirement elicitation frame-work in the smart product-service system context, *Advanced Engineering Informatics* **42** (2019). https://doi.org/10.1016/j.aei.2019.100983.

[16] X.Y. Li, C.H. Chen, P. Zheng, Z.X. Wang, Z.H. Jiang and Z.X. Jiang, A knowledge graph-aided concept-knowledge approach for evolutionary smart product-service system development, *Journal of Mechanical Design* **142** (2020). https://doi.org/10.1115/1.4046807.

[17] X. Zhao, Y. Liu, Y. Xu, Y. Yang, X. Luo and C. Miao, Heteroge-neous star graph attention network for product attributes pre-diction, *Advanced Engineering Informatics* **51** (2022) 101447. https://doi.org/10.1016/j.aei.2021.101447.

[18] K. Ahn and J. Park, Idle vehicle rebalancing in semiconductor fabrication using factorized graph neural network reinforce-ment learning, *IEEE 58th Conference on Decision and Control* (2019), 132–138. https://doi.org/10.1109/CDC40024.2019.9030245.

[19] M. Kuhn, F. Funk, G.L. Zhang and J. Franke, Blockchain-based application for the traceability of complex assembly structures, *Journal of Manufacturing Systems* **59** (2021), 617–630. https://doi.org/10.1016/j.jmsy.2021.04.013.

[20] Q.W. Bao, G. Zhao, Y. Yu and P. Zheng, A node2vec-based graph embedding approach for unified assembly process in-formation modeling and workstep execution time prediction, *Computers & Industrial Engineering* **163** (2022). https://doi.org/10.1016/j.cie.2021.107864.

[21] B. Zhou, B. Hua, X. Gu, Y. Lu, T. Peng, Y. Zheng, X. Shen and J. Bao, An end-to-end tabular information-oriented causality event evolutionary knowledge graph for manufacturing docu-ments, *Advanced Engineering Informatics* **50** (2021), 101441. https://doi.org/10.1016/j.aei.2021.101441.

[22] T.K. Tran, C.D.C. Ta and T.T. Phan, Simplified effective method for identifying semantic relations from a knowledge graph, *Journal of Intelligent & Fuzzy Systems* (2022). https://doi.org/10.3233/JIFS-219288.

[23] Z. Liu, Y. Li, L. Zhao and P. Guo, Construction of intelligent query system for metro electromechanical equipment faults based on the knowledge graph, *Journal of Intelligent & Fuzzy Systems* **41** (2021), 4351–4368. https://doi.org/10.3233/JIFS-189695.

[24] U. Persad, P.M. Langdon and P.J. Clarkson, Inclusive design evaluation and the capability-demand relationship, *Designing Accessible Technology* (2006). https://doi.org/10.1007/1-84628-365-5_18.

[25] Y.H. Wang, S.H. Yu, F.M. Cheng, Z. Liu, D.K. Chen, J.J. Chu, N. Ma and Y.H. Chen, A design knowledge management model for civil aircraft cabin based on markov mogic networks, *International Journal of Computer Integrated Manufacturing* **33** (2020), 732–750. https://doi.org/10.1080/0951192x.2019.1699253.

[26] X.J. Chen, S.B. Jia and Y. Xiang, A review: knowledge reasoning over knowledge graph, *Expert Systems with Applications* **141** (2020). https://doi.org/10.1016/j.eswa.2019.112948.

[27] H.T. Li, Y. Wang, S.H. Zhang, Y.Q. Song and H.M. Qu, KG4Vis: a knowledge graph-Based approach for visualization recom-mendation, *IEEE Transactions on Visualization and Computer Graphics* **28** (2022), 195–205. https://doi.org/10.1109/Tvcg.2021.3114863.

[28] H.W. Wang, M. Zhao, X. Xie, W.J. Li and M.Y. Guo, Knowledge graph convolutional networks for recommender systems, *Pro-ceedings of the World Wide Web Conference* (2019), 3307–3313. https://doi.org/10.1145/3308558.3313417.

[29] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt and B. Scholkopf, Support vector machines, *IEEE Intelligent Systems and their Applications* **13** (1998), 18–28. https://doi.org/https://doi.org/10.1109/5254.708428.

[30] L. Rabiner and B. Juang, An introduction to hidden Markov mod-els, *IEEE ASSP Magazine* **3** (1986), 4–16. https://doi.org/https://doi.org/10.1109/MASSP.1986.1165342.

[31] H.M. Wallach, Conditional random fields: an introduction, *Technical Reports* (2004), 22.

[32] J.H. Dou, J.Y. Qin, Z.X. Jin and Z. Li, Knowledge graph based on domain ontology and natural language processing technol-ogy for chinese intangible cultural heritage, *Journal of Visual Languages and Computing* **48** (2018), 19–28. https://doi.org/10.1016/j.jvlc.2018.06.005.

[33] L.F. Li, P. Wang, J. Yan, Y. Wang, S.M. Li, J.P. Jiang, Z. Sun, B.Z. Tang, T.H. Chang, S.H. Wang and Y.T. Liu, Real-world data medical knowledge graph: construction and applications, *Artificial Intelligence in Medicine* **103** (2020). https://doi.org/10.1016/j.artmed.2020.101817.

[34] J. Devlin, M.W. Chang, K. Lee and K. Toutanova, BERT: pre-training of deep bidirectional transformers for language under-standing, *arXiv* (2018). https://doi.org/10.48550/arXiv.1810.04805.

[35] Z.L. Yang, Z.H. Dai, Y.M. Yang, J. Carbonell, R. Salakhutdinov and Q.V. Le, XLNet: generalized autoregressive pretraining for language understanding, *Proceedings of the Advances in Neural Information Processing Systems* (2019). https://doi.org/10.48550/arXiv.1906.08237.

[36] S. Ronnqvist, N. Schenk and C. Chiarcos, A recurrent neural model with attention for the recognition of chinese implicit discourse relations, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (2017), 256–262. https://doi.org/10.18653/v1/P17-2040.

[37] Z.R. Li, N. Ding, Z.Y. Liu, H.T. Zheng and Y. Shen, Chinese relation extraction with multi-grained information and external linguistic knowledge, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), 4377–4386. https://doi.org/10.18653/v1/P19-1430.

[38] S.C. Zheng, F. Wang, H.Y. Bao, Y.X. Hao, P. Zhou and B. Xu, Joint extraction of entities and relations based on a novel tag-ging scheme, *arXiv* (2017). https://doi.org/10.48550/arXiv.1706.05075.

[39] M. Miwa and M. Bansal, End-to-end relation extraction using LSTMs on sequences and tree structures, *arXiv* (2016). https://doi.org/10.48550/arXiv.1601.00770.

[40] X.R. Zeng, D.J. Zeng, S.Z. He, K. Liu and J. Zhao, Extracting relational facts by an end-to-end neural model with copy mechanism, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (2018), 506–514. https://doi.org/10.18653/v1/P18-1047.

[41] X.R. Zeng, S.Z. He, D.J. Zeng, K. Liu, S.P. Liu and J. Zhao, Learning the extraction order of multiple relational facts in a sentence with reinforcement learning, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Pro-cessing and the 9th International Joint Conference on Natural Language Processing* (2019), 367–377. https://doi.org/10.18653/v1/D19-1035.

[42] B. Du, B. Lin, C.M. Zhang, B.T. Dong and W.D. Zhang, Safe deep reinforcement learning-based adaptive control for USV interception mission, *Ocean Engineering* **246** (2022). https://doi.org/https://doi.org/10.1016/j.oceaneng.2021.110477.

[43] B. Du, Y. Lu, X. Cheng, W. Zhang and X. Zou, The object-oriented dynamic task assignment for unmanned surface vessels, *Engineering Applications of Artificial Intelligence* **106** (2021), 104476. https://doi.org/https://doi.org/10.1016/j.engappai.2021.104476.

[44] T.J. Fu, P.H. Li and W.Y. Ma, GraphRel: modeling text as rela-tional graphs for joint entity and relation extraction, *Proceedings of the 57th Annual Meeting of the Association for Com-putational Linguistics* (2019), 1409–1418. https://doi.org/10.18653/v1/P19-1136.

[45] X.Y. Li, F. Yin, Z.J. Sun, X.Y. Li, A. Yuan, D. Chai, M.X. Zhou and J.W. Li, Entity-relation extraction as multi-turn question answering, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), 1340–1350. https://doi.org/10.48550/arXiv.1905.05529.

[46] Z.P. Wei, J.L. Su, Y. Wang, Y. Tian and Y. Chang, A novel cas-cade binary tagging framework for relational triple extraction, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), 1476–1488. https://doi.org/10.48550/arXiv.1909.03227.

[47] K.Y. Kim, D.G. Manley and H. Yang, Ontology-based assembly design and information sharing for collaborative product de-velopment, *Computer-Aided Design* **38** (2006), 1233–1250. https://doi.org/10.1016/j.cad.2006.08.004.

[48] M. Doerr, The CIDOC conceptual reference module: an onto-logical approach to semantic interoperability of metadata, *AI Magazine* **24** (2003) 75–75. https://doi.org/https://doi.org/10.1609/aimag.v24i3.1720.

[49] S.J. Luo and Y.N. Dong, Integration and management method of cultural artifacts knowledge for cultural creative design, *Computer Integrated Manufacturing Systems* **24** (2018), 964–977.

[50] B. Liang, C.N. van der Wal, K. Xie, et al., Mapping the knowledge domain of soft computing applications for emer-gency evacuation studies: A scientometric analysis and critical review, *Safety Science* **158** (2023), 105955. https://doi.org/10.1016/j.ssci.2022.105955

[51] K. Clark, T.L. Minh, V.L. Quoc and D.M. Christopher, ELEC-TRA: pre-training text encoders as discriminators rather than generators, *Proceedings of the 10th International Conference on Learning Representations* (2020). https://doi.org/10.48550/arXiv.2003.10555.

[52] Y.H. Liu, M. Ott, N. Goyal, J.F. Du, M. Joshi, D.Q. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, Roberta: a ro-bustly optimized bert pretraining approach, *arXiv* (2019). https://doi.org/10.48550/arXiv.1907.11692.

[53] D.P. Kingma and J. Ba, Adam: a method for stochastic optimization, *arXiv* (2014). https://doi.org/10.48550/arXiv.1412.6980.

[54] S.J. Li, W. He, Y.B. Shi, W.B. Jiang, H.J. Liang, Y. Jiang, Y. Zhang, Y.J. Lyu and Y. Zhu, DuIE: a large-scale chinese dataset for information extraction, *International Conference on Natu-ral Language Processing and Chinese Computing* (2019), 791–800. https://doi.org/10.1007/978-3-030-32236-6_72.

[55] G. Bekoulis, J. Deleu, T. Demeester and C. Develder, Joint entity recognition and relation extraction as a multi-head selection problem, *Expert Systems with Applications* **114** (2018), 34–45. https://doi.org/https://doi.org/10.1016/j.eswa.2018.07.032.

[56] Z.H. Huang, W. Xu and K. Yu, Bidirectional LSTM-CRF models for sequence tagging, *arXiv* (2015). https://doi.org/10.48550/arXiv.1508.01991.

[57] T. Mikolov, L. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems* **26** (2013). https://doi.org/10.48550/arXiv.1310.4546.