actuators

MDPI

Article

Hierarchical Understanding in Robotic Manipulation: A Knowledge-Based Framework

Runqing Miao [1], Qingxuan Jia [1,*], Fuchun Sun [2], Gang Chen [1] and Haiming Huang [3]

[1] School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China
[3] College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China
* Correspondence: qingxuan@bupt.edu.cn

Abstract: In the quest for intelligent robots, it is essential to enable them to understand tasks beyond mere manipulation. Achieving this requires a robust parsing mode that can be used to understand human cognition and semantics. However, the existing methods for task and motion planning lack generalization and interpretability, while robotic knowledge bases primarily focus on static manipulation objects, neglecting the dynamic tasks and skills. To address these limitations, we present a knowledge-based framework for hierarchically understanding various factors and knowledge types in robotic manipulation. Using this framework as a foundation, we collect a knowledge graph dataset describing manipulation tasks from text datasets and an external knowledge base with the assistance of large language models and construct the knowledge base. The reasoning tasks of entity alignment and link prediction are accomplished using a graph embedding method. A robot in real-world environments can infer new task execution plans based on experience and knowledge, thereby achieving manipulation skill transfer.

Keywords: robotic manipulation; knowledge representation; knowledge update; knowledge reasoning

check for updates

Citation: Miao, R.; Jia, Q.; Sun, F.; Chen, G.; Huang, H. Hierarchical Understanding in Robotic Manipulation: A Knowledge-Based Framework. *Actuators* **2024**, *13*, 28. https://doi.org/10.3390/act13010028

Academic Editor: Zhuming Bi

Received: 18 December 2023
Revised: 6 January 2024
Accepted: 8 January 2024
Published: 10 January 2024

1. Introduction

Task understanding and skill refinement are crucial abilities for service robots. The decomposition of tasks in a manner similar to human cognition can be transformed into a planning problem within a symbolic space. Task and motion planning (TAMP) is the mainstream method for handling long-term tasks in robotic manipulation, relying on predefined planning domains, symbolic rules, and complex strategy searches. The limitation of such methods lies in the requirement for robots to possess a comprehensive model prior to task execution, thereby impeding their ability to achieve skill transfer and generalization, as well as adaptability to dynamically the evolving task scenes.

In order to address this issue, the knowledge-based approaches have been considered for the representation and task planning of robotic manipulation. Semantic knowledge serves as a medium for skill transfer among humans, providing concise explanations of the world. Knowledge bases effectively express and store the experiences generated in human or robotic manipulation, enabling reasoning and reuse. However, the discrete nature of knowledge poses challenges in directly describing the continuously manipulated data. The existing knowledge-based methods for robotic manipulation focus on characterizing static objects without achieving reasonable decoupling between the different factors. In querying and reasoning, they solely rely on rule-based symbolic computation. To properly represent the complex manipulation knowledge, it is essential to simultaneously consider both the continuous and discrete data as well as the static and dynamic factors. Additionally, robots need to acquire human knowledge and record the existing experiences in order to achieve real-time responses to new tasks and continuous updates.

We propose a knowledge-based framework for a hierarchical understanding of human and robotic manipulation. The framework represents the factors in manipulation in a

hierarchical structure, the object, agent, scene, task, skill, and action, as well as different knowledge types, the ontology, template, and instance. Knowledge is extracted and fused from the life guide dataset wikiHow and external knowledge base DBpedia using large language models (LLMs) to accomplish multi-source knowledge updates. As a result, we create a knowledge graph dataset named SkillKG consisting of 13,154 triples, representing 984 entities and 18 relations. The framework is integrated with the dataset results for the establishment of a comprehensive knowledge base for robotic manipulation. To address entity alignment and link prediction as reasoning tasks, we propose a graph embedding method for representation learning to extract feature and structured information from the nodes in the knowledge bases, thereby generating embeddings. By utilizing prior knowledge, it is possible to predict the action sequence of a robot in similar tasks involving novel objects, thus enabling skill transfer. In the evaluation and experiments, we initially assess the query performance of our knowledge base. Using the generated embeddings, the real-world robot predicts action sequences for new tasks. An UR5 robot achieves an accuracy of 91.7% on action sequence prediction and an accuracy of 81.8% on execution. Figure 1 shows a pipeline of our work in this paper.

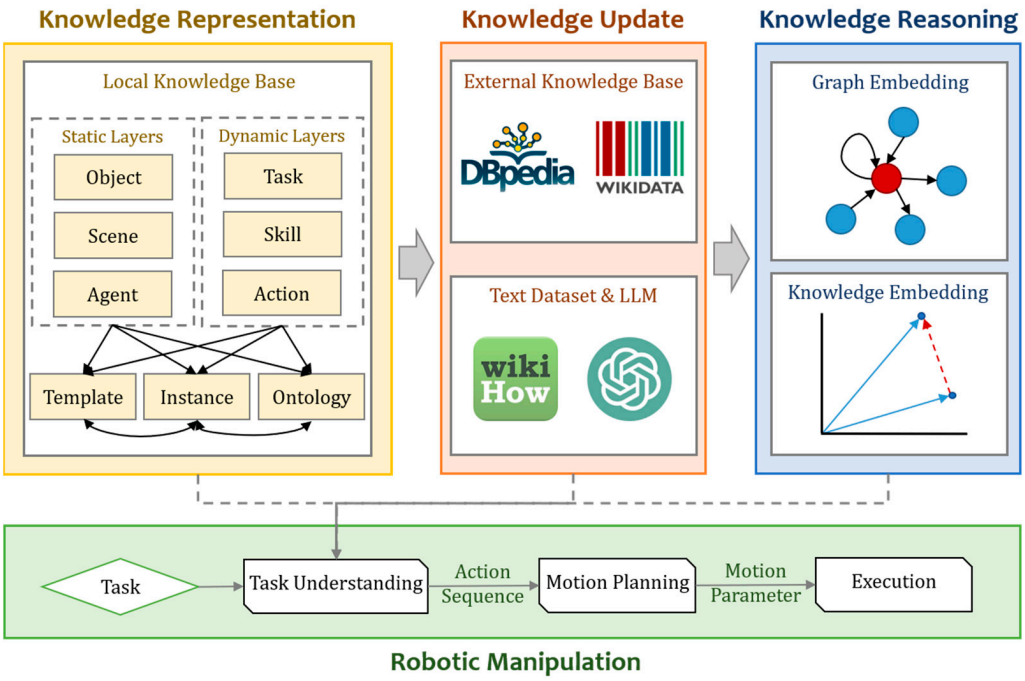


Figure 1. Pipeline of our work. It encompasses the entire process from representation (hierarchical framework) to updating (data fusion), and then inference (graph embedding). Ultimately, it facilitates task understanding in robotic manipulation.

Our contributions encompass: (a) a hierarchical knowledge-based framework that represents the different manipulative factors and knowledge types; (b) a knowledge graph dataset and base that integrates information from the text datasets and external knowledge bases using large language models; (c) a graph embedding method for entity alignment and link prediction; and (d) the evaluation of the proposed framework on a real-world robotic manipulation platform.

The rest of the paper is organized as follows: Section 2 summarizes the recent advances in robotic manipulation with knowledge; Sections 3–5 discuss representation, updating, and reasoning within our framework, respectively; the evaluation, experiments, and discussion are presented in Section 6; and we draw the final conclusions in Section 7.

2. Related Works

2.1. Knowledge Representation in Robotic Manipulation

Robotic manipulation involves three levels, ranging from high-level understanding and task planning to mid-level strategy and behavior planning and down to low-level execution. Knowledge, as a representation of the condensed data and information, primarily corresponds to the top level. The knowledge base provides the semantic context for the robots' input and output in their tasks, including defining the meaning or function of the manipulated objects. The early robot knowledge bases primarily focused on static objects, such as RoboEarth [1,2], KnowRob [3,4], RoboBrain [5], and the recently developed articulated object knowledge base AKB-48 [6]. However, all the aforementioned knowledge bases for robots are large-scale static repositories primarily focused on describing stationary objects in robotic manipulation tasks. The graph composition is excessively intricate and convoluted, leading to a heightened level of complexity when querying. Several knowledge representation methods have been proposed, specifically targeting the behaviors of robots. Action tree bank [7] generates a symbolic high-level representation in the form of a tree, encompassing the knowledge derived from demonstrations. FOON [8] is a structured method for representing the knowledge the models' objects and their movements in manipulation tasks, constructed through the manual annotation of instructional videos. Instead of a comprehensive symbolic representation system, several works apply knowledge to different robotics tasks, such as vision [9], grasping [10,11], assembly [12], and path planning [13,14], to describe robots' behavior processes. Furthermore, the utilization of knowledge graph embedding enables inference and finds application in the field of robotics [15]. However, its efficacy is constrained by the scale of the knowledge graph.

Currently, there are several key challenges facing robot knowledge graphs. Firstly, the discretization of continuous data results in a lack of proper decoupling between the high-level semantics and low-level data. Secondly, the dynamic changes in relations overlook the impact of robot actions on object relations. Thirdly, traditional symbolic computation is the only consideration when querying knowledge graphs. We propose a hierarchical architecture for our knowledge-based framework, which achieves a layered decoupling of knowledge manipulation and the data, while also considering the dynamic factors in robotic manipulation. Moreover, the utilization of a state-of-the-art graph database as the foundation for our knowledge graph enhances the query speed.

2.2. Knowledge Sources of Robotic Manipulation

The hierarchical organization of robotic manipulation skills determines its complexity. In our framework, manipulation knowledge is decoupled into two categories: static and dynamic. Static knowledge describes the stable common-sense knowledge obtained from resources, such as the internet, general knowledge databases, and vertical domain databases. Dynamic knowledge describes the continuously changing entity state process associated with actions generated based on existing experiences or real-time observations. Knowledge can be derived from semantically annotated descriptions by humans or different types of sensors, such as cameras, force sensors, and light-sensitive tactile sensors.

Manipulation knowledge originates from various sources due to its diverse types, primarily encompassing the following three categories: (1) Human-constructed manipulated datasets, which involve focused, single tasks, such as Push [16] and Bigs [17], and those encompassing multiple tasks like RoboTurk [18], MIME [19], and RoboNet [20]. (2) Common knowledge in the general knowledge base. In addition to domain-specific knowledge bases, there are publicly available cross-domain knowledge bases that serve as encyclopedias, such as the language knowledge base WordNet [21]; the concept knowledge base ConceptNet [22,23]; the world knowledge bases Freebase [24], Wikidata [25], DBpedia [26], and YAGO [27]; and others. These common knowledge bases encompass a vast amount of general information, including object definitions, classifications, and functionalities. (3) LLMs refers to transformer language models with trillions or more parameters, which are trained on extensive text data, such as GPT-3 [28], PaLM [29], and LLaMA [30]. These

models exhibit exceptional proficiency in comprehending natural language and tackling intricate problems [31]. Because LLMs have extensive prior knowledge and reasoning capabilities, they facilitate knowledge extraction and update. LLMs have already found applications in robotic manipulation tasks. Text2Motion [32] accomplishes the end-to-end planning of robotic manipulation tasks using LLMs. Wu et al., on the other hand, integrated language-based planning and perception with the limited summarization capability of LLMs to facilitate robots in understanding the users' preferences [33].

2.3. Knowledge Reasoning with Representation Learning

Knowledge reasoning involves deriving new knowledge or conclusions from the existing knowledge using various methods. Knowledge reasoning methods via representation learning are primarily implemented through translating models and graph embedding. Translating models is based on word2vec [34]. TransE [35] utilizes the concept of translating invariance within a word vector space, considering the relations in knowledge bases as translating vectors between entities. Graph networks are primarily used for tasks in non-Euclidean spaces, which align well with the topological graph structure of knowledge graphs. Graph embedding, also known as graph representation learning, expresses the nodes in a graph as low-dimensional dense vectors. It necessitates that the nodes with similar characteristics in the original graph are also close to each other in the low-dimensional representation space. The output expression vector can be used for downstream tasks, such as entity alignment [36] and knowledge fusion [37]. The most classic graph embedding method is DeepWalk [38], which utilizes random walks to sample nodes within the graph and acquire co-occurrence relations among them. Furthermore, there are other methods such as node2vec [39] and LINE [40]. We assess the advantages and disadvantages of the above methods, and combined with the hierarchical structure of our framework, we design a suitable embedding method to accomplish knowledge reasoning.

3. Hierarchical Knowledge Representation

The objective of this work is to establish a comprehensive knowledge-based framework for understanding robotic manipulation. The primary aim of this framework is to systematically analyze and represent the various factors involved in everyday manipulation processes, while aligning with human cognition. On the one hand, it necessitates the consideration of numerous factors pertaining to both human and robotic manipulation. On the other hand, it requires the characterization of multiple types of knowledge data. To address these challenges, we employ a hierarchical design approach that facilitates decoupling between these two dimensions, thereby laying a solid foundation for the ontology construction of the knowledge base.

3.1. Hierarchical Manipulation Factors

We categorize the factors involved in manipulation into two groups: static and dynamic. Static factors refer to concrete and stable elements, such as objects, agents, and scenes in the environment, while dynamic factors pertain to abstract and variable aspects, including the tasks, skills, and actions required for manipulation. We present a definition of the process of manipulation that aligns with human cognition.

$$M = \left\{ S_{object}, S_{agent}, S_{scene}, D_{task}, D_{skill}, D_{action} \right\} \tag{1}$$

In a given scene S_{scene}, an agent S_{agent} aims to accomplish task D_{task} by utilizing a series of skills D_{skill} to manipulate a set of objects S_{object} through performing a sequence of actions D_{action}. We decompose different factors in knowledge representation based on their dynamic and static relations, forming six interconnected levels of knowledge.

3.1.1. Static Layers

The object layer serves as the representation of manipulated objects, which typically refer to physical entities. This aspect of knowledge representation in robotic manipulation has reached a high level of maturity. We make use of common sense knowledge bases, such as DBpedia and Wikidata, along with manipulation datasets like YCB [41] and the object knowledge base AKB-48. The first part expresses the fundamental semantic information about a manipulated object, including its name, description, and superclass. The second part encompasses the physical properties of the object, such as size, weight, shape, color, material composition, etc. The final part includes visual modalities related to appearance, such as 3D mesh and multi-view RGB images. The first two parts are stored in the knowledge base ontology, while the last part is stored in the server, storing the link to the file location in the properties of the object entity.

The agent layer serves as a representation of the subjects involved in manipulation, primarily humans and robots. In terms of human manipulation, body parts such as the hands are commonly utilized. The hardware and software configurations of robots, including various models of mechanical arms, end effectors, cameras, and tactile sensors, play a crucial role in determining the feasibility of manipulation tasks.

The scene layer serves as a representation of the background space for manipulation, encompassing two categories: the broad sense environment, which pertains to indoor spaces where human or robotic activities occur (e.g., kitchens and workshops), and the narrow sense region, which includes relevant information about the manipulation platform, such as desktops, lighting, and artificially divided platform areas.

3.1.2. Dynamic Layers

The task layer serves as a representation of the purpose of manipulation, with tasks in the knowledge base being named using a verb–object structure ("Action_Object") to facilitate direct connection to corresponding entities through their names. The simple tasks involve only one object, such as "Pour_Juice" or "Insert_Key", while the complex tasks may involve multiple objects, such as "Make_ Drink". The task ontology only has a name and description, while the task template and instance contain action sequences and all the starting, intermediate, and ending states—a complete sequence of state transitions. These action sequences consist of action primitives linked to the relevant objects, agents, and scenes. The states are primarily described by visual scene graphs along with coordinates and postures for objects.

The skill layer serves as a representation of the condensed prior knowledge in manipulation. Skills are generalizations of the topological structure underlying similar tasks. Simple tasks are derived from skills at the same level, such as the skill "Peg-in-Hole", which can be used to derive the task "Insert_Key", both involving a sequence of actions like "pick-align-insert". Complex tasks encompass multiple skills and act as their superclass. For instance, the task "Make_Coffee" may incorporate several skills, including "Pour_Powder", "Pour_Water", and "Stirring".

The action layer serves as a representation of the fundamental primitives involved in manipulation. For robots, these action primitives are low-level task units that can be directly implemented with classical motion planning algorithms. For humans, they represent the smallest modules of semantic segmentation. The ordered arrangement of these action primitives enables the composition of tasks and skills, whether they are simple or complex.

3.2. Hierarchical Knowledge Types

We have previously explored hierarchical understanding based on manipulation factors. Considering the knowledge from another perspective, it also encompasses various types, such as conceptual descriptions or specific instances within a task. Consequently, this requires a hierarchical representation based on the types of knowledge, namely from ontology to templates, and then instances.

The ontology is utilized for the representation of abstract conceptual knowledge, which is akin to a dictionary or a classical common sense knowledge base. It serves as a repository for semantic knowledge pertaining to the entities themselves, organizing diverse entity types based on concept definitions, while maintaining strict categorization and hierarchical relations.

The template is utilized to represent a variety of manipulation tasks or skills, wherein each task or skill constitutes a manipulation process with an action sequence as its fundamental structure. This process encompasses the scene where the manipulation takes place, along with the agents and objects associated with each action primitive. Consequently, it necessitates establishing logical or temporal relationships between the nodes at different levels. Each entity within the template is linked to its corresponding counterpart in the ontology through "instanceof", thereby indicating the instantiation of concepts in manipulations. The ontology and template knowledge are enriched by the knowledge graph dataset presented in Section 4.

The instance is utilized to represent each execution, which is akin to a log. They incorporate execution parameters and timestamps based on the corresponding template, with all the entities and relations in the instance mapped from those in said template. Each manipulation task execution generates an instance, thereby allowing for continued knowledge accumulation throughout the process.

In summary, the data structure of the hierarchical knowledge-based framework achieves decoupling from both the manipulation elements and types of knowledge. This ensures the stability and flexibility of the architecture, thereby facilitating ontology construction and subsequent knowledge updates.

4. Multi-Source Knowledge Update

After devising a hierarchical framework, in order to sufficiently expand the scale of the knowledge base for reasoning purposes, we update and complete our framework by incorporating knowledge from various data sources, particularly focusing on template-based manipulation tasks. Upon completing knowledge acquisition and integration, we enhance the hierarchical framework using SkillKG, a task-centric manipulation knowledge graph dataset, and subsequently update the knowledge base. In the construction process, the existing knowledge from multiple sources, such as manipulation text datasets, LLMs, and general knowledge bases, is extensively utilized. The main process is shown in Figure 2.

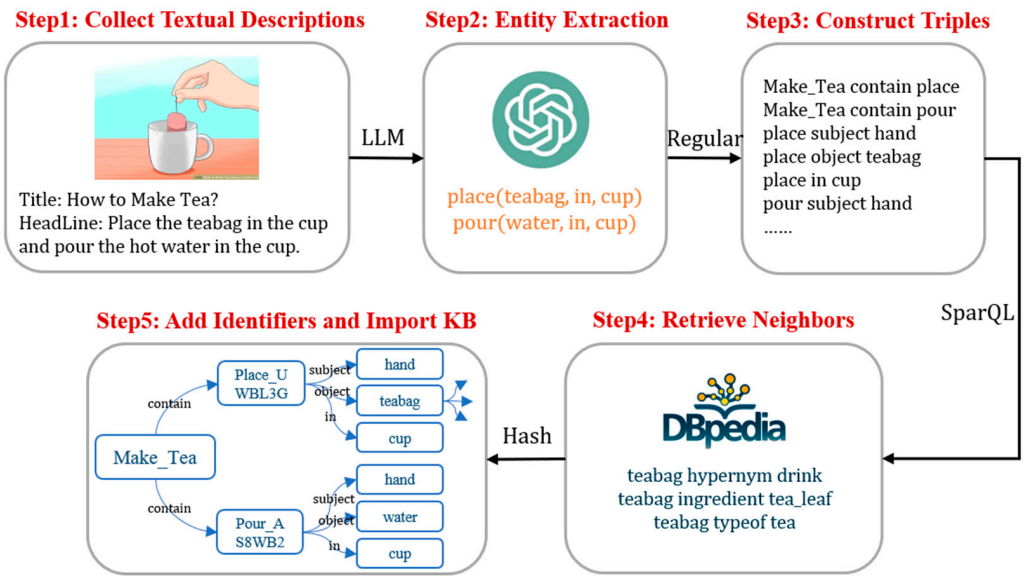


Figure 2. An illustration of data collection and preprocessing steps used to create SkillKG.

4.1. Task Text Collection

Collect Textual Descriptions. We gather manipulation tasks, comprising textual descriptions of the manipulation process. wikiHow serves as an open-source life guide that functions as a large-scale text summarization dataset. We specifically selected guides for physically and simply manipulative tasks, where the object being manipulated undergoes changes in state or position during a task process, such as "how to brew tea" and "how to cut apples". This excludes virtual manipulations like "how to access email", and even more abstract complex non-manipulation tasks like "how to eat healthily". Following this step, we acquired structured text data for 317 manipulative tasks encompassing daily life, kitchen-related activities, and industrial assembly scenes.

4.2. Entity Extraction and Triplet Construction Based on Large Language Models

Entity extraction refers to extracting labels from textual data. We extract the labels from text data in wikiHow, which includes a Title, Headline, and text. The task name is extracted from the Title, while the action sequence is derived from the Headline. We utilize wikiHow's textual data as the input and extracted keywords using LLMs with a prompt, leveraging their text comprehension capabilities. The resulting output, as shown in Figure 3, includes verb sequences with corresponding nouns and prepositions for each manipulation. The LLM we use is text-davinc-003, which is a variant of instructGPT [42] based on GPT-3. Additionally, we introduce the suffixes ".t", ".a", ".i", ".o", and ".l" to differentiate the words related to tasks, actions, subjects, objects, and scenes, respectively.

"Place the teabag in the cup and pour the hot water in the cup. "
Extract all the verbs in the sentence, with the nouns and preposition associated with each verb.
Format: cut(apple, with, knife)

place(teabag, in, cup)
pour(water, in, cup)

Figure 3. An example of LLM inputs, prompts, and outputs.

Subsequently, triplets are constructed based on the identified entities and relations. Triples are the fundamental building blocks of a knowledge graph and take the form of 'entity-relation-entity'. Using a single task template as an example, "Brew_Teabag.t" comprises two actions in a sequence: "place" (with "teabag.o" as the object and "in cup.o" as the target) and "pour" (with "hot_water.o" as the object and "in cup.o" as the target). Based on these data, we derive a set of triples for each manipulation task that represent the parent–child relations between tasks and actions using the relation "contain", sequential relations between actions using "next", action subjects or performers using "subject", action objects using "object", and action targets or locations through relations, such as "from", "in", "on", and "beside".

4.3. Knowledge Fusion Based on External Knowledge Base

We expand knowledge by retrieving neighbors from the common sense knowledge base. Although the entity nodes themselves are singular, connecting the subject, object, and location entities with DBpedia through SparQL allows us to obtain more relevant contextual information. In DBpedia, we retrieve neighbor nodes by restricting the triple relationships with "typeof", "hypernym", and "ingredient" in order to acquire the categories, contexts, and components of local entities as additional knowledge for our dataset. For instance, the examples include "toothbrush.o typeof toiletries.o", "cabbage.o hypernym plant.o", and "noodle.o ingredient leavening_agent.o".

However, the direct retrieval method based on string matching is limited to obtaining nodes from the external knowledge base that have explicit associations with nodes in the local knowledge base. It fails to retrieve differently named nodes representing the same or similar entities. The crucial challenge that knowledge fusion must address is how to

identify the implicitly associated nodes and align them with local entity nodes. We propose a method based on a breadth-first search to construct neighborhoods for generating entity embeddings, thereby achieving alignment and providing implicitly associated nodes for knowledge fusion. For more details about this method, please refer to Section 5.

4.4. Identifier Addition and Knowledge Update

In robotic manipulation execution instances, it is possible to have actions with similar semantics, but different parameters under various tasks. If these actions are all linked to the same node, it can lead to the excessive coupling of data structures in the templates of the knowledge-based framework. To address this issue, unique identifiers in the form of randomly generated hash suffixes are attached to each action within every template. With this approach, dataset construction concludes successfully.

Finally, we import these template data into the knowledge base. The meta nodes representing actions are included in the ontology for aggregating action entities with identical names from templates (e.g., "place_UWBL3G instanceof place"). Additionally, object entities from templates are also replicated within the ontology and connected through "instanceof". The expanded knowledge obtained in Section 4.3 is transferred to object entities within the ontology, as they pertain to specific concepts.

4.5. Dataset Statistic

SkillKG offers a wealth of manipulation knowledge, encompassing both the local knowledge extracted from wikiHow via LLM and the external knowledge retrieved from DBpedia. Please refer to Table 1 for detailed statistics. With a total of 13,154 triples, including 984 entities and 18 relations, SkillKG provides ample high-quality prior knowledge for hierarchical semantic representation aimed at robotic manipulation. The latest version of SkillKG is available at https://github.com/tsingmr/SkillKG, accessed on 1 January 2024.

Table 1. Data statistics and comparison of datasets SkillKG and THOR_U.

Dataset	Triple	Relation	Entity				
			Task	Action	Subject	Object	Scene
THOR_U	1964	15	/	27	/	114	4
SkillKG	13,154	18	317	59	7	595	6

4.6. Dataset Details

In Table 1, we compare SkillKG with THOR_U [15], presenting the number of triples, relations, and entities. The entity subclasses include task, action, subject, object, and location. THOR_U extracts semantic knowledge from the home domain simulation environment AI2THOR [43] for embedding purposes only. SkillKG primarily extracts manipulation skill knowledge from wikiHow and DBpedia for both embedding and constructing a comprehensive knowledge base. SkillKG has significant advantages in data scale and coverage over THOR_U. We also present the distribution of relation categories in Figure 4.

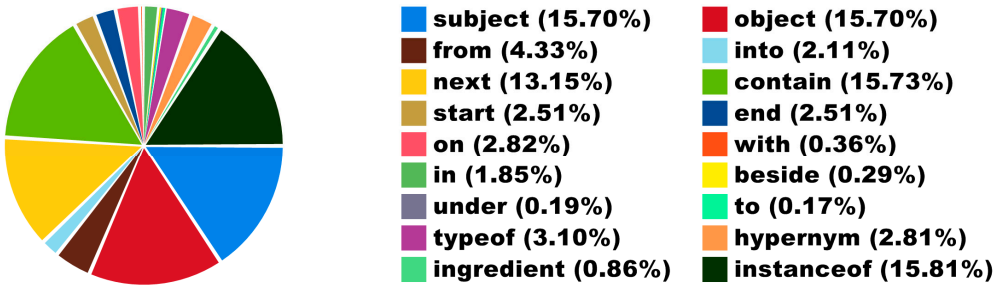


Figure 4. The proportion of relations in the dataset.

5. Embedding-Based Knowledge Reasoning

Retrieval-based reasoning, which relies on knowledge base queries, serves as a fundamental approach for knowledge reasoning. However, its effectiveness is constrained by the size of the knowledge base. Embedding-based methods possess the capability to represent the entities and relations within a continuous feature space. By transforming entities and relations into vectors, the downstream tasks classified as generative reasoning, such as entity alignment and link prediction, can be simplified through vector operations.

The existing dataset represents a small-scale knowledge graph, which can be considered as a multi-relational directed graph. In this graph, the entities are represented as nodes, while the relations are depicted as edges. The knowledge-based framework we propose, specifically designed hierarchically for robotic manipulation, exhibits a higher level of complexity in comparison to that of the general knowledge bases. Hence, when generating the embedding, it is imperative to not only consider explicit relations between the entities, but also comprehensively incorporate the structural information of nodes throughout the entire graph.

In contrast to DeepWalk, which utilizes depth-first search for constructing neighborhoods, we propose an LINE-based method that utilizes breadth-first search to generate entity embeddings for alignment. Embedding serves as prior knowledge for link prediction.

5.1. Entity Alignment

Given a knowledge graph $G(E,R)$ and a triplet set $\left\{(e_i,r,e_j)\middle|e_i \in E, e_j \in E, r \in R\right\}$ within it. The embedding is generated by constructing neighbor similarity. For a directed edge r, the probability of generating a context neighbor entity e_j under a given entity e_i is defined as:

$$p(e_i|e_j) = \frac{exp\left(\vec{v_j}^T \cdot \vec{v_i}\right)}{\sum_{k=1}^{|N|} exp\left(\vec{v_k}^T \cdot \vec{v_i}\right)} \tag{2}$$

where $\vec{v_i}$ is the lower-dimensional vector representation of entity e_i itself, and $|N|$ is the number of neighbor entities. Therefore, the optimization goal is defined as:

$$O = \sum_{i \in N} \lambda_i d(\hat{p_2}(\cdot|e_i), p(\cdot|e_i)) \tag{3}$$

where λ_i is the factor that controls the entity weight, which can be represented by the degree of the entity node. The empirical distribution is defined as:

$$\hat{p}(e_j|e_i) = \frac{w_{ij}}{d_i} \tag{4}$$

where w_{ij} is the weight of relation edge r, and d_i is the out-degree of entity node e_i. Then, using Kullback–Leibler divergence, setting $\lambda_i = d_i$, and omitting the constant terms, the objective function is:

$$O = -\sum_{(i,j) \in E} w_{ij} log p(e_j|e_i)) \tag{5}$$

when calculating similarity, in order to optimize the disadvantage that the denominator calculation of softmax function requires complete traversal, negative sampling optimization is adopted. Then, the objective function is:

$$O = log\sigma\left(\vec{v_j}^T \cdot \vec{v_i}\right) + \sum_{i=1}^{K} E_{v_n P_n(v)}\left[-log\sigma\left(\vec{v_n}^T \cdot \vec{v_i}\right)\right] \tag{6}$$

where K is the number of negative edges. After the above calculation and optimization, the embedding $V(e_i)$ of each entity e_i in the knowledge graph is obtained. Then, the similarity

between the vectors is calculated to find the equivalent entity pair here using the Euclidean Distance, which is formulated as:

$$D = \sqrt{\left(V(e_i) - V(e_j)\right) \cdot \left(V(e_i) - V(e_j)\right)^T} \tag{7}$$

When the distance is less than the threshold, the two entities are regarded as similar entities to achieve alignment, and then the knowledge fusion is realized by retrieving the neighbor of similar entities. The graph-based embedding generation method can automatically extract equivalent entity pairs from knowledge graphs on a large scale without introducing a lot of artificial features.

5.2. Link Prediction

In link prediction, knowledge embedding models, such as DistMult [44] and TransE, define the scoring functions for triplets (e_h, r, e_t), which are continuously optimized to enhance the scores and rankings of the correct triplets. Our knowledge-based framework exhibits a more specific orientation compared to common sense knowledge bases. Therefore, we utilize TransE instead of DistMult as the scoring function for triplets:

$$F_{score}(e_h, r, e_t) = -\|h + r - t\|_{1/2} = -\left\|f(e_h) + r_{e_h,e_t} - f(e_t)\right\|_{1/2} \tag{8}$$

where e_h represents the head entity, e_h represents the tail entity, $\|.\|_{1/2}$ represents the L_1 and L_2 distances, $f(.)$ represents the feature vector after encoding entities, and r_{e_h,e_t} represents candidate relations between the head and tail entities.

During training, the embeddings generated by entity alignment serve as the initial values, and the model is iteratively optimized using cross-entropy loss. After optimization, new relations can be predicted by determining the validity of new triplets, while the accuracy of embeddings can be assessed through various metrics.

5.3. Verification of the Accuracy of Embeddings

The input for this experiment is derived directly from the dataset SkillKG, which encompasses the semantic knowledge stored in our ontology and templates within our knowledge representation framework. This dataset comprises a total of 13,154 triples, 984 entities, and 18 relations. We divide it into a training set, validation set, and test set at a ratio of 12:1:1. The batch size is configured to be 2000, while the epoch is set to 10,000. The Mean Reciprocal Ranking (MRR) and Hit@N are metrics utilized to assess the accuracy of embeddings. The ranking of correct triplets is positively correlated with the magnitude of MRR and Hit@N, indicating a higher level of accuracy in the embeddings. The precomputation process is shown in Algorithm 1: the precomputation process of the metrics of embedding accuracy.

Algorithm 1: Metrics of embedding accuracy

Input: Triples T, Entities E, Scoring function f
Output: Ranks $rank_i$

```
Scores S = ∅
foreach t_i = (e_h, r, e_t) ∈ T do
    add f(t_i) into S
    foreach e in E do
        add f(e_h, r, e) into S
        add f(e, r, e_t) into S
    end
    rank_i ← rank(f(t_i) in S)
end
MRR = (1/|S|) Σ_{i=1}^{|S|} 1/rank_i
Hit@N = (1/|S|) Σ_{i=1}^{|S|} I(rank_i ≤ n)
```

We present the results of MRR and Hits@1, Hits@3, and Hits@10 for SkillKG in Table 2. The baseline includes the bilinear model DistMult and the translating model TransE, which directly acquire explicit relations, while our approach integrates richer, contextualized, structured information. The results demonstrate that our graph embedding method surpasses the baseline in terms of the accuracy of the generated embeddings.

Table 2. Accuracy of embeddings.

Metric\Method		DistMult	TransE	Graph Embedding (Ours)
MRR		34.33%	41.29%	73.84%
Hits@	**1**	20.76%	23.70%	64.31%
	3	41.38%	49.83%	76.68%
	10	59.61%	72.25%	88.12%

In summary, we propose a graph-based embedding generation method for knowledge reasoning. The embeddings are initially generated through entity alignment and subsequently utilized for link prediction to assess the accuracy of the embeddings. Finally, we validate the accuracy of our generated embeddings on our dataset through a comparative experiment.

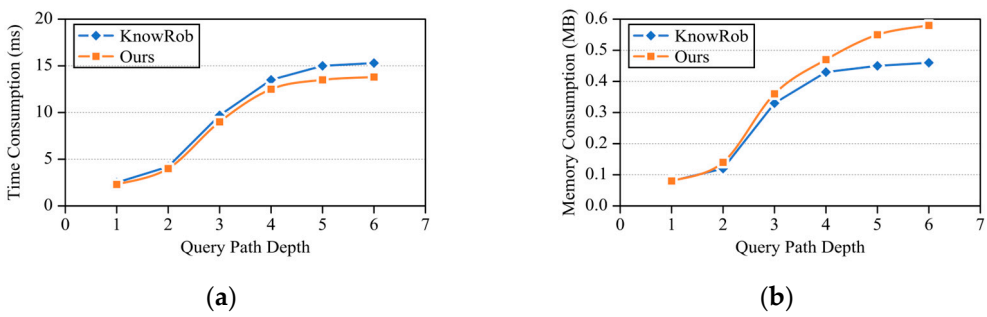
6. Evaluation, Experiments, and Discussion

Firstly, we construct a knowledge base within the hierarchical framework and evaluate its query performance. Subsequently, the embeddings obtained using the graph embedding method on the SkillKG dataset are used to enhance the skill transfer in robotic manipulation.

6.1. Knowledge Base Construction and Query Performance Evaluation

The knowledge-based hierarchical understanding framework was constructed using Neo4j [45], a graph database based on a property graph model. Initially, we established a hierarchical representation structure for the knowledge ontology and subsequently imported the SkillKG dataset, which incorporates the updates from multiple sources to successfully build the knowledge base.

Given that the knowledge-based framework operates within a real-time robotic manipulation scene, it is imperative for its knowledge base to function in real time as well in order to prevent any delays during execution. To assess the query performance of the knowledge base, we conducted tests on the query time and memory consumption at various depths of entity nodes using computers with identical configurations and networks. Each depth was queried five times and averaged, as depicted in Figure 5.



(**a**) (**b**)

Figure 5. Results of knowledge base query performance evaluation: (**a**) time consumption; (**b**) memory consumption.

The baseline used for comparison is KnowRob [3,4], a robot knowledge base based on OWL. The results demonstrate the superior efficiency of our attribute graph-based knowledge base compared to that of KnowRob in terms of the knowledge query time. The

average query time for different depths is consistently below 10 ms, thereby satisfying the real-time manipulation requirements of robots. In terms of knowledge query memory, both the knowledge bases exhibit similar memory consumption (approximately 0.1 MB) when the query path depth is low. However, as the query path depth increases, our knowledge base demonstrates a higher memory occupancy compared to that of KnowRob. The rationale behind this result lies in the fact that the attribute graphs possess higher data dimensions and more complex data storage formats, thereby require more computational resources. Nevertheless, in scenarios where hardware resources are abundant, the disparity in memory consumption exerts a negligible influence on the execution performance.

6.2. Knowledge-Based Robotic Manipulation Skill Transfer

Given a fixed scene, the robot is assigned manipulation tasks. If prior manipulation knowledge for the task exists in the knowledge base, the corresponding templates and instances are queried, and the action sequences and motion parameters are invoked and executed. In cases where there is no prior manipulation knowledge for the task, but similar task templates and instances of other objects have been stored, analogical skill transfer can still be employed to plan the manipulation task. We refer to this as object-centered skill transfer. Similar tasks are generalized from the same skill with similar action sequences.

Based on the aforementioned definition, we aim to manipulate real-world robots by utilizing predictions generated from our knowledge-based framework and knowledge base. We have derived three types of tasks, including simple skills, such as "Pour_Water" and "Stir_Drink", as well as a complex skill called "Make_Drink". The beverages consist of 20 objects in four categories: coffee, tea, soda, and milk. They have different forms and packaging. The forms and packaging correspond to the physical properties of the object attributes in the knowledge-based framework. Such prior knowledge significantly influences the embeddings generated for the knowledge graph, thereby impacting the selection of existing templates as prototypes for new tasks during knowledge inference. The above task categories and objects generate a total of 36 task templates. Figure 6 shows a "Make_Coffee_a" task template derived from the "Make_Drink" task in a manipulation environment, which includes three sub-tasks: "Pour_Coffee_a", "Pour_Water_a", and "Stir_Coffee_a". Each primitive action in the sub-tasks has a subject, object, and object complement. We select three task templates from each type of task as priors. The instances of tasks successfully manipulated by the robot are added to the knowledge base as prior knowledge. The action primitives in the task instance are bound to the physical parameters of the target state. The low-level motion planning for each action primitive is directly generated by an RRT planner [46]. The remaining 27 task templates are utilized as test, with each attempt being conducted five times. The experiment hardware and environment are illustrated in Figure 7. The manipulation equipment includes a Universal Robot (UR5) robotic arm and a Robotiq gripper. Because our focus on skill transfer primarily pertains to higher-level task understanding, we do not address the underlying aspects of motion planning or visual positioning. Consequently, there are four fixed initial areas on the manipulation platform: Production, Material 1, Material 2, and Stirring. The coordinates of these areas are known. We excluded the need for object detection and assumed that both the object labels and positions were provided.

Table 3 shows the accuracy of action sequence prediction and robot execution. The first baseline is pattern matching, where the optimal matching template for generating semantic action sequences and manipulation sequences is chosen by comparing the shortest paths between the task nodes. The second and third baselines, along with our method, all fall within the realm of embedding, where the optimal matching template for generating semantic action sequences and manipulation sequences is chosen using generated feature vectors. These methods incorporate entity and relation features, while utilizing the knowledge base, resulting in a superior performance. The accuracy of representation is enhanced in our method through graph embedding, which incorporates more structured information compared to that of the other methods. As a result, we achieve a 92.59% accuracy rate in

action sequence prediction. According to the accurate sequence of actions, the execution accuracy has also reached 82.96%.

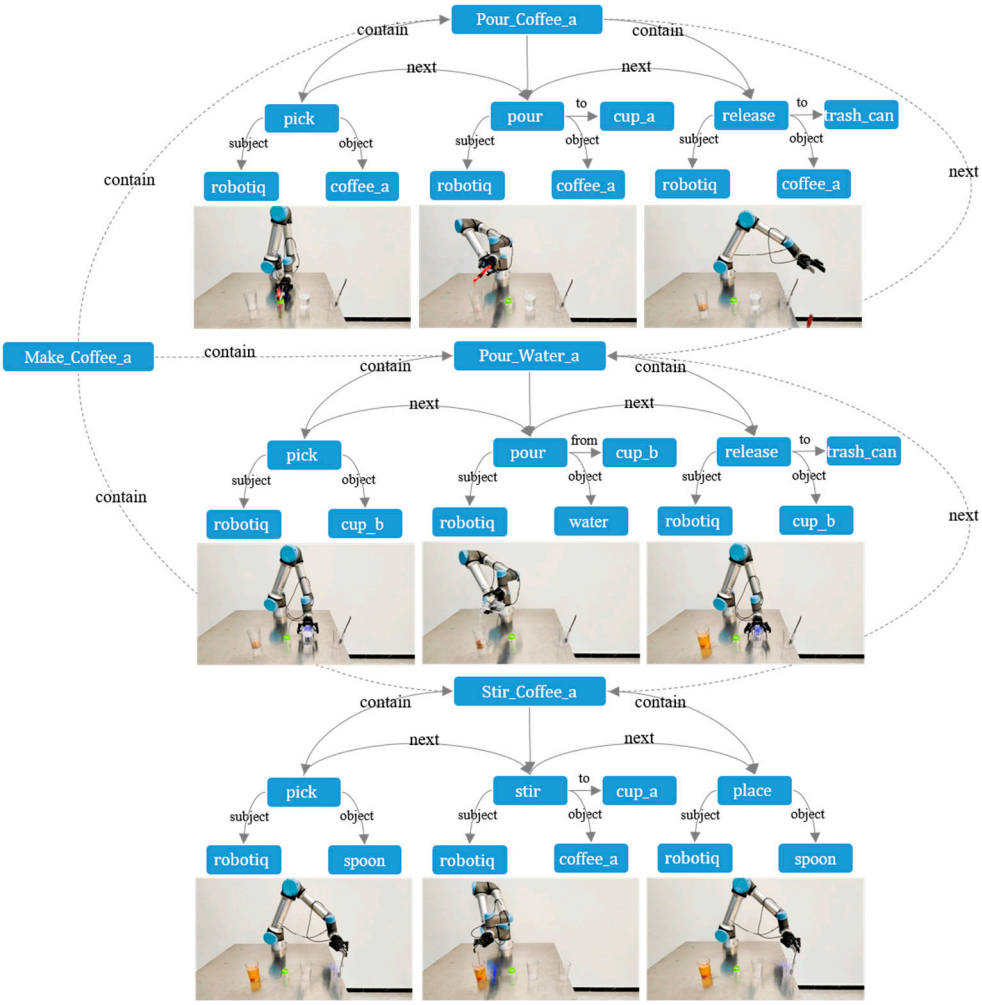


Figure 6. An example of a task template for robotic manipulation in a real environment.

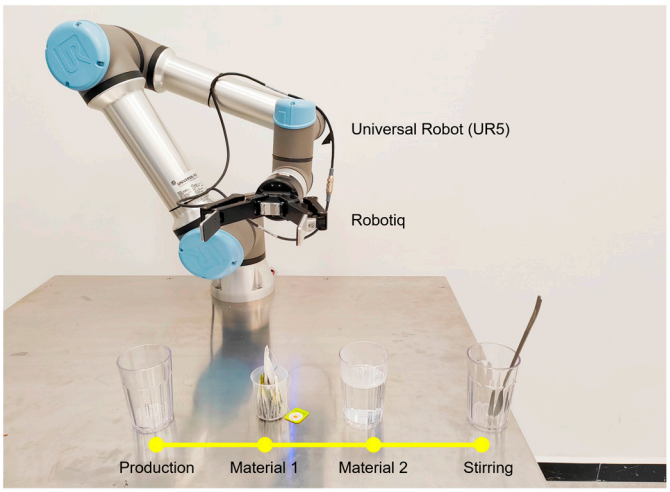


Figure 7. The experiment hardware and environment.

Table 3. Accuracy of skill transfer.

Method	Action Sequence Prediction	Robot Execution
Pattern Matching	51.85%	43.70%
DistMult	74.07%	61.48%
TransE	85.19%	74.81%
Graph Embedding (ours)	92.59%	82.96%

7. Conclusions

In this work, we present a hierarchical framework to comprehensively understand the diverse factors and knowledge types in robotic manipulation. Multi-source knowledge updating is achieved through the utilization of text datasets, LLMs, and external knowledge bases. Based on this foundation, the dataset SkillKG and knowledge base are meticulously constructed. A graph-based embedding method is utilized to generate semantic representations of the entities and relations in SkillKG, followed by an evaluation of the accuracy of feature embedding. Finally, based on the understanding of aforementioned knowledge representation, update, and inference, a robotic system successfully demonstrates skill transfer in a real-world environment. In the future, we aim to transform the visual data into knowledge graphs through scene graph generation and extend the framework to more scenes to explore the methods of induction.

Author Contributions: Conceptualization, R.M.; methodology, Q.J.; software, R.M.; validation, R.M.; formal analysis, R.M.; investigation, Q.J.; resources, F.S.; data curation, R.M.; writing—original draft preparation, R.M.; writing—review and editing, R.M.; visualization, R.M.; supervision, Q.J.; project administration, G.C.; funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by "Autonomous Learning of Complex Skills by multi-degree-of-freedom Agents, grant number 2021B0101410002" and "Major Project of the New Generation of Artificial Intelligence, grant number 2018AAA0102900".

Data Availability Statement: The latest version of our dataset SkillKG is available at https://github.com/tsingmr/SkillKG, accessed on 1 January 2024.

Acknowledgments: Thanks to Institute for Artificial Intelligence, Tsinghua University, for providing financial and equipment support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Waibel, M.; Beetz, M.; Civera, J.; d'Andrea, R.; Elfring, J.; Galvez-Lopez, D.; Häussermann, K.; Janssen, R.; Montiel, J.; Perzylo, A.; et al. RoboEarth—A World Wide Web for Robots. *Roboearth* **2011**, *18*, 69–82.
2. Riazuelo, L.; Tenorth, M.; Di Marco, D.; Salas, M.; Gálvez-López, D.; Mösenlechner, L.; Kunze, L.; Beetz, M.; Tardós, J.D.; Montano, L.; et al. RoboEarth semantic mapping: A cloud enabled knowledge-based approach. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 432–443. [CrossRef]
3. Beetz, M.; Beßler, D.; Haidu, A.; Pomarlan, M.; Bozcuoğlu, A.K.; Bartels, G. Know rob 2.0—A 2nd generation knowledge processing framework for cognition-enabled robotic agents. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 512–519.
4. Tenorth, M.; Beetz, M. KnowRob: A knowledge processing infrastructure for cognition-enabled robots. *Int. J. Robot. Res.* **2013**, *32*, 566–590. [CrossRef]
5. Saxena, A.; Jain, A.; Sener, O.; Jami, A.; Misra, D.K.; Koppula, H. Robobrain: Large-scale knowledge engine for robots. *arXiv* **2014**, arXiv:1412.0691.
6. Liu, L.; Xu, W.; Fu, H.; Qian, S.; Han, Y.; Lu, C. AKB-48: A Real-World Articulated Object Knowledge Base. *arXiv* **2022**, arXiv:2202.08432.
7. Yang, Y.; Guha, A.; Fermüller, C.; Aloimonos, Y. Manipulation action tree bank: A knowledge resource for humanoids. In Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 11–14 November 2014; pp. 987–992.

8. Paulius, D.; Huang, Y.; Milton, R.; Buchanan, W.D.; Sam, J.; Sun, Y. Functional object-oriented network for manipulation learning. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 2655–2662.
9. Jiang, C.; Dehghan, M.; Jagersand, M. Understanding Contexts Inside Robot and Human Manipulation Tasks through Vision-Language Model and Ontology System in Video Streams. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 8366–8372.
10. Mitrevsk, A.; Plöger, P.G.; Lakemeyer, G. Ontology-assisted generalisation of robot action execution knowledge. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 6763–6770.
11. Kwak, J.H.; Lee, J.; Whang, J.J.; Jo, S. Semantic grasping via a knowledge graph of robotic manipulation: A graph representation learning approach. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9397–9404. [CrossRef]
12. Rodríguez, I.; Nottensteiner, K.; Leidner, D.; Durner, M.; Stulp, F.; Albu-Schäffer, A. Pattern recognition for knowledge transfer in robotic assembly sequence planning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3666–3673. [CrossRef]
13. Sun, X.; Zhang, Y.; Chen, J. RTPO: A domain knowledge base for robot task planning. *Electronic* **2019**, *8*, 1105. [CrossRef]
14. Liu, S.; Tian, G.; Zhang, Y.; Zhang, M.; Liu, S. Service planning oriented efficient object search: A knowledge-based framework for home service robot. *Expert Syst. Appl.* **2022**, *187*, 115853. [CrossRef]
15. Daruna, A.; Liu, W.; Kira, Z.; Chetnova, S. Robocse: Robot common sense embedding. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9777–9783.
16. Yu, K.-T.; Bauza, M.; Fazeli, N.; Rodriguez, A. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 30–37.
17. Chebotar, Y.; Hausman, K.; Su, Z.; Molchanov, A.; Kroemer, O.; Sukhatme, G.; Schaal, S. Bigs: Biotac grasp stability dataset. In Proceedings of the ICRA 2016 Workshop on Grasping and Manipulation Datasets, Stockholm, Sweden, 16–21 May 2016; pp. 1–8.
18. Mandlekar, A.; Zhu, Y.; Garg, A.; Booher, J.; Spero, M.; Tung, A.; Gao, J.; Emmons, J.; Gupta, A.; Orbay, E. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In Proceedings of the Conference on Robot Learning, London, UK, 8–11 November 2021; pp. 879–893.
19. Sharma, P.; Mohan, L.; Pinto, L.; Gupta, A. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 906–915.
20. Dasari, S.; Ebert, F.; Tian, S.; Nair, S.; Bucher, B.; Schmeckpeper, K.; Singh, S.; Levine, S.; Finn, C. Robonet: Large-scale multi-robot learning. *arXiv* **2019**, arXiv:1910.11215.
21. Miller, G. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
22. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Fransisco, CA, USA, 4–9 February 2017.
23. Liu, H.; Singh, P. ConceptNet—A practical commonsense reasoning tool-kit. *BT Technol. J.* **2004**, *22*, 211–226. [CrossRef]
24. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
25. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
26. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In Proceedings of the Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Republic of Korea, 11–15 November 2007; pp. 722–735.
27. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
28. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. In Proceedings of the 4th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020; Volume 33, pp. 1877–1901.
29. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
30. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
31. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223.
32. Lin, K.; Agia, C.; Migimatsu, T.; Pavone, M.; Bohg, J. Text2motion: From natural language instructions to feasible plans. *Auton. Robot.* **2023**, *47*, 1345–1365. [CrossRef]
33. Wu, J.; Antonova, R.; Kan, A.; Lepert, M.; Zeng, A.; Song, S.; Bohg, J.; Rusinkiewicz, S.; Funkhouser, T. Tidybot: Personalized robot assistance with large language models. *arXiv* **2023**, arXiv:2305.05658.
34. Church, K. Word2Vec. *Nat. Lang. Eng.* **2017**, *23*, 155–162. [CrossRef]

35. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Proceedings of the NIPS'13: 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013*; Association for Computing Machinery: New York City, NY, USA, 2013; Volume 26.
36. Sun, Z.; Hu, W.; Zhang, Q.; Qu, Y. Bootstrapping entity alignment with knowledge graph embedding. In Proceedings of the IJCAI'18: Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; International Joint Conferences on Artificial Intelligence: San Francisco, CA, USA, 2018.
37. Dong, X.L.; Gabrilovich, E.; Heitz, G.; Horn, W.; Murphy, K.; Sun, S.; Zhang, W. From data fusion to knowledge fusion. *arXiv* **2015**, arXiv:1503.00302. [CrossRef]
38. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
39. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
40. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
41. Calli, B.; Singh, A.; Walsman, A.; Srinivasa, S.; Abbeel, P.; Dollar, A.M. The ycb object and model set: Towards common benchmarks for manipulation research. In Proceedings of the 2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, 27–31 July 2015; pp. 510–517.
42. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A. Training language models to follow instructions with human feedback. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 27730–27744.
43. Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Deitke, M.; Ehsani, K.; Gordon, D.; Zhu, Y. Ai2-thor: An interactive 3d environment for visual ai. *arXiv* **2017**, arXiv:1712.05474.
44. Yang, B.; Yih, W.-t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
45. Miller, J.J. Graph database applications and concepts with Neo4j. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, 23–24 March 2012.
46. Kuffner, J.J.; LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; pp. 995–1001.