Autonomous Robots (2023) 47:529–547
https://doi.org/10.1007/s10514-023-10099-4

Learning instance-level N-ary semantic knowledge at scale for robots operating in everyday environments

Weiyu Liu[1] · **Dhruva Bansal**[1] · **Angel Daruna**[1] · **Sonia Chernova**[1]

Received: 17 June 2022 / Accepted: 26 February 2023 / Published online: 6 April 2023

Abstract

Robots operating in everyday environments need to effectively perceive, model, and infer semantic properties of objects. Existing knowledge reasoning frameworks only model binary relations between an object's class label and its semantic properties, unable to collectively reason about object properties detected by different perception algorithms and grounded in diverse sensory modalities. We bridge the gap between multimodal perception and knowledge reasoning by introducing an n-ary representation that models complex, inter-related object properties. To tackle the problem of collecting n-ary semantic knowledge at scale, we propose transformer neural networks that generalize knowledge from observations of object instances by learning to predict single missing properties or predict joint probabilities of all properties. The learned models can reason at different levels of abstraction, effectively predicting unknown properties of objects in different environmental contexts given different amounts of observed information. We quantitatively validate our approach against prior methods on LINK, a unique dataset we contribute that contains 1457 object instances in different situations, amounting to 15 multimodal properties types and 200 total properties. Compared to the top-performing baseline, a Markov Logic Network, our models obtain a 10% improvement in predicting unknown properties of novel object instances while reducing training and inference time by more than 150 times. Additionally, we apply our work to a mobile manipulation robot, demonstrating its ability to leverage n-ary reasoning to retrieve objects and actively detect object properties. The code and data are available at https://github.com/wliu88/LINK.

Keywords Semantic reasoning · N-ary relation · Object-centric reasoning · Transformer networks

1 Introduction

Robust operation in everyday human environments requires robots to effectively model a wide range of objects and to predict object locations, properties, and uses. Semantic task and object knowledge serves as a valuable abstraction in this context. Perceiving and understanding semantic properties of objects (e.g., a *cup* is *ceramic*, *empty*, located *in kitchen*, and used for *drinking*) aids robots in performing many tasks in the real world, such as inferring missing information in incomplete human instructions (Nyga et al., 2018; Chao et al., 2020), efficiently searching for objects in homes (Zeng et al., 2019; Yang et al., 2019), and manipulating objects based on their affordances and states (Ardón et al., 2019; Liu et al., 2020; Jain et al., 2013).

Prior work has encoded semantic knowledge primarily as pairwise relations between an object's *class* label and its semantic *properties* (e.g., the *cup* is *wet*, the *cup* is *in cabinet*) (Daruna et al., 2019; Chernova et al., 2017; Zhu et al., 2014; Tenorth & Beetz, 2017; Saxena et al., 2014) (Fig. 1 left). These semantic properties can come from a variety of perception methods, such as the use of vision to predict visual attributes (Ferrari et al., 2007; Nazarczuk & Mikolajczyk, 2020) and affordances (Do et al., 2018; Chuang et al., 2018), haptic data to identify object materials (Kerr et al., 2018) and surface textures (Chu et al., 2015), as well as exploratory actions to detect object states (Thomason et al.,

✉ Weiyu Liu
wliu88@gatech.edu

Dhruva Bansal
dbansal36@gatech.edu

Angel Daruna
adaruna3@gatech.edu

Sonia Chernova
chernova@gatech.edu

[1] Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA

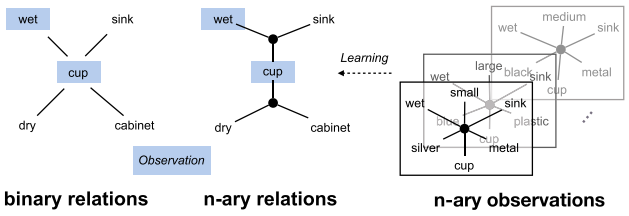


Fig. 1 N-ary relations enable robots to more effectively model complex, inter-related object properties than binary relations. In our framework, we learn generalizable n-ary relations from object instances represented as n-ary observations

2018). However, pairwise encoding of semantic data fails to take full advantage of such multimodal observations because it ignores the complex relational structure between various object properties. For example, observing that a *cup* is *wet* does not help the robot infer that the *cup* more likely should be placed *in sink* than *in cabinet*.

The objective of our work is to enable robots to collectively reason about object properties that can be grounded in different modalities and detected by separate perception algorithms. Specifically, we situate our work in the task of predicting semantic properties of objects based on partial observations. We introduce a novel semantic reasoning framework that uses n-ary relations to model complex, inter-related object properties. In addition to modeling relations between object properties, our framework enables the ability to reason at different levels of abstraction (Fig. 1 middle). For example, a robot searching for a *cup*, with no additional information, is able to perform class-level inference to identify both the *cabinet* and *sink* as likely locations. Given additional information, such as *wet*, n-ary relations enable more refined reasoning and the ability to detect that wet cups are more commonly found *in sink* rather than *in cabinet*.

A key challenge presented by n-ary representations is the collection of semantically meaningful n-ary relations, which require various object properties to be conditioned on each other. Unlike binary relations which can be created by experts or crowdsourced at scale (Gupta et al., 2004; Liu & Singh, 2004; Miller, 1995; Lenat, 1995), n-ary relations are difficult to construct manually. In this work, we obtain n-ary observations, each representing a set of identified semantic properties of an object instance within a particular environmental context (e.g., a *small silver metal cup* that is *wet* and *in sink*), from which we then learn models capturing generalizable n-ary relations (Fig. 1 right). Since the n-ary relations are learned from object instances, they also encode knowledge at the instance-level. To mine generalizable patterns from n-ary observations, we introduce two transformer-based neural networks, inspired by recent advances in contextualized language models (Devlin et al., 2019; Vaswani et al., 2017). The autoencoding model, which we call LINK-AE, is trained to reconstruct hidden properties of object instances. The autoregressive model, which we call LINK-AR, is trained to predict joint probabilities of all observed properties in each n-ary observation. With these self-supervised training objectives, our models learn to generalize n-ary relations, which help predict unobserved properties of novel object instances and perform reasoning at different levels of abstraction. In summary, our work contributes:

- an n-ary instance-level representation of objects, which enables modeling n-ary relations between multimodal object properties and variance between object instances,
- two scalable transformer-based neural networks which learn semantic knowledge about objects from data and are capable of performing inference at different levels of abstraction,
- a dataset, which we call LINK, consisting of 1457 object instances in different situations associated with 15 types of 200 multimodal properties, the richest situated object dataset to date.

We quantitatively validate our approach against five prior methods on the above dataset and demonstrate that our representation and reasoning method leads to significant improvements in predicting unknown properties of novel object instances while significantly reducing computation time. We further show that the autoregressive formulation enables our model to outperform other methods in predicting missing properties of novel n-ary relations with different arities. Additionally, we apply our work to a mobile manipulation robot. We demonstrate that the explicit representation of n-ary knowledge allows the robot to locate objects based on complex human commands. We also show that the learned relations can help the robot infer properties based on observations from multimodal sensory data.

A preliminary version of this work was presented in (Liu et al., 2021). The current version first characterizes the previously introduced transformer network as an autoencoding transformer (i.e., LINK-AE), and then introduces a new autoregressive transformer (i.e., LINK-AR) that explicitly models factorized joint probability of each complete n-ary observation. We also include an additional experiment that directly evaluates relational inference at different levels of abstraction by requiring models to predict novel relations with different arities ranging from binary relations to 16th-order relations. This paper also includes an extended discussion of related work on the topic of object-centric reasoning and a recent baseline based on a pre-trained language model (Devlin et al., 2019).

2 Related work

Our work is related to the following prior efforts.

2.1 Semantic reasoning in robotics

Many ontologies and knowledge graphs have been used across AI and robotics to encode general knowledge about objects (e.g., locations, properties, uses, and class hierarchies) (Lim et al., 2011; Saxena et al., 2014; Tenorth & Beetz, 2017; Lemaignan et al., 2017; Varadarajan & Vincze, 2013). In robotics, a key challenge for semantic reasoning is generalization to previously unseen scenes or environments. Bayesian logic networks have been used to cope with noise and non-deterministic data from different data sources (Chernova et al., 2017). More recently, knowledge graph (KG) embedding models were introduced as scalable frameworks to model object knowledge encoded in multi-relational KGs (Daruna et al., 2019; Arkin et al., 2020). Although the above techniques effectively model objects, they only support reasoning about binary class-level facts, therefore lacking the discriminative features needed to model object semantics in realistic environments.

Other frameworks take a learning approach to modeling object semantics. Methods for learning relations between objects, between object properties, and between objects and their environments have shown to be beneficial for detecting objects on table tops (Kunze et al., 2014; Günther et al., 2018; Nyga et al., 2014), finding hidden objects in shelves (Moldovan & Raedt, 2014), predicting object affordances (Zhu et al., 2014), and semantic grasping (Ardón et al., 2019; Liu et al., 2020). However, most methods leverage probabilistic logic models to learn these relations, which have scalability issues that limit them from modeling interconnected relations in larger domains (Nyga et al., 2014; Moldovan & Raedt, 2014; Zhu et al., 2014; Ardón et al., 2019). In contrast, our proposed framework learns n-ary relations between 15 property types and 200 properties, the richest representation to date.

2.2 Modeling N-ary facts

Our neural network model is closely related to methods developed in the knowledge graph community. Many relational machine learning techniques, including most recent transformer models (Wang et al., 2019; Bosselut et al., 2019), have been developed for modeling KGs and in particular predicting missing links in KGs (Nickel et al., 2015). These techniques treat a KG as set of triples/binary facts, where each triple (h, r, t) links two entities h and t with a relation r (e.g., *(Marie Curie, educated at, University of Paris)*). Despite the wide use of triple representation, many facts in KGs are hyper-relational. Each hyper-relational fact has a base triple (h, r, t) and additional key-value (relation-entity) pairs (k, v) (e.g., *{(academic major, physical), (academic degree, Master of Science)}*). A line of work converts hyper-relational facts to n-ary meta-relations $r(e_1, ..., e_n)$ and leverages translational distance embedding (Wen et al., 2016; Zhang et al., 2018), spatio-translational embedding (Abboud et al., 2020), tensor factorization (Liu et al., 2020) for modeling. Other approaches directly learn hyper-relational facts in their original form using various techniques, including convolutional neural networks, graph neural networks, and transformer models (Rosso et al., 2020; Galkin et al., 2020). Another approach unifies n-ary representation by converting the base triple to key-value pairs; it uses convolutional neural network for feature extraction and then models relatedness of role-value pairs with a fully connected network (Guan et al., 2019). In this work, we represent object properties with key-value pairs. Compared to other representations of higher-order relations, this representation is flexible to model interconnected relations between any types of object properties (e.g., the relation between color and material) and in different levels of abstraction (e.g., the specific relation between object category, color, weight, and material). In addition, we model facts with much higher arities than existing work in the KG community and directly reason about n-ary relations between role-value pairs using the transformer model.

2.3 Object-centric reasoning

Our approach is also related to methods for modeling objects from sensory data. Object-centric datasets has enabled different robotic applications such as object retrieval (Tatsuma et al., 2012; Dyke et al., 2020), grasping (Wade-McCue et al., 2018; Zhang et al., 2021), manipulation (Levine et al., 2016; Huang & Sun, 2019), and object recognition (Singh et al., 2014; She et al., 2020). In computer vision, object attributes are extracted from images (Ferrari et al., 2007; Farhadi et al., 2009; Sun et al., 2013). Recent techniques in visual question answering (Nazarczuk & Mikolajczyk, 2020) and language grounding (Shridhar & Hsu, 2018; Jenkins et al., 2020) allow robots to answer questions about objects and describe objects with natural language. Haptic feedback (Luo et al., 2017; Li et al., 2020) and auditory data (Eppe et al., 2018; Gandhi et al., 2020) have also helped robots interpret salient features of objects beyond vision. Interactive perception and unsupervised exploration can further leverage a robot's exploratory actions to reveal sensory signals that are otherwise not observable (Bohg et al., 2017; Sinapov et al., 2014; Chu et al., 2015; Thomason et al., 2018; Amiri et al., 2018; Bhattacharjee et al., 2018; Tatiya & Sinapov, 2019; Watters et al., 2019). Building on such rich object-level datasets, recent work has also explored multisensory object-centric perception to perform tasks like instance recognition, grasping, and object retrieval (Gao et al., 2021). This work encodes visual, auditory, and tactile sensory data, thereby

moving towards providing a full spectrum of physical object properties. Our work also shares the same goal as a recent work that builds robot-centric object knowledge from multimodal sensory data (Thosar et al., 2021). Built with the tool substitution task in mind, they extract object properties like hollowness, flatness, and roughness, that would be relevant to this real world task. We consider our approach complimentary to the above, as our framework can leverage the rich semantic information extracted from these methods to infer additional unknown object properties.

3 Problem definition

Given a set of observed/known object properties, we aim to predict an unobserved/unknown property of a novel situated object instance using semantic knowledge learned from data. We define a *situated object instance* as a particular specimen of a given object class within a particular environmental context (e.g., the full red Solo cup on the kitchen counter). The object's semantic representation encodes properties grounded in different modalities, and includes both immutable properties (e.g., class, material, shape, and hardness) and mutable properties (e.g., location, clealiness, fullness).

We use the n-ary representation to model all object data. Each n-ary relation is defined by a set of role-value pairs $\{r_i : v_i\}_{i=1}^{n}$, where $r_i \in \mathcal{R}$ is the role set, $v_i \in \mathcal{V}$ is the value set, and $i = 1, ..., n$. The number n represents the arity of the relation. In the context of modeling object semantics, each role corresponds to a property type and each value corresponds to a property value. In this representation, our task can be formally written as $\{r_1 : v_1, ..., r_{n-1} : v_{n-1}, r_n :?\}$, where $n-1$ is the number of known properties, and r_n is the type of the property being queried. The number of known properties n determines the level of abstraction for the query. A smaller n queries more abstract semantic knowledge (e.g., {*class: cup, material: ?*}) and a larger n queries more specific semantic knowledge (e.g., {*class: cup, transparency: opaque, physical property: hard, color: brown, material: ?*}).

Various robotic tasks can benefit from n-ary knowledge about object properties. We, in particular, examine two uses cases. First, a robot can locate specific objects based on users' requests. For example, "find a clean glass cup" can be translated to a query {*class: cup, material: glass, location: ?*}. Second, n-ary knowledge can enable a robot to infer object properties that are hard to be directly observed by collectively reasoning about properties extracted from multimodal sensors. For example, a robot only equipped with object detection and material classification can infer that a glass cup is likely to be fragile with the query {*class: cup, material: glass, physical property: ?*}.

Table 1 Comparing available datasets for learning object semantics

Dataset	Application	# Object classes	# Object instances	# Properties	# Property types	Situated	Complete annotation
Shop-VRB (Nazarczuk & Mikolajczyk, 2020)	Vision & Language	20	66	99	6		✓
GoLD (Jenkins et al., 2020)	Language Grounding	47	207	/	/		
Thomason 2018 (Thomason et al., 2018)	Interactive Perception	4	32	81	6	✓	✓
Zhu 2014 (Zhu et al., 2014)	Knowledge Base	40	4000	97	4		✓
Paolo 2019 (Ardón et al., 2019)	Semantic Grasping	8	30	44	5		✓
AI2Thor (Kolve et al., 2017)	Simulation	/	125	28	6		✓
Ours	Semantic Reasoning	11	98	200	15	✓	✓

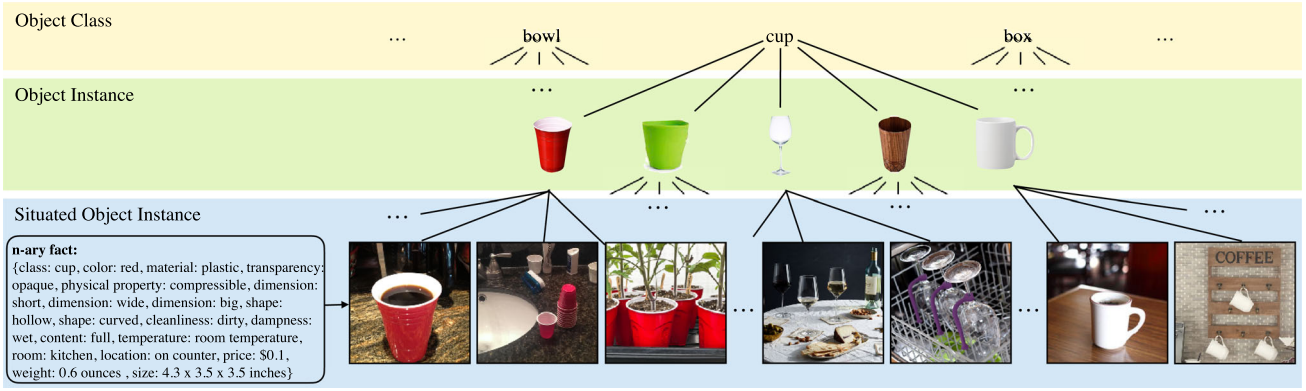


Fig. 2 An example of the collected data showing various cups and diverse environmental contexts each of these instances can be found in. Pictures at the situated object instance level are for illustration but correspond to descriptions of the contexts in our dataset. Each situated object instance has a corresponding fully annotated n-ary observation (bottom left)

Table 2 Object classes and properties in our dataset

Type (# Value)	Values
class (11)	bottle, bowl, box, brush, can, cup, fork, ladle, pan, spatula, sponge
material (8)	ceramic, foam, glass, metal, paper, plastic, porcelain, wood
transparency (3)	opaque, translucent, transparent
dimension (10)	big, deep, long, narrow, shallow, short, small, thick, thin, wide
physical property (6)	absorbent, compressible, elastic, fragile, hard, soft
shape (9)	angular, blunt, curved, flat, forked, hollow, irregular, sharp, straight
*temperature (3)	cold, hot, room temperature
*fullness (3)	empty, full, half
*dampness (3)	damp, dry, wet
*cleanliness (3)	clean, dirty, normal
price (3)	cheap, expensive, medium
weight (3)	heavy, light, medium
size (3)	large, medium, small
*room (11)	balcony, bathroom, bedroom, child's room, closet, dining room, garage, kitchen, laundry, living room, study
color (15)	black, blue, bronze, brown, clear, colorful, gold, green, orange, pink, purple, red, silver, white, yellow
*location (117)	in bag, in basket, in bathtub, in bin, in box, in bucket, in cabinet, in cooler, on bathtub, on bed, on bench, on bookshelf,...

4 LINK dataset

In this section, we present the content and features of the **LINK** dataset for **L**earning **I**nstance-level **N**-ary **K**nowledge. Our dataset contains 1457 fully annotated situated object instances. In Table 1, we compare the content of our dataset to a representative set of data sources from the computer vision, natural language processing, and robotics communities; as can be seen, our dataset has the most diverse set of property types and property values, leading to much richer and more realistic object representations. Properties in our dataset are inherently multimodal, which help bridging robots' perception and reasoning. In addition to visual attributes, we intentionally model properties that are hard to extract from visual data (e.g., dampness and temperature). Our dataset represents variance between object instances by having on average of nine objects per class. Objects in different situations are captured by mutable properties such as location, cleanliness, temperature, and dampness. Furthermore, our dataset provides complete and logically coherent annotations (truth values) of all properties for each situated object instance. Figure 2 illustrates the hierarchy of objects in our dataset, which facilitates the learning of generalizable

n-ary relations between object properties at different levels of abstraction.

4.1 Objects and properties

Our dataset contains 98 instances of everyday household objects organized into 11 object classes. For each object class, we selected objects diverse in sizes, geometries, materials, visual appearances, and affordances from the Amazon product website. We created the initial set of 83 properties (the additional 117 location properties are crowdsourced) from adjectives that people use for describing objects (Lynott & Connell, 2009). We then followed GermaNet,[1] (Hamp & Feldweg, 1997) to categorize these properties into 15 distinct types based on their semantic meanings. Table 2 shows the property values and types in our dataset (mutable properties are labeled with asterisk).

4.2 Collection of N-ary labels

Given 98 object instances and 15 property types, our next step was to collect situated object instances where each object is described by a semantically meaningful combination of properties. We used Amazon Mechanical Turk (AMT) to crowdsource property combinations. The novelty in our crowdsourcing process is that we asked AMT workers to *imagine* objects situated in different environment contexts. Compared to established approaches to collect semantic knowledge, such as asking workers to annotate properties for objects in images and prompting workers to answer commonsense questions about objects, our method is more effective at eliciting multimodal and instance-level knowledge.

More specifically, after we extracted pictures of each object, as well as details of its material, weight, dimension, and price from the Amazon product web page, we conducted a three-stage crowdourcing process. First, for all 98 object instances, we showed pictures of the object to AMT workers and asked them to list the object's immutable properties. Second, we presented AMT workers with an object and a room, and had them imagine and describe three situations in which that object-room combination could be encountered, including details of the location of the object, the associated daily activity, and the object state (e.g., a wet cup on the bathroom counter used for rinsing after brushing teeth). Third, we presented a new set of AMT workers with the above collected situated object descriptions, and had them label mutable properties (e.g., wet, empty, clean) for the associated object. To ensure the quality of the crowdsourced data, we used 3 annotators for each question and filtered workers based on gold standard questions. We manually verified descriptions of situations from stage 2.

[1] GermaNet, the German version of the English lexical database Wordnet (Miller, 1995) provides hierarchical structures for adjectives.

5 Approach

Given $n-1$ properties, we aim to predict the the n^{th} property of type r_n, i.e., $\{r_1 : v_1, ..., r_{n-1} : v_{n-1}, r_n :?\}$, where $n-1$ is the number of observed properties. We develop two transformer-based neural network models based on the following design goals: learning higher-order interactions between typed properties (i.e., role-value pairs), accommodating arbitrary order of properties, supporting inference at different levels of abstraction by accepting arbitrary number of observed properties, representing uncertainties in semantic knowledge, and being scalable. Below, we introduce the autoencoding model **LINK-AE** which we first proposed in (Liu et al., 2021). Then, we propose a new autogressive model **LINK-AR**. We outline the necessary modifications to the autoencoding transformer for the new autoregressive formulation.

5.1 Autoencoding model

The objective of our autoencoding model LINK-AE is to learn conditional probability

$$p(v_n \mid \{\widetilde{r_i : v_i}\}_{i=1}^{n}) \tag{1}$$

where $\{\widetilde{r_i : v_i}\}_{i=1}^{n}$ represents the corrupted version of the n-ary relation $\{r_i : v_i\}_{i=1}^{n}$. Specifically, $\{\widetilde{r_i : v_i}\}_{i=1}^{n}$ corresponds to the input $\{r_1 : v_1, ..., r_{n-1} : v_{n-1}, r_n : [\text{MASK}]\}$, where [MASK] is a special token indicating that the value of the query property has been hidden. As shown in Fig. 3, the masked input is first fed into a transformer encoder (Vaswani et al., 2017), which builds a contextualized representation of the input. The encoding at the n^{th} position is then used to predict the query property via a feedforward layer and a sigmoid function. The autoencoding learning is closely related to masked language models (Devlin et al., 2019) that reconstruct full sentences from corrupted ones where words are randomly replaced with mask tokens. It has been shown that the autoencoding training allows transformer neural networks to capture higher-order distributional information in data (Sinha et al., 2021). Below we describe each component of the autoencoding model in detail and discuss how they help to satisfy the design goals and learn n-ary relations between object properties.

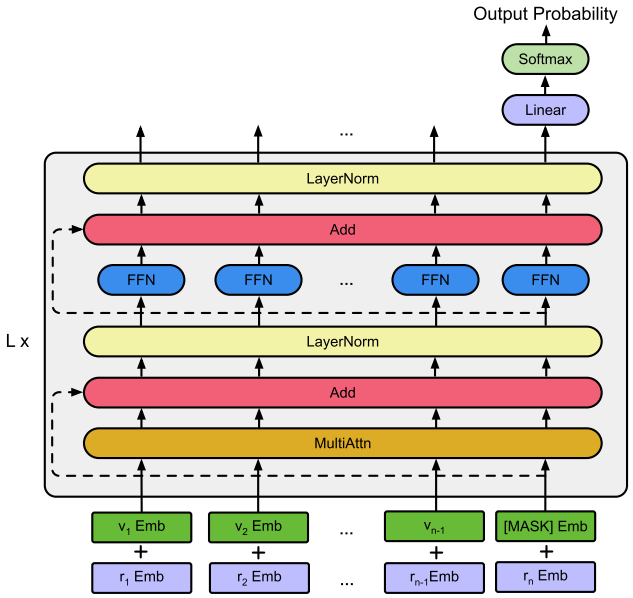


Fig. 3 The architecture of our LINK-AE model includes embedding layers, a transformer encoder, and a feed-forward layer for predicting probabilities of properties

5.1.1 Input encoder

The input encoder uses learned embeddings to convert roles and values in the input to vectors of dimension d_{model}. For each role-value pair, we construct its representation as

$$h_i^0 = x_i^{\text{value}} + x_i^{\text{role}} \tag{2}$$

where x_i^{value} is the embedding for the i^{th} value and x_i^{role} is the embedding for the i^{th} role. At the query position, the value embedding of the [MASK] token indicates that this property is in query. We maintain the role embedding of the query property, allowing the model to condition its reasoning on the type of the query property. Different from existing transformer-based models (Devlin et al., 2019; Bosselut et al., 2019; Wang et al., 2019), we do not use positional embeddings to indicate the position of each role-value pair in the n-ary query since, unlike natural language sentences or KG triples, there is no particular order for object properties. As the latter components of the model are permutation invariant to the order of input data, removing the positional embeddings also allows our model to efficiently learn from object properties represented in n-ary observations.

5.1.2 Transformer encoder

The transformer encoder takes the embedded input $\{h_1^0, ..., h_n^0\}$ and builds a contextualized representation $\{h_1^L, ..., h_n^L\}$ where L is the number of transformer layers in the transformer encoder. We discuss the core components of the transformer encoder below and refer the readers to the original paper for details (Vaswani et al., 2017).

At the heart of the Transformer architecture is the scaled dot-product self-attention function, which allows elements in a sequence to attend to other elements. Each input h_i^l is linearly projected to a query q_i^l, key k_i^l, and value v_i^l. The intermediate output $\hat{h}_i^{l+1}$ is computed as a weighted sum of the values, where the weight assigned to each value is based on the compatibility of the query with the corresponding key. The function is computed on a set of queries simultaneous with matrix multiplication.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where d_k is the dimension of queries and keys. The queries, keys, and values are stacked together into matrix $Q \in \mathbb{R}^{n \times d_{\text{model}}}$, $K \in \mathbb{R}^{n \times d_{\text{model}}}$, and $V \in \mathbb{R}^{n \times d_{\text{model}}}$. We omit the layer index here for clarity.

Instead of computing the attention function once, the multi-head attention has H heads, where each head performs a scaled dot-product attention. This design allows each head to attend to different combinations of the input. As shown in Fig. 3, after the multi-head attention (MultiAttn), a fully-connected feedforward network (FFN) is applied to each position of the sequence separately and identically. Residual connections (He et al., 2016) are applied both after MultiAttn and FFN, which are followed by layer normalizations (Ba et al., 2016).

The transformer encoder is suited to our task because each position can freely attend to all positions in the input, thus aiding in modeling inter-relations between properties. The transformer encoder is also a permutation equivariant function f because for any permutation $z \in \mathcal{Z}_n$, where $\mathcal{Z}_n$ is the set of all permutations of indices $\{1, ..., n\}$, $f(z[\{h_i\}_{i=1}^n]) = z[f(\{h_i\}_{i=1}^n)]$ (Lee et al., 2019). This property supports effective reasoning of order-less object properties.

5.1.3 Final classification

The final layer uses a learned linear transformation and a sigmoid function to convert the encoded input to predicted probabilities of properties. Specifically,

$$p_n = \sigma(E_{\text{value}}\,\text{FCN}(h_n^L)) \tag{4}$$

where FCN is a fully connected layer and E_{value} is the learned embedding matrix used to create input value embeddings. The use of the sigmoid function σ allows the model to accept multiple correct answers, therefore modeling uncertainties in semantic knowledge (e.g., cups can be found in both kitchen and living room)

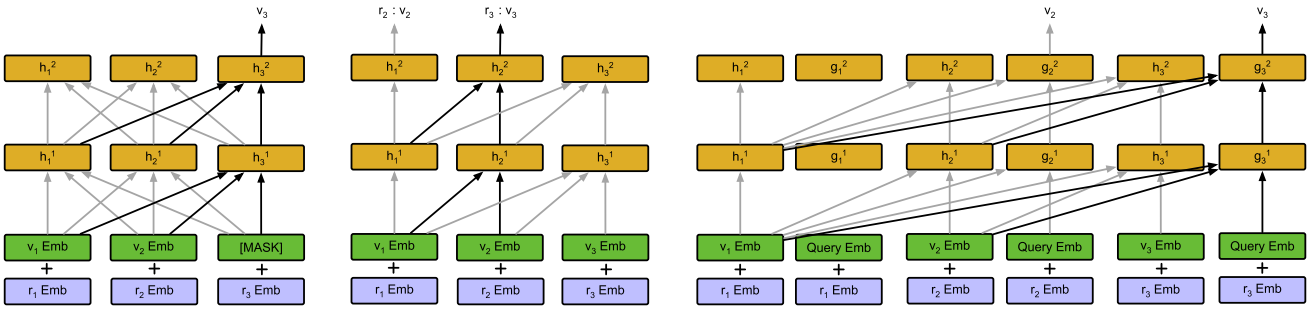


Fig. 4 The arrows visualize the information path for each input position in an autoencoding transformer encoder (LINK-AE), a standard autoregressive transformer encoder, and a two-stream autoregressive transformer encoder (LINK-AR). We highlight the information path for the third role-value pair, which is in query. Comparing with LINK-AE, the lack of arrows in LINK-AR is due to the chosen factorization, the use of causal attention, and the separation of content and query representation. We also illustrate the different input and final classification configurations of the autoencoding and autoregressive models

5.1.4 Autoencoding training

During training, we construct the masked input by replacing only a single value in an n-ary observation with the [MASK] token. We perform this procedure exhaustively for all values and all n-ary observations in the training set. We then group n-ary observations sharing the same masked instances and use their ground-truth values at the query position to construct a one-hot label (continuous-valued properties are discretized). Scoring multiple instances simultaneously is also known as the 1-N setting (Dettmers et al., 2018) and helps reduce training and inference time. Our training objective is to maximize the log-likelihood of Eq. 1. We use cross-entropy between the one-hot label and prediction as training loss. We apply label smoothing (Szegedy et al., 2016) to prevent overfitting.

5.2 Autoregressive model

Compared to the autoencoding model which learns to reconstruct the value of one unknown property from a corrupted version of the input, our autoregressive model does not require input corruption and directly learns the probability of the complete n-ary observation. Given an n-ary relation $\{r_1 : v_1, ..., r_{n-1} : v_{n-1}, r_n : v_n\}$, our autoregressive model LINK-AR factorizes the joint probability of the properties with the chain rule:

$$p(\{r_i : v_i\}_{i=1}^{n}) = \prod_{i=2}^{n} p(v_i \mid r_i, \{r_k : v_k\}_{k=1}^{i-1}) \tag{5}$$

The factors start with role-conditioned probability $p(v_2 \mid r_2, \{r_1 : v_1\})$ modeling binary relations and extend to higher-order relations. Learning the factorized probabilities with increasing numbers of conditional variables help our model learn to perform reasoning at different levels of abstraction. To implement the autoregressive model, we use the same input encoder and final classification layer as the autoencoding model. However, we make some crucial modifications to the transformer encoder to support effective learning of factorized probabilities. Below, we first discuss our new autoregressive transformer encoder and its core component, the two-stream causal attention. Then we describe the training procedure for the new model.

5.2.1 Two-stream causal attention

We propose to use two-stream causal attention to model $\prod_{i=2}^{n} p(v_i \mid r_i, \{r_k : v_k\}_{k=1}^{i-1})$. Built on the two-stream attention proposed in (Yang et al., 2019), our autoregressive transformer encoder conditions the prediction of each unknown property on type-specific information and is permutation invariant to the order of known properties. Before we discuss the full formulation, we first describe how to model $\prod_{i=2}^{n} p(v_i \mid \{r_k : v_k\}_{k=1}^{i-1})$ using causal attention.

The transformer encoder introduced in Sect. 5.1.2 can be used to represent factorized probabilities $\prod_{i=2}^{n} p(v_i \mid \{r_k : v_k\}_{k=1}^{i-1})$ by limiting the attention of each position i to only itself and its previous positions (i.e., $\{1, ..., i\}$). Specifically, this type of causal attention can be achieved by adding a position-wise bias term to QK^T when computing dot-product attention (Vaswani et al., 2017). We compare the information paths of the autoencoding and autoregressive transformer encoders in Fig. 4 (left and middle). Limited by its internal structure, the autoregressive transformer encoder, however, cannot condition the prediction on the type of the query property.

We combine causal attention with two-stream attention to model $\prod_{i=2}^{n} p(v_i \mid r_i, \{r_k : v_k\}_{k=1}^{i-1})$. As illustrated in Fig. 4 (right), we learn two sets of hidden representations. The content representation $h(\{r_k : v_k\}_{k=1}^{i})$ of each position h_i is constructed from content representations its previous positions and itself. The query representation $g(\{r_k : v_k\}_{k=1}^{i}, r_i)$

of each position g_i is built from content representations of its previous positions and the role of the current position. Concretely, we compute both representations with self-attention as follows:

$$\hat{h}_i^l = \text{Attention}(Q : h_i^{l-1}, K, V : \{h_k^{l-1}\}_{k=1}^{i}) \tag{6}$$

$$\hat{g}_i^l = \text{Attention}(Q : g_i^{l-1}, K, V : \{h_k^{l-1}\}_{k=1}^{i-1}) \tag{7}$$

The autoregressive model depends on the order of the input properties because it determines the specific factorization of the joint probability. However, the underlying two-stream causal transformer encoder allows the model to be indifferent to the order of properties serving as conditional variables. Formally, given any permutation $z \in \mathcal{Z}_{i-1}$, where $\mathcal{Z}_{i-1}$ is the set of all permutations of indices $\{1, ..., i-1\}$:

$$\begin{aligned} &p(v_i \mid r_i, z[\{r_k : v_k\}_{k=1}^{i-1}]) \\ =&p(v_i \mid r_i, \{r_k : v_k\}_{k=1}^{i-1}) \end{aligned} \tag{8}$$

5.2.2 Input encoder and final classification

We create h_i^0 similar to the input encoder of the autoencoding model (Eq. 2). We create g_i^0 by adding a special query embedding with the type embedding of the current position.

$$g_i^0 = x^{\text{query}} + x_i^{\text{role}} \tag{9}$$

For final prediction, we apply the final layer introduced previously (Eq. 4) at every position of the sequence starting from $i = 2$. This design allows all factorized probabilities of an n-ary observation to be computed in one forward pass of the neural network model.

5.2.3 Autoregressive training

To learn from different permutation order, for each n-ary observation, we sample a permutation $z \in \mathcal{Z}_n$ and maximize the log-likelihood of the input, i.e.,

$$\mathbb{E}_{z \in \mathcal{Z}_n}\left[\sum_{i=1}^{n} \log p(v_i \mid r_i, \{r_k : v_k\}_{k=1}^{i-1})\right] \tag{10}$$

Similar to prior permutation-based models (Yang et al., 2019; Uria et al., 2016), the trained model serves as an ensemble of models for all possible factorization orders. During inference, the model is able to predict an unknown query property condition on any number of observed properties in any order.

5.3 Implementation details

All components of the model are trained end-to-end. We follow an existing method to use Bayesian optimization (Pelikan et al., 1999) for hyperparameter tuning (Snoek et al., 2012). The best set of parameters is found to be $L = 1$, $H = 4$, $d_{\text{model}} = 240$. We use Adam (Kingma & Ba, 2015) for optimization. We implement our model using PyTorch and train on a Nvidia GTX1080Ti gpu.

6 Experiments on LINK dataset

In this section, we assess the effectiveness of our model for learning n-ary relations between object properties in two different tasks. In the *missing-one* evaluation task, the model is presented with a previously unseen n-ary observation (i.e., situated object instance), and must predict a single missing value given the value's role and all other role-value pairs in the instance. This evaluation task is aligned with the autoencoding training objective used by the LINK-AE model. This task allows us to probe different models' abilities to learn and represent the highest-order relations, which are crucial for robots to accurately model fine-grained correlations between object properties. We use this task to further study the design choices and gain insights into how different types of object properties contribute to reasoning. In the *known-k* evaluation task, which complements the first task, we more closely study reasoning at various levels of abstraction by requiring the model to predict missing values in novel n-ary relations with different arities. For example, a query with two known properties (i.e., $k = 2$) is {*class: cup, material: glass, location: ?*}.

6.1 Experimental setup

Data Preparation: To ensure no test leakage, we first split object instances in the dataset into 70% training, 15% testing, and 15% validation. Situated object instances are then assigned to the correct data split based on its corresponding object instance. For the known-k evaluation task, we create testing n-ary relations by sampling arbitrary numbers of role-value pairs from situated object instances in the test set.[2] To ensure that the n-ary relations are novel, we eliminate a sampled n-ary relation if the combination of its role-value pairs appears as a subset in any situated object instance in the training and validation sets.
Metrics: For each missing value in a test instance, we obtain probabilities of candidate values from the model. Then the candidate values are sorted in descending order based on the probabilities. The rank of the ground-truth value v_n is used to compute metric scores. During ranking, we adopt the filtered setting (Dettmers et al., 2018) to remove any value

[2] We do not enumrate all n-ary relations due to the very large number of combinations of roles and role-value pairs. For example, there are $\binom{200}{5}$ unique combinations of properties when the arity of n-ary is five.

v'_n different from v_n if $\{r_1 : v_1, ..., r_{n-1} : v_{n-1}, r_n : v'_n\}$ exists in the train, validation, or test set. For the missing-one task, this whole procedure is repeated for each value of each testing instance in the test set. For the known-k task, we evaluate the rank for a randomly sampled missing value for each testing instance. We report standard metric Mean Reciprocal Rank (MRR) and proportion of ranks no larger than 1, 2, and 3 (Hits@1, 2, and 3). For both MRR and Hits, a higher score indicates better performance. All results in this section are aggregated results for 9 different random seeds. We also report time spent for running on the whole training and testing sets.

Baselines: We compare against the following baselines:

- **Co-Occur** learns co-occurrence frequency of entities. This model has been used for modeling semantic relations in various robotic applications, including modeling object object co-occurrence (Kunze et al., 2014), object affordance co-occurrence (Chao et al., 2015), and object grasp co-occurrence (Lakani et al., 2018). We apply this model to learn the co-occurrence frequency of object class with object properties in our experiments. The model by design is not able to consider other properties as contextual information.
- **TuckER** is a recent state of the art knowledge graph embedding model (Balazevic et al., 2019). In this paper, we compare to two variants of TuckER. The regular TuckER model follows existing work (Daruna et al., 2019; Arkin et al., 2020) to model binary relations between object class and object properties.
- **TuckER+** is a TuckER embedding model we implement to model binary relations between all pairs of property types (e.g., color and material, shape and location); it approximates an n-ary relation with a combination of binary relations. Specifically, to score an candidate property in an n-ary query, binary relations between the candidate property and each of the known properties are scored and averaged.

Table 4 Ablation on input encoder design

Embeddings			Metric scores			
V	R	Pos	MRR	Hits@1	Hits@2	Hits@3
✓	✓		**76.3**	**63.3**	**79.4**	**89.1**
✓			75.3	62.3	79.0	86.8
✓	✓	✓	74.0	59.9	77.7	87.5
✓		✓	74.0	59.9	77.7	87.5

Bold numbers indicate the best performance

- **NaLP** is a neural network model developed for modeling n-ary relational data in knowledge graphs (Guan et al., 2019). NaLP explicitly models the relatedness of all the role-value pairs in an n-ary observation. We apply this model to learn n-ary relations between object properties.
- **Markov Logic Network (MLN)** represents probabilistic logic languages that have been used to model complex semantic relations in various robotic domains (Nyga et al., 2014; Zhu et al., 2014; Nyga et al., 2018; Ardón et al., 2019; Chernova et al., 2017). We closely follow prior work to specify probabilistic rules for our domain, please see Appendix 1 for details.
- **Masked Language Model (MaskedLM)** shares the same model architecture as the LINK-AE model but uses the pretrained transformer encoder weights from the BERT model (Devlin et al., 2019). Due to its large embedding dimension, this pretrained model has significantly more parameters.

6.2 Results on missing-one task

As shown in Table 3, our **LINK** models outperform existing methods by significant margins on all metrics. Between the two variants, the **LINK-AE** model obtains slightly higher scores. Compared to the second-best model, **MLN**, both our models achieve around 10% increase in MRR while significantly reducing training and testing time; the **LINK-AR**

Table 3 Results% of our model and baseline models

Model	Metric Scores				Time (min)	
	MRR	Hits@1	Hits@2	Hits@3	Training	Testing
Co-Occur	63.0 ± 1.4	44.3 ± 1.9	67.3 ± 1.5	80.2 ± 1.1	<1	3
TuckER	58.7 ± 4.7	38.5 ± 6.5	59.9 ± 6.4	79.3 ± 3.4	<1	3
TuckER+	62.5 ± 3.7	43.0 ± 5.5	65.9 ± 4.2	81.4 ± 1.7	2	3
NaLP	57.9 ± 1.8	38.9 ± 3.1	60.3 ± 2.1	75.8 ± 1.3	8	10
MLN	65.9 ± 2.2	50.1 ± 2.7	68.7 ± 3.0	81.9 ± 1.8	420	487
MaskedLM	61.9 ± 3.4	43.3 ± 4.0	64.8 ± 5.3	78.7 ± 3.8	32	3
LINK-AE	76.3 ± 2.3	63.3 ± 2.7	79.4 ± 2.9	89.1 ± 2.7	3	3
LINK-AR	75.7 ± 1.6	62.3 ± 2.0	79.1 ± 2.5	88.7 ± 1.8	<1	3

Table 5 MRR% of our model and baseline models for each property type

# Values	Class 11	Mat 8	Color 15	Trans 3	Dim 10	Phys 6	Shape 9	Temp 3	Full 3	Damp 3	Clean 3	Room 11	Loc 117	Price 3	Weight 3	Size 3
Random	27.1	33.3	19.9	60.8	18.7	37.7	22.6	61.4	61.5	61.5	61.0	28.5	4.7	60.2	60.7	60.5
Co-Occur	/	55.5	39.2	83.1	17.9	76.1	64.4	91.0	77.4	74.5	67.1	56.1	44.3	56.9	70.7	**70.9**
TuckER	/	56.1	37.7	84.0	16.7	74.3	57.8	83.9	59.9	71.6	61.4	57.3	31.4	55.6	64.6	69.0
TuckER+	53.7	60.0	42.2	85.1	20.1	74.8	60.4	90.9	69.7	72.4	62.6	60.9	39.2	59.7	73.8	65.1
NaLP	45.0	47.4	39.0	84.4	17.7	69.2	52.4	91.0	68.5	70.8	67.7	55.2	23.1	59.0	61.6	61.4
MLN	62.9	**79.1**	72.5	95.7	25.3	79.4	64.3	90.0	69.1	69.5	65.3	67.7	4.6	68.9	75.8	63.7
MaskedLM	47.9	53.3	45.3	85.8	32.0	65.6	68.0	91.1	76.5	74.6	65.5	55.5	40.5	59.5	62.3	66.8
LINK-AE	**72.3**	78.9	**73.2**	**97.1**	**43.8**	84.1	**75.7**	92.3	90.7	82.2	90.2	68.5	59.6	**74.4**	76.6	61.8
LINK-AR	68.0	78.2	72.0	96.6	40.0	85.1	72.3	**92.9**	**90.9**	82.3	**90.6**	**68.8**	**62.5**	71.0	**78.5**	62.7

Bold numbers indicate the best performance

model reduces training time by more than 420 times due to efficient autoregressive training. The significant reduction in computation time allows robots equipped with our models to query a large amount of semantic knowledge in real-time. In comparison with **NaLP**, another model developed specifically for modeling n-ary data, our models' superior performance confirms that the transformer structure and multi-head attention mechanism are more effective at learning the complex semantic relations between object properties. We also observe that **TuckER+**, which learns binary relations between all pairs of object properties, outperforms the regular **TuckER**. This result demonstrates that only modeling class-level semantic knowledge can lead to over-generalization, and that reasoning about the differences between object instances is crucial. It is worth noting that **NaLP**, **TuckER** variants, and **MaskedLM** are not able to outperform the simpler **Co-Occur** model. **TuckER** variants are good at learning the latent structure of binary relations, but the structure does not generalize higher-order relations in this experiment. **NaLP** has shown to be effective at modeling n-ary facts mainly on dataset with 2 to 6 role-values, but it struggles to learn n-ary relations in our data which can have up to 24 role-values. Although finetuning pretrained language models on large binary KGs has shown to be successful in (Bosselut et al., 2019), we notice that this behavior does not generalizes to higher-order relations and a smaller dataset.

Further analyzing MRR for each type of query shown in Table 5, we see that our models outperform existing models in predicting most of the properties. We also notice that the baselines have degraded performance at predicting property types with many candidate values (e.g., location, room, and dimension). **MLN** especially struggles to predict the location role which has 117 possible values. One potential explanation is the closed world assumption being made by **MLN**. Our models learn probabilities of properties and leverage label smoothing to prevent being overconfident at negative training examples. As a result, our models have demonstrated good performance even for these many-valued role types.

6.3 Ablation on input encoder design

We investigate our input encoder design with an ablation study on the **LINK-AE** model. Specifically, we examine the effect of the role embeddings and positional embeddings (discussed in Sect. 5.1.1). Results in Table 4 show that enforcing the order of role-value pairs in an n-ary observation using the positional embeddings results in a drop in performance. The results also confirm that role embeddings are useful for modeling multimodal object properties represented as role-value pairs.

6.4 Visualizing attention

To understand why our transformer-based model is effective at modeling n-ary relational data, we visualize the multi-head attention weights of the **LINK-AE** model, i.e., $\text{softmax}(\frac{QK^T}{\sqrt{d_k}})$. Figure 5 shows the average attention weight assigned to each role when predicting class, physical property, cleanliness, and weight. The attention mechanism exhibits n-ary relational reasoning patterns, which correspond strongly with human intuition-for example, dampness, location, and fullness of an object aids in predicting its cleanliness. Baseline models cannot perform this type of reasoning and thus are not able to model object properties as well as our model.

6.5 Results on known-K task

In this experiment, we compare to a representative subset of the baselines.[3] As shown in Fig. 6, our **LINK-AR** model

[3] We leave **MLN** out because of its exceedingly long inference time on queries with partial evidence (as a large number of properties other than the query properties were missing).

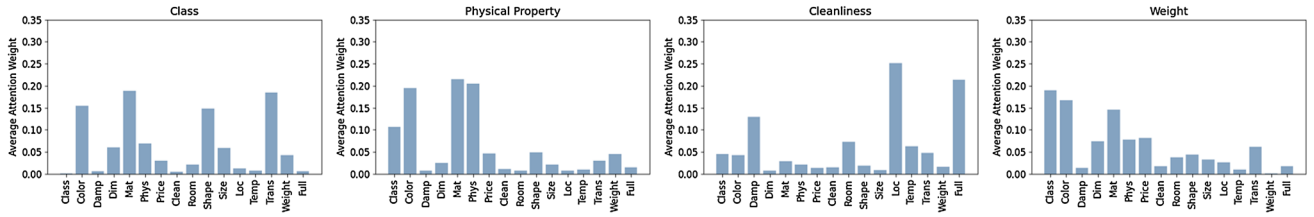


Fig. 5 Visualizations of the attention weights illustrate that different amount of information from each property type is used by our model to predict different types of properties

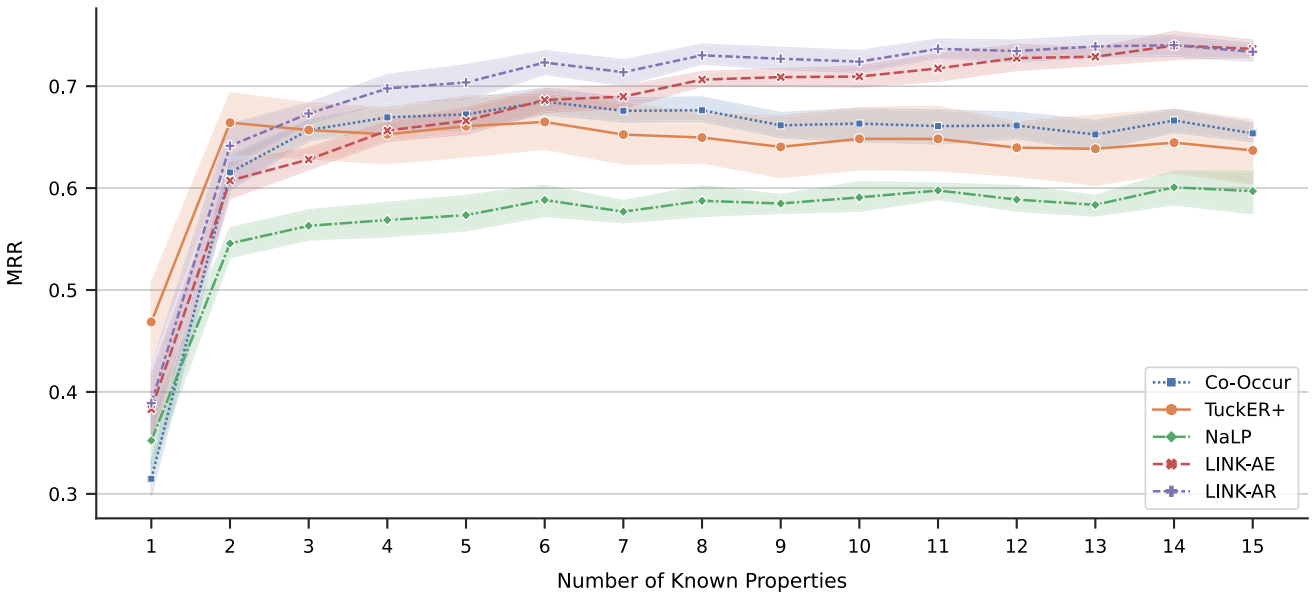


Fig. 6 MRR% of our models and selected baselines on the known-k task. Models are evaluated on predicting missing values of novel n-ary relations with increasingly higher arities. The trendlines show mean and variance

outperforms other models when given more than two known properties. With an increasing number of properties, the performance of all **LINK** variants remain at a high-level while the performance of **Co-Occur** and **TuckER+** starts to slowly decrease. This result corroborates with the finding in the missing-one task that our transformer-based models are more effective than baselines at modeling higher-order relations. Both **Co-Occur** and **TuckER+** rely on binary relations, causing undesirable overgeneralization. It's worthnoting that **TuckER+** has the highest MRRs when given one and two known properties. This result shows that **TuckER+**, as it is designed specifically for pairwise relations, learns a high-quality latent structure that generalizes well to novel binary and ternary relations. However, this superior generalization stops at ternary relations.

7 Robot experiment: object search

In this section, we demonstrate how a household robot can locate specific objects based on users' requests by leveraging the explicit representation of our learned n-ary knowledge. Our experiment serves two purposes, i) to validate our model in a realistic physical setting with non-AMT users, and ii) to test our model's ability to handle queries that reflect realistic use cases, such as a human asking a robot to find a cold beverage or collect dirty dishes. Queries used in this study utilize only a sparse set of known properties,[4] and the robot's task is to predict multiple unknown properties. Specifically, we seek to predict the room and location of each object.

We set up a home environment in our lab with 4 rooms and typical household furniture (Fig. 7). We also generated a corresponding 3D floor plan of the environment (adding an additional bathroom), which listed 24 possible locations for storing objects (Fig. 8). We then recruited 5 users, and had them label their preferred locations for 50 object instances sampled from our dataset. For each object, the user was shown an image of the object, given 1-3 properties describing the state of the object (i.e., cleanliness, temperature, damp-

[4] Human users are unlikely to phrase requests with long adjective sequences.

Fig. 7 Home environment for the object search experiment

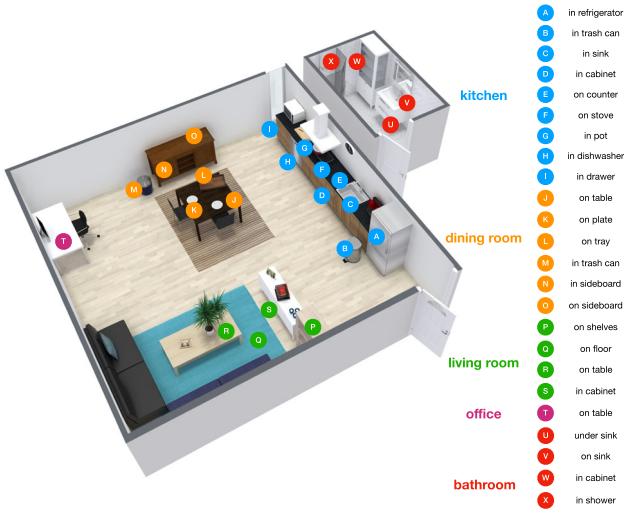


Fig. 8 3D floor plan for remotely collecting preferred locations of objects from five users

ness, and content), and then asked to list 3 ranked likely locations for the object.

We compare the performance of our model against **NaLP**, **Co-Occur**, and **TuckER+**. All models are trained on our complete dataset to validate against collected user data. All models have access to the properties given to the human users as well as the class and material of the object. To predict likely room-location combinations, separately predicted probabilities of the two properties are multiplied and ranked.

We use Hits@K and Hits_Any@K as metrics. Hits@1,2,3 indicate the percentage of times that a model correctly predicts a user's most preferred location of an object within 1, 2, and 3 attempts, respectively. We also introduce Hits_Any@K, which considers a prediction correct if it matches any one of the 3 locations listed by a user, without rank order.

Table 6 summarizes the result of this experiment. We also report the human baseline, which we compute by cross-validating each user against the other users. We observe that the **LINK-AR** model is able to significantly outperform all other models and nearly match human performance at Hits@1 and Hits_Any@1. **Co-Occur** obtains a competitive score on Hits@1 compared to the autoencoding **LINK-AE** model, suggesting that class-level frequency can be an fallback heuristic for finding objects if given only one chance.

Beyond quantitative difference between our model and baselines, we also demonstrate the qualitative improvement on a Fetch robot (Wise et al., 2016). The robot is equipped with the navigation stack developed in (Banerjee et al., 2019) for mapping and navigation, and the method introduced in (Liu et al., 2020) for object detection and grasping. As shown in Fig. 9, the difference (A, B) between our model and **Co-Occur** is clear as our model takes into account of the properties of objects (e.g., cold, dry, clean) while **Co-Occur** searches the same locations for different cups. We also show in Fig. 10 that our model is able to find objects considering both immutable (material in E and F) and mutable properties of objects (dampness in C and D).

8 Robot experiment: integrating with multimodal perception

In this section, we examine whether our model can enable a robot to infer object properties that cannot be directly observed by collectively reasoning about properties extracted from multimodal sensors. This experiment also aims to test whether our model can generalize learned n-ary knowledge to new object instances in the real world.

In this experiment, a robot is tasked to predict either an unknown immutable property of an object based on its class, color, material, and room, or to predict an unknown mutable property based on class, color, material, room, temperature, and location. The robot physically interacts with real objects situated in the environment and leverages different sensing capabilities to extract multimodal observations. We use the same Fetch robot, object detection, and mapping as the previous experiment. Color is detected using OpenCV. Material is detected by the robot using a spectrometer, the SCiO sensor, and the method introduced in (Erickson et al., 2019). Temperature is detected using a Melexis contact-less infrared sensor connected to an Arduino microcontroller. To detect materials and temperatures of objects in real time, the sensors are attached to the end-effector of the robot. The robot uses RRT to plan to poses that allow the sensors to touch the surfaces of the objects. The poses are computed from task-oriented 6-dof grasping poses with the method introduced in (Liu et al., 2020). As shown in Fig. 11, we test on 22 objects which are semantically different from objects in our dataset (e.g., no ceramic pan and plastic box exist in our dataset).

In this experiment, **LINK-AE** and **LINK-AR** are able to correctly predict 27/52 (52%) and 29/52 (56%) of the

Table 6 Results% on object search

	Hits@1	Hits@2	Hits@3	Hits_Any@1	Hits_Any@2	Hits_Any@3
Human Baseline	34.8 ± 6.5	52.0 ± 7.0	64.7 ± 7.3	64.7 ± 7.2	83.2 ± 7.0	90.6 ± 3.9
Co-Occur	19.2 ± 4.6	28.8 ± 3.0	37.6 ± 3.3	50.0 ± 4.2	68.4 ± 7.4	80.0 ± 5.1
TuckER+	12.0 ± 2.0	21.2 ± 3.0	27.2 ± 4.1	42.8 ± 9.2	60.0 ± 8.1	76.6 ± 9.1
NaLP	2.8 ± 3.3	5.6 ± 4.1	10.4 ± 3.8	10.8 ± 13.2	17.2 ± 14.4	19.6 ± 14.0
LINK-AE	20.0 ± 2.0	39.6 ± 10.3	47.6 ± 8.2	55.2 ± 5.4	77.6 ± 8.9	84.8 ± 7.6
LINK-AR	34.0 ± 11.8	46.8 ± 7.3	52.4 ± 7.1	64.0 ± 11.2	77.6 ± 5.2	86.8 ± 4.8

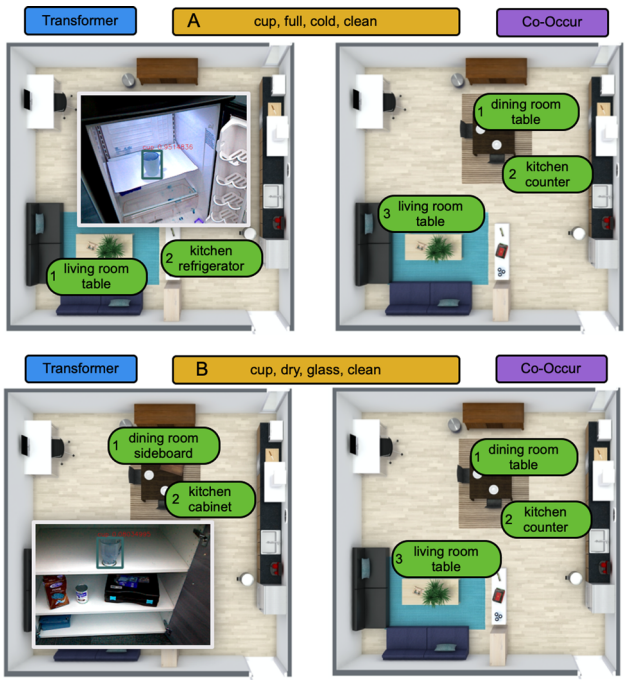


Fig. 9 Two object search tests comparing our model with Co-Occur. Provided properties are shown on top

Fig. 10 Our model predicts different locations based on the given object properties

queried object properties. In comparison, the third best performing models, **TuckER+** and **Co-Occur**, both correctly predict 24/52 (46%). Object materials are correctly detected 45/52 (87%) times. Figure 11 shows examples of the queries and predictions from the **LINK-AE** model.

9 Conclusion

This work addresses the problem of predicting semantic properties of objects based on partial observations. We introduce two scalable transformer neural networks that learn n-ary relations between object properties from n-ary observations, where each represents a set of identified properties of a specific object situated in a particular environmental context. The LINK-AE autoencoding model is trained to predict a single missing property given all other properties in each specific n-ary observation while the LINK-AR autoregressive model directly predicts factorized probabilities of all properties in each n-ary observaton. To train and evaluate our approach, we contribute LINK, a dataset containing objects situated in various environmental contexts and modeled by diverse semantic properties.

Results of the missing-one experiment show that both our models, LINK-AE and LINK-AR, are able to outperform prior state-of-the-art Markov Logic Network with 10% improvement in accuracy and 150 times improvement in computation efficiency. The known-k experiment further demonstrates that prior methods gradually lose accuracy when dealing with increasingly specific relations (i.e., 8th to 17th order relations) while our methods maintain high accuracy. The result also indicates that LINK-AR can better model more abstract relations (i.e., 3rd to 8th order relations) than LINK-AE because LINK-AR is trained to directly predict factorized probabilities conditioning on different numbers

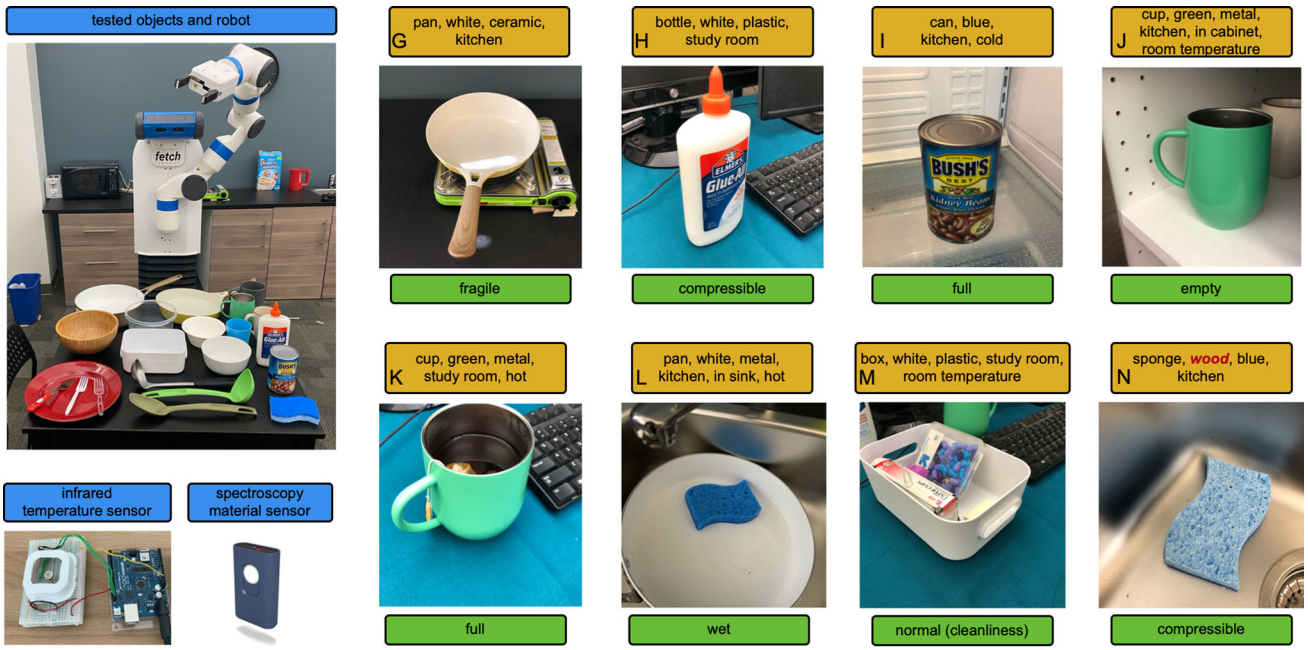


Fig. 11 The Fetch robot uses an infrared temperature sensor to detect the temperature and a spectrometer to detect the material of each novel object situated in different environmental contexts. Our model leverages extracted information (shown on top of each figure on the right) to predict an unknown object property (shown on bottom)

of observed properties. In addition, we evaluate our models in two robot experiments, demonstrating that modeling instance-level knowledge about object properties enables robots to search objects based on object states and properties and jointly reason about properties detected by multimodal perception. The robot experiments confirm that while both our models outperform baselines, LINK-AR obtains higher performance in realistic settings where only a small amount of properties are observed.

Besides the object search and multimodal perception applications examined in this work, our semantic reasoning framework can potentially be applied to a wider variety of robotic tasks that require modeling and inferring object properties, including grounding abstract instructions to task plans (Nyga et al., 2018; Misra et al., 2016) and repairing task plans by substituting objects (Daruna et al., 2021). We are also interested in more closely integrating our methods with multimodal and interactive perception. One potential path is to develop a multimodal transformer network (Xu et al., 2022) which can takes in observations of object properties in the form of detected semantic labels and raw sensor inputs. This approach would allow the model to develop a joint embedding space of high-level symbolic concept and low-level perceptual data. Another direction is to combine n-ary knowledge reasoning with sequential decision making to guide interactive perception of object properties. Most existing methods (Chu et al., 2016; Sinapov et al., 2014) exhaustively perform pre-defined exploratory actions. This strategy assumes a limited set of interactions, is time-consuming, and can cause damage to objects in more realistic settings (e.g., dropping a glass cup reveals a unique sound signal but may also break it). Our n-ary representation of object properties can potentially enable robots to actively select exploratory actions to identify properties of interest.

Funding This work was supported by NSF IIS 1564080, NSF GRFP DGE-1650044, and ONR N00014-16-1-2835.

Declarations

Conflict of interest: The authors have no financial or proprietary interests in any material discussed in this article.

Appendix A Markov logic network: background and implementation

A Markov Logic Network (MLN) combines the ideas of first-order logic with Markov networks by assigning lower probability to worlds that violate more first-order logic formulas (Richardson & Domingos, 2006). In this way, an MLN can learn joint distributions that include first-order logic constraints. For each formula, the MLN learns a weight about relative influence the formula has over the likelihood of a possible world. Given a subset of variable assignments, the MLN can infer the likelihoods of possible complete variable assignments.

Table 7 Example MLN formula templates

Role MLN	MLN formula template in pracmln format (variables in **bold**)
Class	has_class(+**class**) ∧ has_shape(+**shape**)
	has_class(+**class**) ∧ has_specific_place(+**specific_place**)
	has_class(+**class**) ∧ has_color(+**color**)
	has_class(+**class**) ∧ in_room(+**room**)
	has_class(+**class**) ∧ has_dimension(+**dimension**)
	has_class(+**class**) ∧ has_material(+**material**)
	has_class(+**class**) ∧ has_physical_property(+**physical_property**)
Size	has_size(+**size**) ∧ has_physical_property(+**physical_property**)
	has_size(+**size**) ∧ has_shape(+**shape**)
	has_size(+**size**) ∧ has_class(+**class**)
	has_size(+**size**) ∧ has_weight(+**weight**)
Weight	has_weight(+**weight**) ∧ has_class(+**class**)
	has_weight(+**weight**) ∧ has_color(+**color**)
	has_weight(+**weight**) ∧ has_size(+**size**)
	has_weight(+**weight**) ∧ has_material(+**material**)
	has_weight(+**weight**) ∧ has_dimension(+**dimension**)

In pracmln, each formula template in is grounded to multiple formulas to cover the domain of each variable in the formula (e.g., **weight** = {light, medium, heavy}), denoted by the '+' symbol

A naive MLN implementation for this problem using one monolithic MLN for all roles and complex MLN formulas (i.e. non-binary) is not possible because the large number of roles and values per role leads to computationally infeasible grounded markov networks to run MLN training and inference. Our best performing MLN implementation using pracmln (Nyga et al., 2013), was achieved by modeling each role using a dedicated MLN. The formulas of each MLN dedicated to a role were hand-tuned to maximize MLN inference performance for that role while being computationally feasible, see Table 7 for example formula templates we used. By using a dedicated MLN per role, we can include more relationships that improve test performance and remove relationships that degrade test performance, while avoiding computationally infeasible grounded Markov network training and inference due to extraneous relationships modeled by formulas for other roles.

References

Abboud, R., Ceylan, I., Lukasiewicz, T., & Salvatori, T. (2020). Boxe: A box embedding model for knowledge base completion. In *Neurips proceedings*.

Amiri, S., Wei, S., Zhang, S., Sinapov, J., Thomason, J., & Stone, P. (2018). Multi-modal predicate identification using dynamically learned robot controllers. In *IJCAI*.

Ardón, P., Pairet, È., Petrick, R. P., Ramamoorthy, S., & Lohan, K. S. (2019). Learning grasp affordance reasoning through semantic relations. *IEEE Robotics and Automation Letters, 4*, 4571–4578.

Arkin, J., Park, D., Roy, S., Walter, M. R., Roy, N., Howard, T. M., & Paul, R. (2020). Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions. *The International Journal of Robotics Research, 39*, 1279–1304.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Balazevic, I., Allen, C., & Hospedales, T. (2019). TuckER: Tensor factorization for knowledge graph completion. In *Emnlp-ijcnlp*.

Banerjee, S., Daruna, A., Kent, D., Liu, W., Balloch, J., Jain, A., & Chernova, S. (2019). Taking recoveries to task: Recovery-driven development for recipe-based robot tasks. In *ISRR*.

Bhattacharjee, T., Clever, H. M., Wade, J., & Kemp, C. C. (2018). Multimodal tactile perception of objects in a real home. *IEEE Robotics and Automation Letters, 3*, 2523–2530.

Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., & Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics, 33*, 1273–1291.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). Comet: Commonsense transformers for automatic knowledge graph construction. In *ACL*.

Chao, Y.-W., Wang, Z., Mihalcea, R., & Deng, J. (2015). Mining semantic affordances of visual object categories. In *CVPR*.

Chen, H., Tan, H., Kuntz, A., Bansal, M., & Alterovitz, R. (2020). Enabling robots to understand incomplete natural language instructions using commonsense reasoning. In *ICRA*.

Chernova, S., Chu, V., Daruna, A., Garrison, H., Hahn, M., Khante, P., & Thomaz, A. (2017). Situated bayesian reasoning framework for robots operating in diverse everyday environments. In *ISRR*.

Chu, V., Fitzgerald, T., & Thomaz, A. L. (2016). Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *HRI*.

Chuang, C.-Y., Li, J., Torralba, A., & Fidler, S. (2018). Learning to act properly: Predicting and explaining affordances from images. In *CVPR*.

Chu, V., McMahon, I., Riano, L., McDonald, C. G., He, Q., Perez-Tejada, J. M., & Kuchenbecker, K. J. (2015). Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems, 63*, 279–292.

Daruna, A., Liu, W., Kira, Z., & Chetnova, S. (2019). Robocse: Robot common sense embedding. In *2019 international conference on robotics and automation.*

Daruna, A., Nair, L. V., Liu, W., & Chernova, S. (2021). Towards robust one-shot task execution using knowledge graph embeddings. In *International conference on robotics and automation.*

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence* (Vol 32).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies* (Vol. 1, pp. 4171–4186).

Do, T.-T., Nguyen, A., & Reid, I. (2018). Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation* (pp. 5882–5889).

Dyke, R. M., Zhou, F., Lai, Y.-K., & Rosin, P. L. (2020). Shrec 2020 track: Non-rigid shape correspondence of physically-based deformations. In *13th eurographics workshop on 3d object retrieval, 3dor 2020-short papers, Graz, Austria, September 4–5, 2020.* Eurographics Association.

Eppe, M., Kerzel, M., Strahl, E., & Wermter, S. (2018). Deep neural object analysis by interactive auditory exploration with a humanoid robot. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 284–289).

Erickson, Z., Luskey, N., Chernova, S., & Kemp, C. C. (2019). Classification of household materials via spectroscopy. *IEEE Robotics and Automation Letters, 42*, 700–707.

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009). Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1778–1785).

Ferrari, V., & Zisserman, A. (2007). Learning visual attributes. *Advances in Neural Information Processing Systems, 20*, 433–440.

Galkin, M., Trivedi, P., Maheshwari, G., Usbeck, R., & Lehmann, J. (2020). Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 7346–7359).

Gandhi, D., Mulam, H., & Pinto, L. (2020). Swoosh! rattle! thump!—actions that sound. In *Proceedings of robotics: Science and systems.*

Gao, R., Chang, Y.-Y., Mall, S., Fei-Fei, L., & Wu, J. (2021). Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations.

Guan, S., Jin, X., Wang, Y., & Cheng, X. (2019). Link prediction on n-ARY relational data. In *The world wide web conference* (pp. 583–593).

Günther, M., Ruiz-Sarmiento, J., Galindo, C., Gonzalez-Jimenez, J., & Hertzberg, J. (2018). Context-aware 3d object anchoring for mobile robots. *Robotics and Autonomous Systems, 110*, 12–32.

Gupta, R., Kochenderfer, M. J., Mcguinness, D., & Ferguson, G. (2004). Common sense data acquisition for indoor mobile robots. In *AAAI* (pp. 605–610).

Hamp, B., & Feldweg, H. (1997). Germanet—A lexical-semantic net for German. In *Automatic information extraction and building of lexical semantic resources for NLP applications.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Huang, Y., & Sun, Y. (2019). A dataset of daily interactive manipulation. *The International Journal of Robotics Research, 38*, 879–886.

Jain, A., Wojcik, B., Joachims, T., & Saxena, A. (2013). Learning trajectory preferences for manipulators via iterative improvement. In *Advances in neural information processing systems* (pp. 575–583).

Jenkins, P., Sachdeva, R., Kebe, G. Y., Higgins, P., Darvish, K., Raff, E., & Matuszek, C. (2020). Presentation and analysis of a multimodal dataset for grounded language learning. arXiv preprint arXiv:2007.14987.

Kerr, E., McGinnity, T. M., & Coleman, S. (2018). Material recognition using tactile sensing. *Expert Systems with Applications, 94*, 94–111.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings.

Kolve, E., Mottaghi, R., Gordon, D., Zhu, Y., Gupta, A., & Farhadi, A. (2017). Ai2-thor: An interactive 3d environment for visual AI. arXiv preprint arXiv:1712.05474.

Kunze, L., Burbridge, C., Alberti, M., Thippur, A., Folkesson, J., Jensfelt, P., & Hawes, N. (2014). Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding. In *2014 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2910–2915).

Lakani, S. R., Rodríguez-Sánchez, A. J., & Piater, J. (2018). Exercising affordances of objects: A part-based approach. *IEEE Robotics and Automation Letters, 34*, 3465–3472.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., & Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning* (pp. 3744–3753).

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., & Alami, R. (2017). Artificial cognition for social human-robot interaction: An implementation. *Artificial Intelligence, 247*, 45–69.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM, 38*(11), 33–38.

Levine, S., Pastor, P., Krizhevsky, A., & Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.

Li, Q., Kroemer, O., Su, Z., Veiga, F. F., Kaboli, M., & Ritter, H. J. (2020). A review of tactile information: Perception and action through touch. *IEEE Transactions on Robotics, 36*(6), 1619–1634.

Lim, G. H., Suh, I. H., & Suh, H. (2011). Ontology-based unified robot knowledge for service robots in indoor environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41*(3), 492–509.

Liu, W., Bansal, D., Daruna, A., & Chernova, S. (2021). Learning instance-level N-Ary semantic knowledge at scale for robots operating in everyday environments. In *Proceedings of robotics: Science and systems.* Virtual.

Liu, W., Daruna, A., & Chernova, S. (2020). Cage: Context-aware grasping engine. In *International conference on robotics and automation (ICRA).*

Liu, Y., Yao, Q., & Li, Y. (2020). Generalizing tensor decomposition for n-ARY relational knowledge bases. In *Proceedings of the web conference 2020* (pp. 1104–1114).

Liu, H., & Singh, P. (2004). Conceptnet—A practical commonsense reasoning tool-kit. *BT Technology Journal, 22*(4), 211–226.

Luo, S., Bimbo, J., Dahiya, R., & Liu, H. (2017). Robotic tactile perception of object properties: A review. *Mechatronics, 48*, 54–67.

Lynott, D., & Connell, L. (2009). Modality exclusivity norms for 423 object properties. *Behavior Research Methods, 41*(2), 558–564.

Miller, G. A. (1995). Wordnet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41.

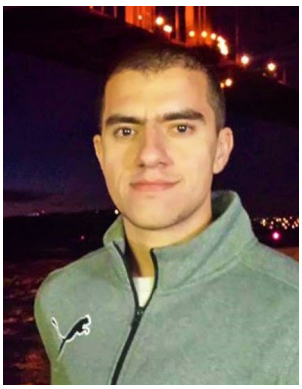
Misra, D. K., Sung, J., Lee, K., & Saxena, A. (2016). Tell me DAVE: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research, 35*(1–3), 281–300.

Moldovan, B., & Raedt, L. D. (2014). Occluded object search by relational affordances. In *IEEE international conference on robotics and automation (ICRA)* (pp. 169–174).

Nazarczuk, M., & Mikolajczyk, K. (2020). Shop-VRB: A visual reasoning benchmark for object perception. In *2020 IEEE international conference on robotics and automation (ICRA)* (pp. 6898–6904).
Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE, 104*(1), 11–33.
Nyga, D., Balint-Benczedi, F., & Beetz, M. (2014). Pr2 looking at things—Ensemble learning for unstructured information processing with Markov logic networks. In *2014 IEEE international conference on robotics and automation (ICRA)* (pp. 3916–3923).
Nyga, D., Picklum, M., & Beetz, M., et al. (2013). *Pracmln—Markov logic networks in Python.* Online Accessed 2022.
Nyga, D., Roy, S., Paul, R., Park, D., Pomarlan, M., Beetz, M., & Roy, N. (2018). Grounding robot plans from natural language instructions with incomplete world knowledge. In *Conference on robot learning* (pp. 714–723).
Pelikan, M., Goldberg, D.E., & Cantú-Paz, E., et al. (1999). Boa: The bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99* (Vol 1, pp. 525–532).
Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning, 62*(1), 107–136.
Rosso, P., Yang, D., & Cudré-Mauroux, P. (2020). Beyond triplets: Hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020* (pp. 1885–1896).
Saxena, A., Jain, A., Sener, O., Jami, A., Misra, D. K., & Koppula, H. S. (2014). Robobrain: Large-scale knowledge engine for robots. arXiv preprint arXiv:1412.0691.
She, Q., Feng, F., Hao, X., Yang, Q., Lan, C., & Lomonaco, V. (2020). Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 IEEE international conference on robotics and automation (ICRA)* (pp. 4767–4773).
Shridhar, M. , & Hsu, D. (2018). Interactive visual grounding of referring expressions for human–robot interaction. In *Proceedings of robotics: Science and systems.*
Sinapov, J., Schenck, C., Staley, K., Sukhoy, V., & Stoytchev, A. (2014). Grounding semantic categories in behavioral interactions: Experiments with 100 objects. *Robotics and Autonomous Systems, 62*(5), 632–645.
Singh, A., Sha, J., Narayan, K.S., Achim, T., & Abbeel, P. (2014). Bigbird: A large-scale 3d database of object instances. In *2014 IEEE international conference on robotics and automation (ICRA)* (pp. 509–516).
Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., & Kiela, D. (2021). Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 2888–2913). Association for Computational Linguistics.
Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25.
Sun, Y., Bo, L., & Fox, D. (2013). Attribute based object identification. In *2013 IEEE international conference on robotics and automation* (pp. 2096–2103).
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
Tatiya, G. , & Sinapov, J. (2019). Deep multi-sensory object category recognition using interactive behavioral exploration. In *2019 international conference on robotics and automation (ICRA)* (pp. 7872–7878).
Tatsuma, A., Koyanagi, H., & Aono, M. (2012). A large-scale shape benchmark for 3d object retrieval: Toyohashi shape benchmark. In *Proceedings of the 2012 Asia pacific signal and information processing association annual summit and conference* (pp. 1-10).
Tenorth, M., & Beetz, M. (2017). Representations for robot knowledge in the Knowrob framework. *Artificial Intelligence, 247*, 151–169.
Thomason, J., Sinapov, J., Mooney, R.J., & Stone, P. (2018). Guiding exploratory behaviors for multi-modal grounding of linguistic descriptions. In *Thirty-second AAAI conference on artificial intelligence.*
Thosar, M., Mueller, C. A., Jäger, G., Schleiss, J., Pulugu, N., Mallikarjun Chennaboina, R., & Zug, S. (2021). From multi-modal property dataset to robot-centric conceptual knowledge about household objects. *Frontiers in Robotics and AI, 8*, 87.
Uria, B., Côté, M.-A., Gregor, K., Murray, I., & Larochelle, H. (2016). Neural autoregressive distribution estimation. *The Journal of Machine Learning Research, 17*(1), 7184–7220.
Varadarajan, K. M. , & Vincze, M. (2013). Afnet: The affordance network. In K. M. Lee, Y. Matsushita, J. M. Rehg, & Z. Hu (Eds.), Computer vision—ACCV 2012 (pp. 512–523). Springer.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
Wade-McCue, S., Kelly-Boxall, N., McTaggart, M., Morrison, D., Tow, A. W., Erskine, J., & Leitner, J. (2018). Design of a multi-modal end-effector and grasping system: How integrated design helped win the amazon robotics challenge.
Wang, Q., Huang, P., Wang, H., Dai, S., Jiang, W., Liu, J., & Wu, H. (2019). Coke: Contextualized knowledge graph embedding. arXiv preprint arXiv:1911.02168.
Watters, N., Matthey, L., Bosnjak, M., Burgess, C. P., & Lerchner, A. (2019). Cobra: Data-efficient model-based RL through unsupervised object discovery and curiosity-driven exploration.
Wen, J., Li, J., Mao, Y., Chen, S., & Zhang, R. (2016). On the representation and embedding of knowledge bases beyond binary relations. In *IJCAI.*
Wise, M., Ferguson, M., King, D., Diehr, E., & Dymesich, D. (2016). Fetch and freight: Standard platforms for service robot applications. In *Workshop on autonomous mobile service robots.*
Xu, P., Zhu, X., & Clifton, D. A. (2022). Multimodal learning with transformers: A survey.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.
Yang, W., Wang, X., Farhadi, A., Gupta, A., & Mottaghi, R. (2019). Visual semantic navigation using scene priors. In *Proceedings of seventh international conference on learning representations (ICLR 2019).*
Zeng, Z., Röfer, A., Lu, S., & Jenkins, O. C. (2019). Generalized object permanence for object retrieval through semantic linking maps. In *IEEE ICRA 2019 workshop on high accuracy mobile manipulation in challenging environments.*
Zhang, R., Li, J., Mei, J., & Mao, Y. (2018). Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 world wide web conference.*
Zhang, H., Yang, D., Wang, H., Zhao, B., Lan, X., Ding, J., & Zheng, N. (2021). Regrad: A large-scale relational grasp dataset for safe and object-specific robotic grasping in clutter.
Zhu, Y., Fathi, A., & Fei-Fei, L. (2014). Reasoning about object affordances in a knowledge base representation. In *European conference on computer vision* (pp. 408–424).

Weiyu Liu is a Ph.D. student at Georgia Institute of Technology, where he works in the Robot Autonomy and Interactive Learning (RAIL) lab and is advised by Professor Sonia Chernova. He received his undergraduate degree in Electrical Engineering from Georgia Institute of Technology in 2017. His research interests are on semantic reasoning for robotic systems and in particular knowledge graph reasoning and semantic manipulation.

Dhruva Bansal is a master student at Stanford University and a Research Assistant at the Stanford Vision Lab. He received his B.S degree from the Georgia Institute of Technology, Atlanta, GA, in 2021. His research interests include robotics and reinforcement learning.



Angel Daruna is a Ph.D. student in the Institute for Robotics and Intelligent Machines at the Georgia Institute of Technology and a graduate researcher in the Robot Autonomy and Interactive Learning (RAIL) lab. He received his B.S. degree from the Georgia Institute of Technology, Atlanta, GA, in 2016. His research interests include robotics, knowledge representations and reasoning, and machine learning.

Sonia Chernova is an Associate Professor in the School of Interactive Computing at Georgia Tech, where she directs the Robot Autonomy and Interactive Learning research lab. Her research spans semantic reasoning, human–robot interaction, interactive machine learning and cloud robotics, with the focus on developing robots that are able to effectively operate in human environments. She is the recipient of the NSF CAREER, ONR Young Investigator, and NASA Early Career Faculty awards.