

SIM-TO-REAL TRANSFER OF VISUAL GROUNDING FOR HUMAN-AIDED AMBIGUITY RESOLUTION

Georgios Tziaras

Department of Artificial Intelligence
University of Groningen
The Netherlands
g.t.tziaras@rug.nl

Hamidreza Kasaei *

Department of Artificial Intelligence
University of Groningen
The Netherlands
hamidreza.kasaei@rug.nl

ABSTRACT

Service robots should be able to interact naturally with non-expert human users, not only to help them in various tasks, but also to receive guidance in order to resolve ambiguities that might be present in the instruction. We consider the task of visual grounding, where the agent segments an object from a crowded scene given a natural language description. Modern holistic approaches to visual grounding usually ignore language structure and struggle to cover generic domains, therefore relying heavily on large datasets. Additionally, their transfer performance in RGB-D datasets suffers due to high visual discrepancy between the benchmark and the target domains. Modular approaches marry learning with domain modeling and exploit the compositional nature of language to decouple visual representation from language parsing, but either rely on external parsers or are trained in an end-to-end fashion due to the lack of strong supervision. In this work, we seek to tackle these limitations by introducing a fully decoupled modular framework for compositional visual grounding of entities, attributes and spatial relations. We exploit rich scene graph annotations generated in a synthetic domain and train each module independently in simulation. Our approach is evaluated both in simulation and in two real RGB-D scene datasets. Experimental results show that the decoupled nature of our framework allows for easy integration with domain adaptation approaches for Sim-To-Real visual recognition, offering a data-efficient, robust, and interpretable solution to visual grounding in robotic applications.

1 INTRODUCTION

Natural Human-Robot Interaction (HRI) remains as one of the most valuable assets for service robots operating in human-centric environments. In such dynamic environments, the robot needs to be able to receive verbal input from non-expert users in order to coordinate its perception and manipulation functions (e.g. user says: “*Give me the mug*”). The user should be also encouraged to assist the robot’s perception in resolving ambiguities (e.g. mug appears twice in the scene), by querying for visual (e.g. color, i.e. “*the red mug*”) or spatial (“*the mug next to the flashlight*”) concepts, as shown in Figure 1. Most deployed robots miss this capability by treating their perception and HRI modules separately, often hard-coding maps from verbal instructions to specific object/scene setups. Such methods suffer from the limitation of not enabling *natural* interaction with non-expert users (e.g. given object query does not appear in the hard-coded object set) and the need to be explicitly re-programmed for each new added object category or way of referring to an object. A bridging of the two modules is viable through end-to-end multi-modal learning models that are trained in image-text pairs. Visual grounding has been a central theme in recent deep learning literature, with an array of methods originating from different communities and for different applications. Most methods employ a two-step training

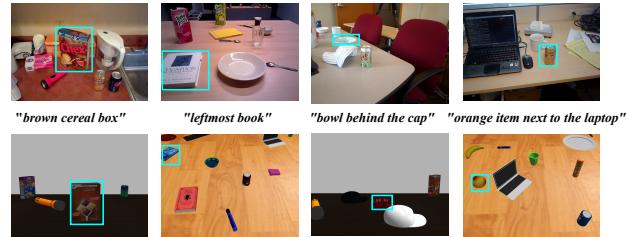


Figure 1: Disambiguation of objects according to natural language query, including (from left to right): reference to visual attribute (color), reference to location, reference to spatial relation with another object and combined. Examples are included from a real RGB-D scene (top-row), as well as from synthetically generated scenes (bottom-row).

* We thank NVIDIA Corporation for their generous donation of GPUs which was partially used in this research.

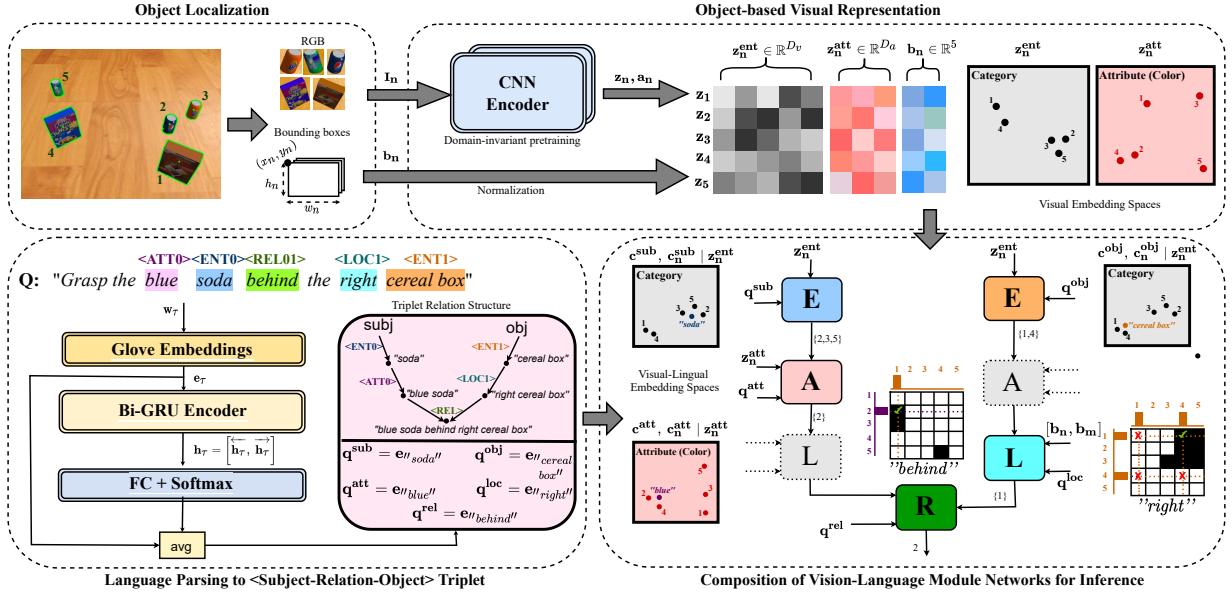


Figure 2: A schematic of the proposed framework, illustrated with a running example for the given image and the query: “*Grasp the blue soda behind the right cereal box*”. First, a localization module extracts object crops. Each crop is passed through a pair of CNN encoders (one for entity and one for attributes) and is concatenated with its bounding box features to constitute an object-based representation. For the language input, a word tagger assigns roles on each word in the sentence which indicates their grammatical role in the phrase. Role tags are replaced by (averaged) word embeddings and the triplet structured resolving the relation is mapped to an assembly of modules for inference.

strategy, with the first step being the pre-training of an object detector, therefore suffering from being bound to a pre-defined object list. More recent works address this with representation learning techniques, such as zero/few-shot grounding (Sadhu et al., 2019; Yang et al., 2019c), self-supervised (Li et al., 2019; Lu et al., 2019a) or multi-task (Lu et al., 2019b) pre-training. However, associated benchmark datasets struggle to cover generic, domain-agnostic content and thus rely on massive scale to be transferable effectively. Moreover, their transfer performance often suffers due to the domain discrepancy between the ideally pre-processed images of the training benchmarks and the noisy real-time sensory images. The alternative would be to design a domain-specific dataset directed towards HRI applicability (as in Hatori et al. (2017)), something which comes at the great cost of having to manually collect data and provide linguistic annotations depending on the environment/scenarios that the agent is operating. In contrary, robotic manipulation and perception have been drastically benefited by the *Sim-to-Real Transfer* paradigm, which offers a platform for data-efficient, domain-specific solutions that omit the need for manual annotation by utilizing tools to simulate the real world.

In this paper, we propose a novel deep learning approach for Sim-to-Real transfer of visual grounding. The model breaks down the overall task to grounding of entities, attributes, and spatial relations separately. We develop a synthetic domain of household objects in a simulation environment and extract scene graph annotations that provide pair-wise relations for objects and their properties. We use this dataset to train our modules in a fully decoupled fashion and integrate it with a domain adaptation approach for visual recognition in order to explore its Sim-to-Real transfer potential. In summary, our key contributions are threefold:

- We propose a new deep learning architecture that combines two modular formulations for decoupled visual grounding of entities, attributes, and spatial relations, while enabling strong supervision from pair-wise relation information.
- We generate a synthetic dataset of household objects in tabletop scenes, paired with a small-scale real counterpart, aimed for evaluating Sim-to-Real transfer for visual recognition and grounding. Moreover, we establish a baseline for these tasks using our model combined with a popular domain adaptation technique.
- We perform extensive sets of experiments on synthetic and real datasets to show the benefits of our model in terms of data-efficiency and robustness to input query complexity. Furthermore, we integrate our system with a simulated and a real robot to use it as a proxy for visual grounding in an HRI scenario.

2 RELATED WORK

Although an in-depth review is beyond the scope of this work, we discuss a few recent efforts in *visual grounding*, *module networks* and *robotics-related synthetic domains*.

Visual grounding: Also called *referring expression comprehension*, visual grounding concerns the task of localizing a target region in an image according to a natural language query. Their key objective is that the model interprets linguistic information to reason about the scene, including entity properties and relations. Many different approaches have been developed for visual grounding that can be organized into four main categories, including: (i) joint-embedding models, which use either CNN-LSTMs or attention mechanisms to rank regions according to the conditional probability given the expression (Mao et al., 2015; Yu et al., 2016; Rohrbach et al., 2015; Luo & Shakhnarovich, 2017); (ii) approaches that use an external parser to retrieve the order of referential reasoning exploiting linguistic cues (Liu et al., 2018; Cirik et al., 2018); (iii) graph-based methods, which represent objects as nodes and their relations as edges (Wang et al., 2018; Yang et al., 2019a;b), attempting to model complex relationships by updating neighbouring information with message passing; and (iv) transformer-based pre-trained models, that follow the success of pre-training large transformer-based models in NLP for vision-and-language data, either in a self-supervised (Li et al., 2019; Lu et al., 2019a; Chen et al., 2019) or a multi-task (Lu et al., 2019b) formulation. Most recently, transformer architectures have been employed for standalone visual grounding (Du et al., 2021).

Module networks Architectures based on neural module networks (Andreas et al., 2015) have been studied for vision-and-language learning tasks, both in visual grounding (Hu et al., 2016; Yu et al., 2018) and in VQA context (Li et al., 2020; Mao et al., 2019; Yi et al., 2018). Similar to our approach, module networks break the overall architecture into separate modules, each responsible for grounding a specific concept in the input query. We base our approach on CMN (Hu et al., 2016), which assumes a triplet subject-relation-object structure in the input query and ground entities and relations separately by aggregating pair-wise matching scores. To deal with absolute relations that break the assumed triplet structure (e.g. “*The right bowl*” is equivalent to “*The bowl right from the other bowls*”), Yu et al. (2018) introduces a location module that aims at grounding relations between objects of the same category. Moreover, they aid entity grounding by treating attribute prediction as a separate branch in the entity module. In this work we aim to combine ideas from both methods to develop a model that deals with attributes and absolute location while maintaining the triplet assumption, which provides a strong prior for generalizing to more complex queries. Both mentioned models are trained end-to-end in the weakly supervised setting, using only the ground truth bounding box as the supervisory signal, as pair-wise relation annotations are rarely provided in benchmark datasets. In contrary, we further exploit rich supervision from a synthetic dataset and train all modules in a decoupled fashion, which as we show in section 4.3 allows for data-efficient adaptation to real scenes, while entertaining high interpretability.

Robotics-related synthetic domains Pre-training in synthetic datasets is a broadly adapted practise for robotic manipulation (Kaspar et al., 2020; Zhao et al., 2020; Kadian et al., 2020), planning (Truong et al., 2020; Zhu et al., 2019) and 3D vision tasks (Doersch & Zisserman, 2019; Raghavan et al., 2020). For visiolinguistic data, the field is much drier, with barely any benchmarks for vision-and-language tasks in RGB-D data. An exception is the VQA benchmark SHOP-VRB (Nazarczuk & Mikolajczyk, 2020) which models a kitchen environment of 20 distinct objects, with rich attribute annotations (e.g weight, mobility etc.). The development of the benchmark is based on CLEVR (Johnson et al., 2016), a synthetic dataset of abstract object shapes in arbitrary spatial arrangements. The dataset is equipped with question-answer-functional program annotations, used mainly to diagnose the visual reasoning abilities of established models. Similarly, we build on top of the CLEVR annotation engine to automatically sample visual grounding queries from scene graph representations that we generate in a simulation environment.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

We formulate the visual grounding task in the retrieval setting. Given the RGB frame of the input scene, represented as a set of N extracted *Region of Interest (ROI)* proposals, \mathbf{o}_n , and a natural language query, Q , that refers to a specific object depicted, the goal is to find the $n^* = \text{argmax}_n S(\mathbf{o}_n, Q)$, where $n \in \{1, \dots, N\}$ and $S_n := S(\mathbf{o}_n, Q)$ is a similarity function between any ROI and the query.

We adopt the subject-relation-object triplet composition method of Hu et al. (2016), but adapt our modules to also include absolute location queries, such as Yu et al. (2018). More specifically, we formulate the similarity function as

$S_n := R(S_n^{sub}, S_n^{obj})$, where:

$$R(S_n^{sub}, S_n^{obj}) = \max_{m \in \{1, \dots, N\}} (S_n^{sub} + S_{nm}^{rel} + S_m^{obj}) \quad (1)$$

This function estimates the similarity between each ROI and the query (e.g. “*the bowl next to the bottle*”), by aggregating unitary matching scores between the ROIs and isolated words / phrases that correspond to the two syntactic roles, i.e. subject \mathbf{q}_n^{sub} (e.g. “*bowl*”) and object \mathbf{q}_n^{obj} (e.g. “*bottle*”), with pair-wise scores between the ROIs and a relation word $S_{nm}^{rel} = S_{nm}^{rel}(\mathbf{o}_n, \mathbf{o}_m, \mathbf{q}_n^{rel})$ (e.g. “*next to*”). The unitary scores for each ROI are computed after composing two vision-language matching modules, one for *entity* (E) and one for *attribute* (A) matching, as well as potentially one spatial-language for *location* (L) matching, such that for $k = \{sub, obj\}$:

$$S_n^k = E(\mathbf{o}_n, \mathbf{q}_n^{ent}) \circ A(\mathbf{o}_n, \mathbf{q}_n^{att}) \circ L(\mathbf{o}_n, \mathbf{q}_n^{rel}) \quad (2)$$

This formulation combines benefits from both related modular approaches, on the one hand enabling training with *strong supervision* through pair-wise relation annotations for each scene (as discussed in [Hu et al. \(2016\)](#)), while at the same time incorporating relation cases of absolute location (e.g. “*the left bowl*”) that break the assumed triplet structure. Similar to [\(Yu et al., 2018\)](#), we treat visual attribute grounding as a standalone module, disentangled from semantic grounding (i.e. the category of the object). Another departure from the previous methods is the replacement of the soft language parser with a standalone hard parser, trained solely in synthetic data. We highlight that this is not equivalent to methods utilizing external parsers ([Andreas et al., 2016; 2015](#)), as even though the parser is not trained end-to-end with the rest of the architecture, it is yet optimized for the specific domain and its representations are used for training the three matching modules.

3.2 OBJECT-BASED REPRESENTATION

In our visual grounding setup, all appearing objects are first localized, and visual features for each ROI crop are then extracted. Object localization is most popularly achieved by integrating advanced methods for detection (e.g. Mask-RCNN ([He et al., 2017](#))) or clustering (e.g. UCN ([Xiang et al., 2020](#))). However, we exploit our robotic setup and localize objects directly in 3D space. In particular, we first remove the planar point cloud (table), cluster the remaining points ([Rusu & Cousins, 2011](#)), and finally calculate a local reference frame by applying PCA on the normalized covariance matrix of each cluster ([Kasaei et al., 2018](#)). The 3D coordinates for each object are translated to the RGB camera reference frame and the corresponding ROI, \mathbf{I}_n , is cropped from the input image using the translated 2D bounding box coordinates.

We treat visual attributes of appearing entities (such as color, material, etc.) as a means of disambiguating objects of the same category (e.g. “*red bowl*”, “*plastic cup*”) and attempt to ground them independently of the object categories ([Yao et al., 2016; Liu et al., 2017](#)). To obtain object-based representations \mathbf{o}_n for each ROI, we pass the image crop \mathbf{I}_n to two visual feature extractors, one associated with the entity F^E and one with the attribute F^A modules, such that:

$$\mathbf{o}_n = [\mathbf{z}_n^{ent}, \mathbf{z}_n^{att}, \mathbf{b}_n,] \quad (3)$$

$$\mathbf{z}_n^{ent} = F^E(\mathbf{I}_n; \Theta_E), \quad \mathbf{z}_n^{att} = F^A(\mathbf{I}_n; \Theta_A) \quad (4)$$

with $[,]$ denoting the concatenation operation and Θ_E, Θ_A represent trainable parameters of the two CNN encoders $F^E : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{D_v}$ and $F^A : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{D_a}$. For entities, we start from a pre-trained CNN checkpoint and keep until the penultimate layer of the network. For attributes, an earlier (second or third) layer is more appropriate, as lower-level layers provide more abstract features (e.g. color, shape etc.) that are more useful for attribute matching ([Yu et al., 2018](#)). Alternative representation methods can also be utilized (e.g. color histograms). We further inject all spatial information lost due to cropping ROIs by concatenating the visual representations with normalized bounding box features, $\mathbf{b}_n \in \mathbb{R}^5$, where:

$$\mathbf{b}_n = \left(\frac{x_n}{W} \quad \frac{y_n}{H} \quad \frac{x_n + w_n}{W} \quad \frac{y_n + h_n}{H} \quad \frac{w_n \cdot h_n}{W \cdot H} \right)^T \quad (5)$$

We adopt such object-based visual representations to aid their Sim-to-Real adaptation, as training in ROI crops, instead of the entire RGB frame, acts as a first step to reduce the domain gap between the source and target domains.

3.3 LANGUAGE PARSER

We parse the input query by mapping each word $\mathbf{w}_{1:T}$ in a phrase of length T to a tag $\mathbf{u}_{1:T}$, which serves as an indicator for the word’s role in the sentence (incl. entity, relation, location and attribute). We utilize Glove embeddings ([Pennington et al., 2014](#)) to embed the input words and a bi-directional GRU ([Chung et al., 2014](#)) of hidden size D_h to

contextualize embeddings within the phrase. The hidden state at each step $\tau = 1, \dots, T$ is given by the concatenation of the hidden states for the two directions, such that:

$$\mathbf{e}_\tau = \text{GloVeEmbedding}(\mathbf{w}_\tau) \quad (6)$$

$$\mathbf{h}_\tau = [\overleftarrow{\mathbf{h}}_\tau, \overrightarrow{\mathbf{h}}_\tau] \quad (7)$$

$$\overleftarrow{\mathbf{h}}_\tau = \text{GRU}(\mathbf{e}_\tau, \overleftarrow{\mathbf{h}}_{\tau-1}), \quad \overrightarrow{\mathbf{h}}_\tau = \text{GRU}(\mathbf{e}_\tau, \overrightarrow{\mathbf{h}}_{\tau-1}) \quad (8)$$

We extend our tag set with two additional concepts, a *mask* tag that tells our model to ignore the specific word (e.g. “*The*” in “*The red bowl*”) and an *abstraction* tag that is used for words that generalize the concept of an object (e.g. “*item*” in “*The item next to the bowl*”). The role tag is followed by an integer $0 - 2$ that indicates which attribute/location should be grouped with which entity phrases. The resulting tag vocabulary enumerates a total of $D_u = 3 \times 6 = 18$ tags. Unlike soft parsing methods, this allows the model to extend the structure recursively up to two-hop relation queries (e.g. “*The bowl behind the item that is left from the bottle*”). Overall, we tag each word by applying a linear projection $\Theta_u \in \mathbb{R}^{2 \cdot D_h \times D_u}$ on of the GRU output, followed by a softmax activation: $\mathbf{u}_\tau = \text{softmax}(\Theta_u^T \cdot \mathbf{h}_\tau)$.

We recover the desired triplet structure by composing all $\mu \in \{ent, att, loc\}$ tags per unique group index $k = 0, \dots, 2$ and replace the role tags with (averaged) word embeddings of the corresponding words / phrases:

$$\mathbf{q}_k^\mu = \text{avg}_\tau \{ \mathbf{e}_\tau \mid \mathbf{u}_\tau = [\text{TagVocab}(\mu), k] \} \quad (9)$$

$$Q_k = \{ \mathbf{q}_k^{ent}, \mathbf{q}_k^{att}, \mathbf{q}_k^{loc} \} \quad (10)$$

For standard single-hop relation queries $k \in \{0, 1\}$, the two identified groups will result in the $\{\mathbf{q}_k^{ent}, \mathbf{q}_k^{att}, \mathbf{q}_k^{loc}\}$ embeddings for the subject/object module evaluations appearing in equation 2. For a detailed example parse, see bottom-left of Fig. 2.

3.4 MATCHING MODULES

We implement a two-layer MLP to embed the object-based and the extracted query embeddings into the same space, where we compute a unitary matching score. Specifically, for unitary modules E, A , given the visual representation \mathbf{z}_n^μ and a word embedding \mathbf{q}^μ , the final matching score will be given by:

$$\mathbf{c}_n^\mu = (\Theta_1^\mu \cdot \mathbf{z}_n^\mu) \odot \mathbf{q}^\mu, \quad \mu = \{ent, att\} \quad (11)$$

$$E(\mathbf{q}^{ent}) = \Theta_2^{ent} \cdot \frac{\mathbf{c}_n^{ent}}{\|\mathbf{c}_n^{ent}\|_2}, \quad A(\mathbf{q}^{att}) = \Theta_2^{att} \cdot \frac{\mathbf{c}_n^{att}}{\|\mathbf{c}_n^{att}\|_2} \quad (12)$$

$$S_n^k = E(\mathbf{q}^{ent}) \odot A(\mathbf{q}^{att}) \quad (13)$$

where \odot denotes element-wise multiplication and $\Theta_1^{ent} \in \mathbb{R}^{D_v \times D_{jv}}, \Theta_2^{ent} \in \mathbb{R}^{D_{jv} \times 1}, \Theta_1^{att} \in \mathbb{R}^{D_a \times D_{jv}}, \Theta_2^{att} \in \mathbb{R}^{D_{jv} \times 1}$ trainable matrices of the the entity and attribute modules respectively. We formulate the location (L) module on a similar matching network B , which is used for spatial-language matching. Given two objects $n, m \in \{1, \dots, N\}$, we concatenate their location features $b_{nm} = [b_n, b_m]$ and use the same architecture as above for getting one score per object pair:

$$B(\mathbf{q}^{spt}) = \Theta_2 \cdot \frac{(\Theta_1 \cdot \mathbf{b}_{nm}) \odot \mathbf{q}^{spt}}{\|(\Theta_1 \cdot \mathbf{b}_{nm}) \odot \mathbf{q}^{spt}\|_2} \quad (14)$$

with $\Theta_1^{spt} \in \mathbb{R}^{10 \times D_{js}}, \Theta_2^{spt} \in \mathbb{R}^{D_{js} \times 1}$ trainable parameters of B . The resulting $N \times N$ matrix gives highest scores for objects that respect the spatial relation encapsulated in \mathbf{q}^{spt} . In order to locate the input object, the aggregated entity-attribute scores of equation 13 are given as input to the location module and the final output is given according to equation 2:

$$L(\mathbf{q}^{spt}) = \text{softmax} \left(\sum_m S_n^k \cdot S_m^k \cdot B(\mathbf{q}^{spt}) \right) \quad (15)$$

The spatial-language module B provides the pair-wise scores S_{nm}^{rel} that combined with the aggregated unitary scores for subject/object are used to express the training objective of equation 1. A running example of module composition for inference is shown on the bottom right of Fig. 2.

4 EXPERIMENTAL RESULTS

To assess the performance of the proposed approach, we perform three rounds of experiments, including: (i) evaluation of visual grounding performance in a synthetic dataset generated in simulation, (ii) evaluation of the adaptability of our method in real-world scenes after few-shot fine-tuning of the visual representations using two RGB-D datasets and (iii) evaluation of the applicability of our method in a HRI scenario for human-aided robotic grasping in both simulation and a real robot.

4.1 DATASETS

We focus our experiments on tabletop domains where objects are in arbitrary spatial arrangements. Objects are annotated both in category (e.g. bowl, soda can, etc.) as well as instance-level (red/white bowl, coca-cola/pepsi etc.) and the most dominant color of the object is used as the only attribute for visual disambiguation. For disambiguating through spatial relations, we include the following spatial concepts: { “left”, “right”, “further”, “closer”, “behind”, “front”, “bigger”, “smaller”, “next to”}. As in most natural settings, humans often refer to objects using various words other than the predefined category label (e.g.. “Bring me the coca-cola” is more natural than “Bring me the red soda can”), we also include annotations for appropriate items according to their brand, variety or flavour (e.g. “Coca-Cola” vs “Pepsi”, “Chex” vs “Crunch”, “strawberry juice” vs. “mango juice”).

4.1.1 SYNTHETIC SCENES

We collect a set of 32 object models from available resources and organize them into 8 object categories, with a total of 10 different colors. We render synthetic scenes in the Gazebo environment (Koenig & Howard, 2004) and develop tools for automated scene annotation, including class labels for categories and color, bounding boxes and segmentation masks for localization, as well as pair-wise spatial relations for all objects in the scene. Each scene contains 5 to 10 objects and all parsed scene information including object attributes and their relations are gathered in symbolic scene graph representations. We generate a total of 8k scenes for the training and 2k for the development set and multiple test, splits aiming to evaluate different components of the proposed method.

In order to generate natural language queries from the gathered scene graphs, we develop on top of the CLEVR (Johnson et al., 2016) data generation engine with the following adaptations: a) we treat object category as the main object identifier (instead of shape), b) we exclude material attributes and treat size as a spatial concept, c) we sometimes replace category with special instance-level words if applicable, d) we extend the vocabulary of spatial concepts to the ones aforementioned and e) we add absolute location annotations phrased as attributes. We use the $\{0, 1, 2\}$ -hop question templates after changing the text from the QA to the desired grounding format and implement constraints to control the ambiguity of the generated queries.

4.1.2 REAL SCENES

We evaluate the transfer performance of our model in real scenes from two RGB-D datasets, the publicly available Washington RGB-D scenes v1 dataset (Song et al., 2015b), and a manually collected dataset (Custom).

Washington Scene V1 We evaluate our approach on the Washington RGB-D scenes dataset (Song et al., 2015a), comprised of 8 different scene setups (three tables, three desks, a kitchen and a meeting room table). The dataset contains six object categories from a total of 19 different object instances, usually discriminated by visual cues (e.g. yellow vs. blue flashlight), although also containing object instances of the same category and color (e.g. two white bowls).

Custom The object catalogue is a subset of the objects included in Gazebo, but include novel attribute combinations (e.g purple soda can, orange Pringles etc.), containing a total of 17 object instances in 5 categories (soda can, Pringles, juice box, book and milk box). We place each object in the table in uniform positions and orientations and collect up to 20 samples per object instance, aimed as a fine-tuning resource for the visual representation backbone. We further



Figure 3: Synthetic recreation of a tabletop scene in Gazebo. The object catalogue (top), an example front view RGB (bottom-left) and depth (bottom-middle) pair, as well as the corresponding point cloud visualized in RViz (bottom-right).

record 206 test scenes containing 5 – 15 arbitrarily chosen objects placed randomly across the workspace and annotate them for location and visual attributes manually. We then follow the same process as in simulated data to construct annotated scene graphs as well as sample natural language queries.

4.2 VISUAL GROUNDING EVALUATION IN SIMULATION

In this section we explore the benefits of the proposed approach for visual grounding in the synthetic domain. We compare our method with two end-to-end baselines, namely: (i) **feature-fusion**, following the CNN-LSTM + MLP architecture of Rohrbach et al. (2015), and (ii) **soft-parser-based** modular networks, based on CMN Hu et al. (2016) and MattNet Yu et al. (2018). We highlight that even though modular, the second baseline’s networks require training of all modules in an end-to-end fashion, unlike our decoupled approach. For architectural and training details of our implementations for this experiment, see Appendix B.

In order to diagnose the effectiveness of adding explicit modules and hard parsing, we ablate the modules of each compared method and report results broken down by type of disambiguation used in the query. In particular, we include: (i) **Category (Cat)**, where there is no present ambiguity and the category of the object is used (e.g. “*bowl*”), (ii) **Color (Col)**, referring to the color of an object (e.g. “*blue bowl*”) (iii) **Location (Loc)**, where an absolute spatial relation between objects of the same category is used (e.g. “*rightmost bowl*”), (iv) **1-Hop**, where there is a single subject-relation-object triplet without further attributes/locations (e.g. “*bowl behind flashlight*”), as well as (v) **2-Hop**, where one triplet acts as the object phrase for another triplet and attributes/locations might be present within all phrases (e.g. “*blue bowl behind the flashlight that is closer than the red mug*”). We evaluate our models in the development set and select the best performing model using early stopping. We then report the performance of the selected models in various test splits, using top-1 accuracy of the predicted ROI index as the evaluation metric.

The results are summarized in Table 1. A first observation is that the feature-fusion baselines struggle to comprehend spatial relations, even when concatenating location features in the objects representations. This is due to lack of contextualization between locations of different objects. Second, the explicit incorporation of attribute features, either just concatenating in the holistic or as a separate module for the modular approaches, results in very high accuracy for color queries. This result positively reinforces using independent representations for visual category and attributes, as very commonly high-level pre-trained representations have “dropped” abstract features (such as color, shape) in favour of more category-specific ones (see 67.8% over 95.0 + % in column 2 of the table). A similar result is observed regarding absolute spatial relations, where using a separate location module for absolute relations resolves the grammatical discrepancy of relevant queries from the standard triplet structure. This leads to big accuracy gap in the locations split. Finally, we observe that even though end-to-end approaches successfully ground single-hop queries, they consistently cannot generalize to two-hop relations. We attribute this result to the fact that these approaches use a single relation concept assigned, resulting in both relation words present in the sentence aggregated and the entire second subject-relation-object triplet treated as single object. On the other hand, our decoupled formalism naturally generalizes to higher-order hops, as each hop is predicted uniquely in the word tagging step, resulting in a huge performance gap with the other baselines (96.3% vs. < 56.4%).

4.3 SIM-TO-REAL ADAPTATION EVALUATION

In this subsection we evaluate the potential of our method for few-shot adaptation to real scenes. The decoupled nature of our model allows us to battle the gap between sim and real solely on the visual domain, using a domain adaptation method for object recognition as a representation technique and directly transferring the rest of the architecture without any further training. We highlight that this is a unique benefit of our approach compared to soft-parser-based baselines,

Model	Cat	Col	Loc	1- Hop	2-Hop	Average
(E) + GRU + MLP	98.7	67.8	51.4	59.1	56.4	66.7
(E+A) + GRU + MLP	98.5	95.4	51.3	58.4	54.9	71.7
(E+A+L) + GRU + MLP	98.1	95.4	65.5	60.0	55.6	74.0
(E) + SoftParser	98.8	72.9	50.7	66.3	51.4	68.0
(E+A) + SoftParser	98.8	95.5	50.7	72.4	52.7	74.0
(E+A+L) + SoftParser	98.8	95.5	98.7	96.3	50.2	87.9
(E+A+L) + HardParser	98.4	95.7	98.1	96.9	96.3	97.0

Table 1: Visual grounding accuracy (%) evaluated for the test split of the synthetic data, organized by type of reference. Baselines include CNN-LSTM feature fusion (1-3), end-to-end modular using a soft language parser (4-6) and modular using hard language parsing (7). Compared to fusion-based, modular approaches solve relation queries by treating spatial relation comprehension separately. Without incorporation of explicit modules (i.e. for *attributes (A)* and *location (L)*), all baselines struggle to perform in corresponding splits. Disentangling language parsing from assembling modules for inference allows generalization to multiple relation steps entailed in the input query (*see 2-Hop*).

as due to their end-to-end nature they also need additional text data to fine-tune the soft-parser as part of the architecture (see Appendix C) for a discussion and data-efficiency comparisons).

In this round of experiments, we pre-train our CNN encoder paired with a classification head, in order to align the representations of objects in the two domains, using only a few real labeled images. We replace *MobilenetV2* features with the pre-trained domain-invariant representations, and then fine-tune the entity/attribute modules of our hard-parser-based model in synthetic data. We use the DDIRL (Tanwani, 2020) domain adaptation method and compare it with four baselines, namely: (i) using only synthetic, (ii) using only the few real data, (iii) training in both sim and real datasets, and (iv) the unsupervised method DANN (Ganin et al., 2016). Both mentioned techniques employ adversarial training to align the distributions between the two domains, with DDIRL exploiting a few labeled examples to further model the conditional (domain discrimination per class) together with the marginal (unsupervised domain discrimination) distribution. We refer the reader to the original paper for more theoretical insight on domain adversarial training. For implementation details and visualizations of the training progress, see Appendix C.

We report object recognition accuracy of the DDIRL classifier after pre-training, the accuracy of our entity and attribute modules after training using the adapted representations and the overall visual grounding accuracy for synthetic queries generated for the test splits of the two target datasets. The last experiment includes queries from all the ambiguity types of Table 1. Our results are presented in Table 2. We observe that DDIRL outperforms other baselines in both datasets and overall provides robust representations for extending to the visual grounding task. Performance for matching and grounding closely follows that of visual recognition in all baselines, suggesting that the only bottleneck in performance is the quality of the visual representations. Overall, the system demonstrates high data-efficiency for adaptation to natural images. Notably, the system achieves 81.3% accuracy on Washington v1 and 80.5% on our custom dataset, while only being exposed to real supervision from 20 labeled images per object category. Most importantly, no language data for the target datasets are required in order to transfer the grounding system. We believe that this result strongly suggests in favour of utilizing modeling and learning in synthetic domains for vision-and-language tasks, paired with domain adaptation techniques for visual data only.

4.4 INTEGRATION WITH A ROBOT FRAMEWORK FOR HUMAN-AIDED OBJECT PICKING

We integrated the proposed visual grounding pipeline into the RACE cognitive robotic framework (Kasaei et al., 2018) and evaluated the end-to-end system in a HRI scenario, where a non-expert human user instructs a robot to grasp an (ambiguous) object from a crowded scene. In particular, we use RACE as a world model that we call to obtain object locations and grasp proposals in real-time. The user makes a query and after an object ROI is proposed, the most suitable grasp pose is sampled and the actuator is moved to the grasp position using an inverse kinematics solver. The robot then attempts to pick the object and return back to default position.

We generate random scenes in simulation and evaluate the end-to-end behavior of the system. We report results separately for **simple** (i.e only color, only location, color+location or entity-only 1-hop) queries in images of 5 – 8 objects and **mixed** (1- and 2-hop relations including colors and locations in arbitrary parts of the phrase) queries in images of 9 – 12 objects. All queries are given by a human supervisor in real-time after inspecting the simulated scenes. To emulate potential discrepancies of our synthetic templates with natural language, we repeat the experiment for a smaller split where 3 human supervisors gave arbitrary queries without being aware of the template structure. The interpretable nature of our model’s inference allows us to diagnose the source of failure in case of errors, namely: a) **parsing** failure, where the input query is parsed incorrectly as a result of a mis-classification in the tagger network, b) **perception** failure, where the parse is correct but the matching modules gave incorrect scores, and

Table 2: Evaluation results for sim-to-real adaptation of visual representations for the test splits of both RGB-D datasets. Results are reported as top-1 predicted accuracy (%) evaluated using adapted embeddings, including (from left to right): recognition accuracy (*Rec*), average of entity-word (*Ent*) and attribute-word (*Att*) matching accuracy, as well as overall performance for visual grounding (*VG*). Final row indicates manually selecting the labeled subset of target images and extending it with random cropping - illumination augmentations.

Model	Washington v.1			Custom		
	Rec	Ent/Att	VG	Rec	Ent/Att	VG
Sim only	39.9	38.5	29.1	22.4	20.0	12.9
Real only	61.9	58.4	55.0	78.2	74.9	71.3
Sim+Real	58.6	56.2	52.1	74.8	72.3	68.8
DANN	73.4	72.7	66.9	66.8	65.3	60.2
DIRL	88.6	86.2	81.0	86.1	85.9	80.1
DIRL (+augm.+sel.)	90.8	88.0	81.3	87.1	85.7	80.5

Table 3: Evaluating the system for coupling visual grounding with robotic grasping in synthetic scenes.

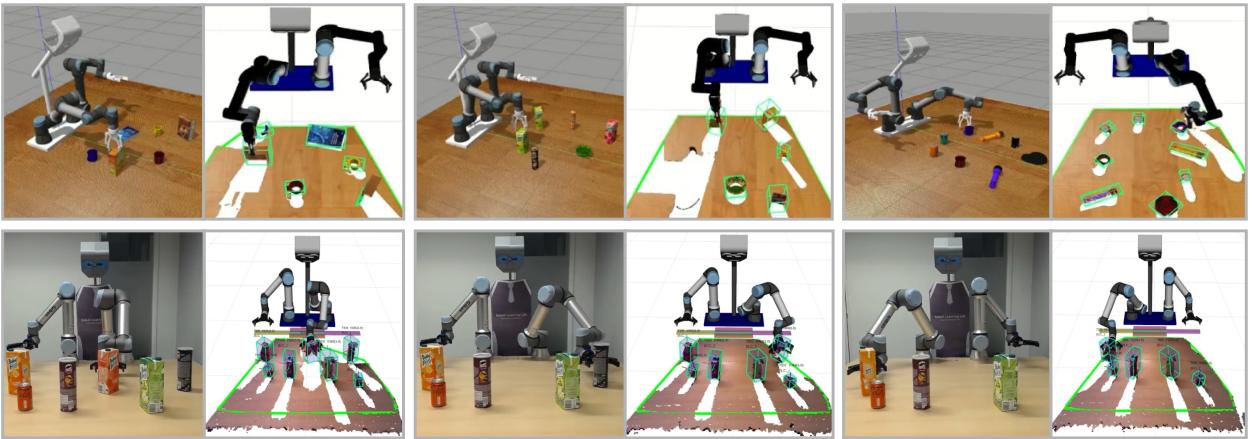


Figure 4: A sequence of snapshots capturing the setup of our robot framework in Gazebo (*top*) and in a real-world environment (*bottom*). We generate a random scene and command the robot to grasp a specific item with a text instruction, using a disambiguation query. For our examples the queries are, from *top-left* to *right-bottom*: “*cereal box behind the blue mug*”, “*juice box next to the rightmost Pringles*”, “*mug left from the soda can behind the flashlight*”, “*pink juice box*”, “*leftmost Pringles*”, “*juice box right from the Pringles*”. In the snapshots we demonstrate the robot during the picking action (*left*) and the localization results in RViz (*right*”).

c) **grasping failure**, where the system behaves correctly but the grasping behaviour fails. Results are summarized in Table 3.

We observe that in the template splits, the visual grounding system is robust to the grasping instructions across all trials, with the exception of a few perception failures. Language parsing as well as spatial relation grounding completed all trials without any errors. In the natural queries, we observe an increase in the parsing errors of the system, especially in the case of the mixed split. This is caused in almost all cases because of the use of an unknown spatial concept (i.e. “*middle*”, “*between*” etc), which leads in arbitrary spatial-language matching scores. Perception failures remain very low (< 5%) and as in the previous split are always entity mismatches due to partial views of objects caused by occlusion. The overall results showcase that the system can indeed be effectively utilized as a visual grounding utility for robotic grasping, with the main bottleneck being the grasping behaviour. Even in the case of human users not being aware of the robot’s vocabulary, results suggest that the parsing module is general enough (12.2% on average for both splits) to capture the majority of different variations of language humans used to relate objects in the tabletop setups of the experiment.

Furthermore, we perform similar end-to-end grasping experiments with a real robot. Our setup is shown in Fig. 4. For visual representation, we use the CNN encoder that we adapted for our Custom dataset. We place objects randomly within the workspace of the robot and query for different instances based on color, location and spatial relations between them. A video demonstrating the robot behaviour has been attached to the paper as supplementary material.

5 CONCLUSION

In this work, we present a novel modular deep learning framework for visual grounding in RGB-D domains. We focus on an ambiguity resolution HRI scenario, where the input query can always be decomposed in terms of grounding entities, attributes, and spatial relations between different objects. We develop a synthetic domain and tools for automatic annotation, which we use to train our modules independently. In particular, the controlled setup of simulation is used to generate elaborate scene graph representations, which are used to generate appropriate query expressions. For inference, we combine the modules by exploiting the grammatical priors of relations in natural language. Experimental results carried in simulation show that the proposed method achieves high accuracy in all versions of the ambiguity resolution task. Additionally, we show that, compared to other modular baselines, our method is robust to input query complexity, while the decoupled formulation of the modules enables high interpretability at test time. We also assess the sim-to-real adaptation of the proposed approach on two real RGB-D scene datasets. Results showed that our method allows for data-efficient adaptation to real-scenes using only few-shot visual supervision. We finally integrate the implemented model with a robot framework and demonstrate its utility for interactive object picking.

One limitation of the proposed system is potential discrepancies between our synthetic language templates and natural queries given arbitrarily by humans. In a relatively specialized domain (e.g., tabletop scenes of objects), this discrepancy can be alleviated by composing suitable modules based on language structure. However, the more one wants to cover generic content, the more refining the system architecture needs, which grants the work of modeling the domain and identifying useful linguistic priors a more cumbersome task, in which case end-to-end learning is favored. In the future, we would like to investigate the possibility of incremental acquisition of entity, attribute, and spatial concepts, e.g. by using a replay mechanism to update visual representations while tackling catastrophic forgetting. Additional directions ahead include extending the range of vision-and-language tasks tackled by the proposed framework, including VQA and visual reasoning.

REFERENCES

- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Deep compositional question answering with neural module networks. *CoRR*, abs/1511.02799, 2015. URL <http://arxiv.org/abs/1511.02799>.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *CoRR*, abs/1601.01705, 2016. URL <http://arxiv.org/abs/1601.01705>.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: learning universal image-text representations. *CoRR*, abs/1909.11740, 2019. URL <http://arxiv.org/abs/1909.11740>.
- Junyoung Chung, Çaglar Gülcöhre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL <http://arxiv.org/abs/1412.3555>.
- Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. *CoRR*, abs/1805.10547, 2018. URL <http://arxiv.org/abs/1805.10547>.
- Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d pose estimation: motion to the rescue. *CoRR*, abs/1907.02499, 2019. URL <http://arxiv.org/abs/1907.02499>.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL <https://arxiv.org/abs/2010.11929>.
- Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Visual grounding with transformers. *CoRR*, abs/2105.04281, 2021. URL <https://arxiv.org/abs/2105.04281>.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL <http://jmlr.org/papers/v17/15-239.html>.
- Jun Hatori, Yuta Kikuchi, Sosuke Kobayashi, Kuniyuki Takahashi, Yuta Tsuboi, Yuya Unno, Wilson Ko, and Jethro Tan. Interactively picking real-world objects with unconstrained spoken language instructions. *CoRR*, abs/1710.06280, 2017. URL <http://arxiv.org/abs/1710.06280>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. *CoRR*, abs/1611.09978, 2016. URL <http://arxiv.org/abs/1611.09978>.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016. URL <http://arxiv.org/abs/1612.06890>.

- Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.
- S Hamidreza Kasaei, Miguel Oliveira, Gi Hyun Lim, Luís Seabra Lopes, and Ana Maria Tomé. Towards lifelong assistive robotics: A tight coupling between object perception and manipulation. *Neurocomputing*, 291:151–166, 2018.
- Manuel Kaspar, Juan David Muñoz Osorio, and Jürgen Bock. Sim2real transfer for reinforcement learning without dynamics randomization. *CoRR*, abs/2002.11635, 2020. URL <https://arxiv.org/abs/2002.11635>.
- Nathan P. Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 3:2149–2154 vol.3, 2004.
- Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *CoRR*, abs/2107.07651, 2021. URL <https://arxiv.org/abs/2107.07651>.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019. URL <http://arxiv.org/abs/1908.03557>.
- Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. A competence-aware curriculum for visual concepts learning via question answering. *CoRR*, abs/2007.01499, 2020. URL <https://arxiv.org/abs/2007.01499>.
- Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Explainability by parsing: Neural module tree networks for natural language visual grounding. *CoRR*, abs/1812.03299, 2018. URL <http://arxiv.org/abs/1812.03299>.
- Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4866–4874, 2017.
- Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. URL <http://arxiv.org/abs/1711.05101>.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265, 2019a. URL <http://arxiv.org/abs/1908.02265>.
- Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. *CoRR*, abs/1912.02315, 2019b. URL <http://arxiv.org/abs/1912.02315>.
- Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. *CoRR*, abs/1701.03439, 2017. URL <http://arxiv.org/abs/1701.03439>.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *CoRR*, abs/1904.12584, 2019. URL <http://arxiv.org/abs/1904.12584>.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. *CoRR*, abs/1511.02283, 2015. URL <http://arxiv.org/abs/1511.02283>.
- Michał Nazarczuk and Krystian Mikolajczyk. Shop-vrb: A visual reasoning benchmark for object perception. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6898–6904, 2020.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL <https://arxiv.org/abs/2103.00020>.
- Nithin Raghavan, Punarjay Chakravarty, and Shubham Shrivastava. Sim2real for self-supervised monocular depth and segmentation. *CoRR*, abs/2012.00238, 2020. URL <https://arxiv.org/abs/2012.00238>.
- Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. *CoRR*, abs/1511.03745, 2015. URL <http://arxiv.org/abs/1511.03745>.
- Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pp. 1–4. IEEE, 2011.
- Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4694–4703, 2019.
- Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. URL <http://arxiv.org/abs/1801.04381>.
- Shurhan Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015a. doi: 10.1109/CVPR.2015.7298655.
- Shurhan Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015b. doi: 10.1109/CVPR.2015.7298655.
- Sanjay Subramanian, Will Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension, 2022.
- Ajay Kumar Tanwani. DIRL: domain-invariant representation learning for sim-to-real transfer. *CoRR*, abs/2011.07589, 2020. URL <https://arxiv.org/abs/2011.07589>.
- Joanne Truong, Sonia Chernova, and Dhruv Batra. Bi-directional domain adaptation for sim2real transfer of embodied navigation agents. *CoRR*, abs/2011.12421, 2020. URL <https://arxiv.org/abs/2011.12421>.
- Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. *CoRR*, abs/1812.04794, 2018. URL <http://arxiv.org/abs/1812.04794>.
- Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning RGB-D feature embeddings for unseen object instance segmentation. In *Conference on Robot Learning (CoRL)*, 2020.
- Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. *CoRR*, abs/1909.08164, 2019a. URL <http://arxiv.org/abs/1909.08164>.
- Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019b.
- Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. *CoRR*, abs/1908.06354, 2019c. URL <http://arxiv.org/abs/1908.06354>.
- Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. *CoRR*, abs/1611.01646, 2016. URL <http://arxiv.org/abs/1611.01646>.
- Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. *CoRR*, abs/1810.02338, 2018. URL <http://arxiv.org/abs/1810.02338>.

- Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. *CoRR*, abs/1608.00272, 2016. URL <http://arxiv.org/abs/1608.00272>.
- Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. Mattnet: Modular attention network for referring expression comprehension. *CoRR*, abs/1801.08186, 2018. URL <http://arxiv.org/abs/1801.08186>.
- Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *CoRR*, abs/2009.13303, 2020. URL <https://arxiv.org/abs/2009.13303>.
- Fengda Zhu, Linchao Zhu, and Yi Yang. Sim-real joint reinforcement transfer for 3d indoor navigation. *CoRR*, abs/1904.03895, 2019. URL <http://arxiv.org/abs/1904.03895>.

A VISUALIZATION EXAMPLES OF MODEL INFERENCE

An important benefit of the implemented framework is that it enables full interpretability of the underlying reasoning steps taken to perform a grounding. The compositional nature of the modules during test time allows us to inspect the output of each module and diagnose the source of error in case of failure. In Fig. 5 we demonstrate the predicted tags, the corresponding module composition and intermediate module outputs for grounding examples of all types of ambiguity resolution considered in this work. In Fig. 6, we illustrate how one could add exceptions in the outputs of the modules to capture specific failures and query them back to the human instructor. This allows extending the system to a dialogue setting, potentially enabling human users to incrementally teach the agent novel object-word pairs.

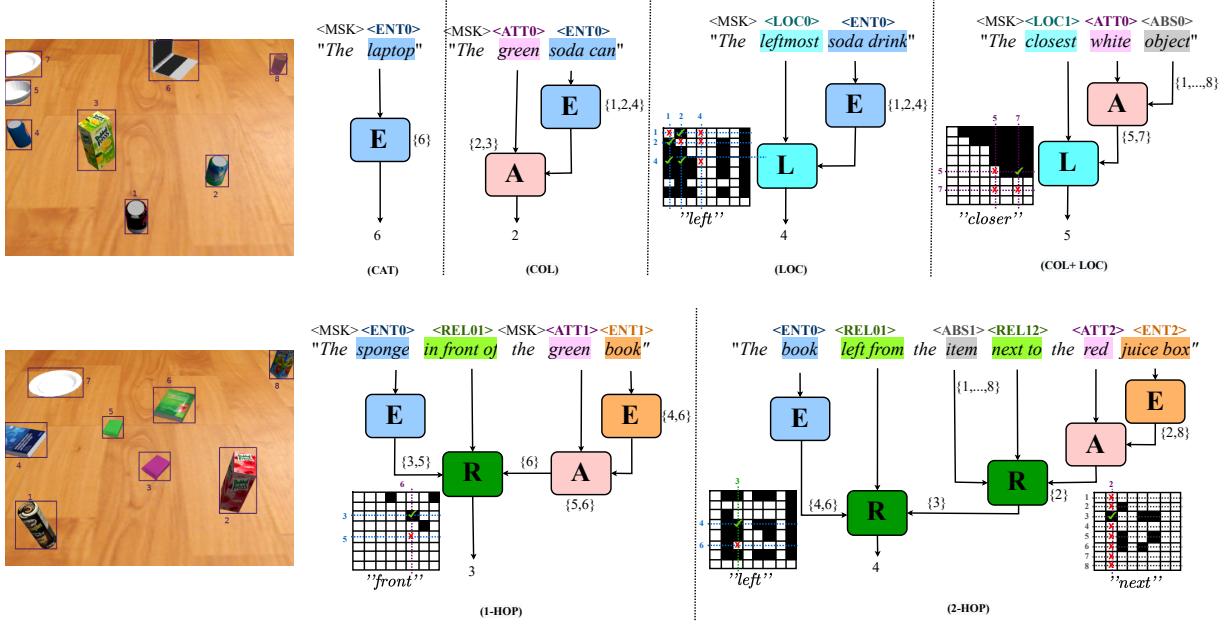


Figure 5: Output tags (illustrated on top of the input query) and corresponding module composition for inference. Examples include all different types of ambiguity resolution, from top-left to bottom-right: category (no ambiguity), color, location, color + location, mixed (1-hop) and mixed (2-hop)

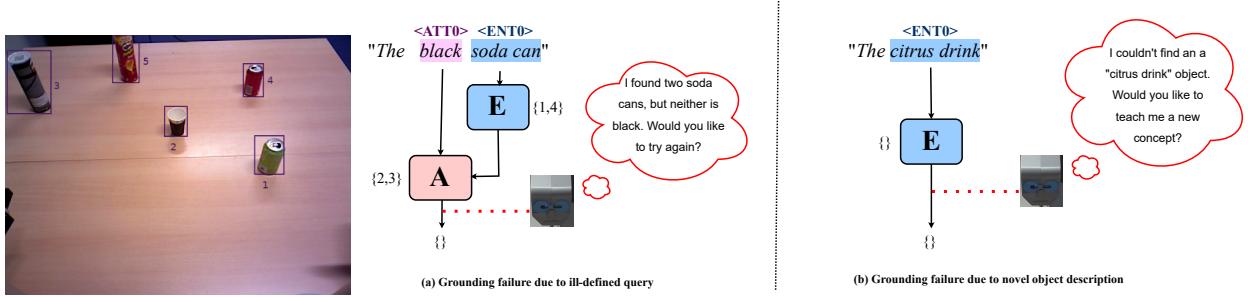


Figure 6: Illustration of how adding exception traces in the outputs of each module, combined with suitable responses, could extend the present framework to a dialogue setup, enabling users e.g. to repeat in case of ill-defined instructions (left) or even teach the robot a new concept (right).

B VISUAL GROUNDING EVALUATION IN SIMULATION EXPERIMENT

For entity recognition (category), we use a pre-trained *MobileNet v2* network (Sandler et al., 2018) and fine-tune it on segmented crops from the synthetic scenes. We isolate the feature extraction layers and use them as the visual

representation backbone for this round of experiments. For attribute representation , since we only experiment with color, we encode each input crop with a $\mathbb{R}^{3 \times 256}$ color histogram vector, concatenated per channel. We have found experimentally that this representation is robust for both synthetic and real data.

In the feature-fusion baselines the input query is encoded into a single vector with a GRU encoder and concatenated with each entity (E) representation before passed to an MLP for ranking. We use pre-trained Glove (Pennington et al., 2014) embeddings for word representation and a single bi-directional GRU layer with hidden size 256. The MLP has two hidden layers of sizes 1024, and 128, respectively and rectified linear unit (ReLU) activation is used in both layers. We repeat our experiments but further concatenate each visual-query representation with attribute (A) and bounding box location (L) features.

Additionally, we include modular approaches, based on CMN (Hu et al., 2016), where we keep the same visual representation but now use a soft language parser to decompose the input query into separate modules for subject-relation-object matching, potentially adding attribute and location modules explicitly, as in (Yu et al., 2018). We implement the parser as the same GRU network paired with three attention layers for each module and use hidden sizes of 256 for the unitary and 50 for the pair-wise spatial relation matching networks. We train the model end-to-end in the weakly supervised setting, using the ground truth ROI index with a softmax loss over the final predictions per object.

For our hard parser-based model, we replace the soft parser with a neural module for word tagging and exploit the assumed triplet structure to compose modules in test time based on the predicted tags. First, we train the word tagger using a batch size of 64 with a softmax loss over the vocabulary of tags we describe in Section 3.3 (as e.g. in part-of-speech tagging). As we train using synthetic data generated through templates, the ground truth tag annotations are taken from the template that generates the query. To train entity/attribute modules, we exploit the CLEVR-like functional program annotations of our generated dataset that provide us with execution traces for resolving each query. Such traces include annotations of which object indices and words are inputs/outputs to all modules separately. We extract and express these input/output indices as binary vectors (over the set of objects) and treat each object as a Bernoulli variable. We train the modules with a Binary Cross Entropy (per-logit) loss over the accumulated datasets for entity and attribute input/output pairs separately. For the spatial relation module B , since we have pair-wise scores for object pairs, we first flatten the pair-wise matrix to 1D and train with a BCE (per-logit) loss, using the flattened pair-wise matrix of the queried relation (again retrieved from the execution traces of our synthetic dataset) as supervision. For the spatial module we use a batch size of 8. All models are implemented in PyTorch (Paszke et al., 2019) and trained using the AdamW optimizer (Loshchilov & Hutter, 2017), with a batch size of 256 unless stated otherwise, and an original learning rate of 5×10^{-4} and a linear decay scheduler.

We highlight that in the original CMN network the authors mentioned that their model is able to be trained using *strong* supervision, meaning pair-wise annotations between the objects in the scene for the specific relation mentioned in the query. This ground truth signal is the same we use to train the spatial-language module B , as described above. This is in fact a more relaxed notion of *strong* supervision as meant in the context of this work, as in CMN all modules and the soft parser are trained jointly by this signal. Instead, in our work we train each module independently. When adding explicit attribute and location modules, the soft-parser baseline moves to the MattNet Yu et al. (2018) formalism, where all modules compute unitary scores and there is no pair-wise contextualization in any part of the network. That is because the CMN work only studies simple relation queries (without attributes/locations) and treats the subject / object as separate entities to be grounded. On the contrary, MattNet joins the relation and the object modules into a single relation module, removing the pair-wise computation between subject and object. As a result, the only variation of this baseline that can be trained in the strong supervision setting (in the CMN notion - meaning pair-wise annotations) is the (E) + *SoftParser*. We confirm experimentally that using the same dataset size the “strong” version of this baseline achieves marginally same results as the weakly trained one, although converging must faster.

C SIM-TO-REAL VISUAL DOMAIN ADAPTATION EXPERIMENT

For our experiments, we adapt the CNN encoder architecture of the digit experiments from the original DIRL paper, after we add an additional convolutional-pooling-block in the input and fix image resolution to 75×75 . We use 20 labeled examples from the real domain and train using AdamW with a learning rate of 10^{-3} . We split the batch size with 50% sim and 50% real data, while randomly shuffling the few labeled examples within the 25% of the real part. We augment the labeled batch with random cropping and illumination distortions. For the rest 75%, we sample unsupervised images with replacement. We set the weights for all the constituent losses to 1 and use a margin of 0.6 for the triplet KL loss. We skip the pseudo-labeling process during training, as it experimentally did not provide

improvements. Fig. 7 presents the incremental progression of the learned representations towards domain-invariant representations for both real datasets.

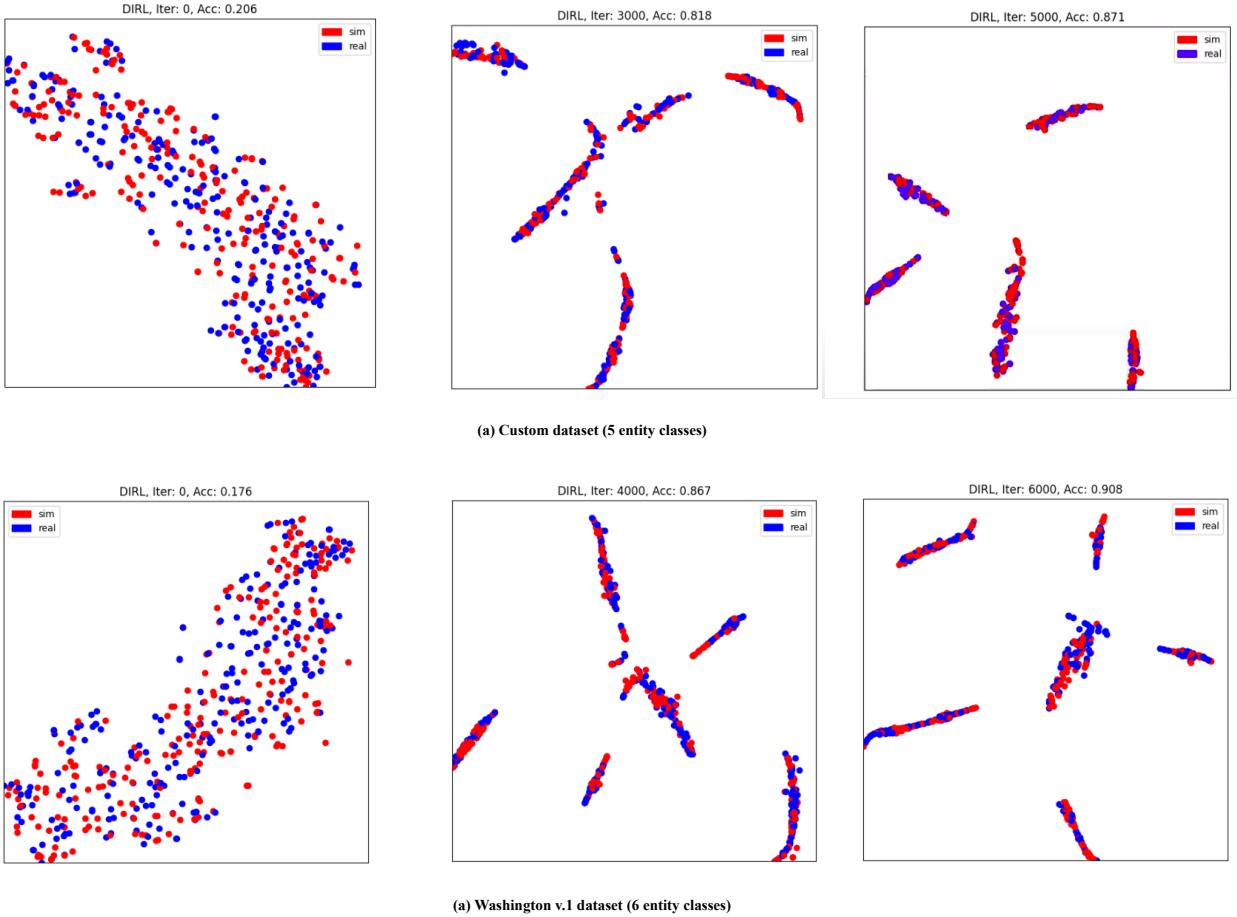


Figure 7: t-SNE projections of the embedding spaces learned by our CNN encoder during adversarial pre-training for Sim-To-Real adaptation. We visualize results for our Custom dataset (*top*) and the Washington v.1 dataset (*bottom*) during the start of training (“*left*”), after 3-4 thousand iterations respectively (“*middle*”) and after reaching maximum classification accuracy in the test split (“*right*”). We randomly sample 500 images from each domain to generate the t-SNE projections.

Additionally, we perform a comparative study of visual grounding performance of our model versus the soft-parser-based baseline, for different dataset sizes. As we have already highlighted in Section 4.3, the soft-parser baseline requires fine-tuning of all modules in an end-to-end fashion, including the language parser. In absence of real natural language annotations for the scenes in the target domains, one could use the synthetic queries and train the model from scratch after only adapting the visual representations. We perform this experiment for different volumes of image-query samples and compare with our hard-parser-based model. We note that as in the original experiment, our model here is not trained end-to-end, but the parser and the matching modules are trained independently and combined only for evaluation in the real datasets. In order to provide fair comparisons, we exclude from the test splits of the target datasets all queries that the soft-parser baseline cannot handle (mixed 1-hop, 2-hop). Fig. 8 illustrates the top-1 accuracy of both datasets for different dataset sizes.

We observe that given enough data, the soft-parser is indeed capable or reaching parallel performance to our model (in the excluded split). However, our model is a lot more data-efficient, as we observe from our results. Notably, we validate that using only $20k$ image-query pairs (25% of the training data) our word tagger network achieves 99.7% token-level and 95.1% sentence-level accuracy when evaluated using the ground truth tags of the synthetic dataset. This is a benefit from utilizing explicit supervision for training the language module, versus training it end-to-end with the rest of the network. Even though additional annotations are generally considered a bottleneck, we highlight that

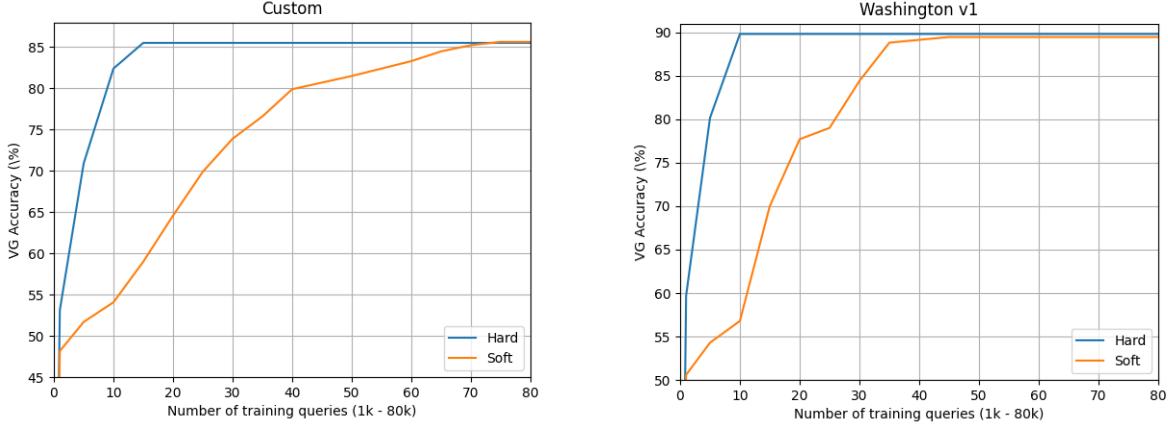


Figure 8: Top-1 predicted accuracy vs. number of synthetic image-query samples used for fine-tuning both models after adapting the visual representations. Results are presented for our Custom (*left*) and the Washington-v.1 (*right*) dataset. The test splits include only "simple" cases of queries (category, location, attribute and entity-only relation). Even though both models eventually reach the same performance, our hard-parser-based model is much more data efficient, due to decoupling training for language parsing and the matching modules

since we train with language data *only* in simulation, this limitation is alleviated by the fact that in with the simulator we can extract such rich annotations "for-free", in large scale and in a controlled fashion.

D COMPARISONS WITH ZERO-SHOT LARGE-SCALE TRANSFORMERS

A well established methodology in modern machine learning focuses around using large-scale representation learners, usually transformers, most popularly pre-trained in a self-supervised manner in massive datasets (i.e. *Foundation models*). The model is then fine-tuned for a downstream task or applied in a zero-shot setting. This methodology is also popular in vision-and-language learning, with an array of different transformer architectures and pre-training objectives (Lu et al. (2019a); Li et al. (2019); Chen et al. (2019); Lu et al. (2019b)). Compared to modular approaches such as ours, that are very closely tight to the application domain, such pre-trained methods have shown to provide tremendous generalization capabilities in arbitrary downstream tasks. This benefit is due to the representational strength of the pre-trained embeddings, as the model is exposed to massive amounts of data. In order to explore trade-offs between such methods and the one we propose in this work, we evaluate two zero-shot transformer approaches in our synthetic dataset after adapting them for the visual grounding task.

We experiment with two models, CLIP (Radford et al., 2021) and ALBEF (Li et al., 2021). CLIP employs one image-only and one text-only encoder and uses a contrastive loss on top of them to learn representations that are closer in the embedding space for matched image-text pairs. Similarly, ALBEF further employs a multi-modal encoder and pre-trains with three losses, a) an image-text contrastive loss (ITC) such as CLIPs, b) an image-text matching loss (ITM) in the output of the multi-modal encoder, and c) a masked language modeling loss for the text encoder.

In order to evaluate the models for zero-shot visual grounding, we replicate the *text-pair* experiment introduced in the ReCLIP (Subramanian et al., 2022) paper for the CLEVR dataset. In this setup, a false version of the input query is generated, both for spatial (e.g. positive: "*bowl behind flashlight*", negative: "*bowl in front of flashlight*") and for visual only queries (e.g. positive "*a blue bowl and black flashlight*", negative: "*a red bowl and a black flashlight*"). Both queries are passed to the model and we measure accuracy as the percentage of queries that the positive version scores higher than the negative. For CLIP, we experiment with different ResNet-50 (He et al., 2015) and Visual Transformer (ViT) (Dosovitskiy et al., 2020) encoders. Following the original paper, we experiment with ALBEF using both the ITC and the ITM scores for computing similarity scores for image-text pairs. We present results separated in two splits, namely: a) **Visual**, where the query contains only object entity and color and b) **Spatial**, where the query is a 1-hop entity-only relation. Results are

Model	Visual	Spatial
<i>CLIP-RN50x4</i>	90.3	50.8
<i>CLIP-RN50x16</i>	95.1	52.8
<i>CLIP-RN50x64</i>	92.6	49.8
<i>CLIP-ViT-B/32</i>	88.0	47.5
<i>CLIP-ViT-B/16</i>	91.4	48.7
<i>ALBEF-ITM</i>	89.8	48.4
<i>ALBEF-ITC</i>	73.3	48.4
<i>ours</i>	95.6	96.9

Table 4: Accuracy (%) on our synthetic dataset for the image-text matching task.

summarized in Table 4. It can be seen that both CLIP and ALBEF perform well in the visual version of the task but poorly in relation queries. The evaluation split contains 895 scene instances.

Our experiment reproduces the findings of the ReCLIP paper, namely that even though pre-trained models perform well on the visual task, they cannot comprehend spatial relations. The authors alleviate this by extending the model to also include a ROI proposal generator and a rule-based relation extractor. With this formulation the grounding model becomes very similar to ours, with the exception of using a hand-crafted relation module and CLIP/ALBEF for computing visual matching scores, instead of a module trained in a supervised fashion in the domain. With the insight of this result, combined with additional benefits of our approach, namely: a) computational efficiency, as our method has less than 0.2% the parameter count of CLIP and b) interpretability, due to full decoupling of modules in our approach, grants our model favourable for application. We highlight however that this is expected, as our model is designed with domain knowledge about the task (grammatical structure of relations in natural language) injected in its architecture. Foundation models still offer a strong baseline, especially considering their zero-shot capabilities, although still lacking when compared with domain-specific architectures for a given task.