Naturally Supervised 3D Visual Grounding with Language-Regularized Concept Learners

Chun Feng* Stanford University
Joy Hsu* Stanford University
Weiyu Liu Stanford University
Jiajun Wu Stanford University

Abstract

*3D visual grounding is a challenging task that often requires direct and dense supervision, notably the semantic label for each object in the scene. In this paper, we instead study the naturally supervised setting that learns from only 3D scene and QA pairs, where prior works underperform. We propose the Language-Regularized Concept Learner (LARC), which uses constraints from language as regularization to significantly improve the accuracy of neuro-symbolic concept learners in the naturally supervised setting. Our approach is based on two core insights: the first is that language constraints (*e.g.*, a word's relation to another) can serve as effective regularization for structured representations in neuro-symbolic models; the second is that we can query large language models to distill such constraints from language properties. We show that LARC improves performance of prior works in naturally supervised 3D visual grounding, and demonstrates a wide range of 3D visual reasoning capabilities—from zero-shot composition, to data efficiency and transferability. Our method represents a promising step towards regularizing structured visual reasoning frameworks with language-based priors, for learning in settings without dense supervision.*

1. Introduction

3D visual reasoning models often require direct supervision during training to achieve faithful 3D visual grounding, for example, in the form of classification labels for each ground truth object bounding box in the scene. However, dense visual annotation for 3D scenes is difficult and expensive to acquire. In this paper, we study the more practical setting of *naturally supervised* 3D visual grounding, where models learn by looking at only scene and question-answer pairs, and do not use object-level classification supervision. Prior state-of-the-art works for 3D referring expression comprehension [5, 23, 26, 29, 31, 43] do not show strong visual reasoning capabilities, such as generalization and data efficiency, in this indirectly supervised setup (See Figure 1).

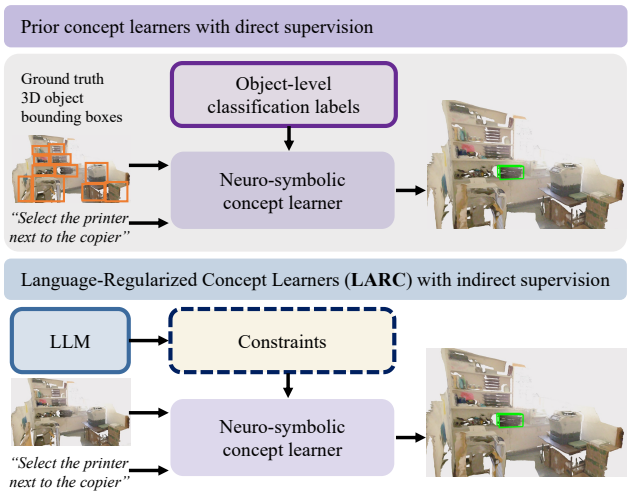

Figure 1. Compared to prior works, LARC conducts 3D visual grounding in the naturally supervised setting, by training neuro-symbolic concept learners with language regularization.

To this end, we propose a neuro-symbolic concept learner that leverages constraints from language (*e.g.*, a word's relation to another), as regularization in low guidance settings. Compared to visual annotations, language-based priors are cheap to annotate, and free when distilled from large language models (LLMs). We show that especially in settings with indirect supervision, models benefit from such regularization as it reduces overfitting on noisy signals. Notably, we leverage the structured and interpretable representations from neuro-symbolic methods to effectively inject constraint-based regularization.

Neuro-symbolic concept learners decompose visual reasoning queries into modular functions, and execute them with neural networks. Each network outputs representations that can be indexed by its concept name (*e.g.*, *chair*) and arity (*e.g.*, that the concept describes a *binary* relation) [36]. Due to this modularity, prior neuro-symbolic works have shown strong generalization, data efficiency, and transferability in the 3D domain [26]. However, they require direct

*Equal contribution.

classification supervision to train intermediate networks for concept grounding, and do not take into account the taxonomy of the concept name in relation to other concepts in language, nor its effect on other concepts. For example, previous methods do not capture from language that *wardrobe* and *dresser* are visually similar and synonymous, or that a vase *left* of a painting indicates that the painting is *right* of the vase. Although recent neuro-symbolic works have used LLMs for the interpretation of language to function structures [25, 26, 49], they purely translate the query and do not encode language priors in execution.

To enable learning in naturally supervised settings, we propose a **La**nguage-**R**egularized **C**oncept Learner (**LARC**). LARC builds upon the 3D neuro-symbolic concept learner, NS3D [26], and introduces regularization on intermediate representations based on language constraints. Our method leverages general constraints derived from well-studied semantic relations between words, for example, symmetry [21], exclusivity [38], and synonymity [40], which are broadly applicable across all language-driven tasks. We show that we can effectively encode such semantic contexts from language into neuro-symbolic concept learners through regularization. Additionally, we demonstrate that concepts satisfying these constraints can be distilled from LLMs based on language priors (*e.g.*, the concepts *near* and *far* are symmetric), and that language rules also enable execution of novel concepts from composition of learned concepts.

LARC significantly improves performance of neuro-symbolic concept learners in two datasets for 3D referring expression comprehension. In addition, we demonstrate that using language-based rules allows LARC to zero-shot execute unseen concepts, while all previous models fail to generalize. Importantly, LARC significantly improves data efficiency and transferability between datasets compared to prior works, critical for 3D visual reasoning systems, while not requiring object-level classification labels.

In summary, our key contributions are the following:

- We propose language-based constraints that capture the semantic properties of concepts, and show that such priors can be distilled from large language models.
- We introduce constraint-based regularization that can be effectively applied on the structured intermediate representations of neuro-symbolic concept learners.
- We show that we can leverage language rules to compose unseen concepts from learned concepts.
- We empirically validate LARC's improvement upon prior works on two datasets, and show strong generalization, data efficiency, and transferability between datasets.

2. Related Works

3D grounding systems. Priors works have proposed methods for 3D grounding, leveraging context from the full 3D scene [28, 31, 54, 55]. Many of such methods take on an end-to-end approach, and jointly attend over language and 3D point clouds [7–10, 35]. More specifically, several works leverage the Transformer [47] architecture to solve the 3D referring expression comprehension task [1, 23, 29, 43, 52]. Notably, TransRefer3D [23] uses a Transformer-based network to extract entity-and-relation-aware representations, and LanguageRefer [43] employs a Transformer architecture over bounding box embeddings and language embedding from DistilBert [44]. The Multi-View Transformer [29] projects the 3D scene to a multi-view space to learn robust representations. Other methods have used 2D information to augment grounding in 3D [52], including LAR [5], which uses 2D signals generated from 3D point clouds to assist its 3D encoder. Recently, NS3D [26] proposed a modular, neuro-symbolic approach to 3D grounding, which shows strong data efficiency and generalization compared to end-to-end works, but requires dense supervision.

All of the prior works for 3D grounding leverage classification labels in the training stage, many in the form of direct classification losses [2, 5, 23, 26, 29]. BUTD-DETR [31] and SAT [52] use outputs from pre-trained detection networks, including predicted bounding boxes and their class labels, as additional inputs. LanguageRefer [43] applies a semantic classifier to obtain the class label of each object, which are tokenized and used as input. Zhao et al. [55] use pre-trained segmentation networks to filter candidates based on predicted categories from segmentation networks. While LARC follows NS3D's neuro-symbolic concept learning paradigm, LARC works in the naturally supervised setting, and is neither trained with classification labels nor uses classification predictions from other pre-trained networks.

Neuro-symbolic concept learners. Neuro-symbolic methods have shown strong visual reasoning performance and a wide range of capabilities. They parse visual reasoning queries into programs, and execute such programs with modular networks on a variety of visual domains [4, 11, 18, 22, 25, 26, 30, 32, 33, 37]. Neuro-Symbolic VQA [53] first proposed program execution for 2D visual question answering, and the Neuro-Symbolic Concept Learner [36] improved the training paradigm by removing direct supervision on scene representations and program traces in the 2D domain. A recent work, Logic-Enhanced Foundation Model [25], reduced prior knowledge required by replacing domain-specific languages with domain-independent logic. LARC further lowers the guidance required for 3D concept learners by regularizing intermediate representations, enabling strong performance in naturally supervised settings with only 3D scene and QA pairs as supervision.

Constraints in neural networks. Many works have proposed strategies for models to effectively encode knowledge-based rules, for the purposes of improving interpretability, sample efficiency, and compliance with safety con-

straints [13, 14, 20, 48]. Von Rueden et al. [48] describes existing methods which range in sources of constraints (*e.g.*, expert knowledge), representations (*e.g.*, graphs), and methods for integration (*e.g.*, feature engineering). Most related to our work are methods that incorporate discrete rules into models' learning processes. Several works use rules as bases for model structures [19, 34, 39, 46], while others learn latent embeddings of symbolic knowledge that can be naturally handled by neural networks [3, 50]. Logical rules have also been transformed into continuous and differentiable constraints, for example via t-norm [17], to serve as additional loss terms in training [16, 24, 27, 45, 51]. Different from existing approaches, LARC takes advantage of neuro-symbolic concept learners with structured representations, and utilizes language-based rules to regularize learned features.

3. Methods

To describe LARC, we first provide preliminary background on our task and method in Section 3.1. Then, we detail how LARC distills constraints from LLMs in Section 3.2, and applies constraints on structured neuro-symbolic representations in Section 3.3. After, we present our method's training details in Section 3.4. Finally, we discuss LARC's language composition during inference in Section 3.5.

3.1. Preliminaries

Neuro-symbolic concept learners [18, 26, 36] are methods that decompose input language queries into symbolic programs, and differentiably execute the programs with concept grounding neural networks on the input visual modality. These approaches have shown strong 3D visual grounding capabilities, but often require dense supervision in complex domains, *e.g.*, ground truth object bounding boxes and classification labels for objects in order to train intermediate programs. In this paper, we build LARC from NS3D [26], a neuro-symbolic concept learner that conducts grounding in the 3D domain, in which such dense supervision is expensive and difficult to annotate. NS3D was proposed to tackle the task of 3D referring expression comprehension (3D-REC).

Problem statement. In 3D-REC, we are given a scene $\mathcal{S}$, which is represented as an RGB-colored point cloud of C points $\mathcal{S} \in \mathbb{R}^{C\times 6}$, and an utterance $\mathcal{U}$ describing a target object and its relationship with other objects in $\mathcal{S}$. The goal is to localize the target object $\mathcal{T}$. Notably, there exists distractor objects of the same class category as $\mathcal{T}$, such that understanding the full referring expression is necessary in order to answer the query. For example, in a cluttered living room, $\mathcal{U}$ may be "the chair beside the shelf", which requires a neuro-symbolic concept learner to first find the *shelf*, then find the object *beside* it which is of the class *chair*. There may be many *chairs* in the room, from which the model must identify the target object from.

Concept learners. The overall architecture of LARC follows that of NS3D, and its concept learning framework consists of three main components. The first is a semantic parser that parses the input language $\mathcal{U}$ into a symbolic program $\mathcal{P}$. The symbolic program consists of a hierarchy of primitive operations defined in a domain-specific language for 3D visual reasoning tasks, and represents the reasoning process underlying $\mathcal{U}$. Given the utterance example above, the semantic parser (here, a large language model) will yield the program `relate(filter(scene(), chair), filter(scene(), shelf), beside)`, which indicates the functions that should be run to output the answer.

The second is a 3D feature encoder that extracts structured object-centric features for each scene. The 3D-REC task gives segmented object point clouds for each scene as input, which NS3D leverages; however, LARC instead uses VoteNet [42] to reduce required annotations by detecting objects directly from $\mathcal{S}$. For each detected object point cloud of M_i points $\mathcal{O}_i \in \mathbb{R}^{M_i\times 6}$, a 3D backbone (here, PointNet++ [41]) takes $\mathcal{O}_i$ as input and outputs its corresponding feature $f_i^{\text{obj}} \in \mathbb{R}^d$, where d is the dimension of the feature. For each pair and triple of objects, an encoder learns relational features $f_{i,j}^{\text{binary}}$ and $f_{i,j,k}^{\text{ternary}}$, which represent the binary relations (*e.g.*, beside) and ternary relations (*e.g.*, between) among objects respectively. Given N objects in the scene, the unary representation f^{obj} is a vector with N features, one representing each object. The binary representation f^{binary} is matrix of $N \times N$ features, representing relations between each pair of objects, and similarly for ternary features f^{ternary}.

The third component of concept learners is a neural network-based program executor. The executor takes the program $\mathcal{P}$ and learned features $(f^{\text{obj}}, f^{\text{binary}}, f^{\text{ternary}})$, and returns the target object $\mathcal{T}$. During the execution of the symbolic program, the entity-centric features and learned concept embeddings will be used to compute score vectors, for example, $y^{\text{chair}}, y^{\text{shelf}} \in \mathbb{R}^N$; and a probability matrix, for example, $prob^{\text{beside}} \in \mathbb{R}^{N\times N}$, where N denotes the number of objects in the scene. Elements of y^{chair} denote the likelihood of objects' belonging to category *chair*, and elements of $prob^{\text{beside}}$ denote the likelihood of object pairs' satisfying the relation *beside*. Given the aforementioned symbolic program, the executor reasons with

$$y = \min\left(y^{\text{chair}}, prob^{\text{beside}} \times sx(y^{\text{shelf}})\right),$$

where min is element-wise minumum and $sx()$ is the softmax function applied to y^{shelf}. Finally, the index of the referred object can be found with argmax y.

Naturally supervised setting. In the canonical 3D-REC setup, class labels for each ground truth object bounding box in the scene are given, and used in all previous works as well as in NS3D to supervise intermediate programs. NS3D hence is trained with an object classification loss $\mathcal{L}_{cls}$ along

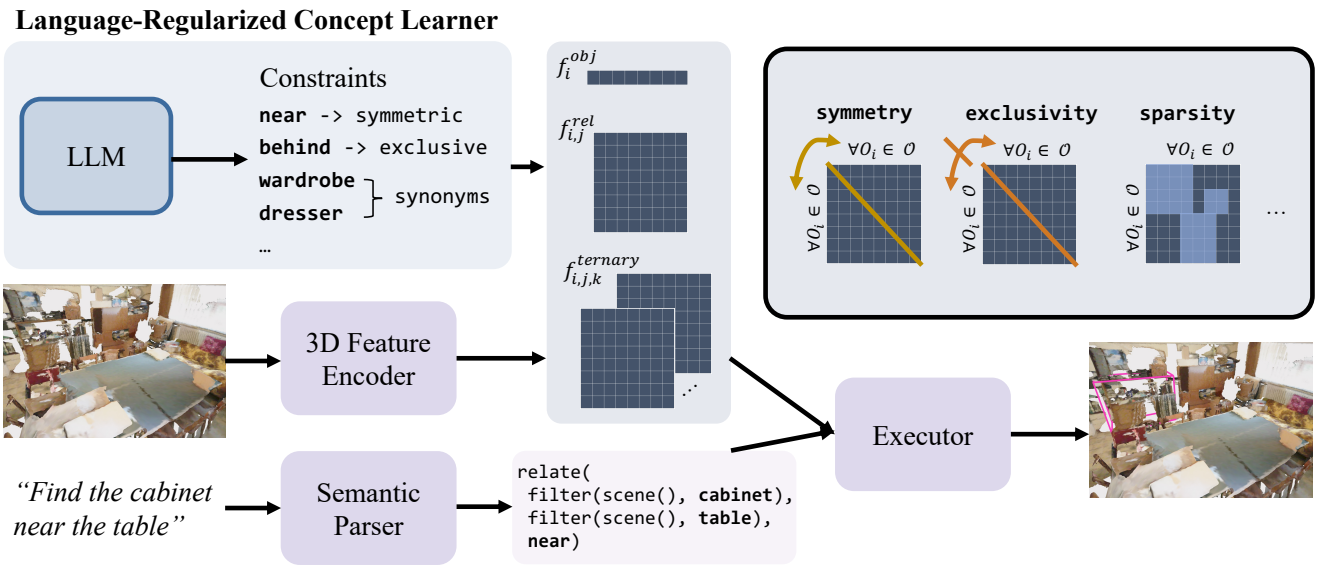

Figure 2. LARC distills constraints from large language models, and injects these rules as regularization into the learning process of structured neuro-symbolic concept learners.

with the final target object prediction loss $\mathcal{L}_{pred}$. In contrast, LARC operates in a naturally supervised setting, supervised by only $\mathcal{L}_{pred}$ and a set of constraint-based regularization losses. In our low guidance setting, ground truth bounding boxes are not known, and we instead use VoteNet [42] to generate object detections.

Constraints. In order to learn in indirectly supervised settings, LARC leverages language-based constraints as regularization. These constraints are based on language priors (*e.g.*, derived from synonyms), hence distilled from LLMs, or are based on general priors (*e.g.*, sparsity). We can enforce such rules on the structured and interpretable representations of concept learners, by way of regularization losses and data augmentation. We describe the definition and application of constraints below.

3.2. Constraint Generation with LLMs

LARC encodes language-based constraints from LLMs in concept learners. These constraints are generally applicable across all language-based tasks, regardless of input visual modality. To this end, LARC uses LLMs to extract concepts names that satisfy a set of language-based rules. We propose symmetry [21], exclusivity [38], and synonymity [40] as general categories of language priors that capture a broad set of language properties and semantic contexts. At a high level, these rules specify the taxonomy of the concepts in relation to one another, as well as its effect on other concepts. Importantly, the constraints are broad, are applicable for all language queries, encode a wide range of properties, and can be used across different datasets.

We query LLMs to classify concepts into these aforementioned constraints, by providing the set of concepts automatically extracted from the semantic parser, as well as definitions of the language rules. LLMs can accurately extract concepts that satisfy the constraints, as the models capture common usage of the concepts. Note that these rules can also be cheaply annotated by humans, compared to the dense supervision otherwise required in the form of 3D bounding boxes or classification labels. Below, we describe the definition for each category of language-based constraints, give examples of concepts from each category, and provide details on how we integrate the rule into LARC.

Symmetry. Relations between objects can be symmetric, in which the same relation holds when the order of the objects in the given relation is reversed. For example, given the relational concept *close*, language priors dictate that the table *close* to the chair implies that the chair is also *close* to the table. Hence, the probability matrix $prob^{\text{close}}$ of size $N \times N$, which describes the likelihood of object pairs' satisfying the relation *close*, should be symmetric.

To distill a set of concepts that are symmetric, we prompt an LLM (here, GPT-3.5 [6]) with the parsed set of relational concepts, and ask it to output the subset of concepts that exhibit such reciprocity. Given the definition of the constraint as prompt, the LLM is able to automatically and accurately determine whether the relational concept satisfies the symmetry prior. Examples of concepts extracted by the LLM include *near*, *beside*, *far*, etc. Then, LARC encodes the symmetric property with the proposed concepts as constraints to regularize LARC's intermediate representations during training. We include prompts in the Appendix.

Exclusivity. We define exclusive relations as concepts that indicate opposing relations when the order of objects is

reversed. For example, given the concept *above*, the box *above* the cabinet implies that the cabinet *cannot* be *above* the box. By querying the LLM for relational concepts that are exclusive and enforcing such constraints during training, we encourage LARC to learn relational representations that are consistent with language-based priors. For example, LARC's $prob^{\text{above}}$ matrix should not yield high probabilities for the relationships "box above cabinet" and "cabinet above box" given the same scene. Examples of exclusive relations proposed by the LLM include *left*, *behind*, *beneath*, etc. See Figure 3 for visualizations of LARC's learned concepts.

Synonymity. Humans have developed an extended set of vocabulary, in which there exist synonyms that represent visually similar 3D objects. End-to-end models typically leverage such nuanced taxonomy and word semantics through pre-trained language encoders, such as BERT [15]. However, such integration of similar language properties is not explicitly modeled in neuro-symbolic frameworks, which ground modular symbols. To enable neuro-symbolic concept learners to encode these language priors in structured representations, we first query LLMs for visually similar synonyms within the concepts of object categories. For example, concepts such as *wardrobe* and *dresser*, as well as *table* and *dining table* are visually similar and synonymous.

LARC then encourages the object-centric representations for the similar concepts to be closer to one another. To do so, LARC first parses utterances into symbolic programs, then selects for programs that contain concepts with an LLM-defined synonym. For each of these programs, we augment the original program with a synonymized version, with a randomly selected synonym concept generated by the LLM substituted in. This encourages the answer, and hence execution trace of LARC, to be similar for synonyms.

3.3. Constraints on structured representations

We introduce two core methods for incorporating constraints into structured neuro-symbolic representations. The first is through regularization losses, and the second through data augmentation. We describe both approaches below. Notably, LARC's intermediate representations can be indexed by concept name and by arity, which enables effective injection of these constraints into the training process.

Regularization losses. Recall that during the execution of neuro-symbolic programs, relations between objects are represented as probability matrices. Here, we use $prob^{\text{binary}} \in \mathbb{R}^{N\times N}$ and $prob^{\text{ternary}} \in \mathbb{R}^{N\times N\times N}$ to denote the probability matrices of binary and ternary relations respectively. Their elements are interpreted as the likelihood that the referred relation exists between the pair or triple of objects. For example, $prob^{\text{beside}}_{i,j}$ specifies the probability that object *i* is beside object *j*, and $prob^{\text{between}}_{i,j,k}$ specifies the probability that object *i* is between objects *j* and *k*, where *i, j, k* are indices of objects. Note that we mask diagonal elements, which represents objects' relations with itself. We ignore them not only in the calculation of losses, but also during execution. Based these notations, we introduce the following regularization losses.

We first define a constraint on the sparsity of the probability matrix $prob^{\text{rel}}$ for each relational concept. Due to the noise in VoteNet object detections, LARC must learn to parse out bounding boxes that are not valid objects in the scene. Hence, we encourage $prob^{\text{rel}}$ to be sparse, keeping large values and ignoring small values. We treat small probability values, where the model is uncertain about the relation between objects, as noise in object bounding box predictions. Therefore, we inject a sparsity regularization loss to denoise LARC's execution on the object-centric representations. The loss $\mathcal{L}_{spar}$ is applied to both binary and ternary relations, and encourages the probability matrices to be sparse as to remove noise from VoteNet object detections. The sparsity regularization loss is defined as

$$\mathcal{L}_{spar} = ||prob^{\text{rel}}||_1.$$

We then describe the symmetry regularization loss $\mathcal{L}_{sym}$, which encourages symmetric relations, as proposed by LLMs in the previous section, by enforcing symmetry on the relation matrix. At a high level, this regularization decreases the difference between $prob^{\text{rel}}_{i,j}$ and $prob^{\text{rel}}_{j,i}$, given that the order of the object does not affect the relation prediction. We define $\mathcal{L}_{sym}$ as

$$\mathcal{L}_{sym} = ||prob^{\text{rel}} - (prob^{\text{rel}})^T||_2^2.$$

Finally, we introduce the exclusivity regularization loss $\mathcal{L}_{excl}$. The exclusivity loss can be interpreted as the opposite of the symmetric loss, where $prob^{\text{rel}}_{i,j}$ and $prob^{\text{rel}}_{j,i}$ is encouraged to not both have high values for a given scene. We define $\mathcal{L}_{excl}$ as

$$\mathcal{L}_{excl} = ||\max(0, prob^{\text{rel}}) \odot \max(0, prob^{\text{rel}})^T||_1.$$

Together, these constraint-based regularization losses are strong signals for neuro-symbolic concept learners in indirectly supervised settings.

Data augmentation. LARC also learns to encode objects that represent similar concepts to closer object-centric representations through data augmentation. LARC augments symbolic programs by replacing parsed concepts with a randomly selected synonym concept, if exists. As an example, given the program `relate(filter(scene(), wardrobe), filter(scene(), cabinet), close)`, and similar concepts of *wardrobe* and *dresser* distilled from an LLM, we will augment the data with an additional program `relate(filter(scene(), dresser), filter(scene(), cabinet), close)`. This program is supervised with the same

answer as the original program, given the same scene. The LARC execution trace will hence be encouraged to be similar between programs with synonym concepts, leading to similar intermediate representations.

3.4. Training

We train LARC in the naturally supervised 3D-REC setting, with only scene and question-answer pairs. LARC is trained with the target object prediction loss $\mathcal{L}_{pred}$, a standard cross-entropy loss, along with the proposed regularization losses from the above sections, which do not require additional annotation. Overall, the loss can be computed as

$$\mathcal{L} = \mathcal{L}_{pred} + \alpha\mathcal{L}_{sym} + \beta\mathcal{L}_{excl} + \gamma\mathcal{L}_{spar}.$$

3.5. Language-based composition

During inference, LARC can also query LLMs for language rules when presented with novel concepts not seen during training. Given an unseen concept, for example *center*, an LLM can specify how it can be composed from a set of learned concepts, automatically extracted by LARC's semantic parser. As an example, the ternary relation of *center* can be decomposed into a series of spatial *left* and *right* relations. Hence, LARC can execute a program of "couch in the *center* of a lamp and a desk" as a combination of: couch *left* of lamp and *right* of desk, or couch *right* of lamp and *left* of desk. Then, LARC can use the learned probability matrices of $prob^{\text{left}}$ and $prob^{\text{right}}$ to compose $prob^{\text{center}}$ as

$$prob^{\text{center}}_{i,j,k} = \max\left(prob^{\text{left}}_{i,j} + prob^{\text{right}}_{i,k}, prob^{\text{right}}_{i,j} + prob^{\text{left}}_{i,k}\right).$$

Similarly, for an unseen concept combination of *not* with a concept such as *behind*, the LLM can specify execution with the learned *front* concept. In practice, LARC builds a lookup table of all antonym pairs queried from LLMs. When presented with new concepts, LARC will query the lookup table and find the antonym concept to execute.

In our experiments, we choose a subset of relations that we know can be composed with learned concepts as a proof of concept. We can also manually specify such language rules to enable execution of new concepts.

4. Experiments

We evaluate LARC on the ReferIt3D benchmark [2], which tests 3D referring expression comprehension on the ScanNet dataset [12]. We specifically focus on the SR3D setting that leverages spatially-oriented referential language, and measure accuracy by matching the predicted objects with the target objects. As we use VoteNet object detections, target objects are calculated as the VoteNet detection that has the highest IOU with the ground truth bounding box. Notably, we study the naturally supervised setting, where all models are not given object-level classification labels during training. Our indirectly supervised setting removes the need for dense annotations in the 3D domain.

In Section 4.1, we compare LARC to NS3D [26], the prior state-of-the-art neuro-symbolic method. We show that simple constraints from language significantly improves the performance of NS3D, and retains all benefits that structured frameworks yield. We additionally present comparisons to end-to-end methods on a variety of metrics, and show qualitative visualizations of LARC's learned concepts. In Section 4.2, we present ablations of LARC's rules and train setting. More visualizations can be found in the Appendix.

4.1. Comparison to prior work

In this section, we evaluate LARC and report test accuracy, generalization accuracy, data efficiency, and transfer accuracy on a new dataset. We compare LARC with top-performing methods: BUTD-DETR [31], MVT [29], NS3D [26], LAR [5], TransRefer3D [23], and LanguageRefer [43]. We additionally present qualitative visualizations of LARC's learned concept representations in comparison to that of NS3D. We note that BUTD-DETR uses labels from pre-trained detectors directly as input to the model; we modified the architecture accordingly and use our VoteNet detections.

Accuracy. We first evaluate LARC's performance on ReferIt3D, compared with prior methods. We also create two additional test subsets, which specifically report the accuracy of queries with symmetric and exclusive concepts. These subsets are selected from the original test set; the symmetric subset consists of 6,487 examples, while the exclusive subset consists of 1,256 examples.

In Table 1, we see that LARC significantly outperforms the prior neuro-symbolic concept learner, NS3D, in the naturally supervised setting. LARC improves performance of NS3D by 9 point percent with our language regularization. LARC also performs comparably to prior end-to-end methods, despite evaluation in the indirectly supervised setting, where neuro-symbolic concept learners tend to underperform. Importantly, we see a significant improvement in all other metrics of interest, described in sections below.

NS	Model	Test acc.	Sym.	Excl.
✗	LanguageRefer [43]	29.8	29.1	31.5
	TransRefer3D [23]	30.6	29.4	32.9
	LAR [5]	32.2	31.3	34.5
	MVT [29]	35.7	32.2	37.3
	BUTD-DETR [31]	**38.5**	**36.3**	**40.2**
✓	NS3D [26]	27.6	24.9	31.2
	LARC (Ours)	**36.6**	**34.5**	**38.4**

Table 1. Comparison of LARC with prior works in the naturally supervised setting of 3D referring expression comprehension. LARC improves performance of NS3D significantly.

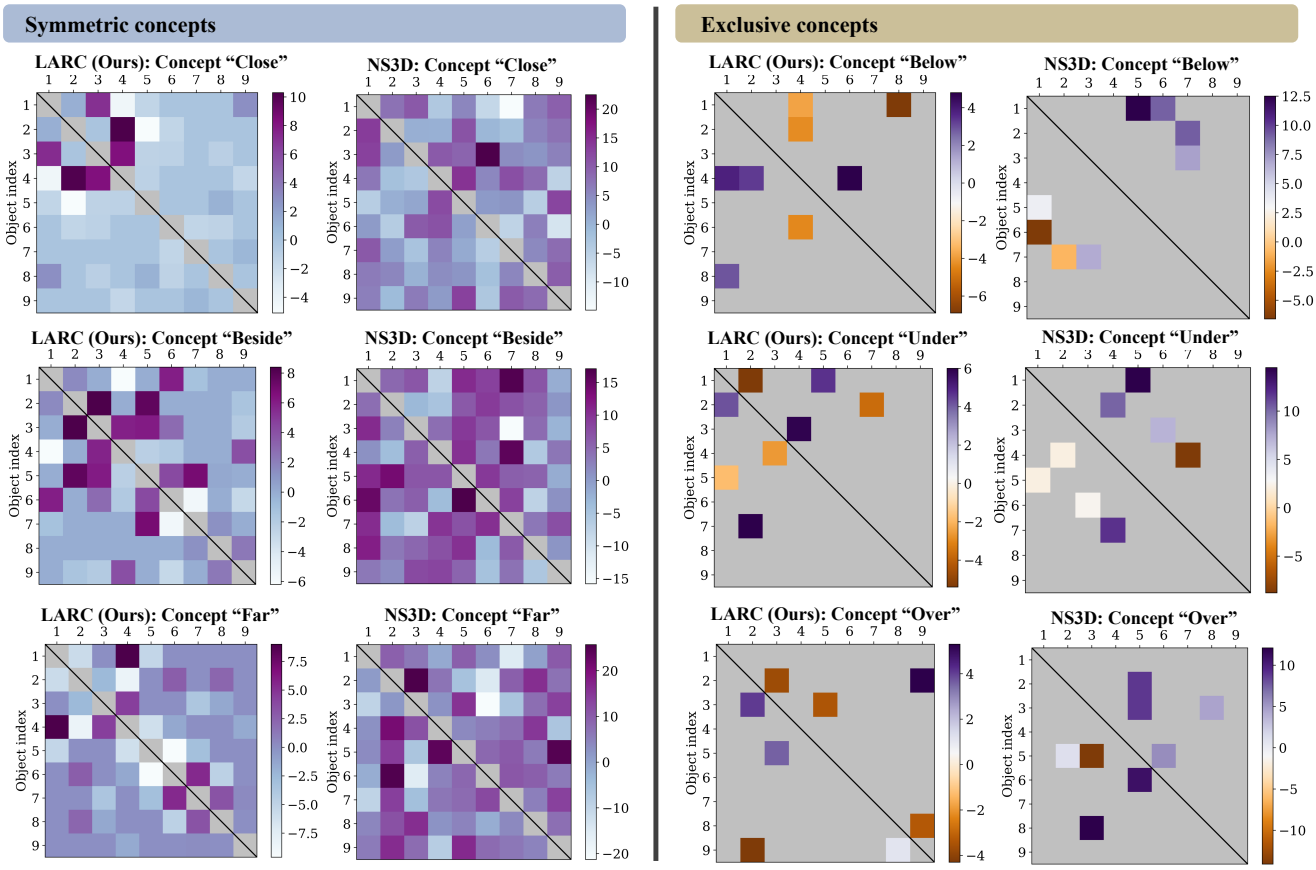

Figure 3. Visualizations of LARC's and NS3D's learned features for symmetric (left two columns) and exclusive (right two columns) concepts; each matrix represents likelihood of pairs of objects' relations adhering to the given concept. LARC features learn to encode constraints from language priors significantly more effectively than that of the NS3D baseline.

Generalization. We evaluate LARC's ability to zero-shot generalize to unseen concepts based on language composition rules. We create two test sets with concepts not seen during training: ternary relations (*e.g.*, *center*, *between*) and antonyms (*e.g.*, *not behind*, *not left*). Queries with these concepts are removed from the train set. The ternary relations test set consists of 1,145 examples from the test set, and the antonym test set consists of 50 annotated examples.

In Table 2, we see that it is easy to compose learned concepts to execute novel concepts in LARC. In comparison, prior works suffer significantly and fail to generalize, even when leveraging word embeddings from powerful, pre-trained language encoders as input to the model.

Data efficiency. We additionally demonstrate that LARC retains strong data efficiency due to its modular concept learning framework. In Table 3, we see that LARC is significantly more data efficient than prior works at 5% (1,423 examples), 10% (2,846 examples), 15% (4,268 examples), 20% (5,691 examples), and 25% (7,113 examples) of train data used. Notably, LARC sees a 6.8 point percent gain from the top-performing prior work with 10% of data.

	Ternary acc.	Antonym acc.
LanguageRefer [43]	8.9	17.9
TransRefer3D [23]	8.5	18.2
LAR [5]	9.7	21.8
MVT [29]	10.5	22.0
BUTD-DETR [31]	9.3	19.5
NS3D [26]	19.1	16.0
LARC (Ours)	**37.4**	**32.6**

Table 2. LARC can use language-based rules to generalize execution of new concepts with composition of learned concepts.

Transfer to an unseen dataset. We also evaluate LARC's transfer performance to an unseen dataset, ScanRefer [8], which contains new utterances on scenes from ScanNet [12]. We retrieve a subset of 384 ScanRefer examples that reference the same object categories and relations as in ReferIt3D, such that all methods can be run inference-only.

In Table 4, we present results to show that LARC's neuro-symbolic framework enables effective transfer to the ScanRefer dataset, while only being trained on the ReferIt3D dataset. Prior methods, even the best end-to-end ones of

	5%	10%	15%	20%	25%
LanguageRefer [43]	17.7	18.5	19.7	21.4	22.8
TransRefer3D [23]	16.4	18.1	19.5	22.1	23.3
LAR [5]	17.5	19.9	21.8	22.5	24.3
MVT [29]	19.5	22.1	24.2	25.0	27.0
BUTD-DETR [31]	19.4	22.3	25.9	28.2	31.5
NS3D [26]	15.7	21.6	22.2	22.7	23.1
LARC (Ours)	**24.9**	**29.1**	**32.4**	**33.9**	**34.8**

Table 3. LARC yields stronger data efficiency compared to prior works at five different percentage points of train data.

BUTD-DETR [31] and MVT [29], do not enable such generalization. We see that LARC outperforms BUTD-DETR by 14.8 point percent and MVT by 15.2 point percent.

Qualitative visualizations. Our proposed regularization method enables LARC to learn representations that are consistent with constraints from language properties. To examine this qualitatively, we present visualizations of LARC's learned concepts in Figure 3. We visualize the probability matrix for each concept, where each value in the $N \times N$ matrix, $prob^{\text{rel}}_{i,j}$, represents the likelihood that the relation between objects of index i and index j adheres to the given concept. For exclusive concepts, we highlight high percentile values as well as their symmetric complements.

In the left two columns of Figure 3, we see that LARC learns symmetric matrices for concepts in which object order in the relation does not matter, such as *close* and *far*. In contrast, NS3D's matrices are noisy and do not exhibit the same consistency. For concepts in which relations are reversed when objects are reversed in order, such as *under* and *over*, LARC captures the exclusive relation between $prob^{\text{rel}}_{i,j}$ and $prob^{\text{rel}}_{j,i}$. In the right two columns of Figure 3, LARC's matrices yield opposing values, and hence colors, in symmetric indices, while NS3D does not encode this knowledge in its representations.

	ScanRefer
LangRefer [43]	13.9
TransRefer [23]	14.7
LAR [5]	15.4
MVT [29]	17.7
BUTD [31]	18.1
NS3D [26]	22.4
LARC (Ours)	**32.9**

Table 4. LARC enables transfer of learned concepts to the ScanRefer dataset [8], while prior works notably drop in performance.

4.2. Ablations

Finally, we present ablations of LARC's performance without each constraint. We additionally ablate LARC's improvement on NS3D in settings with classification supervision.

Constraints. In Table 5, we compare LARC with different variants of LARC trained without each constraint. We see that each of the general rules is important to encode in LARC, as the removal of any constraint leads to worse performance. The synonym prior yields a strong effect on LARC, while the sparsity prior affects LARC at a smaller margin. We hypothesize that this is because the synonym prior is applied on concepts that encode object categories, which are more difficult to learn without classification supervision. Noise in VoteNet object detections may be more trivial in comparison.

Supervision. In Table 6, we present results on a train setting that gives models access to object-level classification labels, to evaluate LARC's performance under denser supervision. The classification label of each VoteNet detected object is assigned the label of the ground truth object with the highest IOU to it. While LARC still improves NS3D by 6.6 point percent in this setting, we see less of a performance gain from our proposed regularization, compared to the 9 point percent improvement in the naturally supervised setting. We hypothesize that this is because object-level classification supervision reduces uncertainty in LARC's object-centric representations during training, hence lessens the need for regularization. We note that as the class labels for supervision are applied on predicted bounding boxes with potentially incomplete object point clouds, classification supervision does not yield significant improvements.

	Acc.
LARC (Ours)	36.6
w/o Symmetry	34.9
w/o Exclusivity	35.1
w/o Sparsity	35.8
w/o Synonymity	33.4

Table 5. LARC's accuracy without each constraint.

	Cls.	Acc.	Cls.	Acc.
NS3D [26]	✓	31.6	✗	27.6
LARC (Ours)	✓	**38.2**	✗	**36.6**

Table 6. Comparisons under a train setting with object-level classification supervision on predicted bounding boxes.

5. Conclusion

3D visual grounding models perform poorly in naturally supervised settings, without access to object-level semantic labels or ground truth bounding boxes. We propose the Language-Regularized Concept Learner as a neuro-symbolic method that uses language regularization to significantly improve performance in indirectly supervised settings. We demonstrate that simple constraints, with concepts distilled from large language models, can significantly increase accuracy by way of regularization on structured representations. We show that LARC is extremely generalizable, highly data efficient, and effective transfers learned concepts to different datasets, by only looking at 3D scene and question-answer pairs in the naturally supervised setting.

Acknowledgments. This work is in part supported by NSF RI #2211258, ONR N00014-23-1-2355, ONR YIP N00014-24-1-2117, and AFOSR YIP FA9550-23-1-0127.

References

[1] Ahmed Abdelreheem, Ujjwal Upadhyay, Ivan Skorokhodov, Rawan Al Yahya, Jun Chen, and Mohamed Elhoseiny. 3DRefTransformer: Fine-grained Object Identification in Real-world Scenes Using Natural Language. In *WACV*, pages 3941–3950, 2022. 2

[2] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural Listeners for Fine-grained 3D Object Identification in Real-world Scenes. In *ECCV*, pages 422–440. Springer, 2020. 2, 6, 11

[3] Miltiadis Allamanis, Pankajan Chanthirasegaran, Pushmeet Kohli, and Charles Sutton. Learning continuous semantic representations of symbolic expressions. In *ICML*, 2017. 3

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to Compose Neural Networks for Question Answering. In *NAACL-HLT*, 2016. 2

[5] Eslam Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. Look Around and Refer: 2D Synthetic Semantics Knowledge Distillation for 3D Visual Grounding. In *NeurIPS*, pages 37146–37158, 2022. 1, 2, 6, 7, 8, 11

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. *NeurIPS*, 33:1877–1901, 2020. 4, 11

[7] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3DJCG: A Unified Framework for Joint Dense Captioning and Visual Grounding on 3D Point Clouds. In *CVPR*, pages 16464–16473, 2022. 2

[8] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. In *ECCV*, pages 202–221. Springer, 2020. 7, 8, 11

[9] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X. Chang. D3Net: A Speaker-listener Architecture for Semi-supervised Dense Captioning and Visual Grounding in RGB-D Scans, 2021.

[10] Jiaming Chen, Weixin Luo, Xiaolin Wei, Lin Ma, and Wei Zhang. HAM: Hierarchical Attention Model with High Performance for 3D Visual Grounding. *arXiv preprint arXiv:2210.12513*, 2022. 2

[11] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding Physical Concepts of Objects and Events Through Dynamic Visual Reasoning. In *ICLR*, 2021. 2

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, pages 5828–5839, 2017. 6, 7, 11

[13] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040, 2022. 3

[14] Changyu Deng, Xunbi Ji, Colton Rainey, Jianyu Zhang, and Wei Lu. Integrating machine learning with human knowledge. *Iscience*, 23(11), 2020. 3

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ACL*, 2019. 5

[16] Michelangelo Diligenti, Marco Gori, and Claudio Saccà. Semantic-based Regularization for Learning and Inference. In *Artificial Intelligence 244 (2017) 143–165*, 2015. 3

[17] Michelangelo Diligenti, Soumali Roychowdhury, and Marco Gori. Integrating prior knowledge into deep learning. In *International Conference on Machine Learning and Applications*, 2017. 3

[18] Mark Endo, Joy Hsu, Jiaman Li, and Jiajun Wu. Motion question answering via modular motion programs. *ICML*, 2023. 2, 3

[19] Manoel VM França, Gerson Zaverucha, and Artur S d'Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine learning*, 94:81–104, 2014. 3

[20] Eleonora Giunchiglia, Mihaela Catalina Stoian, and Thomas Lukasiewicz. Deep learning with logical constraints. In *IJCAI*, 2022. 3

[21] Lila R Gleitman, Henry Gleitman, Carol Miller, and Ruth Ostrin. Similar, and similar concepts. *Cognition*, 58(3):321–376, 1996. 2, 4

[22] Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. Visual Concept-Metaconcept Learning. In *NeurIPS*, 2019. 2

[23] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. TransRefer3D: Entity-and-Relation Aware Transformer for Fine-Grained 3D Visual Grounding. In *ACM International Conference on Multimedia*, pages 2344–2352, 2021. 1, 2, 6, 7, 8, 11

[24] Nick Hoernle, Rafael Michael Karampatsis, Vaishak Belle, and Kobi Gal. MultiplexNet: Towards Fully Satisfied Logical Constraints in Neural Networks. In *AAAI*, pages 5700–5709, 2022. 3

[25] Joy Hsu, Jiayuan Mao, Joshua B Tenenbaum, and Jiajun Wu. What's Left? Concept Grounding with Logic-Enhanced Foundation Models. *NeurIPS*, 2023. 2

[26] Joy Hsu, Jiayuan Mao, and Jiajun Wu. NS3D: Neuro-Symbolic Grounding of 3D Objects and Relations. In *CVPR*, pages 2614–2623, 2023. 1, 2, 3, 6, 7, 8, 11

[27] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing Deep Neural Networks with Logic Rules. In *ACL*, 2016. 3

[28] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided Graph Neural Networks for Referring 3D Instance Segmentation. In *AAAI*, pages 1610–1618, 2021. 2

[29] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-View Transformer for 3D Visual Grounding. In *CVPR*, 2022. 1, 2, 6, 7, 8, 11

[30] Drew Hudson and Christopher D Manning. Learning by Abstraction: The Neural State Machine. In *NeurIPS*, 2019. 2

[31] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom Up Top Down Detection Transformers for Language Grounding in Images and Point Clouds. In *ECCV*, pages 417–433. Springer, 2022. 1, 2, 6, 7, 8, 11

[32] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and Executing Programs for Visual Reasoning. In *ICCV*, 2017. 2

[33] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning. In *ICML*, 2020. 2

[34] Tao Li and Vivek Srikumar. Augmenting Neural Networks with First-Order Logic. *ACL*, 2019. 3

[35] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3D-SPS: Single-stage 3D Visual Grounding via Referred Point Progressive Selection. In *CVPR*, pages 16454–16463, 2022. 2

[36] 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. In *ICLR*, 2019. 1, 2, 3

[37] Jiayuan Mao, Tomas Lozano-Perez, Joshua B. Tenenbaum, and Leslie Pack Kaelbing. PDSketch: Integrated Domain Programming, Learning, and Planning. In *NeurIPS*, 2022. 2

[38] Ellen M Markman and Gwyn F Wachtel. Children's use of mutual exclusivity to constrain the meanings of words. *Cognitive psychology*, 20(2):121–157, 1988. 2, 4

[39] Giuseppe Marra, Michelangelo Diligenti, Francesco Giannini, Marco Gori, and Marco Maggini. Relational neural machines. In *ECAI*, 2020. 3

[40] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 2, 4

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NeurIPS*, 30, 2017. 3

[42] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *CVPR*, pages 9277–9286, 2019. 3, 4, 11

[43] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. LanguageRefer: Spatial-Language Model for 3D Visual Grounding. In *CoRL*, pages 1046–1056. PMLR, 2022. 1, 2, 6, 7, 8, 11

[44] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*, 2019. 2

[45] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 3

[46] Geoffrey G Towell and Jude W Shavlik. Knowledge-based artificial neural networks. *Artificial intelligence*, 70(1-2): 119–165, 1994. 3

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *NeurIPS*, 30, 2017. 2

[48] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021. 3

[49] Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and Joshua B Tenenbaum. From Word Models to World Models: Translating from Natural Language to the Probabilistic Language of Thought. *arXiv preprint arXiv:2306.12672*, 2023. 2

[50] Yaqi Xie, Ziwei Xu, Mohan S. Kankanhalli, Kuldeep S. Meel, and Harold Soh. Embedding Symbolic Knowledge into Deep Networks. In *NeurIPS*, 2019. 3

[51] Jingyi Xu, Zilu zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *ICML*, 2018. 3

[52] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. SAT: 2D Semantics Assisted Training for 3D Visual Grounding. In *ICCV*, pages 1856–1866, 2021. 2

[53] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In *NeurIPS*, 2018. 2

[54] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. InstanceRefer: Cooperative Holistic Understanding for Visual Grounding on Point Clouds through Instance Multi-level Contextual Referring. In *ICCV*, pages 1791–1800, 2021. 2

[55] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation Modeling for Visual Grounding on Point Clouds. In *ICCV*, pages 2928–2937, 2021. 2

Language-Regularized Concept Learners

Supplementary Material

The appendix is organized as the following. In Appendix A, we specify the prompts used to query LLMs in LARC. In Appendix B, we present additional visualizations of LARC's performance. In Appendix C, we include additional results of LARC on different levels of box prediction noise. In Appendix D, we discuss the ScanRefer [8] dataset.

A. Prompts

Below, we provide the prompts used to query large language models, specifically, GPT-3.5 [6], for concepts that satisfy LARC's constraints.

Symmetry and exclusivity. We use the following prompt to categorize relational concepts, where *[relations]* is the list of relational concepts automatically extracted from the input language by LARC's semantic parser:
We define two kinds of spatial relations: Asymmetric relations are relations that don't exhibit reciprocity when the order of the objects is reversed. Symmetric relations are relations that exhibit reciprocity when the order of the objects is reversed. Here are some relations: [relations]. For each relation, specify whether it is a symmetric relation or an asymmetric relation.

Synonyms. We use the following two-round query to find visually similar synonyms in object categories, where the *[object categories]* list is automatically extracted:
First round: *Here are some object categories: [object categories]. List categories that have similar meanings.*
Second round: *Within each group, list categories that have similar appearances.*

B. Visualizations

In this section, we present additional visualizations of LARC's performance. First, we compare LARC's predictions to that of prior works on the ReferIt3D [2] dataset. Then, we provide execution trace examples of LARC. After, we demonstrate failure cases of LARC and include analyses. Finally, we show examples of VoteNet [42] object detections in comparison to ground truth bounding boxes.

Comparison to prior works We present examples of LARC's predictions as well as baselines' on the ReferIt3D [2] dataset. We see samples in Figure 4 where LARC outperforms baselines, including NS3D [26], BUTD-DETR [31], MVT [29], LAR [5], TransRefer [23], and LangRefer [43], in the naturally supervised 3D grounding setting.

Execution traces In Figure 5, we present examples of LARC's execution trace. LARC first parses input instruction utterances into symbolic programs, then hierarchically executes each modular program to retrieve the answer.

Failure cases We provide several examples of LARC's failure cases in Figure 6. In the top row, we see cases where LARC finds target objects of the correct object category, but with incorrect relations. In the bottom row, we see cases where LARC yields target objects of incorrect object categories. LARC is likely to fail in 3D visual grounding when the target object category is one without data-augumented synonyms during training, as it is difficult to learn with few examples in the naturally supervised setting.

VoteNet detections In Figure 7, we show examples of VoteNet [42] object detections, used in our low guidance setting, in comparison to ground truth bounding boxes. We see that VoteNet detections often result in incomplete point clouds, due to size corruption or center shift. This noise leads to additional challenges in 3D visual grounding; however, VoteNet detections significantly reduce the amount of labelled 3D data required during inference.

C. Experiments

Noisy detection experiments. We report results of NS3D and LARC over 6 different levels of box prediction noise in Table 7, with each column representing ratio of perturbation on the original box. LARC consistently improves NS3D under all settings.

Noise level	0.0	0.1	0.2	0.3	0.4.	0.5
NS3D	27.6	22.9	20.7	19.6	13.8	10.7
LARC (Ours)	**36.6**	**35.6**	**33.5**	**30.2**	**24.7**	**20.1**

Table 7. Comparisons under different levels of box prediction noise.

D. ScanRefer

Here, we describe how LARC uses the ScanRefer [8] data for zero-shot transfer from ReferIt3D [2].

Data construction We first create a subset of ScanRefer with queries that contain the same objects and relations as in ReferIt3D, such that we can run all method inference-only. This ScanRefer subset consists of 384 unseen utterances, on the same ScanNet [12] scenes.

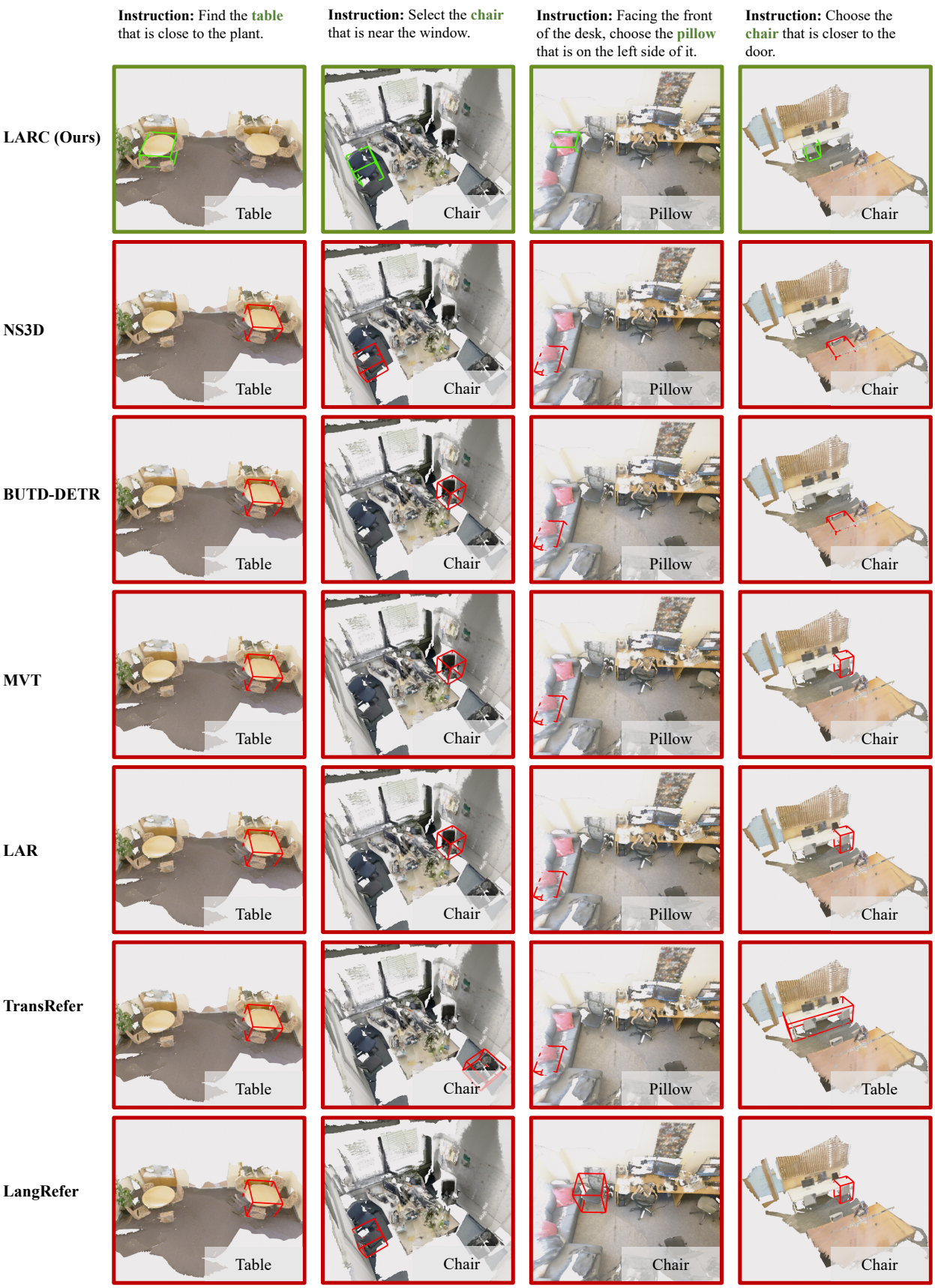

Figure 4. LARC's performance compared to prior works in the naturally supervised setting; each column shows every model's prediction for a given instruction.

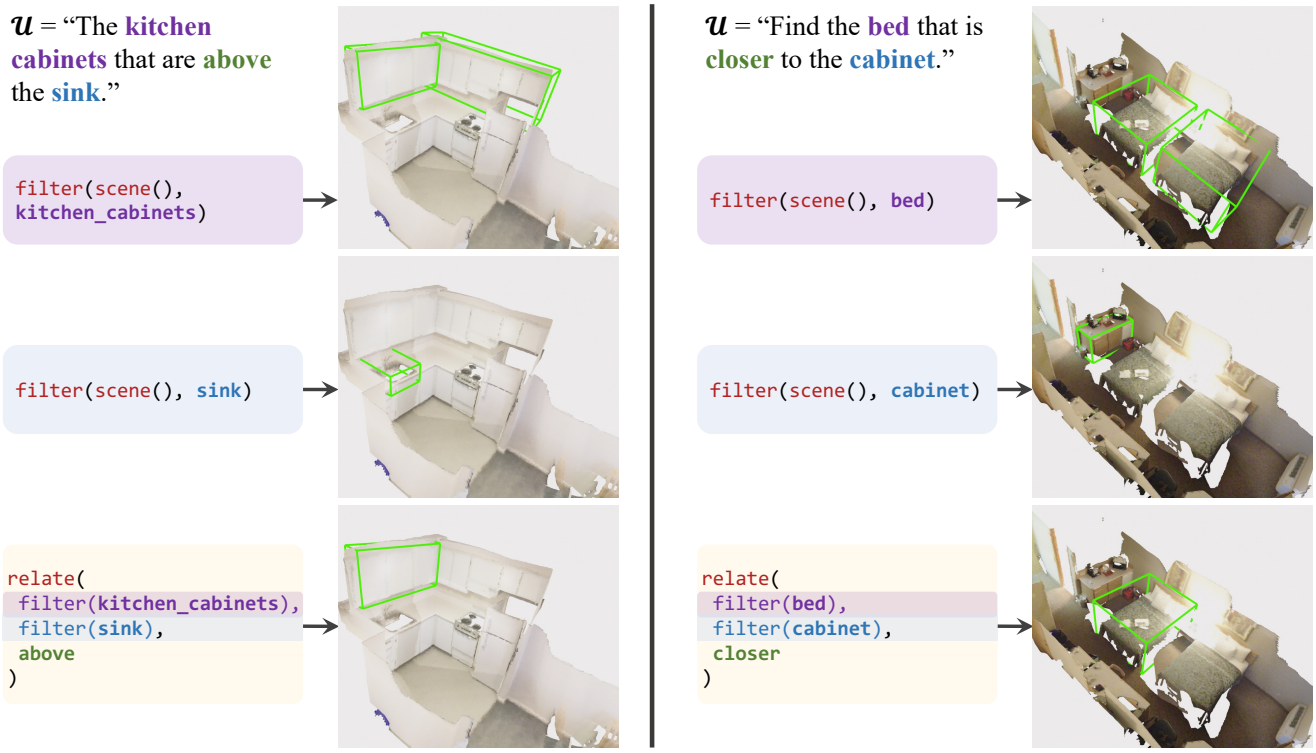

Figure 5. LARC's neuro-symbolic framework executes symbolic programs hierarchically to retrieve the target answers.

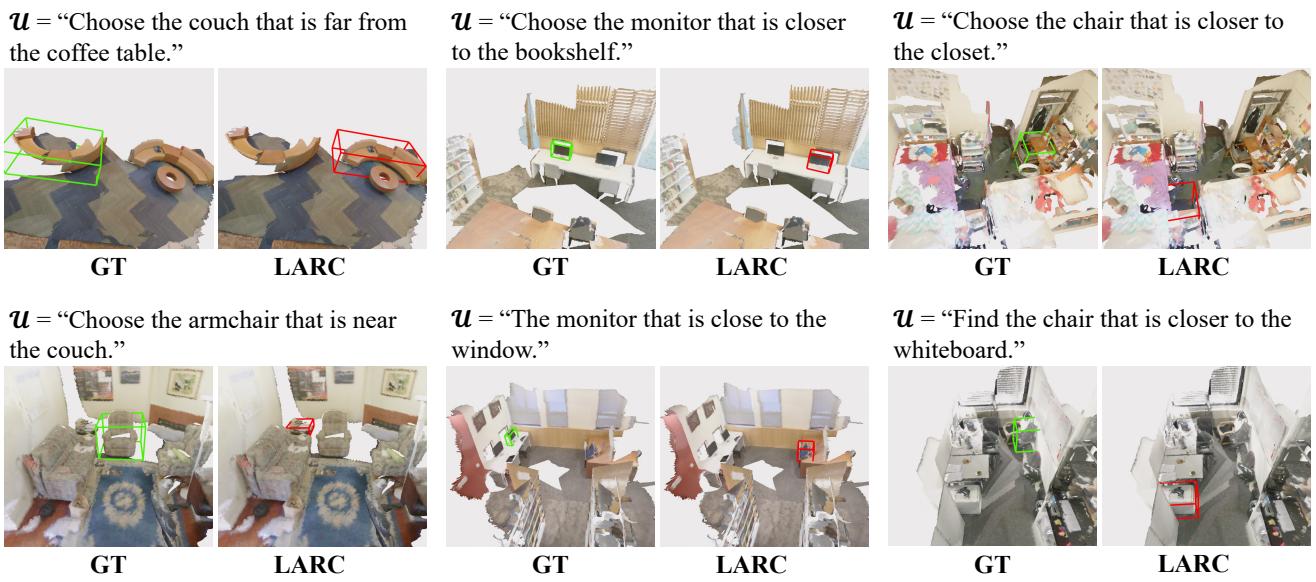

Figure 6. LARC can fail in understanding 3D relations (top row) or 3D object categories (bottom row); its modularity enables such analyses.

Implementation To transfer learned concepts to ScanRefer, we use GPT as LARC's semantic parser to generate programs from input language. The programs are executed as described in the main paper. LARC relies on the generalization abilities of LLMs to zero-shot transfer to ScanRefer, by decomposing new language into learned programs, without requiring any additional training or finetuning of neural networks. In comparison, end-to-end methods significantly underperform when faced with unseen input language.

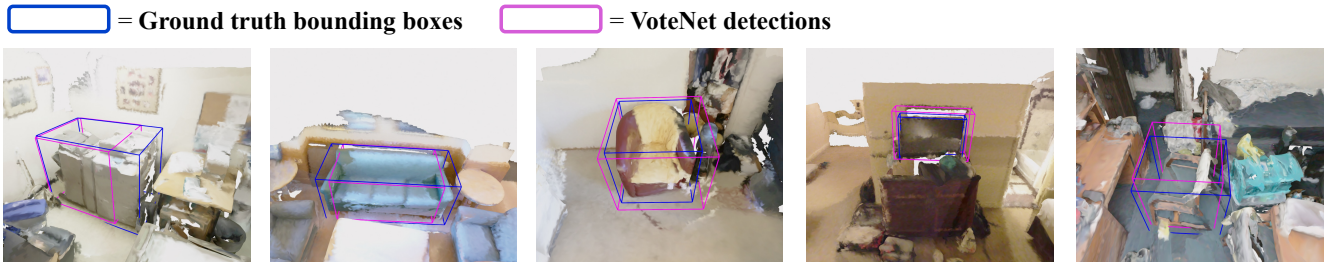

Figure 7. Comparison of ground truth bounding boxes (in blue) and VoteNet detections (in purple) used in the low guidance 3D visual grounding setting.