IEEE Robotics and Automation Letters (RAL) paper presented at
the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2022)
October 23-27, 2022, Kyoto, Japan

SGL: Symbolic Goal Learning in a Hybrid, Modular Framework for Human Instruction Following

Ruinian Xu, Hongyi Chen, Yunzhi Lin, and Patricio A. Vela

***Abstract*— This paper investigates human instruction following for robotic manipulation via a hybrid, modular system with symbolic and connectionist elements. Symbolic methods build modular systems with semantic parsing and task planning modules for producing sequences of actions from natural language requests. Modern connectionist methods employ deep neural networks that learn visual and linguistic features for mapping inputs to a sequence of low-level actions, in an end-to-end fashion. The hybrid, modular system blends these two approaches to create a modular framework: it formulates instruction following as symbolic goal learning via deep neural networks followed by task planning via symbolic planners. Connectionist and symbolic modules are bridged with Planning Domain Definition Language. The vision-and-language learning network predicts its goal representation, which is sent to a planner for producing a task-completing action sequence. For improving the flexibility of natural language, we further incorporate implicit human intents with explicit human instructions. To learn generic features for vision and language, we propose to separately pretrain vision and language encoders on scene graph parsing and semantic textual similarity tasks. Benchmarking evaluates the impacts of different components of, or options for, the vision-and-language learning model and shows the effectiveness of pretraining strategies. Manipulation experiments conducted in the simulator AI2THOR show the robustness of the framework to novel scenarios.** [1]

I. Introduction

Ideally robot agents sharing the same working space with humans and assisting them would be capable of interpreting human instructions and performing their corresponding tasks. The main challenge in human instruction following comes from the diversity of communication and interpretation, which permits incomplete or ambiguous natural language. This paper proposes to disambiguate natural language via visual information within a hybrid, modular framework.

Early symbolic works employ semantic parsing and task planning to first map natural language into certain representations and then generate a sequence of actions. Attempts to address the ambiguity of natural lanugage include incorporating knowledge bases [1], [2], dialogue systems [3], and vision [4]. Dialogue systems shift the burden of disambiguation to the user. Alternatively, information from visual sensors might provide the missing information at the cost of slower processing via symbolic methods, due to growth in the candidate instruction space [4]. With sufficient data, connectionist approaches to parsing multi-domain information might improve parsing performance without sacrificing inference speed.

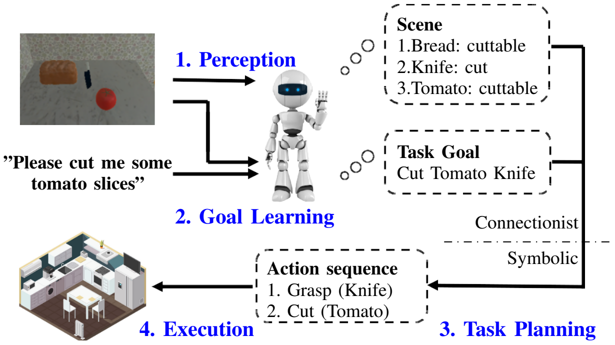


Fig. 1. Illustration of SGL, a hybrid, modular framework for human instruction following. SGL consists of four main components: perception, goal learning, task planning, and execution. Best viewed in color.

Connectionist approaches avoid processing natural language and vision based engineered symbolic representations by learning visual and linguistic features via deep neural networks. Sequence-to-sequence models learn to map image and text input to a sequence of low-level actions [5], [6]. End-to-end designs exhibit exposure bias caused by using complete target actions as training inputs but not having access to them for the inference stage [7]. An error in one step affects future predictions. To enhance performance, researchers break the network into sub-modules or separately consider different types of tasks [8]–[11], which requires richer annotations.

To leverage the strengths of symbolic and connectionist approaches, we propose a hybrid, modular framework as depicted in Figure 1. The modules perform perception, vision-aided goal learning via a deep network, task planning via a symbolic planner, and task execution. Inspired by previous methods for manipulation task completion via object affordance recognition [12], [13], we formulate goal learning as predicting symbolic goal representations for Planning Domain Definition Language (PDDL) specifications, to bridge connectionist goal learning and symbolic task planning. This paper's main contributions are:
(1) A hybrid, modular system for human instruction following. It leverages the semantic feature learning properties of deep neural networks and symbolic computation of task planners. Modularity facilitates component-level analysis and upgrading.
(2) Benchmarking of component-level impacts on vision-and-language robotic task goal learning. To enhance visuo-linguistic network performance via pretraining tasks [14]–

Ruinian Xu, Hongyi Chen, Yunzhi Lin and Patricio A. Vela are with Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, GA, USA. {rxu72, hchen657, yunzhi.lin, pvela}@gatech.edu

This work was supported in part by NSF Award #2026611.

[1] All code is publicly available at https://github.com/ivalab/mmf

[16], two strategies are proposed to separately pretrain the visual and linguistic encoders: scene graph parsing and semantic textual similarity. They outperform standard pretraining methods.
(3) Manipulation experiments conducted in the AI2THOR simulator [17], with five daily activities and unseen scenarios, demonstrate the robustness of the proposed framework to novel objects and environments.

II. Related Work

A. Human Instrution Following

Human instruction following requires robotic agents to understand human instructions and perform the requested tasks. Common robotic tasks include navigation and manipulation. Each has different scene understanding needs. Navigation requires identifying landmarks to understand where the agent is and how it should move. The reviews [18], [19] describe existing work on Vision-and-Language Navigation. Manipulation requires interpreting interactions between objects and how to manipulate them. The focus here is on instruction following for robotic manipulation. Some work on this front focuses on visually-derived ambiguity for a primitive action given a specific request [20], [21] as opposed to commands with differing command specificity (*i.e.*, lingusitic ambiguity) but unambiguous action sequences given the scene. This section will first review existing symbolic and connectionist methods for human instruction following. Afterwards, it reviews connectionist methods for joint feature learning in vision-and-language neural networks and common pretraining tasks for learning generic visual and linguistic features.

1) Symbolic Method: Human instruction following requires translating human language into robot understandable language. Based on manually defined symbols, early works employ semantic parsing to transform natural language into logical representations which perserves the meaning. With well-structured input language, there are works parse natural language into formal semantic expressions such as a list of templates [22], which is unscalable with the growth of the complexity of the manipulation task. Instead of parsing natural language into formal representation, researchers have explored the direction of intermediate representations such as Spatial Description Clause (SDC) [23] and Linear Temporal Logic (LTL) [24], [25], which will also be the direction of symbolic representation in this work. However, instructions provided by non-expert human users can be vague or incomplete. Realizing the ambiguity of natural language, some researchers attempt to incorporate external information such as knowledge bases [1], [2], dialogue systems [3], visual scene information [4] or multi-source information [26]. Among these auxiliary information from robotic vision, serving as a simple and straightforward but rich way to disambiguate natural language, will be studied in this work. Semantic parsing, which relies on syntax of language to perform symbolic computation, can't well capture the semantic meaning of language and has the difficulty of translating abstract sentences such as human intents. Meanwhile, symbolic approaches using rule-based task planning achieve high accuracy for computing action sequences for manipulation when the symbols are correct.

2) Connectionist Method: With the significant evolution of connectionist methods in recent years, deep neural networks show impressive strengths in learning semantic and high-dimensional features, which improves robustness to various types of input data. Packing everything into one network, end-to-end learning models [5], [6] are first proposed to directly map natural language and vision to a sequence of low-level actions. The sequence-to-sequence model suffers from the well-known issue of teacher forcing, which leads to the poor performance under test scenarios. Observing the great performance drop from training to testing stages of end-to-end learning models, researchers start to break the end-to-end network design and modularize the framework into several networks. There are different designs of modular networks focus on different natures of robotic tasks, such as decomposing the model into perception and action policy streams [8], [9], modularizing the model into separate submodules for sub-tasks [10], [11], decomposing the problem into sub-goal planning, scene navigation, and object manipulation [27] and constructing the model into observation model, high-level controller and low-level controller [28]. Modular systems with purely connectionist modules should benefit from symbolic elements. Reason for this assertion is that the uncorrected error propagation from one module through subsequent modules when they consist purely of connectionist modules [7]. The symbolic modules in a hybrid system can be designed offset the issue by recognizing inconsistent inputs.

B. Vision-and-Language Feature Learning

Learning symbolic goal representation via vision-and-language deep networks requires learning generic visual and linguistic features to assist generalization to unseen scenarios. Here, we review existing methods in visual question answering for visual and linguistic feature encoding and their pretraining tasks.

Visual feature learning methods used in V&L models can be categorized into Object Detector(OD)-based region, CNN-based grid and Vision Transformer(ViT) patch features. Due to the computational and time cost of pretraining vision transformer, this type of methods will not be explored and benchmarked. Most previous works [15], [16], [29], [30] employ OD-based region features which are extracted via pretrained Faster R-CNN [31] based object detectors. Concerns for these types of methods are frozen parameters and time cost of object detectors during the training and inference stage, respectively. To address these two issues, works [32], [33] have explored grid extracted visual features via CNNs such as ResNet [34], which makes the vision-and-language model end-to-end trainable. One-stage designs for visual feature learning also reduce inference time but sacrifice a small amount of performance. Alternatively, pretraining CNNs using similar but different tasks enhances feature learning for downstream tasks [35]. Options are object detection [36], semantic segmentation [37], and instance segmentation [38]. Though existing pretraining tasks help to capture object information in imagery, they ignore potential interactions between objects important to robotic tasks. Better pretraining tasks should be identified.

For linguistic feature learning, early research [39]–[41] focused on learning word-level feature embeddings. To learn high-level semantic embedding for sentences, based on Recurrent Neural Networks (RNN), LSTM, Bidirectional LSTM, GRU [42] and other similar designs are proposed. The main concern of RNN-based methods is forgetting past information for modeling long sequence data. The rise of Transformers [43] led to a new family of approaches, such as GPT [44], BERT [45], RoBERTa [46], etc. Among them, BERT model and its pretraining strategy of masked language modeling (MLM) is most widely used due to its simple network design and superior performance. Modeling natural language without clustering sentences with similar semantic meanings, linguistic encoders might have the difficulty of interpreting similarity between explicit human instruction and implicit human intent.

The above review of symbolic and connectionist approaches for human instruction following suggests considering a hybrid, modular system to leverage the strengths of both methods and compensate each other's limitations: we propose to address human instruction following via connectionist goal learning and symbolic task planning. Employing Planning Domain Definition Language (PDDL) as the symbolic representation, connectionist and symbolic approaches are bridged with PDDL goal specifications. The vision-and-language connectionist framework, consisting of a visual encoder, a linguistic encoder, multi-modal fusion and a classifier, is to learn symbolic goal states (not actions). The output goal representation feeds to a symbolic task planner to generate a sequence of actions. To improve feature learning in the vision-and-language network, we propose to separately pretrain the visual and linguistic encoders on scene graph parsing and semantic textual similarity tasks. Scene graph parsing forces visual encoders to capture relationships between objects, while semantic textual similarity helps linguistic encoders learn similar semantic embeddings between human instructions and intents. The modular design of goal learning and instruction following frameworks enables simpler replacement and upgrading of individual components and analysis for failures.

III. Preliminaries

A. Planning Domain Definition Language

For task planning, we employ the Planning Domain Definition Language (PDDL), a widely used symbolic planning language. With a list of pre-defined **objects** and their corresponding **predicates** (such as dirty, graspable, etc.), a **domain** consists of primitive actions and corresponding effects. Here, affordances and attributes serve to define available **predicates** for subsequently specifying object-action-object relationships. Planning requires establishing a **problem**, which is composed of the initial state and a desired goal state of the world. The initial state is formed with a list of objects with corresponding predicates. The goal state is structured in the form of action, subject and object. From the **domain** and **problem** specification, a PDDL planner produces a sequence of primitive actions leaving the world in the goal state when executed.

B. Problem Statement

Given an RGB image and a sentence of natural language, the objective of this framework is to generate a sequence of manipulation actions that achieve the task indicated by the sentence. Processing of RGB image generates an initial state estimate using an object detector. Completing the problem specification involves the proposed vision-and-language deep learning framework, whose function is to convert the paired image and natural language input into a symbolic goal representation compatible with PDDL. Once the problem specification is built, the symbolic PDDL planner solves it to generate the action sequence. The robot then performs the ordered actions in the environment.

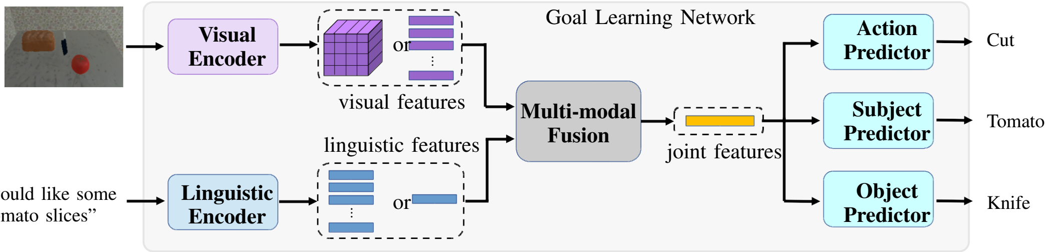
IV. Approach

This section works from the bottom up, first describing the vision-and-language deep learning architecture proposed for learning symbolic goal representations for PDDL specification. Section IV-A describes the modular elements and candidate options. Sections IV-B and IV-C describe customized pretraining tasks for the visual and linguistic encoder modules, respectively. The section concludes with a description of the integrated hybrid, modular framework for human instruction following, taking vision and language inputs to then output a sequence of corresponding manipulation actions.

A. Vision-and-Language Task Goal Learning

The proposed symbolic goal learning architecture is depicted in Figure 2. It outputs a simple PDDL goal consisting of action a, subject s and object o from an RGB image I and a natural language string L. The underlying vision-and-language deep learning network adopts a modular design consisting of a visual encoder, a linguistic encoder, a multi-modal fusion module and a classification module. The visual and linguistic encoders learn visual and linguistic features, respectively. As these features are embedded in different domains, the encoder outputs require fusion into a joint feature space. These joint features feed to three classifiers for predicting the action, subject and object of the PDDL goal representation. Each classifier is a 2-layer Multi-Layer Perception (MLP) with 256 hidden dimensions. The action classifier has 5 categories, while the subject and object classifiers have 35 categories.

Visual Encoder. The visual encoder produces a set of local features $V = \{v_1, \cdots, v_n\}$, from a RGB image $I \in \mathbb{R}^{H \times W \times 3}$. There are two principal types of visual features, grid and region. Grid features arise from block-wise image processing that leads to a feature map $V \in \mathbb{D}^{H_1 \times W_1 \times C_1}$ where H_1 and W_1 are the grid height and width and C_1



Request Types	Explicit	Complete	Grasp the bread, place the bread in the toaster, and turn the toaster on.
		Incomplete	Help me place the bread in the toaster to toast it.
	Implicit		I am hungry and would like toasted bread. OR I want toasted bread.

Fig. 2. Vision and language symbolic goal learning network architecture. From RGB image and natural lanugage inputs, it outputs a PDDL goal state (action, subject and object). Blocks with solid line represent components. Blocks with dashed line represent visual, linguistic and joint features.

is the dimension of a grid feature. A linearly indexed grid feature is v_i. The network for grid features will be ResNet [34]. Region features arise from convolutional networks that scan the entire image to identify candidate regions containing targets of interest; The outputs are regional features $V \in \mathbb{R}^{N \times C_1}$ extracted from the candidate regions, where N is the number of regions. Faster R-CNN is one such network [31]. Both are tested in §V-C.

Linguistic Encoder. Given a natural language sentence L composed of K words, the linguistic encoder can either generate the corresponding embedding set $Q = \{q_1, \cdots, q_k\}$ which represents each word or a single embedding vector q which represents the semantic meaning of the entire sentence. The encoder commonly involves word embedding and feature encoding. For word embedding, each word in the sentence will be mapped to an embedding based on some pretrained embedding tables, such as GloVe [40]. Feature encoding options include LSTM and transformers. LSTM is a common network design for language modeling capable of connecting past information to the current task. More recently, attention-based transformers have gained popularity for improved memorization of long sentences and capture of semantic meaning in language. Both approaches are evaluated in §V-C.

Multi-modal Fusion. Visual and linguistic features lie in different domains and require extra operations to fuse into a joint representation. The simplest fusion operations are concatenation, element-wise addition, and multiplication. While linguistic features represent the entire sentence, the set of visual features does so implicitly. Additional processing of the visual features is needed to obtain a single image-wide feature representation. Pooling operations such as max or average pooling, or simple addition achieve this outcome. A potential issue is that not all local features contribute equally to the final prediction. Some visual features are related to irrelevant pixels or regions, which should be ignored or have less influence on the pooled output. Further, linguistic features extracted from instructions can provide guidance in identifying latent regions whose information should be preserved. Attention modules [43] in the form of self-attention and cross-attention are widely used to correlate features in the same domain and across different domains, respectively. Section V-C evaluates and discusses simple and attention-based fusion methods.

B. Pretraining on Scene Graph Parsing

The vision-and-language task learning framework considers the different roles of multi-modal information. Vision captures the information of objects and their interactions, which reflects potential robotic tasks in the scene. Language provides context. It helps to narrow down or determine the target task over the task space inferred from vision. We apply this insight and propose to pretrain the visual encoder on scene graph parsing to help learn generic features that encode attributes and relationships for objects. A scene graph G consists of:

- a set of bounding boxes $B = \{b_1, \cdots, b_k\}, b_i \in \mathbb{R}^4$;
- a set of corresponding attributes $A = \{a_1, \cdots, a_k\}$ where the tuple a_i include object category o_i, affordance d_i and general attribute t_i; and
- a set of relationships $R = \{r_1, \cdots, r_j\}$ between bounding boxes.

Using the Stacked Motif Network [47], factor the probability of constructing the graph G given the RGB image I as

$$P(G \,|\, I) = P(B \,|\, I) P(A \,|\, B, I) P(R \,|\, A, B, I) \tag{1}$$

The bounding box generation model $P(B \,|\, I)$ is based on the Faster R-CNN object detection model [31]. It is pretrained as described in SectionV-A.2 and keeps parameters frozen during the training stage for attribute and relation prediction. The attribute prediction model $P(A \,|\, B, I)$ involves encoding contextual representation for each bounding box and decoding corresponding attribute information. Predicted bounding boxes B will be ordered from left to right by the central x-coordinate in the image and fed into a biLSTM for learning contextual representation $C = \{c_1, \cdots, c_k\}$:

$$C = \text{biLSTM}([f_i; W_l l_i]_{i=1,\cdots,n}) \tag{2}$$

where C contains the hidden states of the final LSTM layer, W_l is the projection matrix to predicted class distribution l_i and f_i represents the regional feature. A separate LSTM decodes object category $\hat{o}_i$, affordance $\hat{d}_i$ and attribute $\hat{t}_i$ based on the encoded context C:

$$h_i = \text{LSTM}_i([c_i; \hat{o}_{i-1}]) \tag{3}$$

$$\hat{o}_i = \text{argmax}(W_o h_i) \tag{4}$$

where $\hat{o}_i$ represents the prediction of the object category. Affordance and attribute prediction follows the same design. The relation prediction model $P(R \mid A, B, I)$ also involves encoding and decoding stages. A biLSTM encodes representation D for each bounding box from object context C, category $\hat{o}_i$, affordance $\hat{d}_i$ and attribute $\hat{t}_i$. The relation r_{ij} between the i-th and j-th bounding box is:

$$r_{ij} = \text{softmax}(W_r g_{ij}) \tag{5}$$

where $g_{ij} = (W_h d_i) \circ (W_t d_j) \circ f_{ij}$, f_{ij} is the feature vector for the union boxes and W_r, W_h, W_t are projection matrices. See public code for more details[1].

C. Pretraining on Semantic Textual Similarity

For ambiguous natural language, previous methods mainly consider issues of missing partial information, anaphoric reference, and high-level verb. Such natural language descriptions still contain partial task-related information indicating what objects to be manipulated and what actions to be performed. In this work, we consider implicit requests that convey the terminal state but not the actions. As shown in Fig. 2, we propose to admit explicit human instructions or two types, and implicit human intents, the last of which might require incorporating environmental information for full understanding. Explicit human instructions are divided into complete and incomplete instructions. The complete instruction describes ordered sub-steps with full actions and objects, while the incomplete one has partial information. There are four main reasons for missing information: missing object, missing action, high-level verb and anaphoric reference. Though composed with different low-level words, explicit instruction and implicit intent have the same high-dimensional semantic meaning in the robotic task domain. Semantic textual similarity tackles determining how similar two texts' semantic meanings are. We apply this insight and propose to pretrain the linguistic encoder on semantic textual similarity between explicit human instructions and implicit human intents.

The Siamese network is a network consists of twin networks which take different inputs but are coupled by a common objective function. Following the design of Sentence-BERT [48], we employ BERT followed by a pooling layer as the language modeling network to learn separate embeddings for each sentence in the pair. With two embeddings, we compute cosine similarity between them and use the mean squared error as the objective function:

$$\mathcal{L}_{\text{sts}}(s_{ex}, s_{im}; \epsilon) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{s_{ex} \cdot s_{im}}{\max(\|s_{ex}\|_2 \cdot \|s_{im}\|_2, \epsilon)}\right)^2 \tag{6}$$

where s_{ex} and s_{im} are embeddings for explicit instruction and implicit intent, n is the number of sentence pairs and ϵ is set to $1e-8$. The implementation is with the code[1].

D. Instruction Following Framework

Figure 3 depicts the flow of the proposed hybrid, modular instruction following framework whose four components are: Perception, Goal Learning, Task Planning, and Execution. The hybrid design leverages the strengths of semantic feature learning from deep neural networks and of symbolic manipulation from symbolic planners. The **Perception** module interprets the visual scene and its contents. The **Goal Learning** module uses visuo-linguistic input to output symbolic goal specifications for the **Task Planning** module. The **Task Planning** module generates a sequence of low-level actions. The **Execution** module performs planned actions based on operational information obtained from the **Perception** module. Modular design benefits include: easy analysis of failure modes; easy component replacement with better methods; and easy augmentation with other components, such as lifelong learning; to make the entire framework more complete and powerful.

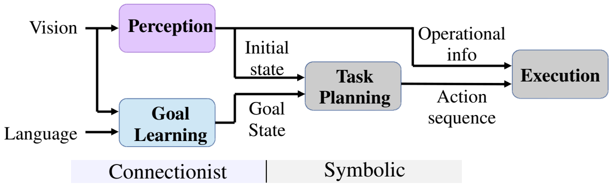


Fig. 3. Human instruction following framework, designed for performing manipulation tasks by following human instructions. It takes vision and language as input to decode what sequence of actions to execute.

This work uses Mask R-CNN [49] as the **Perception** module to detect objects and their category segmentation masks. Categorical information is detected and corresponding affordances and attributes are retrieved from a knowledge base to build the initial state for PDDL. The **Goal Learning** module uses the vision-and-language learning network described in §IV-A for goal state prediction. The **Task Planner** module is a PDDL planner that outputs a primitive action sequence from the detected initial and goal states. In addition to the action plans, robotic **Execution** requires operational information, which comes from **Perception** module detection masks.

V. Vision-and-Language Model Benchmark

This section introduces three training datasets with corresponding training policies for learning three tasks. The evaluation metric is then discussed along with benchmarking to evaluate each component in the vision-and-language goal learning framework, and the two proposed pretraining tasks. For model and training details, see the public code [1].

A. Datasets

The three datasets are symbolic goal learning, scene graph parsing, and semantic textual similarity datasets. The dataset, its generation code and training code is public[1].

1) Symbolic Goal Learning Dataset: For learning symbolic goal representation from vision and language, we created a dataset containing 32,070 images paired with natural language, which could be either an explicit instruction or implicit intent. It covers five daily activities: picking and placing, object delivery, cutting, cooking, and cleaning. We employ the simulator AI2THOR to generate image and sentence pairs, which is automatically annotated with PDDL

TABLE I
BENCHMARKING VISION-AND-LANGUAGE SYMBOLIC GOAL LEARNING

Model	**RGL Accuracy(%)**	Speed (fps)
Grid-LSTM-Concat	77.24	476.19
Grid-LSTM-Add	80.29	476.19
Grid-LSTM-Mul	72.47	476.19
Region-LSTM-Concat	81.35	9.14
Grid-BERT-Concat	89.02	230.95
Grid-BERT-Add	85.69	226.41
Grid-BERT-TDAtt-v1 [29]	92.61	215.98
Grid-BERT-TDAtt-v2 [29]	93.54	210.71
Grid-BERT-CoAtt [30]	92.30	120.70
Grid-BERT-Concat (SGP)	93.55	230.95
Grid-BERT-Concat (STS)	90.96	230.95
Grid-BERT-Concat (SGP+STS)	94.54	230.95

goal states. Besides imperfect natural lanugage, we also include imperfect vision where one or both objects involved in the task are not in the image. With such input, the vision-and-language symbolic goal learning network is expected to predict the missing object to be "unknown" in the goal state output.

2) Scene Graph Parsing Dataset: Based on the Symbolic Goal Learning Dataset, the Scene Graph Parsing dataset was created via annotating data with object category, affordance, attribute and their relationships. It covers 32 categories, 4 affordances, 5 attributes and 4 relationships in total.

3) Semantic Textual Similarity Dataset: The created Semantic Textual Similarity dataset is for learning the similar semantic meaning between explicit human instructions and implicit human intents for robotic tasks. The same five daily activities are in the dataset: picking and placing, object delivery, cutting, cleaning, and cooking. It contains 90,000 pairs of explicit instruction and implicit intent, generated from a manually created list of templates. For the purpose of improving the diversity, sentences are automatically paraphrased by Parrot [50] during the generation process. Ranking sentence similarity uses 5 for similar sentences (same action, subject, and object) and 0 for dissimilar sentences (different actions). Partial scores for when the actions agree are: 3.3 when only one of subject or object agree, and 1.7 when neither agree. The scores are automatically generated from known goal classes.

B. Evaluation Metric

Evaluating the prediction accuracy of the vision-and-language models should test for symbolic matching to the PDDL goal state entities, which are action, subject, and object. We propose the Robotic Goal Learning (RGL) accuracy score:

$$\text{RGL} = \delta(\hat{a}, a, \hat{s}, s, \hat{o}, o) \equiv \delta(\hat{a}, a) \cdot \delta(\hat{s}, s) \cdot \delta(\hat{o}, o). \quad (7)$$

where $\delta(\cdot)$ is the Kronecker delta function, $\hat{a}$ and a are predicted and ground-truth action label, and the same holds for the subject s and object o labels.

C. Benchmarking Vision-and-Language Goal Learning

The test configurations in Table I permit comparison of different implementation choices regarding the core components, plus the effect of attention models and pre-training. The baseline visual and and language encoders will employ Grid features and LSTM, respectively. Baseline pretraining is image classification and masked language modeling. For reference, the LSTM-only and BERT-only models perform at 67.07% and 55.91%, which shows the value of adding visual information.

Regarding the fusion component for the baseline model, three simple strategies were tested: concatenation (concat), addition (add) and multiplication (mul). The best of the three tested for the baseline LSTM implementation is addition. Switching from Grid to Region features, with concatenation, leads to a small boost in performance of 4.11% but a 50x drop in processing rate. Grid feature encoders show better trade-off between prediction accuracy and inference speed if real-time is important. Considering a change in the language encoder to BERT, there is a boost in performance to 89.02% and 85.69%, for fusion by concatenation and addition, respectively. The 2x drop in timing is not serious, thus BERT+concat would be the more sensible option to use. It provides a 11.78% boost in performance and still operates beyond frame-rate.

Regarding attention versus pre-training, the attention model implemented were Top-Down Attention (TDAtt) [29] and Co-Attention (CoAtt) [30]. There are two top-down attention variants: directly feeding the fused embedding for classification, and concatenating the fused embedding with extracted visual and linguistic features. Pre-training tests involved Scene Graph Parsing (SGP) and Semantic Textual Similarity (STS) tasks. Independent SGP+STS pre-training of the two encoders provided the best boost over the baseline pretraining methods without affecting processing time. While attention models did improve the outcomes, they are known to require customized training policies to operate well [51].

VI. MANIPULATION EXPERIMENTS

A. Experimental Setup

Manipulation experiments in AI2THOR evaluate the robustness and generalization of the proposed instruction following framework to novel scenarios. Five different daily activities are conducted, which include Picking and Placing, Object Delivery, Cutting, Cleaning and Cooking. There are four different levels of scenarios for each task. Easy scenario only contains involved objects in the scene. Medium scenario incorporates irrelevant objects. The first hard scenario further includes multiple candidates while the second hard scenario misses partial or all objects required to perform the task. Due to missing objects in the scene, task planning is not expected to find valid solutions and execution is also not required for the second hard case. There are 10 scenarios for each level and either novel instruction or intent will be paired with the image. The model, which consists of grid feature encoder, BERT and concatenation and is pretrained on both tasks, are employed.

TABLE II

RESULTS OF MANIPULATION EXPERIMENTS IN AI2THOR. (P: PERCEPTION; GL: GOAL LEARNING; TP: TASK PLANNING; E: EXECUTION)

	Pick_n_Place				**Object Delivery**				**Cut**				**Cook**				**Clean**				**Average (%)**			
(%)	P	GL	TP	E	P	GL	TP	E	P	GL	TP	E	P	GL	TP	E	P	GL	TP	E	P	GL	TP	E
VSR	96.7	80.0	80.0	80.0	96.7	86.7	83.3	83.3	100.0	86.7	86.7	86.7	96.7	73.3	70.0	70.0	96.7	90.0	90.0	83.3	97.3	83.3	82.0	80.7
ISR	90.0	70.0	100.0	100.0	100.0	60.0	100.0	100.0	100.0	80.0	100.0	100.0	90.0	50.0	100.0	100.0	100.0	90.0	100.0	100.0	96.0	70.0	100.0	100.0
SR	95.0	77.5	85.0	80.0	97.5	80.0	87.5	83.3	100.0	85.0	90.0	86.7	95.0	67.5	77.5	70.0	97.5	90.0	92.5	83.3	**97.0**	**80.0**	**86.5**	**85.5**

TABLE III

COMPARISON OF MANIPULATION EXPERIMENTS TO EXISTING METHODS

	V2A [5]		ALFRED [6]		Mod [10]		HiTUT [27]		SGL		
(%)	S	U	S	U	S	U	S	U	S_GL	U_GL	U_E
SR	44.7	40.2	70.3	49.9	71.9	63.0	87.7	80.6	94.5	80.0	85.5
VSR	23.8	6.4	70.3	49.9	71.9	63.0	87.7	80.6	96.7	83.3	80.7
ISR	65.0	79.9	-	-	-	-	-	-	88.8	70.0	100.0

B. Manipulation Metrics

To evaluate each module in the instruction following framework, each manipulation experiment trial is considered as successful if it satifies four conditions. For **Perception**, all involved objects are required to be correctly detected, which constructs the initial state for PDDL. For **Goal Learning**, PDDL goal state should be correctly predicted. For **Task Planning**, generated action sequence is composed of correct ordered actions. Given that AI2THOR does not support physical modeling of robot-object interaction, **Execution** evaluation requires the Intersection-of-Union (IoU) of detected and ground-truth masks for objects to be over the 0.5 threshold. Based on [5], Valid Success Rate (VSR) and Invalid Success Rate (ISR) are employed for easy, medium and the first hard, and the second hard scenarios, respectively. VSR evaluates tasks with valid solutions while ISR evaluates ones where there is no valid solution. Success Rate (SR) is used to take the average over all valid and invalid tasks.

C. Outcomes and Analysis for Manipulation Experiments

Results of manipulation experiments are collected in the Table II, with a more detailed breakdown in the public repository[1].For the easy, medium and the first hard scenarios which have valid solutions (VSR row), results show that the 82.0% success rate of task planning is close to the product of 97.3% for perception and 83.3% for goal learning. Closeness indicates approximate independence of the perception and goal learning modules. The average success rate (SR row) includes all four scenarios. The 97.0% success rate for perception and 1% performance drop from task planning to execution shows that the existing perception module works well. The goal learning module is the main performance bottleneck. Further study into training methods and network design for symbolic goal learning is needed to improve the performance of the SGL human instruction following framework.

For the second hard scenario which has no valid solutions (ISR row), the 100% ISR of task planning is higher than the product of perception and goal learning. Though the goal learning module fails to predict a missing object as *unknown*, the Perception module does not detect it (Perception has a low false positive rate). With an incomplete initial representation the symbolic task planner correctly outputs *no solution*, which shows the value of the symbolic component. With a symbolic module computing the primitive actions sequence, the system can decide whether the task is achievable. Connectionist approaches instead predict incorrect sequential actions since illogical outputs are not recognized.

D. Outcomes and Analysis for Comparison

Since the proposed method focuses on manipulation tasks which do not include navigation, we collect experimental results of manipulation sub-tasks for existing connectionist approaches in Table III for performing approximate comparative analysis. Seen and Unseen scenarios are denoted as S and U. Table III shows that SGL outperforms all methods with an 85.5% Unseen task success rate (SR), thereby supporting the hypothesized benefits of the proposed hybrid, modular instruction following framework. Since several of the baseline methods cannot handle invalid requests, consider only the valid requests (VSR row). SGL Unseen performance matches that of HiTUT. Both are the top performing methods. HiTUT has self-monitoring and backtracking; the robotic agent may try again if it observes a failure. As a simple, single-pass approach, SGL matches a more complex, multi-pass approach.

Analyzing the Seen performance for valid requests, SGL has the highest VSR. While this means SGL has a relatively high Seen/Unseen performance gap, it also means that removing the gap would result in much higher task understanding and execution performance (by 9% relative to the next best Seen performer, HiTUT). The higher upper bound suggests that using joint visuo-linguistic features to predict goal state predicate labels (action, object, subject) may be preferred to the output types in the baseline methods. In short, a language-to-language translation process may be better than a language-to-action conversion process.

The SGL VSR drop from 96.7% to 83.3% is caused by two factors. First, the training dataset doesn't include images with multiple candidate objects, which causes domain shift. Grid-based feature encoding may have trouble localizing the correct regions of interest. Secondly, the main issue is with the *cook* tasks (see VSR row for Cook columns). One reason is the appearance of the microwave and stove burner in the same image leading to confusion about which device to cook with. For cooking tasks with the microwave as *object*, the network mispredicted to use stoveburner by 44.4%. The other reason could be the imbalance training data between unknown and microwave cases such that *microwave* was predicted into *unknown*. The performance drop for the ISR case is less relevant because the symbolic reasoning pipeline correctly rejects these cases. A benefit of this property is that the goal learning module can focus on boosting valid request reasoning over invalid request reasoning. Comparing the SGL VSR performance drop (13.4%) to sequential action predictors V2A (17.4%) and ALFRED (20.4%), indicates that symbolic goal learning is less sensitive to Unseen scenarios than sequential action prediction.

VII. Conclusion

To address human instruction following with diverse natural language inputs, we propose to compensate for implicit or missing information via vision and present a hybrid, modular framework consisting of symbolic goal learning via deep netural networks and task planning via symbolic planners. We propose a vision-and-language goal learning framework, which consists of the visual encoder, linguistic encoder, multi-modal fusion and classification. Benchmarking compares the impacts of different techniques for the different components. For learning generic features and boosting the performance when fine-tuning on specific tasks, we propose to separately pretrain the visual and linguistic encoder on scene graph parsing and semantic textual similarity tasks. We show the effectiveness of the two pretraining tasks on a model with visual grid features, BERT, and fusion by concatenation, Evaluation of the instruction following framework in the AI2THOR simulator shows robustness to novel scenarios. The hybrid framework combines the strength of semantic feature learning from deep neural networks and capability of rejecting invalid tasks from symbolic planners. The modular design of the framework enables easy analysis of the cause of failure, simple replacement of each component, and incorporation of more modules. For future work, we will work on incorporating modules such as feedback mechanism to deal with dynamic environments and domain adaptation for real world application.

References

[1] M. Tenorth and M. Beetz, "Knowrob: A knowledge processing infrastructure for cognition-enabled robots," *IJRR*, vol. 32, no. 5, pp. 566–590, 2013.

[2] A. Antunes, L. Jamone, G. Saponaro, A. Bernardino, and R. Ventura, "From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning," in *ICRA*, 2016, pp. 5449–5454.

[3] P. Pramanick, C. Sarkar, and I. Bhattacharya, "Your instruction may be crisp, but not clear to me!" in *RO-MAN*, 2019, pp. 1–8.

[4] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me dave: Context-sensitive grounding of natural language to manipulation instructions," *IJRR*, vol. 35, no. 1-3, pp. 281–300, 2016.

[5] M. Nazarczuk and K. Mikolajczyk, "V2a-vision to action: Learning robotic arm actions based on vision and language," in *ACCV*, 2020.

[6] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *CVPR*, 2020, pp. 10 740–10 749.

[7] R. Marc'Aurelio, C. Sumit, A. Michael, and Z. Wojciech, "Sequence level training with recurrent neural networks," in *ICLR*, 2016.

[8] K. M. Dipendra, B. Andrew, B. Valts, N. Eyvind, S. Max, and A. Yoav, "Mapping instructions to actions in 3d environments with visual goal prediction," in *EMNLP*, 2018, pp. 2667–2678.

[9] K. P. Singh, S. Bhambri, B. Kim, R. Mottaghi, and J. Choi, "Factorizing perception and policy for interactive instruction following," in *ICCV*, 2021, pp. 1888–1897.

[10] R. Corona, D. Fried, C. Devin, D. Klein, and T. Darrell, "Modular networks for compositional instruction following," *arXiv preprint arXiv:2010.12764*, 2020.

[11] S. Zhou, P. Yin, and G. Neubig, "Hierarchical control of situated agents through natural language," *arXiv preprint arXiv:2109.08214*, 2021.

[12] F.-J. Chu, R. Xu, L. Seguin, and P. A. Vela, "Toward affordance detection and ranking on novel objects for real-world robotic manipulation," *RA-L*, vol. 4, no. 4, pp. 4070–4077, 2019.

[13] F.-J. Chu, R. Xu, and P. A. Vela, "Recognizing object affordances to support scene reasoning for manipulation tasks," *arXiv preprint arXiv:1909.05770*, 2020.

[14] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *ICCV*, 2019, pp. 7464–7473.

[15] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *NIPS*, vol. 32, 2019.

[16] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[17] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[18] A. Mogadala, M. Kalimuthu, and D. Klakow, "Trends in integration of vision and language research: A survey of tasks, datasets, and methods," *JAIR*, vol. 71, pp. 1183–1317, 2021.

[19] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language: A survey," 2020.

[20] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively picking real-world objects with unconstrained spoken language instructions," in *ICRA*, 2018, pp. 3774–3781.

[21] M. Shridhar, D. Mittal, and D. Hsu, "Ingress: Interactive visual grounding of referring expressions," *IJRR*, vol. 39, no. 2-3, pp. 217–232, 2020.

[22] S. Guadarrama, L. Riano, D. Golland, D. Go, Y. Jia, D. Klein, P. Abbeel, T. Darrell *et al.*, "Grounding spatial relations for human-robot interaction," in *IROS*, pp. 1640–1647.

[23] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *AAAI*, vol. 25, no. 1, 2011, pp. 1507–1514.

[24] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "From structured english to robot motion," in *IROS*, 2007, pp. 2717–2722.

[25] C. Finucane, G. Jing, and H. Kress-Gazit, "Ltlmop: Experimenting with language, temporal logic and robot control," in *IROS*. IEEE, 2010, pp. 1988–1993.

[26] P. Lindes, A. Mininger, J. R. Kirk, and J. E. Laird, "Grounding language for interactive task learning," in *Proceedings of the First Workshop on Language Grounding for Robotics*, 2017, pp. 1–9.

[27] Y. Zhang and J. Chai, "Hierarchical task learning from language instructions with unified transformers and self-monitoring," *arXiv preprint arXiv:2106.03427*, 2021.

[28] V. Blukis, C. Paxton, D. Fox, A. Garg, and Y. Artzi, "A persistent spatial semantic representation for high-level natural language instruction execution," in *CoRL*, 2022, pp. 706–717.

[29] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018, pp. 6077–6086.

[30] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *CVPR*, 2019, pp. 6281–6290.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NIPS*, vol. 28, 2015.

[32] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen, "In defense of grid features for visual question answering," in *CVPR*, 2020, pp. 10 267–10 276.

[33] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv preprint arXiv:2004.00849*, 2020.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[35] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014, pp. 647–655.

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.