Learning Generalizable Visual Task Through Interaction

Weiwei Gu
Arizona State University
weiweigu@asu.edu

Anant Sah
Arizona State University
asah4@asu.edu

Nakul Gopalan
Arizona State University
ngopalan6@asu.edu

Abstract

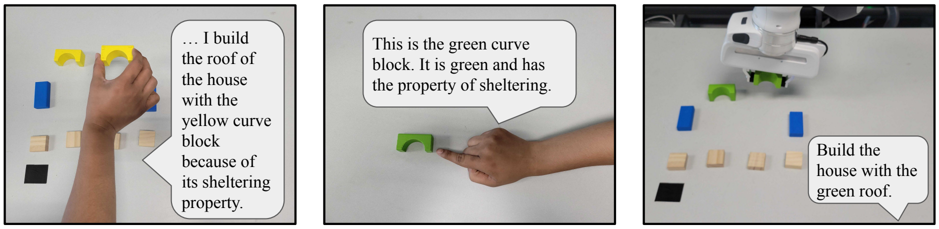
We present a framework for robots to learn novel visual concepts and visual tasks via in-situ linguistic interactions with human users. Previous approaches in computer vision have either used large pre-trained visual models to infer novel objects zero-shot, or added novel concepts along with their attributes and representations to a concept hierarchy. We extend the approaches that focus on learning visual concept hierarchies and take this ability one step further to demonstrate novel task solving on robots along with the learned visual concepts. To enable a visual concept learner to solve robotics tasks one-shot, we developed two distinct techniques. Firstly, we propose a novel approach, Hi-Viscont(HIerarchical VISual CONcept learner for Task), which augments information of a novel concept, that is being taught, to its parent nodes within a concept hierarchy. This information propagation allows all concepts in a hierarchy to update as novel concepts are taught in a continual learning setting. Secondly, we represent a visual task as a scene graph with language annotations, allowing us to create novel permutations of a demonstrated task zero-shot in-situ. We compared Hi-Viscont with the baseline model (FALCON [19]) on visual question answering(VQA) in three domains. While being comparable to the baseline model on leaf level concepts, Hi-Viscont achieves an improvement of over 9% on non-leaf concepts on average. Additionally, we provide a demonstration where a human user teaches the robot visual tasks and concepts interactively. With these results we demonstrate the ability of our model to learn tasks and concepts in a continual learning setting on the robot.

1 Introduction

Robots in a household will encounter novel objects and tasks all the time. For example, a robot might need to use a novel vegetable peeler to peel potatoes even though it has never seen, let alone used such a peeler before. Our work focuses on teaching robots novel concepts and tasks one-shot via human-robot interactions, which include demonstrations and linguistic explanations. We then want the robot to generalize to a similar but unseen visual task. A robotic system that can learn generalizable tasks and concepts from few natural interactions from a human-teacher would represent a large leap for robotics applications in everyday settings. In this work we aim to take a step in the direction of generalizable interactive learning as demonstrated Fig. 1.

Previously, large image and language models have been extended to robotics to manipulate novel objects, and create visual scenes [20, 2]. These methods recognize novel objects by using their underlying large language and visual models to extract task-relevant knowledge. However, they are not capable of learning to create a novel visual scene from an in-situ interaction with a human user. There is also significant work in few-shot learning of visual concepts in computer vision [19, 21, 27, 23, 29, 25], albeit without extensions to robotics domains. These approaches focus on learning novel concepts for image classification, but ignore the fact that the novel concepts also bring new information to update our understanding of concepts already known to the robot. The reverse path of

NeurIPS 2023 Workshop on Generalization in Planning (GenPlan 2023).


… I build the roof of the house with the yellow curve block because of its sheltering property.
This is the green curve block. It is green and has the property of sheltering.
Build the house with the green roof.

(a) (b) (c)

Figure 1: This figure demonstrates how Hi-Viscont learns from users interactively. (a) First the user demonstrates a structure, say a "house," with its sub-components such as its "roof" and the concepts used to make the "roof" such as a "yellow curve block". (b) The user then teaches a novel concept such as a "green curve block" and describes its properties. (c) The user can now ask the robot to create a new structure ("house with green roof") zero-shot with the taught component without explicitly asking for the object of interest.

knowledge propagation, that is, from novel concepts to previously known concepts is equivalently important in performing tasks in the real-life scenarios, especially when the agent has little knowledge of the world and needs to continually add information to known concepts.

In this work, we propose a novel framework, Hi-Viscont, that enables robots to learn visual tasks and visual concepts from natural interactions with a human user. We learn the task type and concepts from users one shot, and then generalize to tasks within the task type zero-shot. We do this by connecting our insights on *one-shot visual concept learning* and the use of *scene graphs*. The robot learns the structure of a visual task by converting linguistic interactions with a human user into a contextualized scene graph with language annotations. Moreover, Hi-Viscont updates parental concepts of the novel concept being taught. Such updates allow us to generalize the use of the novel concepts in to solve novel tasks.

The contribution of this work is listed as below:

1. We present visual concept results on VQA tasks that are comparable to the state-of-the-art FALCON model. More specifically, Hi-Viscont improves on FALCON on all non-leaf concepts across all domains with significance.
2. We enable the robot agent to learn a visual task from in-situ interactions with a scene graph, allowing zero shot generalization to an unseen task of the same type, as demonstrated in Fig 1.

2 Related Work

Language conditioned manipulation. Significant work exists in learning concepts and tasks for robots in interactive settings even with the use of dialog [4, 18]. Our work differs from previous works as it is attempting to learn visual concepts for manipulation one-shot, while updating other known concepts to improve generalization. Moreover, our approach is completely differentiable and can start with zero known concepts, which is important for a continual learning setup. Previous work has focused on language conditioned manipulation [20, 15, 3, 2]. Shridhar et al. [20] computes a pick and place location conditioned on linguistic and visual inputs. Liu et al. [15] focuses on semantic arrangement on unseen objects. Ahn et al. [1], Brohan et al. [3] train on large scale of linguistic and visual data and can perform real-life robotic task based on language instructions, however our work is focused on interactive teaching of tasks and concepts and not on emergent behaviors from large models. Daruna et al. [6] learn a representation of a knowledge graph by predicting directed relations between objects allowing a robot to predict object locations. To the best of the author's knowledge ours is the first paper that learns concepts and tasks one shot to generalize to novel task scenarios on a robot making our contributions significant compared to other related works.

Visual reasoning and visual concept learning. Our work is related to visual concept learning [19, 16, 30, 7, 14] and visual reasoning [17, 8, 10, 9]. To perform the visual reasoning task, traditional methods [17, 8, 10, 9] decompose the visual reasoning task into visual feature extraction and reasoning by parsing the queries into executable neuro-symbolic programs. On top of that, many

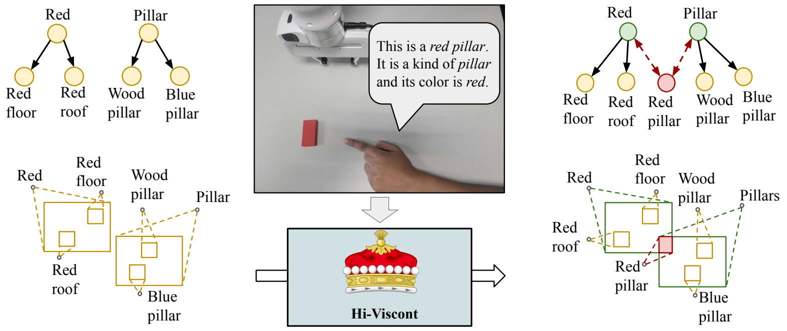


Figure 2: We demonstrate the updates to the box embedding space and the parent concepts when a novel concept is taught to our robot using Hi-Viscont. Existing approaches only edit the leaf nodes as those represent novel concepts.

concept learning frameworks [19, 16, 30, 7, 14] learn the representation of concepts by aligning concepts onto objects in the visual scene. As far as we know, FALCON[19] is the most similar work to our work in this line of research. However, when introducing a new concept, our work continually updates the representation of all related concepts, whereas Mei et al. [19] does not, which makes it ill-suited for continual learning settings. Our work is also related to the area of few-shot learning [21, 25, 27], which learns to recognize new objects or classes from only a few examples but does not represent a concept hierarchy which is useful in robotics settings.

Scene graph. Scene graphs are structural representations of all objects and their relationships within an image. The scene graph representation [5] of images is widely used in the visual domains for various tasks, such as image retrieval [10], image generation [11], or question answering [24]. This form of representation has also used in the robotics domains for long-horizon manipulation [32].

3 Methods

We first present the baseline FALCON model and then introduce our Hi-Viscont model. We based our model on concept learners as they can be taught concepts few shot, and they can reason over the attributes of chosen (and their parent) concept classes. FALCON is the SOTA concept learner which learns novel concepts one-shot.

3.1 FALCON

Mei et al. [19] developed FALCON, a meta-learning framework for one-shot concept learning in visual domains. FALCON learns a new visual concept with one or a few examples, and uses the learned concept to answer visual reasoning questions on unseen images. There are three components for the FALCON model: a visual feature extractor that extracts the object-centric features for the input image, a graph neural network (GNN) based concept learner, and a neuro-symbolic program executor that executes the input neuro-symbolic program.

Natural language sentences describing objects and their queries are represented as structured neuro-symbolic programs. FALCON learns novel concepts by interpreting the images presented and the relationships between known concepts and the unknown concept being learned using a neuro-symbolic program. After learning, the model performs reasoning over questions, by converting these questions into neuro-symbolic programs that are executable by the model.

FALCON uses a pre-trained ResNet-34 as visual feature extractor. The visual feature extractor computes a feature for each object in a scene seperately, which can then be used for downstream visual reasoning. FALCON uses a box embedding[26] to represent concepts and their object visual features.

Finally, the concept learning module of FALCON is composed of two separate Graph Neural Networks(GNNs), the relational GNN and the Example GNN. To predict a embedding for a novel concept c, FALCON first samples random prior embedding as the representation for c from a Dirichlet distribution. Then, FALCON updates the embedding of c by computing messages from parent nodes based on their factor weights or relationship and also computing a message from the visual feature (represented as a node within the Example GNN) for the concept being learned. This computed representation for a novel concept c can then be used for VQA tasks.

FALCON has two major issues for interactive task learning on the robot. Firstly, the model lacks scene information to solve tasks. We address this in our work. Secondly, FALCON assumes concepts are learned perfectly and do not need to be updated as the model learns more concepts. For example, when we teach the model the concept of "container" with an image of a "cup," FALCON cannot update the features of the "container" concept when the concept of "bowl" is taught as a child to the "container" concepts. This might mean that FALCON assumes that "containers" have handles which is untrue.

3.2 Hi-Viscont

We present our concept net model, Hi-Viscont (HIerarchical VISual CONcept learner for Task), which actively updates the related known concepts when we introduce the novel concept to improve upon FALCON's generalization capabilities. We adopted several modules from the framework of FALCON, including the visual feature extractor, the neuro-symbolic program executor, the box embedding space, and the novel concept learner. Moreover, we introduce an additional GNN module, Ancestor Relational GNN (ARGNN), that updates the related known concepts as a novel concept is introduced. ARGNN predicts a new embedding for the related known ancestor concepts to the novel concept. To do this update we compute a message from the visual feature of novel concept's instance to the embedding of the related nodes using the relations between the parent concepts and the novel concept.

When a novel concept c is inserted to Hi-Viscont, the extracted visual feature o_c of concept c and its relations with known concepts R_c are fed to Hi-Viscont as input. Each relation $rel = (c', c, r)$, where c' denotes the related concept, and r describes its relationship with c. We compute embedding e_c for novel concept c using the same method as FALCON. Then, using the additional ARGNN, we predict a new embedding for each related concept c' by computing a message from the visual feature o_c to the current embedding of the related concept $e^0_{c'}$ using the same relationship rel. The formula for this update is denoted as follows:

$$e^1_{c'} = \text{ARGNN}(o_c, rel, e^0_{c'})$$

The resulted embedding $e^1_{c'}$ will be used as the representation for concept c' for future task or updates.

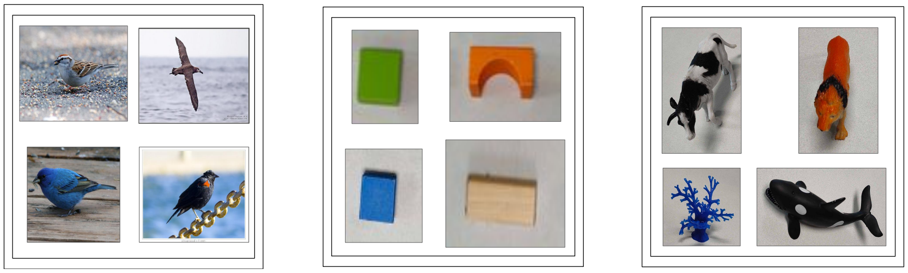
To provide gradient flow to train ARGNN, we extended the concept learning task proposed by FALCON by adding validation questions for each related concept, that is when a new concept is added all concepts in the concept net are tested for accuracy over the novel concept. For example, from our previous discussion the newly inserted "bowl" concept's object instance is checked with the "container" parent to see if the presented "bowl" also tests as a "container." A more detailed description of our training pipeline and methodology can be found in the appendix.

While FALCON was evaluated solely on the newly inserted concept, we evaluate all concepts (leaf and parent nodes) of our model on unseen images. Such an evaluation ensures consistency between parent and child concepts which is a necessity in continual learning settings. This evaluation mechanism allows us to evaluate the quality for the embedding of all concepts in the resulted knowledge graph, which is closer to how these knowledge are used in the real world setting.

3.3 Learning Visual Task via Scene Graph

To learn a visual task from a single in-situ interaction with human user, we first convert the user's demonstration (Fig. 1.a) into an initial scene graph. Each node of the initial scene graph corresponds to an object that the user placed, and it contains the bounding box information of the object and the user's linguistic description of the object. For each node of the initial scene graph, we also store the positional relations w.r.t. other nodes, to allow for object placements when reconstructing the scene. We mark a fixed location with black tape on the table, which serves as the origin and is treated as the zeroth object. All other objects placed by the user will be to the top right of the origin.

Based on the initial scene graph and the user's linguistic request for the desired variant of the visual scene, we infer a goal scene graph modelled as a node-wise classification task. Since the variant of



(a) CUB dataset

(b) House domain

(c) Zoo domain

Figure 3: Sample images from the three domains we are targetting in this work.

the visual task from the user request share the same structure as the demonstration, the goal scene graph is to have the same number of nodes as the initial scene graph.We take the user's description of the corresponding node of the initial scene graph t_i and the user's linguistic request of the variant of the structure q as inputs, and perform a two-step inference: First we decide if the node in the goal graph is different from the one in demonstration; Subsequently if the node is different we decide which object satisfies the node location with another classification step.

To decide whether the concept of a node within the scene has changed given the user's description of the node and the user's current request q we perform a binary classification at each node. The result of this classification decides if we are changing a node's concept or not. We use a pretrained $\text{BERT}_{\text{base}}$ model to encode the context request pair which is then fed into a multi-layer perceptron (MLP) with a Cross-Entropy loss. The second step of the inference extracts the related concepts from the context if the node's concept needs to be changed as per the request. We convert the concept extraction problem into a classification problem by providing concept candidates as a part of the input again with BERT model and an MLP with a Cross-Entropy loss. The related concepts of each node is fed as input for the concept net model to decide the object to pick, and the positional relations with other nodes are used to compute the placement location. The robot will reconstruct the scene following the order of the nodes. For each node, the robot picks the object according to the concept net model. The placement location of each object is at a fixed distance to the direction indicated by the relation with its closest neighbor that is placed. Pairing the concept net model with scene graph, the robot is able to learn the placement of a scene in one single demonstration and perform variants of the scene without demonstration.

3.4 Robotics Setup

We integrate our visual task learning and concept learning model with a Franka Emika Resarch 3 arm(FR3). This pipeline allows us to show the generalizability with which Hi-Viscont learns visual concepts when compared to FALCON [19] in learning and solving novel tasks. To set this demonstration up we use a Franka Emika Research 3 arm (FR3), two calibrated realsene D435 depth cameras, and a mono-colored table to allow for background subtraction. We use the SAM(Segment Anything Model) [13] to separate the foreground and the background and get individual bounding boxes for each of the blocks on the table. For pick and place initially, we experimented with Transporter networks [31] but used a simpler Visuo-Motor Servoing mechanism for reliability. We expected users to maintain about an inch of space between each object in the scene to allow the robot to pick objects without collisions and for SAM to segment objects from the background accurately. In the process of picking and placing if an error is made the robot recovers autonomously. Once the object is grasped we then place the object into the Task scene, with the position calculated relatively with respect to the previously placed object nodes or zeroth origin object. This process is done iteratively until we have completed the whole scene graph.

4 Results

In this section, we present results on concept learning on the visual question answering task on three different domains. The experiment results demonstrates that our concept net model learns better representation for concepts than our baseline model, and is more robust for continual learning across all domains. Additionally, we present a demonstration of a human user teaching our system visual concepts and visual task through interactions.

4.1 Domains

We first present experimental results on VQA tasks for three domains: the **CUB-200-2011 dataset**, a **custom house-construction domain** with building blocks, and a **custom zoo domain** with terrestrial and aquatic animals.

CUB-200-2011 dataset [28] is a standard dataset to demonstrate visual concept learning. It contains $11,788$ images for 200 bird classes. Using the following bird taxonomy [22], we added the hypernyms of the bird classes and expanded the number of concepts to 365. Following the design of the dense graph propagation [12], the relation of each concept includes all of its ancestors.

The house construction domain includes 31 types of building block objects. Each object has 10 different images. To introduce relations between concepts, we additionally introduced 6 different concepts and 3 different affordances of objects. The dataset on the house construction domain includes 310 images and 40 concepts in total.

The zoo domain includes 28 different types of objects. Similar to the house domain, we took 10 images for each object, and introduce 6 general concepts to introduce a hierarchy for the concepts. The dataset on the zoo domain includes 280 images and 34 concepts in total.

We created the House-Construction and Zoo domains because they allow us to construct arbitrarily hard tasks with different types of objects that a robot can grasp. Following FALCON's data creation protocol, we procedurely generate training and testing examples for each domain. We generate descriptive sentences and questions based on the ground truth annotations of images and external knowledge, which is the relationship between concepts. For all the descriptive sentences and the questions, we also generate the corresponding neural-symbolic programs. We directly compared our concept net model with FALCON on all three domains.

To demonstrate that Hi-Viscont is better for continual learning, we compare these models with no pre-trained concepts. Results across the three datasets are obtained from different splits of concepts and image. Images used for testing are never seen by the model in any phase of training for both train concepts and test concepts. We present the standard deviation and the pairwise t-test result in the appendix.

4.2 VQA Tasks

We evaluate the question-answer pairs for all concepts for all the three domains on images that are not shown in the pre-train or the train phase. In Table 1, we present the results on the VQA task for test concepts. Our model, Hi-Viscont achieves comparable results to the baseline state-of-the-art FALCON model on test concepts in all three domains. Given that in a concept network there are fewer parent concepts than leaf concepts the performance of both models is comparable in such a general test case. However, when we split the concepts by their depth in the hierarchy, Hi-Viscont shines and achieves a significantly better performance with the parental nodes, which will be discussed by each domain separately.

Method	CUB-200-2011	House Construction	Zoo
Hi-Viscont	74.39±7.04	86.41±5.28	83.50±8.44
FALCON	73.40±5.77	87.17±4.17	85.12±6.64

Table 1: The average F1 score and standard deviation of Hi-Viscont and FALCON on the test concepts across all the three domains, each on five different splits of concepts.

CUB dataset: We present our results for concepts by their level in the taxonomy in Table 2. Hi-Viscont is better with significance for concepts in the level of Genera($p < 0.001$), Family($p = 0.001$), and Order($p = 0.001$) according to paired t-tests. Species are the leaf level concepts where the models again perform comparably as expected. This is because the leaf level updates of Hi-Viscont and FALCON do not differ significantly. As there is only one highest level ancestor for the Class with CUB there is no negative example for it in the dataset leading to similar performance by both models as the answer is always `True`.

Mtd.	Species	Genera	Family	Order	Class
HV	87.1±2.0	**90.4±0.6**	**90.7±1.7**	**92.0±0.8**	95.9±8.2
FCN	86.5±1.4	88.2±1.0	84.3±1.4	84.3±3.2	99.3±1.0

Table 2: The average F1 score and standard deviation of Hi-Viscont (HV) and FALCON (FCN) on the test set of the CUB dataset by the depth of concepts in the hierarchy on five different splits.

Method	Object	Color	Affordance
Hi-Viscont	88.46±1.58	**99.24±0.70**	**89.86±9.12**
FALCON	89.28±0.93	87.27±5.83	57.35±9.23

Table 3: The average F1 score and standard deviation of Hi-Viscont and FALCON on the test set of the our custom dataset of house construction domain by type of concepts on five different splits of concepts.

House construction domain: In this domain, the Color and Affordance concepts are non-leaf nodes in the hierarchy, whereas the object concepts are the leaf nodes. Following expectations, as demonstrated in Table 3, Hi-Viscont has a comparable performance to FALCON in the leaf node object concepts, while achieving significant improvements in both Color ($p = 0.005$) and Affordance (non-leaf) concepts ($p = 0.002$) according to the pairwise t-tests.

Zoo Domain In the zoo domain leaf concepts are not at equivalent depths from the root node forcing us to analyze the performance crudely w.r.t. leaf and non-leaf nodes in Table 4. Again Hi-Viscont achieves a comparable performance at leaf level concepts, but becomes significantly better than FALCON in the non-leaf concepts ($p = 0.001$).

4.3 Demo

In addition to the VQA experiment results, we also present a demonstration of a human user teaching visual tasks and visual concepts to the robot through in-situ interactions. The demonstration can be found in the associated webpage [1].

5 Limitations

There are three major limitations in our work. Firstly, although that we test Hi-Vicont on a large VQA dataset, we conducted our robotics demo of visual task learning only on the House domain, which contains a small number of objects. We would like to increase the task complexity and the number of objects available in the domain in the future. Secondly, the interaction between users and the robots is controlled and not completely open and dynamic. Even though a fixed template for their language is not required we ask the users to interact with the robot in specific ways. Finally, a thorough human subject study is needed to measure the system's capability of performing visual tasks.

Method	Leaf	Non-leaf
Hi-Viscont	87.93±3.40	**85.84±5.79**
FALCON	88.99±3.75	66.15±5.34

Table 4: The average F1 score and standard deviation of Hi-Viscont and FALCON on the test set of the zoo domain by type of concepts.

6 Conclusion

In conclusion, we present Hi-Viscont, a novel concept learning framework that actively updates the representations of known concepts which is useful in continual learning settings such as robotics. Hi-Viscont achieves comparable performance to SOTA FALCON model on VQA task across three domains in leaf level concepts, and is significantly better on non-leaf concepts. Our model also enables robots to learn a visual task from in-situ interactions by representing visual tasks with a scene graph, which allows zero-shot generalization to an unseen task of the same type.

[1] `https://sites.google.com/view/ivtl`

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[4] Joyce Y Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. Language to action: Towards interactive task learning with physical agents. In *IJCAI*, pages 2–9, 2018.

[5] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1–26, jan 2023. doi: 10.1109/tpami.2021.3137605. URL `https://doi.org/10.1109%2Ftpami.2021.3137605`.

[6] Angel Daruna, Weiyu Liu, Zsolt Kira, and Sonia Chernova. Robocse: Robot common sense embedding, 2019.

[7] Chi Han, Jiayuan Mao, Chuang Gan, Joshua B. Tenenbaum, and Jiajun Wu. Visual concept-metaconcept learning, 2020.

[8] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. *CoRR*, abs/1807.08556, 2018. URL `http://arxiv.org/abs/1807.08556`.

[9] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. *CoRR*, abs/1803.03067, 2018. URL `http://arxiv.org/abs/1803.03067`.

[10] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Inferring and executing programs for visual reasoning. *CoRR*, abs/1705.03633, 2017. URL `http://arxiv.org/abs/1705.03633`.

[11] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[12] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. Rethinking knowledge graph propagation for zero-shot learning, 2019.

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[14] Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. A competence-aware curriculum for visual concepts learning via question answering, 2020.

[15] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects, 2021.

[16] 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 *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJgMlhRctm`.

[17] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. doi: 10.1109/cvpr.2018.00519. URL `https://doi.org/10.1109%2Fcvpr.2018.00519`.

[18] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1435–1442, 2012.

[19] Lingjie Mei, Jiayuan Mao, Ziqi Wang, Chuang Gan, and Joshua B. Tenenbaum. FALCON: Fast visual concept learning by integrating images, linguistic descriptions, and conceptual relations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=htWIlvDcY8`.

[20] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation, 2021.

[21] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

[22] Brian L. Sullivan, Christopher Wood, Marshall J. Iliff, Rick Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142:2282–2292, 2009. URL `https://api.semanticscholar.org/CorpusID:85401998`.

[23] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning, 2018.

[24] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering, 2017.

[25] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need?, 2020.

[26] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures, 2018.

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2017.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[29] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs, 2018.

[30] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding, 2019.

[31] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[32] Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, and Yuke Zhu. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs, 2021.

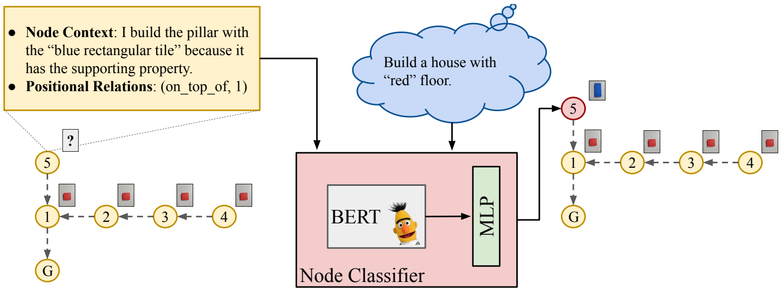

Node Context: I build the pillar with the “blue rectangular tile” because it has the supporting property.
Positional Relations: (on_top_of, 1)
Build a house with “red” floor.
?
5
1
2
3
4
G
BERT
MLP
Node Classifier
5
1
2
3
4
G

Figure 4: We demonstrate how our pipeline decide the object to pick for one node in the scene graph. We feed the node context and the request into a node classifier, which is composed of a BERT encoder and a MLP layer, to decide which concept to pick up. In this example, the object for node 5 is "blue rectangular tile" because it is not mentioned in the request.

A Implementation Details

A.1 Node Classifier

Figure 4 describes the process of how our pipeline choose an object to pick. The inference is a two-step process, both using a BERT$_{\text{base}}$ model and a MLP layer, and taking the node context t_i and he linguistic request q as inputs. In the first step we use the BERT$_{\text{base}}$ model and a MLP layer to decide whether the node in the goal graph is different from the corresponding node in the demonstration. Then we use another BERT$_{\text{base}}$ model and MLP layer to extract the object from t_i and q for this node location.

In the example of Figure 4, we are trying to decide the object that should be placed in position 5. Based on the node context t_5 and the request q, the node classifier decides that node 5 in the goal graph should remain the same as the demonstration. Then, we use the concept extractor to extract the object from t_5, and we found that the object that should be placed at node 5 is ”blue rectangular tile”.

A.2 Training Pipeline

We explain our training pipeline in this section. The concepts from the dataset is divided into three groups: $C_{pretrain}$, C_{train} and C_{test}, where the pre-train concepts $C_{pretrain}$ represent the pre-existing nodes in the knowledge graph. The training of the concept net model is consists of three stages, the pre-training for the visual feature extractor, the pre-training for the embedding of pre-train concepts $C_{pretrain}$, and the training to update the knowledge graph with train concepts C_{train}.

Pre-training the Visual Feature Extractor. In the first pre-training stag, we generate a VQA dataset on both the pre-train concepts $C_{pretrain}$ and the train concepts C_{train}. The purpose of this stage is to expose the visual feature extractor with a larger variation of visual features. We jointly pre-train the visual feature extractor and the embedding for the pre-train concepts $C_{pretrain}$ and the train concepts C_{train} with the visual question answering task in this stage. After this pre-training stage, the embeddings of all pre-train concepts and train concepts will be discarded.

Pre-training Pre-train Concepts. The embedding of pre-train concepts $C_{pretrain}$ is obtained through gradient descent in this pre-train phase. For this phase, we generate a VQA dataset on the pre-train concepts $C_{pretrain}$ only. After we warmup the visual feature extractor in the first pre-training phase, we jointly train the visual feature extractor and the embedding for the pre-train concepts in this phase using the same VQA task. This pre-training step is skipped under the setting where the concept net has zero prior knowledge of the concepts, which is the setting of all of our experiments.

Training. After we have pre-trained the visual feature extractor and the embedding for the pre-train concepts, we train the concept learner module during the training stage. We freeze the weights of the visual feature extractor at this stage because otherwise the embeddings for the pre-train concept will not be usable. Because we hope to train ARGNN to update the embedding for known concepts with information from unseen instances, we have to reset the embedding for all the pre-train concepts and train concepts,$C_{pretrain}$ and C_{train}, after all the train concepts are inserted to the network. After inserting all the concepts within the train set in the final round, we do not reset the embedding for the train concepts and insert the concepts in the test set C_{test}.

A.3 Training Configurations

In this section we describe the training configuration of the experiments for all the three domains. During the training phase, the model completes one round of training if it finishes to insert all the concepts in the training set once. For simplicity, we unify the steps of training with rounds of insertion. For all the experiment results we report in this work, we adopted the configuration where there is no pre-train concepts. As a results, the second phase of pre-training is skipped for all the three datasets. For all the three domains, we train our model for completing the concept graphs 100 rounds, and the number of concept insertions varies depending on the split of the concepts. We start the training with a learning rate of 0.001 and decrease the learning rate by a factor of 0.1 in every 25 rounds of completing the knowledge graph in the training stage. For CUB-200-2011 dataset, we train our model for 50000 iterations with a batch size of 10. We use an Adam optimizer with learning rate of 0.0001 in the pre-training phase of the visual feature extractor. For the house construction domain and the zoo domain, we train our model for 5000 iterations with a batch size of 10. We use an Adam optimizer with learning rate of 0.0001 in the pre-training stage of the visual feature extractor.

A.4 Robot Setup

In this section, we describe the details for camera calibration. We need to calibrate cameras with respect to the FR3 base frame. We take multiple pictures in different configurations of the FR3 end-effector to which an acuro market is attached. This allows us to find a Transformation Matrix which converts the coordinates from the camera frame to the Robot base frame. The place scene camera is used to find the length of the object occupying the current node of the scene graph.

In this section, we describe how we compute the plcaement location for each object in detailed. SAM is used to segment the objects placed in the place scene and find the bounding boxes of each placed object which are also the nodes of our scene graph. This allows us to calculate the position of the next object by finding the relative position of the next node with respect to the current object being placed. , referencing the position of the node of the scene, and calculating the length of the bounding box of the referenced node. we use a formula of shift = 1/2*max(bounding box of the referenced node length)+50 pixel space **Next node position**= Relation to the reference node(Reference node position,shift). The function relation to the reference node adds a shift to the reference node position based on its relation to the next node. For example, it adds the shift only to the x coordinate if there is "to the top of" relation, or in the case of "to the right of" relation, it adds only the y coordinate of the current position. In our scene graph, we are able to identify "to the top of ", "to the bottom of"," to the right of"," to the left of", "to the top right of"," to the top left of"," to the bottom right of", and "to the bottom left of" relations.
The Segment Anything Model is capable of separating the foreground from the background. This allows us to find the table mask and the segment of each object placed in the camera frame on the table.
The flow of our pipeline requires us to first demonstrate the visual scene with all the objects placed in the Task scene to make a structure with linguistic inputs. We have to make sure that the objects are placed at a distance that allows SAM to create separate segment boxes for the objects. Then we pass each segmented object to either FALCON or Hi-Viscont classifiers to classify conditioned on the given language query. The robot then picks the object with simple visuo-motor servoing. by the node information of the scene graph. Once we find the object to be picked we then calculate the center of the bounding box of that object and convert it to the Robot frame with the help of the transformation matrix. If in the process there is an incomplete or erroneous grasp, we reattempt the whole classification again autonomously. Once the object is grasped we then place the object into the Task scene, with the position calculated relatively with respect to the previously placed object nodes or the ground. This process is iteratively done until we have completed the whole scene graph.

B Detailed Results

In this section, we presents the statistic tests between Hi-Viscont and FALCON for all the three domains.

CUB-200-2011. Results of paired t-test suggest that Hi-Viscont achieves higher F1 scores with significance for concepts in Genera($p < 0.001$), Family($p < 0.001$), and Order($p = 0.005$).

House Construction Domain. Results of paired t-test suggests that Hi-Viscont achieves higher F1 scores with significance for color concepts($p = 0.005$) and affordance concepts($p = 0.002$).

Zoo Domain. Results of paired t-test suggests that Hi-Viscont achieves higher F1 scores with significance for non-leaf concepts($p = 0.001$).