Learning Action-Effect Dynamics for Hypothetical Vision-Language Reasoning Task

Shailaja Keyur Sampat, Pratyay Banerjee, Yezhou Yang and **Chitta Baral**
Arizona State University, USA
{ssampa17,pbanerj6,yz.yang,chitta}@asu.edu

Abstract

'Actions' play a vital role in how humans interact with the world. Thus, autonomous agents that would assist us in everyday tasks also require the capability to perform 'Reasoning about Actions & Change' (RAC). This has been an important research direction in Artificial Intelligence (AI) in general, but the study of RAC with visual and linguistic inputs is relatively recent. The CLEVR_HYP (Sampat et al., 2021) is one such testbed for hypothetical vision-language reasoning with actions as the key focus. In this work, we propose a novel learning strategy that can improve reasoning about the effects of actions. We implement an encoder-decoder architecture to learn the representation of actions as vectors. We combine the aforementioned encoder-decoder architecture with existing modality parsers and a scene graph question answering model to evaluate our proposed system on the CLEVR_HYP dataset. We conduct thorough experiments to demonstrate the effectiveness of our proposed approach and discuss its advantages over previous baselines in terms of performance, data efficiency, and generalization capability[1].

1 Introduction

Humans interact with their environment to accomplish desired goals. Object manipulation (i.e., performing "actions" over the objects) is a fundamental concept that makes this interaction possible. In other words, actions in their simplest form have the power to change the state of a world and hence play a vital role in enabling humans to perform day-to-day tasks. As we are developing autonomous agents that can assist us in everyday tasks, they would also require to interact with complex environments. Hence, the development of autonomous agents that can perform actions to effectively manipulate objects and understand corresponding effects is of great importance. As a result, Reasoning about Action and Change (RAC) has been a long-standing research problem, since the rise of AI.

The work of McCarthy et al. (1960) was the earliest to emphasize the importance of reasoning about actions. They developed an advice taker system that can do deductive reasoning about scenarios such as "going to the airport from home" requires "walking to the car" and "driving the car to airport". Since then, many real-life use cases have been identified which require AI models to understand interactions among the current states of the world, actions being performed over various objects, and most likely following states (Banerjee et al., 2020). While RAC has been more popular among knowledge representation and logic communities, it has recently piqued the interest of researchers in NLP and vision domains. A recent survey by Sampat et al. (2022) compiled a comprehensive list of works that explore neural network's ability to reason about actions and changes, provided a dataset of linguistic and/or visual inputs.

In a recent tweet, Prof. Yann LeCunn also emphasized the importance of this research direction. He mentions that "while we progress towards human-level AI, I believe we need to find new concepts that would (i) allow machines to learn to predict how one can influence the world through taking actions, (ii) learn hierarchical representations that allow long-term predictions in abstract spaces, (iii) enable agents to predict the effects of sequences of actions so as to be able to reason & plan - *all of this in ways that are compatible with gradient-based learning*" (LeCun, 2022).

In this work, we aim to better tackle the hypothetical action reasoning task of CLEVR_HYP dataset (Sampat et al., 2021). An example from this dataset is shown in Figure 1). The key objective is to understand changes caused over a visual scene by an action described in the natural language and answer a reasoning question. In Figure 2, we describe two possible action-effect learning strategies (LS1 and

[1]Dataset setup scripts and code for baselines are available at https://github.com/shailaja183/ARL

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5943 - 5953

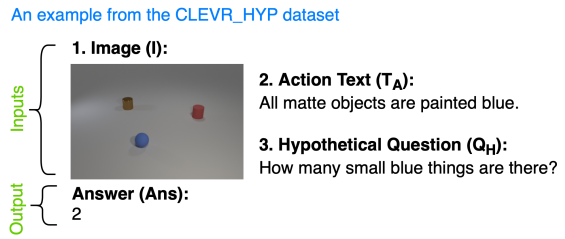

Figure 1: Revisiting CLEVR_HYP task (Sampat et al., 2021): Answer a reasoning question (Q_H) about changes caused over the given image (I) by performing a hypothetical action (T_A).

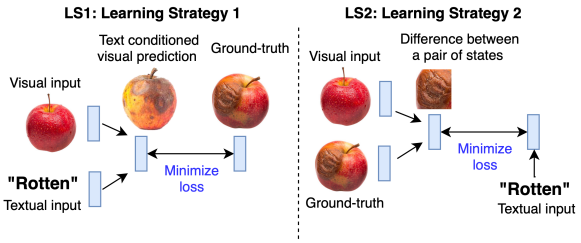

Figure 2: Two possible ways (LS1 and LS2) to learn action-effect dynamics in a supervised learning setting. In this paper, we implement LS2 which demonstrates improvements over LS1. Blue box denotes vector representation.

LS2) through a toy example to convey our intuition behind this work. LS1 uses visual features (i.e. features from the image of an apple) and a representation of actions (i.e. text "rotten") through sentence embeddings to imagine the effects (i.e. how a rotten apple would look like). This can be an intuitive choice to model the CLEVR_HYP task using pre-trained vision-language models.

In our hypothesis, LS1 does not improve the model's understanding of what effects the actions will produce. Thus, we propose an alternative strategy, LS2[2]. Specifically, we let the model observe the difference between pairs of states before and after the action is performed (i.e. decayed portion of the apple that distinguishes a good apple from the rotten one), then associate those visual differences with the corresponding linguistic action descriptions (i.e. text "rotten"). LS2 is likely to better capture action-effect dynamics, as action representations are learned explicitly.

To empirically test the above hypothesis, we develop a model (which is described in Section 3) and evaluate it on the CLEVR_HYP dataset. We hope that our exciting results would enable the development of AI agents that can better collaborate with humans in the physical world and encourage further investigations in this research area. In summary, our key contributions are as follows;

- We propose a novel learning strategy for predicting the "effects of actions" in the vision-language domain (shown in Figure 2).
- We develop a 3-stage model to implement the proposed learning strategy and evaluate on the existing CLEVR_HYP (Sampat et al., 2021) dataset.
- Through ablations and analysis, we demonstrate the effectiveness of our model in terms of performance (5.9% accuracy improvements), data efficiency (one-third of training data required), and better generalization capability in comparison with the best existing baselines.

[2]Figure 2 is meant to convey our intuition behind the proposed model in this work at a very high level. Figure 5 is more complex in comparison and accurately describes the working of our model, considering the format of CLEVR_HYP dataset, decomposing the task into various neural components, and measuring them using appropriate loss functions.

2 CLEVR_HYP

In this section, we briefly summarize important aspects of the CLEVR_HYP dataset (Sampat et al., 2021) and related terminologies used in the subsequent sections.

2.1 Problem Formulation

The task aims at understanding changes caused over an image by performing an action described in natural language and then answering a reasoning question over the resulting scene. Consult Figure 1 for better understanding of the following;

- *Inputs:*
  1. Image (I)- Visual scene with rendered objects
  2. Action Text (T_A)- Textual modality describing action(s) to be performed over I
  3. Hypothetical Question (Q_H)- Textual question that will assess the system's capability to understand changes caused by T_A on I
- *Output:* Answer (A) for the given Q_H
- *Answer Vocabulary:* [0-9, yes, no, cylinder, sphere, cube, small, big, metal, rubber, red, green, gray, blue, brown, yellow, purple, cyan]
- *Evaluation:* 27-class Answer Classification / Accuracy (%)

2.2 Dataset Details and Partitions

The CLEVR_HYP dataset assumes to have a closed set of object attributes, action types, and question

Object Attributes in Visual Scenes	Action Text Types	Question Reasoning Types
1. **Shape**: cylinder, sphere or cube 2. **Size**: small or big 3. **Material**: metal or rubber 4. **Spatial**: left, right, front, behind or on 5. **Color**: red, green, gray, blue, brown, yellow, purple or cyan	1. **Add** new objects to the scene 2. **Remove** objects from the scene 3. **Change attribute** of the objects 4. **Move** objects in or out of plane	1. **Counting** objects fulfilling the condition 2. **Verify existence** of certain objects 3. **Query attribute** of a particular object 4. **Compare attributes** of two objects 5. **Integer comparison** of two object sets (same, larger or smaller)

Table 1: Summary of object attributes, actions, and reasoning types in CLEVR_HYP dataset (Sampat et al., 2021)

reasoning types which are summarized in Table 1. The dataset is divided into the following partitions;

- *Train* (67.5k) / *Val* (13.5k) sets have <I, T_A, Q_H, A> tuples along with the scene graphs as a visual oracle and functional programs[3] as a textual oracle.
- *Test* sets consist of only <I, T_A, Q_H, A> tuples, and *no oracle annotations are available*. There are three different test sets,
  1. Ordinary test (13.5k) consists of examples with the same difficulty as train/val
  2. 2HopT_A test (1.5k) consists of examples where two actions are performed ex. 'Move a purple object on a red cube *then* paint it cyan.'
  3. 2HopQ_H test (1.5k) consists of examples where two reasoning types are combined ex. 'How many objects are *either* red *or* cylinder?'

2.3 Baseline Models

Following is a brief description of two top-performing baselines reported in Sampat et al. (2021), to which we will compare the results of our proposed approach in this paper.

- **(TIE) Text-conditioned Image Editing:** Text-adaptive encoder-decoder with residual gating (Vo et al., 2019) is used to generate new image conditioned on the action. Then, new image along with the question is fed into LXMERT (Tan and Bansal, 2019) (which is a pre-trained vision-language transformer), to generate an answer. The model can be visualized in Figure 3.

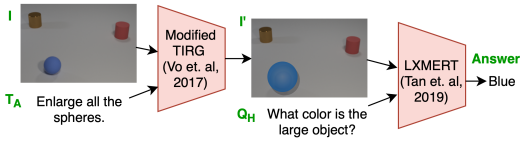

Figure 3: Architecture of TIE baseline

- **(SGU) Scene Graph Update:** In this model, understanding changes caused by an action text is considered as a graph-editing problem. First, an image is converted into a scene graph and action text is converted into a functional program (FP). Sampat et al. (2021) developed a module inspired by Chen et al. (2020) that can generate an updated scene graph based on the original scene graph and a functional program of an action text. It is followed by a neural-symbolic VQA model (Yi et al., 2018) that can generate an answer to the question provided the updated scene graph. The model can be visualized in Figure 4.

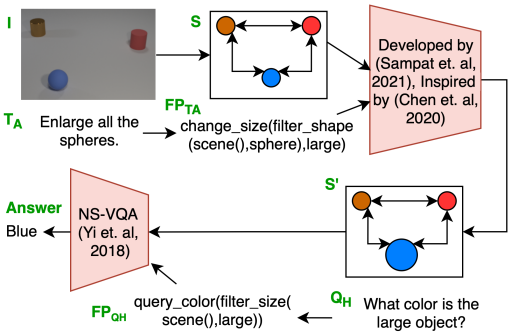

Figure 4: Architecture of SGU baseline

There are important distinctions between baselines developed by Sampat et al. (2021) and our proposed method. Both the above baselines rely on pre-learned word representation of actions- either by a word-vector algorithm or a learned functional program and use that to conditionally update the visual scene (at pixel level or through graph operations). Thus, TIE and SGU resembles more to LS1 in Figure 2. Note that in SGU baseline, individual functions (in their functional program representations) are human authored i.e. what kind of inputs it accepts and what it will return when executed. For example, 'remove <attribute>' function will take a set of objects as input and return a subset of objects which do not have <attribute>.

In contrast, we learn action representations through two-step process. First, we learn to predict

[3]Originally introduced in CLEVR (Johnson et al., 2017). For example, a question 'How many red metal things are there?' can be represented as a functional program 'count(filter_color(filter_material(scene(),metal),red))'

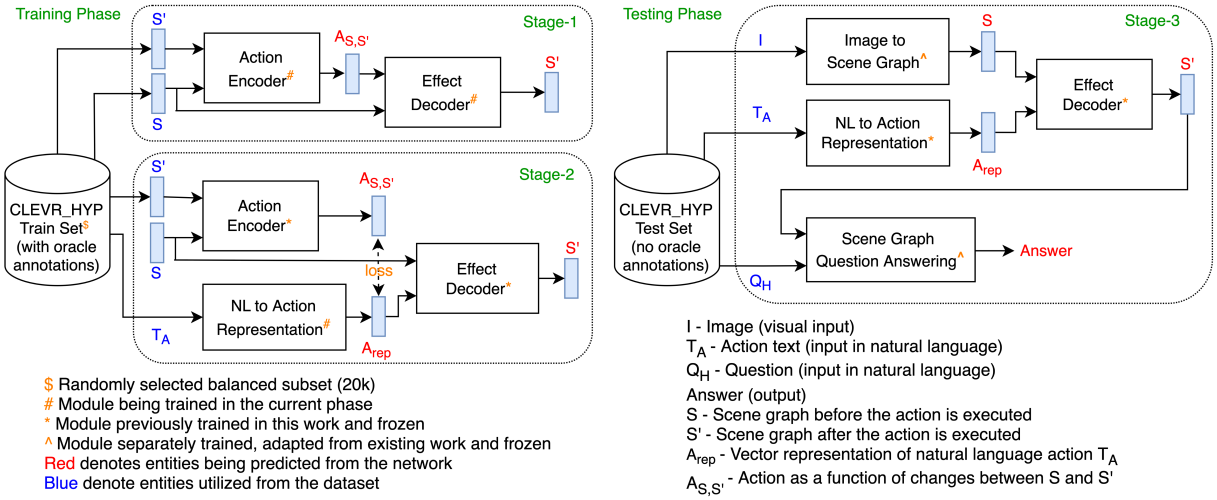

Figure 5: Detailed visualization of our proposed 3-stage Action Representation Learner (ARL) model: (left) training phase (right) testing phase (bottom) terms and notations used. Best viewed in color.

changes in a pair of scene-graphs (before and after the action is performed). And second, we minimize the loss between changes in the scene with the representation of linguistic action descriptions. Thus, our proposed model resembles more to LS2 in Figure 2. Our method is purely based on data and does not require any human intervention. Also, note that oracle scene graphs post-actions are not available at the test-time. By enforcing two-way representation learning, we are able to predict changes in the scene graph using and action vector from linguistic action description for a given test instance.

3 Proposed Model: Action Representation Learner (ARL)

In this section, we describe the architecture of our proposed model Action Representation Learner (ARL).

In our point of view, the most critical component of a model that attempts to solve CLEVR_HYP is the one where mapping between visual changes and actions are learned. In Figure 2, we graphically demonstrated our intuition behind how we can do so. Our hypothesis is that a model can learn better action representations by observing difference between a pair of states (before and after the action is performed) and then associate those visual differences with given linguistic description of actions. In this regard, we attempt to create a 3-stage model shown in Figure 5, which we believe would better capture the causal structure of this task. Detailed description of each individual component is provided below and can be visualized in Figure 6.

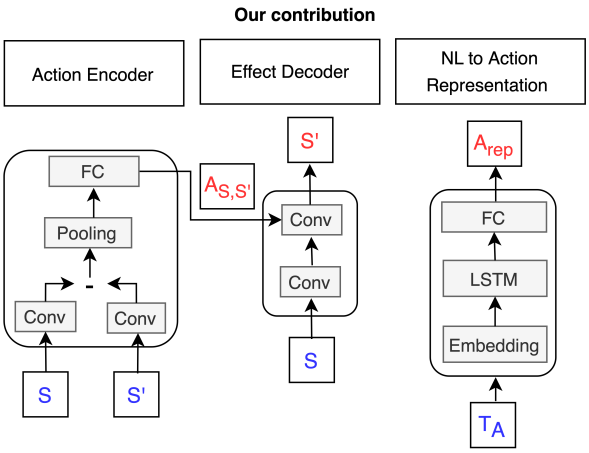

Figure 6: Internal details of the components used in ARL model. Red denotes entities predicted, Blue denotes entities utilized from the dataset.

3.1 Stage-1

Actions have the power to change the state of the world. In other words, difference between a pair of states can be considered as a function of performing actions. For example, consider two states 'green cube' and 'red cube'. The change between above states is 'green→red', which is a function of action 'paint' or 'change color'. Conversely, if one is provided with a state 'green cube' and knowing that the 'paint red' action is performed, then one can visualize that the next state would be 'red cube'. We aim to capture such relationship between state changes and actions in this stage.

Specifically, we setup an encoder-decoder model to achieve this objective. Since our objective is to

learn action and effects, we refer to them as 'Action Encoder' and 'Effect Decoder'. As described in Section 2.2, the training set of CLEVR_HYP provides oracle annotations for an initial scene graph S (using which the image is rendered) and a scene graph after executing the action text S′. We take a random subset of 20k[4] scene graph pairs from CLEVR_HYP training set that are balanced by action types (add, remove, change, move) to train this encoder-decoder.

We capture the difference between states S and S′ i.e. $A_{S,S'}$ using the encoder. At the test time, we do not have the updated scene graph S′ available. To address this issue, the encoder is followed by a decoder, which can reconstruct S′ provided S and the learned scene difference $A_{S,S'}$ in the encoder network. Formally,

$$A_{S,S'} = ActionEncoder(S, S') \quad (1)$$

$$S' = EffectDecoder(S, A_{S,S'}) \quad (2)$$

Where $(S, S') \in$ CLEVR_HYP training set, red denotes entities predicted, blue denotes entities utilized from the dataset. We jointly train action encoder-effect decoder networks with the following objective;

$$argmax_{\Theta_{ActionEncoder}\Theta_{EffectDecoder}} [logP(S' | S,\ ActionEncoder(S, S'))] \quad (3)$$

Note that, in common applications involving encoder-decoder architecture, the decoder part of the model is removed once the desired performance is achieved and the encoder is used to encode input sequences to a fixed-length vector at test-time. Contrary, here we discard the encoder part and keep the decoder part to obtain updated scene representation from the initial scene and learned action vector.

3.2 Stage-2

As explained in stage-1, we do not have the S′ at the test time and we cannot compute $A_{S,S'}$. However, provided that changes in the scene are a function of the action, we can approximate A_{rep} A_{rep} is a vector representation corresponding to natural language action. A network is trained which can convert 'Natural language to Action Representation' with the help of encoder-decoder network trained in Stage-1 that maximizes the log probability that outputs the correct state S′ as below.

$$argmax_{\Theta_{NL2ActionRep}} [logP(S' | S,\ NL2ActionRep(T_A))] \quad (4)$$

At the core, lies LSTM encoder, which precedes by an embedding layer and followed by dense layers. During model training, in addition to finding the values for the weights of the LSTM and dense layers, the word embeddings for each word in the training set are computed. This is achieved using nn.Embedding(vocabulary_size, embedding_size) layer defined in pytorch. This way, a fixed length one-hot vector of given length is generated for each word in the vocabulary depending on the position of the word in context and updated using back-propagation. Embedding layer is similar to a linear layer, which returns the index where one is located instead of returning the whole one-hot vector. It takes an action text T_A as a sequence of learned word embeddings, runs an LSTM over them, then projects from the final cell state to get the output A_{rep}. The LSTM has a hidden layer of size 200.

3.3 Stage-3

For each image, we use Mask R-CNN (He et al., 2017) to generate segment proposals of all objects. Along with the segmentation mask, the network also classifies the objects based on their visual attributes- color, material, size, and shape. The threshold for segment proposals is set to 0.9 i.e. segments with bounding-box score less than 0.9 are dropped. The segment for each single object, paired with the original image (resized to 224x224) is sent to ResNet-34 (He et al., 2016) to extract 3D coordinates of objects in the scene. Inclusion of original full image is observed to enhance the performance by incorporating contextual information. Note that scene-parsing is pre-trained and not fine-tuned with rest of the network.

4 Results and Analysis

In this section, we discuss the performance of our model quantitatively and qualitatively. Additionally, we discuss our findings from three ablations for our model.

4.1 Quantitative Results

Once we complete the aforementioned 3-stage training process, we leverage a couple of existing

[4]We experiment with different data sizes and discuss results in Section 4.3, but obtain the optimal results for 20k samples when action vector length is 125

models along with the trained components to make predictions on CLEVR_HYP (Sampat et al., 2021) test data (as shown in the right part of the Figure 5). The CLEVR_HYP has three test sets- Ordinary, 2HopT$_A$ and 2HopQ$_H$. Refer to Section 2.2 for the description of each test setting with examples.

Test performance on CLEVR_HYP			
	TIE	*SGU*	*ARL*
Ordinary	64.7	70.5	76.4
2HopA$_T$	55.6	64.4	69.2
2HopQ$_H$	58.7	66.5	70.7

Table 2: Performance of two baselines (TIE, SGU) reported in (Sampat et al., 2021) and our proposed model (ARL) on three test sets of CLEVR_HYP

Accuracy(%) by Action Types			
Validation	TIE	SGU	ARL
Add	58.2	65.9	70.3
Remove	89.4	88.6	94.1
Change	88.7	91.2	95.8
Move	61.5	69.4	72.6
2HopT$_A$	*TIE*	*SGU*	ARL
Add + Remove	53.6	63.2	66.7
Add + Change	55.4	64.7	70.6
Add + Move	49.7	57.5	63.2
Remove + Change	82.1	85.5	91.6
Remove + Move	52.6	66.4	68.3
Change + Move	53.8	63.3	67.1

Table 3: Performance breakdown of models by different action types in validation and 2HopT$_A$ test set

Accuracy(%) by Reasoning Types			
Validation	TIE	SGU	ARL
Count	60.2	74.3	78.6
Exist	69.6	72.6	77.3
CompareInteger	56.7	67.3	70.7
CompareAttribute	68.7	70.5	73.4
QueryAttribute	65.4	68.1	74.9
2HopQ$_H$	*TIE*	*SGU*	*ARL*
And	59.2	67.1	70.3
Or	58.8	67.4	71.5
Not	58.1	65.0	68.4

Table 4: Performance breakdown of models by different question types in validation and 2HopQ$_H$ test set

The CLEVR_HYP dataset is formulated as a classification task with exactly one correct answer. Therefore, the exact match accuracy (%) metric is used for evaluation. Table 2 demonstrates the performance of our proposed model in comparison with the two best performing existing models TIE and SGU, described in Section 2.3. Our proposed approach outperforms those baselines by 5.9%, 4.8% and 4.2% on *Ordinary*, *2HopT$_A$ Test* and *2HopQ$_H$ Test* respectively. This demonstrates that our model not only achieves better overall accuracy but also has improved generalization capability when multiple actions have to be performed on the image or understand logical combinations of attributes while performing reasoning.

In Table 3, we analyze the ability of models to perform a particular action. For validation set, the model is expected to perform one of the four actions- add objects, remove objects, change in attributes or move objects. Overall, we can observe that our proposed method ARL achieves better fine-grained accuracy for all action types compared to existing models. All three models do quite well on 'remove' and 'change' actions whereas struggle when new objects are added or existing objects are moved around. Yet, our model shows 4.4% and 3.2% improvements on 'add' and 'move' respectively compared to the best previous baseline.

For 2HopT$_A$ test set, the model is expected to perform two different actions (among add, remove, change and move) one after the other. Our observation from the Validation results remains consistent when multiple actions are combined. In other words, models were able to achieve relatively high accuracy for actions 'remove' and 'change', hence their combination 'remove+change' also has high model performance. Whereas other combinations of actions accomplish relatively lower performance. It leads to the conclusion that understanding the effect of different actions is of varying complexity. It also indicates that the learned action representations in our proposed model are helpful as it shows better generalization by action types.

Though understanding changes caused by actions is the core challenge in CLEVR_HYP (Sampat et al., 2021) task, it has a question answering downstream task. To answer questions in the CLEVR_HYP dataset, models should be able to perform counting, check the existence of objects given the criteria, compare sets of objects, or retrieve attributes of the desired objects. We carry out a similar analysis of models based on their capability to perform above-mentioned reasoning tasks.

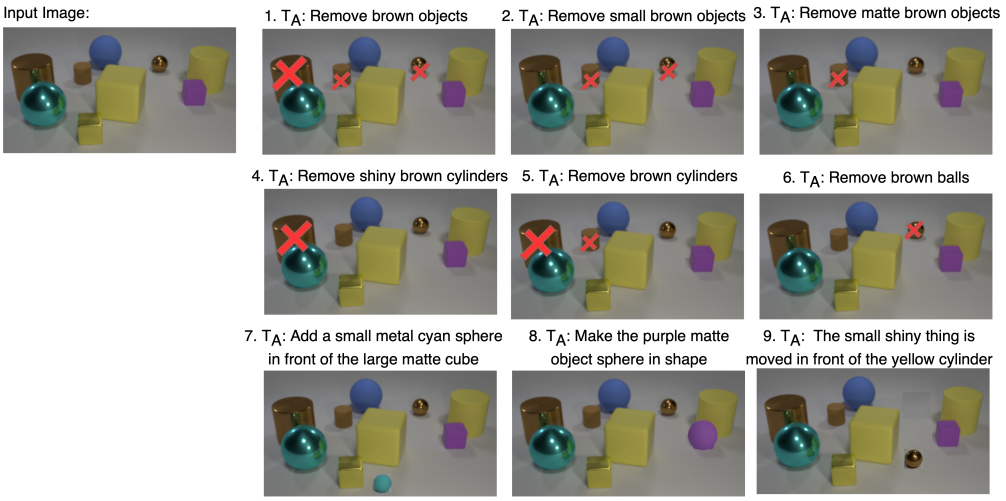

Figure 7: Correct scene graph predictions for the given (input image, action text) by our ARL model

The results are summarized in Table 4.

For validation set, our proposed method ARL has better fine-grained accuracy across all reasoning types, but improvements on 'query attribute' and 'exist' types are maximum (6.8% and 4.7% respectively). For the 2HopQ$_H$ test set, the model is expected to perform logical operations within a particular reasoning type. For example, 'How many objects are either red or cylinder?' and 'Are there any rubber cubes that are not green?'. Though we see some gains here as well, our overall pipeline is limited by the capabilities of the question answering model of Yi et al. (2018) we use in stage-3.

4.2 Qualitative results

In Figure 7, we visually demonstrate scene graphs predicted by our ARL model over a variety of action texts. From examples 1-6, we can observe that the model can correctly identify objects that match the object attributes (color, size, shape, material) provided in the action text. Examples 4 and 6 demonstrate that our system is consistent in predictions when we use synonyms of various words (e.g. sphere∼ball, shiny∼metallic) in the dataset. Finally, examples 7-9 show that our model does reasonably well on other actions (add, change, move).

We further generate a t-SNE plot of action vectors learned by our best proposed model, which is shown in Figure 8. At a first glance, we can say that the learned action representations formulate well-defined and separable clusters corresponding to each action type. Clusters for add, remove and change actions are closer and somewhat overlapping. We observed that many samples of type 'change' is interpreted by the reasoner as 'remove+add' action. For example, if a color of 'small blue metal sphere' is changed to 'red', the action reasoner interprets it as removal of the 'small blue metal sphere' followed by an addition of a 'small red metal sphere' on the same location.

4.3 Ablations

Importance of stage-1 training Cause-effect learning with respect to actions is a key focus in CLEVR_HYP. In existing models, it is formulated as a updated scene graph prediction task (i.e. given an initial scene and an action, determine what the resulting scene would look like after executing the action). In our opinion, stage-1 plays a critical role in learning causal structure of the world. To demonstrate this, we set up two experiments; first, where training takes place in a sequential manner (stage-1 followed by stage-2), where trained encoder-decoders from stage-1 are frozen and utilized in stage-2. Second experiment, where there is no separate stage-1 training and encoder-decoder in stage-2 are randomly initialized.

The results are summarized in Table 5. We can observe that inclusion of stage-1 training improves the accuracy of scene graph prediction by ∼30% compared to the stage(2 only) model. To evaluate question answering task of CLEVR_HYP, both setups are followed by stage-3 where the image parser and scene-graph question answering modules are combined to predict the answer. It is known that (Yi et al., 2018) has near-perfect performance on the scene graph question answering task over CLEVR (Johnson et al., 2017). As a result, the

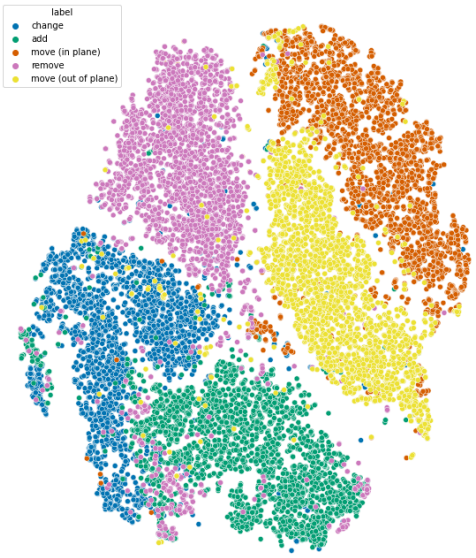

Figure 8: The t-SNE plot of learned action vectors

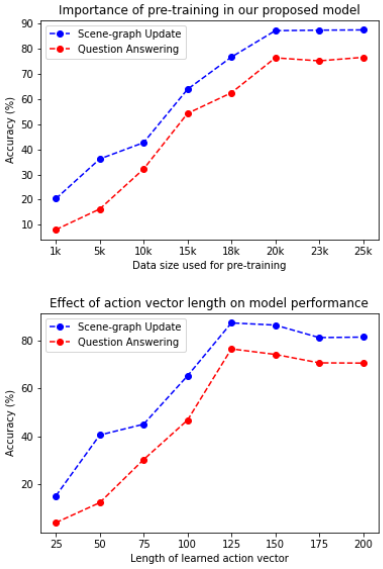

Figure 9: Performance of our model with varying (top) data size and (bottom) action vector lengths

gains achieved in the scene graph task directly benefit the question answering performance without much of a loss. In other words, there are only 0.2% instances where the scene prediction is correct but the final answer is incorrect.

Performance with different lengths of learned action vector in stage-1 In this ablation, the goal is to find out optimal length of action vectors that can reasonably simulate the effects of the actions. We experiment with different lengths of learned action vector- from 25 to 200 in increment of 25. Figure 9 (bottom) shows the effect of training with diverse action vector lengths on scene graph update and downstream question answering task. The model learns better initially when the vector length is increased, however performance reaches at peak for the action vector length of 125.

Task	Experiment	Accuracy (%)
Scene Graph Update	Stage(2 only)	56.3
	Stage(1+2)	87.2
Question Answering	Stage(2+3 only)	45.7
	Stage(1+2+3)	76.4

Table 5: Performance of our model in the absence and presence of stage-1 over ordinary test set

5 Related Works

In this section, we discuss existing research efforts that align with our work in this paper. Specifically, we elaborate on tasks involving learning action representations, counterfactual reasoning, and what-if question answering datasets.

Tasks involving learning representation of actions: Better representation learning is key to success in all kinds of artificial intelligence problems (Banerjee et al., 2021; Gan et al., 2020; Chen et al., 2021; Lee et al., 2018b,a). Learning a mapping from the goal (provided in natural language) to a sequence of actions to be performed in a visual environment is a common task in robotics (Kanu et al., 2020; Shridhar et al., 2020). Specifically, human-in-the-loop methods for training robots to perform various actions involve learning a mapping between verbal commands and low-level motor controls of a robot (Stepputtis et al., 2020). Another relevant task is vision-and-language navigation (Anderson et al., 2018; Chen et al., 2019; Nguyen et al., 2019), where an agent navigates in a visual environment to find the goal location by following natural language instructions. Navigation tasks focus on selecting the right actions to achieve desired goals provided a visual environment and natural language instructions. Our focus in this paper is to develop models that can implicitly reason about the effect of actions rather than determining which action to perform.

Counterfactual vision-language reasoning: Counterfactual reasoning is termed as an ability to develop mental representations to generate alternate consequences about an event that happened in the past based on given criteria. Inspired by this human ability, there have been efforts to utilize this concept to improve many aspects of language and vision-language research; Kusner et al. (2017) and Garg et al. (2019) proposed methods to measure counterfactual fairness of models. A few recent works incorporated counterfactual augmentation of training sets (Zmigrod et al., 2019; Fu et al., 2020b) to improve the robustness of models and discourage biases. Contrary to that, the work of Fu et al. (2020a) was the first to utilize counterfactual instructions in training (ex. multiple questions asked to the same image set) to deal with the data scarcity issue and improve the generalization.

What-if question answering datasets: WIQA (Tandon et al., 2019) is a testbed for what-if reasoning over natural language contexts. Provided a procedural paragraph, the task is to answer the question "Does change in X result in change in Y?" (where X and Y are two events from the paragraph) as a 3-way choice- correct, opposite, or no effect. In the vision-language domain, TIWIQ (Wagner et al., 2018) was among the earliest works. Given a synthetically rendered table-top scene, the task is to generate a textual response to the what-if question when an action (push, rotate, remove or drop) is performed on an object. However, the evaluation of open-ended text generation is challenging. To fill in this gap, Sampat et al. (2021) created CLEVR_HYP dataset. It shares similarities with TIWIQ for having rendered images, limited action types, and QA as a task. However, the key difference is that in CLEVR_HYP, an action can cause changes to multiple objects in the scene, which is not the case with TIWIQ.

6 Conclusion

In the vision and language domain, several tasks are proposed that require an understanding of the causal structure of the world. In this work, we propose an effective way of learning action representations and implement a 3-stage model for the what-if vision-language reasoning task CLEVR_HYP. We provide insights on the learned action representations and validate the effectiveness of our proposed method through ablations. Finally, we demonstrate that our proposed method outperforms existing baselines while being data-efficient and showing some degree of generalization capability. By extending our approach to a larger set of actions, we aim to develop AI agents which are equipped with action-effect reasoning capability and can better collaborate with humans in the physical world.

Limitations

In the CLEVR_HYP dataset, all actions are considered to be independent of each other and execution of actions is always guaranteed. However, we found few instances in the CLEVR_HYP dataset where two different actions taken over an initial scene leads to the same resulting effects. Our proposed 3-stage model has higher error rates and low confidence for such samples. Further, in real world situations, actions can have inter-dependencies on world conditions or have other properties such as order-sensitivity, symmetry with respect to other actions, reversibility etc. Exploring hypothetical reasoning problems from aforementioned aspects is still an open research direction.

Computing Infrastructure

All experiments are done over Tesla V100-PCIE-16GB GPU. Total time for all experiments (including parameter search for best model) utilized approximately 70 GPU hours.

Ethical Considerations

In this paper, our experiments are limited to publicly available CLEVR_HYP dataset that is synthetically generated through controlled environment. Thus, there are no ethical violations or known bias issues, to our best knowledge.

Acknowledgements

We are thankful to the anonymous reviewers for the constructive feedback. This work is partially supported by the grants NSF 1816039 and NSF 2132724.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018,*

Salt Lake City, UT, USA, June 18-22, 2018, pages 3674–3683. IEEE Computer Society.

Pratyay Banerjee, Chitta Baral, Man Luo, Arindam Mitra, Kuntal Pal, Tran C Son, and Neeraj Varshney. 2020. Can transformers reason about effects of actions? *arXiv preprint arXiv:2012.09938*.

Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2021. Weakly supervised relative spatial reasoning for visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1908–1918.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

Lichang Chen, Guosheng Lin, Shijie Wang, and Qingyao Wu. 2020. Graph edit distance reward: Learning to edit scene graph. In *European Conference on Computer Vision*, pages 539–554. Springer.

Nuo Chen, Chenyu You, and Yuexian Zou. 2021. Self-Supervised Dialogue Learning for Spoken Conversational Question Answering. In *Proc. Interspeech 2021*, pages 231–235.

Tsu-Jui Fu, Xin Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. 2020a. Iterative language-based image editing via self-supervised counterfactual reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4413–4422.

Tsu-Jui Fu, Xin Eric Wang, Matthew F Peterson, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. 2020b. Counterfactual vision-and-language navigation via adversarial path sampler. In *European Conference on Computer Vision*, pages 71–86. Springer.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 6616–6628. Curran Associates, Inc.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society.

John Kanu, Eadom Dessalene, Xiaomin Lin, Cornelia Fermuller, and Yiannis Aloimonos. 2020. Following instructions by imagining and reaching visual goals. *arXiv preprint arXiv:2001.09373*.

Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*.

Yann LeCun. 2022. Some obvious facts related to human level ai.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018a. Odsqa: Open-domain spoken question answering dataset. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 949–956. IEEE.

Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. 2018b. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. *Proc. Interspeech 2018*, pages 3459–3463.

John McCarthy et al. 1960. *Programs with common sense*. RLE and MIT computation center.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. 2019. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12527–12537. Computer Vision Foundation / IEEE.

Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. 2021. Clevr_hyp: A challenge dataset and baselines for visual question answering with hypothetical actions over images. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3692–3709.

Shailaja Keyur Sampat, Maitreya Patel, Subhasish Das, Yezhou Yang, and Chitta Baral. 2022. Reasoning about actions over visual and linguistic modalities: A survey. *arXiv preprint arXiv:2207.07568*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A benchmark for interpreting grounded instructions for

everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. IEEE.

Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. 2020. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085. Association for Computational Linguistics.

Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. 2019. Composing text and image for image retrieval - an empirical odyssey. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6439–6448. Computer Vision Foundation / IEEE.

Misha Wagner, Hector Basevi, Rakshith Shetty, Wenbin Li, Mateusz Malinowski, Mario Fritz, and Ales Leonardis. 2018. Answering visual what-if questions: From actions to predicted scene descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1039–1050.

Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661.