Generating Natural Language Descriptions from Multimodal Data Traces of Robot Behavior

by

Kamalesh Kalirathinam

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2021 by the
Graduate Supervisory Committee:

Heni Ben Amor, Chair
Mariano Phielipp
Yu Zhang

ARIZONA STATE UNIVERSITY

December 2021

ABSTRACT

Natural Language plays a crucial role in human-robot interaction as it is the common ground where human beings and robots can communicate and understand each other. However, most of the work in natural language and robotics is majorly on generating robot actions using a natural language command, which is a unidirectional way of communication. This work focuses on the other direction of communication, where the approach allows a robot to describe its actions from sampled images and joint sequences from the robot task. The importance of this work is that it utilizes multiple modalities, which are the start and end images from the robot task environment and the joint trajectories of the robot arms. The fusion of different modalities is not just about fusing the data but knowing what information to extract from which data sources in such a way that the language description represents the state of the manipulator and the environment that it is performing the task on. From the experimental results of various simulated robot environments, this research demonstrates that utilizing multiple modalities improves the accuracy of the natural language description, and efficiently fusing the modalities is crucial in generating such descriptions by harnessing most of the various data sources.

DEDICATION

I dedicate this thesis to my parents and mentors who have supported me across my journey towards completing this thesis.

ACKNOWLEDGMENTS

I would like to thank Dr. Heni Ben Amor for his incredible support across my journey in completing this thesis and giving me an opportunity to work in the Interactive Robotics Lab. I am also grateful to Dr. Mariano Phielipp for his constant support and guidance in my research internship at Intel A.I Labs. Lastly I would like thank Simon Stepputtis for his knowledgeable insights and ideas in helping me complete this work.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

Language is instrumental in human communication, where it plays a crucial role in conceptualizing and structuring human thought. Thus, language has an intrinsic connection with human thinking. Moreover, as society evolves to incorporate intelligent systems, it is only natural that language describes the thoughts or intentions of these systems. Robots will inevitably come into contact with humans one day, and they need to have human-like qualities. Robots are already starting to be employed in the healthcare industry as well as at-home care facilities like hospitals or nursing homes where they can help around simple tasks such as taking care of patients while assisting the nurses or looking after the children by keeping their rooms clean. With the ever-increasing need for robots to communicate with humans, it is necessary that they do so in such a manner that everyone can understand. As it will allow these machines not only to function more efficiently but also interact naturally and comfortably instead of relying on an expert's projection or control over their actions.

In the past, most of our research concentrated on generating robot control sequences Stepputtis *et al.* (2020); Lynch and Sermanet (2021) from a given natural language command where the robot's objective was to understand the human language and perform the task accordingly. However, just like a human should be able to communicate with their thoughts, robots must also have the ability for two-way conversations. The only thing that will make this bi-directional communication complete is if there are mechanisms in place so that both humans and robots can talk back to each other. This work primarily focuses on completing the other part of the bi-directional communication by modeling a system that can generate natural lan-

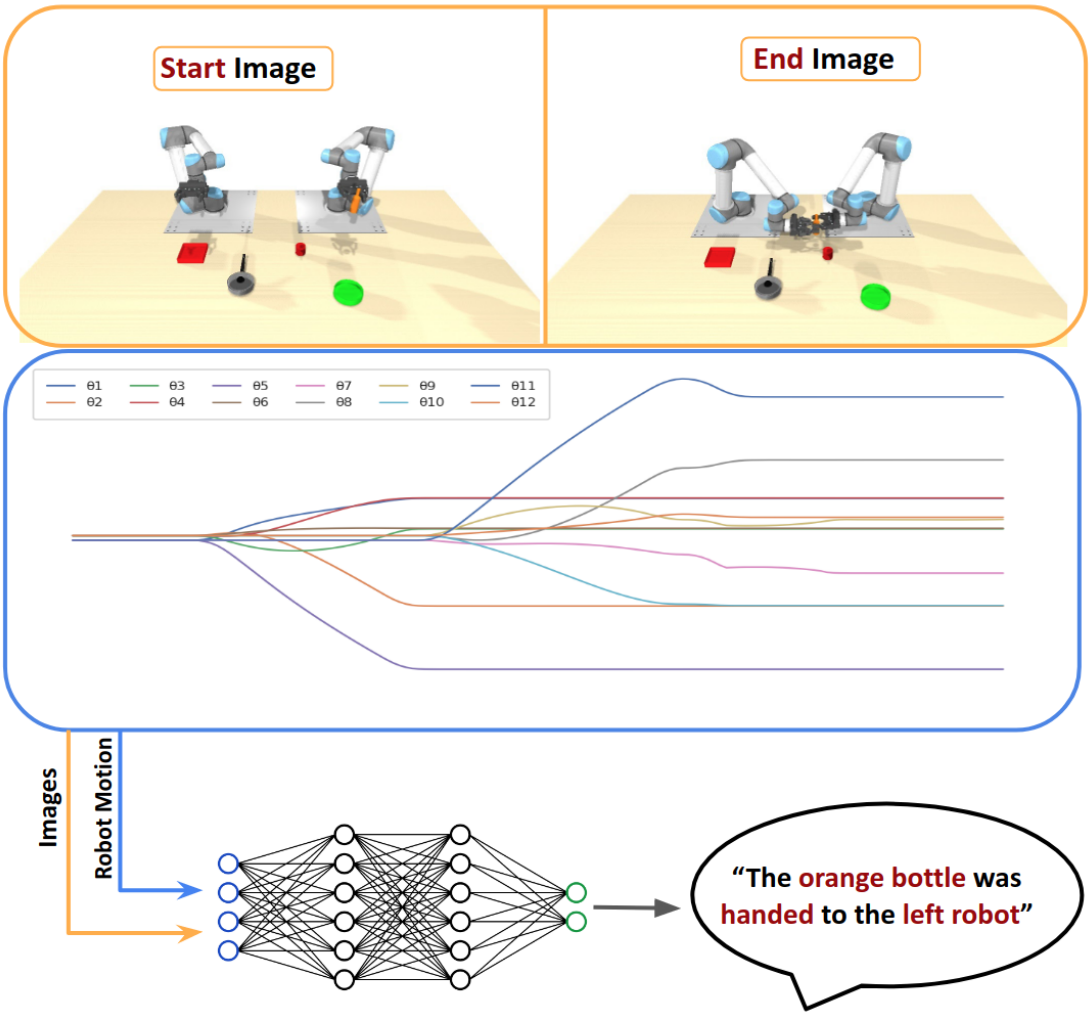

Figure 1.1: Natural Language Description Using Images and Robot Motion

guage descriptions for a robot task. This approach takes multiple modalities such as control and images from their environment to create concise, coherent accounts about how the robot behaves, which is consistent with the simulated environment. Fig 1.1 shows the two manipulators performing a task through collaboration. Here the model takes in the pre and post conditioned image of the task and the joint trajectory of the two robots to produce a concise description such as *"The red jar was handed to the right robot"*.

These natural language descriptions can have various use cases, such as enabling elderly and visually impaired people by describing tasks. It also helps with a rescue operation when humans cannot be put in danger, for example, during a fire or other sensitive situations where clarity of communication is key to success. Generating clear description reduces the cognitive load of a person and can help them do the task more efficiently. Describing robot tasks and their environments in the real world is challenging because the generated language needs to reflect what is happening with the entities in the task environment while also considering behavioral context, which can change based on current states or interactions between them.

To summarize there are two major contributions in this work:

- We prove that adding extra modality of joint control significantly improves the natural language description of the robot task as compared to the current state of the art models which uses image only.
- We show that our approach of fusing the image and joint control modality is efficient and leads to better results than just simply concatenating the data sources.

Chapter 2

BACKGROUND

This work lies at the intersection of vision, language, and robotics. The advancement of vision and language in deep learning has allowed us to solve complex problems like image captioning and vqa. This fascinating area in machine learning allows computers to learn how to process unstructured data such as images and language, where it was earlier challenging to learn. Traditionally rule-based systems were used to solve natural language tasks. One such example is SHRDLU Winograd (1972), a computer program that allowed humans to interact with natural language commands. This was one of the early examples of artificial intelligence where an average human could interact with a computer without being an expert or needing to code. Algorithms have been a vital part of our society for decades. However, plenty of approaches focused on symbolic language processing through formal grammar, parsing, or semantic analysis in the early days Jurafsky and Martin (2009). The invention of grounded communication protocols Steels (2003) and translating English language commands to robot executable plan specifications Kress-Gazit *et al.* (2008) was an important step in the improvement of robots. Whereas some methods also used language to learn from demonstrations Dillmann and Friedrich (1996).

Human-robot interactions are often vulnerable to flaws in the process when people do not trust what they say or do. The work Martelaro *et al.* (2016); Javaid and Estivill-Castro (2021) shows use of natural language helps alleviate these issues, allowing for greater transparency and honesty between humans and robots. However, most of the research in this area is mainly towards generating robot control sequences using natural language. The work Stepputtis *et al.* (2020); Lynch and Sermanet (2021) uses

natural language commands to generate robot control signals. Commands such as: "quickly go the blue bowl" require the model to localize the position of the bowl and understand the contextual meaning of the words "quickly," "blue bowl," and execute a robot trajectory. Therefore, learning from vision and language to generate robot control signals. Papers like Tellex *et al.* (2020); Liu and Zhang (2019) show some difficulties as well as progress in the field of language and robotics.

Some of the early image captioning methods Vinyals *et al.* (2017); Johnson *et al.* (2015) involved using a recurrent neural network amalgamated with convolution neural networks to serve as language and vision models respectively. The CNNs were used to extract features from the image whereas RNNs were used to take the features as input and produce a sentence sequentially. Afterwards the NLP domain exploded with the invention of attention and transformer networks Vaswani *et al.* (2017), where models were able to attend to the words of interest and learn the context in the sentence. From the advent attention and transformers, BERT Devlin *et al.* (2019) and GPT like models were introduced which showed superior performances in language based tasks. Some of the work got extended towards analysing image streams or videos to generate language which required understanding dynamics of actions done by humans. The work in Huang *et al.* (2020) uses the extraction of captions from various instructional image streams.

Our work uses inspiration from the most recent advances of natural language processing and deep learning, significantly leveraging the attention mechanism at its core which is also common in many of the state-of-the-art models like BERT and GPT. We share the same motivation in Yoshino *et al.* (2021) but differ in our approach as we mainly focus on making an attention-based model as compared to clustering through K-means and chunking the sentence . This work is also similar to XAI Chakraborti *et al.* (2021) but differ in addressing the scope, our work is focused

on generating language description of the robot task rather than focusing on how the robot failed in it's task or why a robot cannot complete it's assigned task. Our research is emphasized on generating the robot scene description from the available modalities of vision and robot control.

Chapter 3

PROBLEM STATEMENT AND APPROACH

In our work we propose a function $f(x)$ which takes in the start image $\boldsymbol{I}_{start}$, end image $\boldsymbol{I}_{end}$ and the joint angles of the bimanual robot $\boldsymbol{J}_\theta$ across t time steps where $\boldsymbol{J}_\theta \in \mathbb{R}^{t\times 14}$, to produce a sentence S describing attributes like action, agent, object, object color, direction and quantity of the robot task. $\boldsymbol{I}_{start}$, $\boldsymbol{I}_{end}$ are the images captured at the start and end of the robot task respectively whereas $\boldsymbol{J}_\theta$ is the joint angles and gripper states captured across time steps t of the entire robot task from each of the 6-DOF UR5 robots. With images and robot motions as the two input modalities we learn a function $f(x)$ through a neural network frame work to produce a sentence S. We model the function in a recurrent manner such that

$$\boldsymbol{w}_{n+1} = f\left((\boldsymbol{I}_{start}, \boldsymbol{I}_{end}), \boldsymbol{J}_\theta, \boldsymbol{w}_{0\to n}\right) \tag{3.1}$$

where w_{n+1} is the next tokenized word and w_0 is $< sos >$, the start of sentence token. The network keeps producing the word w until $< eos >$, the end of sentence token is generated or the sequence reaches the max length of 27.

We train the network until convergence, once trained given a start $\boldsymbol{I}_{start}$, end $\boldsymbol{I}_{end}$ image of the robot task and joint angle sequence $\boldsymbol{J}_\theta$ our network is able to generate a sentence describing the robot task. The description contains key attributes of the robot task such as action, agent, object, object color, direction and quantity poured which the network is able to predict under new robot task environment. We also test our model for generalizability by varying robot motion, lighting conditions, geometry of objects and unseen object, action pair. Figure 3.1 shows the architecture of our model where can see the overview of our approach.

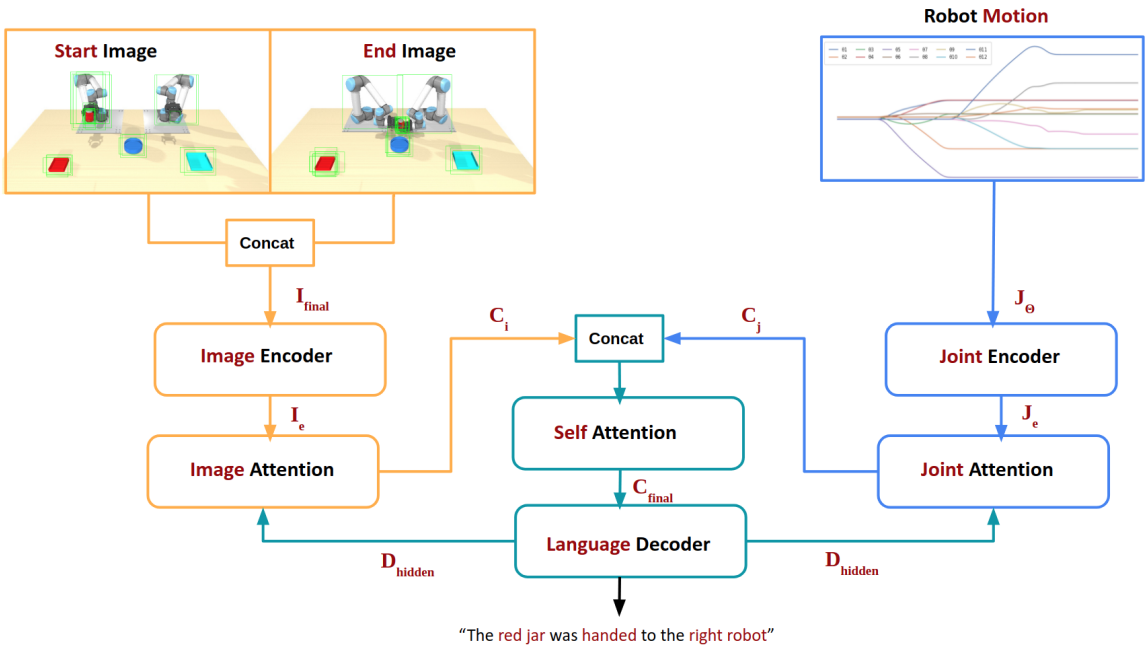

Figure 3.1: Overview of the Model Architecture

For Data pre-processing we use FRCNN Wu *et al.* (2019) that uses a pre-trained ResNet-101 backone trained on the imagenet dataset and then fine tuned on our own dataset to get regions of interest from the start and end image. The input images are resized to 299×299 with a normalized range of $[-1, 1]$ and centring each color channel to zero. We take the feature vector of first 25 regions of interest from FRCNN of each start, end image and concatenate them to make the final image feature vector. The joint angle sequences are padded with zeroes to have a max length of 270. Similarly, the sentences used for training are tokenized and padded to a max length of 27 where we add $< sos >$, start of sentence and $< eos >$, end of sentence token respectively. After the tokenization , we end up creating a dictionary D of words N, where our recurrent language decoder predicts the next word by generating a distribution of probabilities across the N words or classes. At inference we convert the tokens to words from the dictionary D and append each word to create the sentence. We

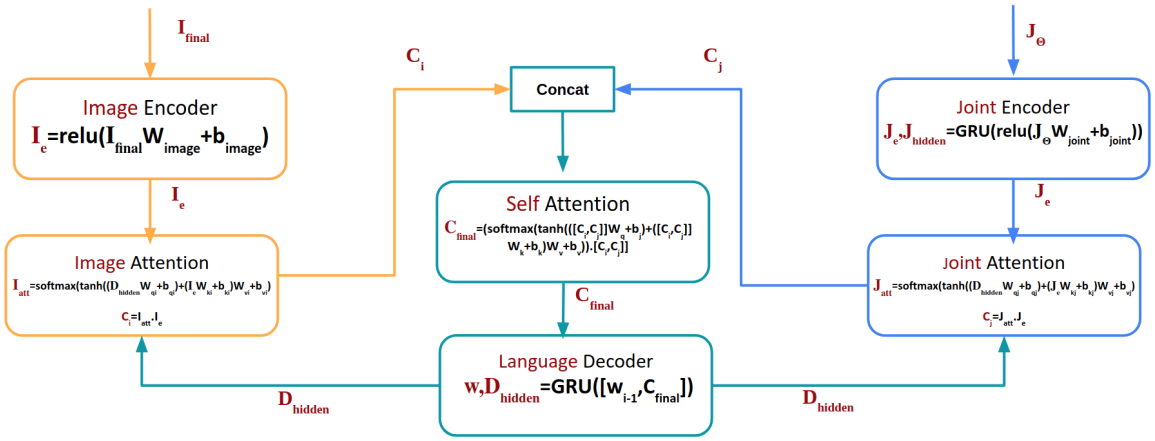

Figure 3.2: Working of Each Module in the Model Architecture

terminate the prediction of words when the sentence reaches a max length of 27 or an $< eos >$ is generated. Figure 3.2 describes in depth working of each of the modules in the network architecture.

3.1 Image Encoder

We retrieve the feature vectors of the start image I_{start} and end image I_{end} from the FRCNN respectively. We then choose the top 25 regions of interest from each image where each feature vector is of 1024 dimensions i.e $I_{start} \in R^{25\times1024}$ and $I_{end} \in R^{25\times1024}$. Finally we concatenate both of the these latent representations of the images such that we get $I_{final} \in R^{50\times1024}$. I_{final} is processed by going through a fully connected layer to get an encoded image embedding I_e, where

$$\boldsymbol{I}_e = \text{relu}(\boldsymbol{I}_{final}\boldsymbol{W}_{image} + \boldsymbol{b}_{image}) \tag{3.2}$$

here $\boldsymbol{W}_{image} \in \mathbb{R}^{50\times1024}$ and $\boldsymbol{b}_{image} \in \mathbb{R}^{1024}$ are variables which are learned by the network.

3.2 Joint Encoder

The joint encoder module encodes the robot motion $J_\theta \in R^{270\times14}$ by processing it to a GRU cell containing 1024 hidden units such that

$$\boldsymbol{J}_e, \boldsymbol{J}_{hidden} = \text{GRU}(\text{relu}\,(\boldsymbol{J}_\theta \boldsymbol{W}_{joint} + \boldsymbol{b}_{joint})) \tag{3.3}$$

here $\boldsymbol{W}_{joint} \in \mathbb{R}^{14\times1024}$ and $\boldsymbol{b}_{joint} \in \mathbb{R}^{1024}$ are variables which are learned by the network. J_e is the encoded robot motion representation where as J_{hidden} is the hidden state of the GRU cell, this hidden state would later be used to initialize the language decoder.

3.3 Attention

We use an attention mechanism to generate our natural language descriptions, out of all the available attention mechanisms we incorporate an additive approach, where our key and query are added to get the value. The value is then processed through a softmax function which we treat as attention weights. These attention weights are multiplied with our latent representations of our modalities, which is then further processed by our language decoder to generate sentences. In total we perform image attention, joint attention and self attention, the following sections would discuss each of the attention modules in detail.

3.3.1 Image Attention

We incorporate image attention by treating the hidden state of our language decoder D_{hidden} as query and the image encoded vector I_e as key and perform additive attention such that

$$\begin{aligned}\boldsymbol{I}_{att} = \text{softmax}(\tanh\,((\boldsymbol{D}_{hidden}\boldsymbol{W}_{qi} + \boldsymbol{b}_{qi}) + \\ (\boldsymbol{I}_e\boldsymbol{W}_{ki} + \boldsymbol{b}_{ki})\boldsymbol{W}_{vi} + \boldsymbol{b}_{vi})\end{aligned} \tag{3.4}$$

here $\boldsymbol{W}_{vi} \in \mathbb{R}^{1024\times1}$ and $\boldsymbol{W}_{qi}, \boldsymbol{W}_{ki} \in \mathbb{R}^{1024\times1024}$ and $\boldsymbol{b}_{vi} \in \mathbb{R}^{1}$ and $\boldsymbol{b}_{ki}, \boldsymbol{b}_{qi} \in \mathbb{R}^{1024}$ are variables which are learned by the network. I_{att} are the image attention weights which is then multiplied by the encoded image representation I_e to get the image context vector such that

$$\boldsymbol{C}_i = \boldsymbol{I}_{att} \cdot \boldsymbol{I}_e \tag{3.5}$$

where the multiplication is the dot product between I_{att} and I_e

3.3.2 Joint Attention

Just like the image attention we incorporate joint attention by treating the hidden state of our language decoder D_{hidden} as query and the joint encoded vector J_e as key and perform additive attention such that

$$\begin{aligned}\boldsymbol{J}_{att} = \text{softmax}(\tanh((\boldsymbol{D}_{hidden}\boldsymbol{W}_{qj} + \boldsymbol{b}_{qj}) + \\ (\boldsymbol{J}_e\boldsymbol{W}_{kj} + \boldsymbol{b}_{kj})\boldsymbol{W}_{vj} + \boldsymbol{b}_{vj})\end{aligned} \tag{3.6}$$

here $\boldsymbol{W}_{vj} \in \mathbb{R}^{1024\times1}$ and $\boldsymbol{W}_{qj}, \boldsymbol{W}_{kj} \in \mathbb{R}^{1024\times1024}$ and $\boldsymbol{b}_{vj} \in \mathbb{R}^{1}$ and $\boldsymbol{b}_{kj}, \boldsymbol{b}_{qj} \in \mathbb{R}^{1024}$ are variables which are learned by the network. J_{att} is the joint attention weights which is then multiplied by the encoded robot motion representation J_e to get the joint context vector such that

$$\boldsymbol{C}_j = \boldsymbol{J}_{att} \cdot \boldsymbol{J}_e \tag{3.7}$$

where the multiplication is the dot product between J_{att} and J_e

3.3.3 Self Attention

Finally we perform an additive self attention mechanism by concatenating the image context vector C_i and the joint context vector C_j and treat the concatenated vector as both the query and key. The earlier attention modules extract all the information necessary from both of the modalities which are images and robot motion.

The self attention module extracts the information from the context vectors of each of these modalities and attends to the features which are necessary to generate an optimal sentence. We get the final context vector C_{final} such that

$$\begin{aligned} \boldsymbol{C}_{final} = (\text{softmax}(\tanh\left(([C_i, C_j]\boldsymbol{W}_q + \boldsymbol{b}_q) + \right. \\ \left. ([C_i, C_j]\boldsymbol{W}_k + \boldsymbol{b}_k)\boldsymbol{W}_v + \boldsymbol{b}_v\right)) \cdot [C_i, C_j] \end{aligned} \tag{3.8}$$

I_{final} and J_θ remains the same for a given predicted sentence where as the image context vector C_i , the joint context vector C_j and the final context vector C_{final} keeps changing as the language decoder predicts new words and hidden states for each of its previous words and hidden states from the images and robot motion as inputs. The following section would discuss the language decoder in detail.

3.4 Language Decoder

The attention modules and the language decoder work together to generate sentences. We start with tokenizing the start of sentence token $< sos >$ and processing with an embedding layer initialized with our vocab length 90 and embedding dimensions of 256. After processing the tokenized word to get a word embedding w, we concatenate the word embedding with the final context vector C_{final} and process it through a GRU cell of 1024 units such that

$$\boldsymbol{w}, \boldsymbol{D}_{hidden} = \text{GRU}([\boldsymbol{w}_{i-1}, \boldsymbol{C}_{final}]) \tag{3.9}$$

we initialize the language decoder's hidden state D_{hidden} with the hidden state of the joint encoder J_{hidden} where w_{i-1} is the previous predicted word token processed after passing through embedding layer. Finally we process w through a fully connected layer of 90 units and choose the word with the maximum score. The decoder keeps on generating the words until it predicts the end of sentence token $< eos >$ or the sequence length reaches a max length of 27, we then append the predicted words to

form the final sentence. To get better results, we incorporate beam search with a beam width of 5 to get sentences with the optimal probability by summing up the log probabilities of each word and choosing the sentence containing the maximum score.

3.5 Training

Our training set contains the concatenated start and end image of the robot task I_{final}, the robot motion of the entire robot task J_θ and the sentence S. I_{final} and J_θ are inputs to our model where our objective is to learn to predict the sentence S. For training our network we choose a batch size of 32 and predict the sentences by optimizing over the sparse categorical cross-entropy loss. We use teacher forcing by giving the correct target input to the decoder and remove it while evaluation. This helps the model for faster convergence and learning the correct sequence of words. Finally we use adam optimizer with a learning rate of 0.001. The dense layer in the image encoder, and the GRU cells in the joint encoder and language decoder are initialized to 1024 units. We train the whole network on a single P100 GPU until it reaches convergence.

Chapter 4

EVALUATION AND RESULTS

We train and evaluate our model in a simulation environment where we create a bi-manual robot setup with two UR5 robots. Both the robots have 6 degrees of freedom, we also collect the gripper states of each of these robots resulting in a 14 dimensional robot control vector for each time step. We collect the robot motion data as well as the start and end image of the robot task where the robot tasks involve actions like picking, pouring, opening, twisting and exchange. Tasks like picking and pouring involve only one robot executing the task where as tasks like twisting and exchange involve a joint collaboration between the two UR5 robots in-order to execute the task successfully. We randomize the positions of the objects as well as their colors and collect the training data. We train our model on the following robot tasks:

- **Pick:** Here one of the robot agent picks up an object
- **Exchange:** Here one of the robot agent exchange an object with the other robot agent
- **Twist:** Here one of the robot agent twists the jar lid while the other robot agent holds the jar
- **Open:** Here one of the robot agent opens the lid of the saucepan and places at either right or left of the saucepan
- **Pouring:** Here one of the robot agent pours an object into one of the bowl either partially or fully.

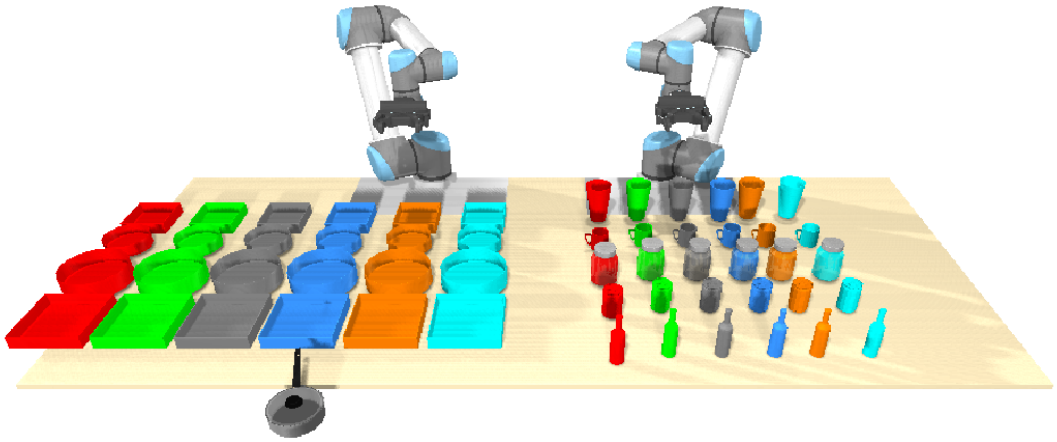

Figure 4.1: All Objects Used for the Robot Tasks

The sentences predicted from our model describes the robot task incorporating various attributes of the task environment such as object, object color, direction, agent, action and quantity poured. One such example is *The left robot partially poured the red mug into the small orange circular bowl at the left of the table* containing all the attributes necessary for describing the robot task.

We use Coppeliasim Rohmer *et al.* (2013); James *et al.* (2019) to generate our robot task enivronment, this simulation environment helps us to perform the robot task with a physics engine capable to capture the dynamics in the environment. In total, we use 6 colors which are red, turquoise, blue, green, gray , orange and 5 objects that can be picked up: mug, sod can, jar, glass, bottle and 5 bowls to pour these objects in: big circular bowl, small square bowl, small circular bowl and big square bowl. We use a jar that has a lid which can twisted to open. We also have a saucepan lid that can opened. Figure 4.1 shows all the possible objects used to perform the robot tasks. We collect 10,000 demonstrations of the 5 robot tasks for training and use 1500 demonstration for evaluation.

To train our model we generate sentences that describes the robot task. The sen-

tences are generated by a python template by taking in the attributes from the simulation environment and creating a sentence. The template uses synonyms and various different sentence structure to incorporate a good variety.This variety is achieved by randomly selecting synonyms for different words and also selecting random template structure. Each demonstration is annotated with 4 sentence resulting in 40,000 data points from the 10,000 demonstrations for training.

4.1 Sentence Generation

Sentence generation for training our model is done through a template which has the simulation environment variables as its input. Overall there are 6 variables or tags: <Action>,<Agent>,<Direction>,<Object>,<Color>,<Quantity>. Each tag can have different values depending on the randomization of the robot task, for example the <Action> tag can be <Pick>,<Exchange>,<Twist>,<Open>,<Pour> whereas the <Agent> tag can be <Left Robot>,<Right Robot>. Depending on each task a different set of tags are used to create a sentence, for example the <Quantity> tag is only used in the Pour task whereas the <Color> tag is not used in the Open task. These tags are later retrieved from the predicted sentence to compare the model accuracies which we will be discussing in the Evaluation Metric section. Table 4.1 shows the tags used in each task and below are some example sentences generated by the template for their respective task:

- **Pick:** The $< left\ robot > < took >$ the $< red > < mug >$ from the $< front >$ of the table.
- **Exchange:** The $< blue > < mug >$ was $< handed >$ to the $< right\ robot >$.
- **Twist:** A $< jar\ lid >$ was turned by the $< right\ robot >$

- **Open:** The $< left\ robot >$ opened the $<$ *saucepan lid* $>$ and place it near $< right >$ of the saucepan.

- **Pouring:** The $<$ *blue* $> <$ *bottle* $>$ was $<$ *partially* $> <$ *poured* $>$ by the $<$ *right robot* $>$ into the $<$ *small* $> <$ *gray* $> <$ *circular* $>$ bowl at the $<$ *top right* $>$ of the table.

Note that the <Direction> tag used in pick and pour task defines the location of the object on the table whereas the same tag for the open task is used to define the relative position of the saucepan lid with respect to the saucepan. Also, twist and open task has only one object associated with the <Object> tag i.e jar lid and saucepan lid and no color tag is being used in these tasks.

4.2 Evaluation Metric

To test our model on new environments, new scenarios are generated and given to the trained model to produce a verbal description of the task. This description is evaluated by reducing all predicted words to their respective tags and then compare them with the ground truth tags. Intuitively, the template engine used to generate sentences from the tags which we get from the simulation environment is now being used as an inverse template engine to get the tags from the predicted sentence. After getting the predicted tags we compare it with the ground truth tags for each task and calculate the percentage of the correct comparisons. The inverse template engine accounts for the usage of different tags for each task as described in Table 4.1 and retrieves the tags across all the 5 tasks. We add up all the correct comparisons in each task and put them in the Action, Agent, Direction, Object, Color and Quantity Metric accordingly. Note that the <Agent1>,<Agent2> tag in Exchange task (refer 4.1) would be counted in the Agent metric and the <Color1>,<Color2>,<Object1>,<Object2>

Task	Tags
Pick	<Action>,<Agent>,<Color>,<Object>,<Direction>
Exchange	<Action>,<Agent1>,<Agent2>,<Color>,<Object>
Twist	<Action>,<Agent>,<Object>
Open	<Action>,<Agent>,<Object>,<Direction>
Pour	<Action>,<Agent>,<Color1>,<Object1>,<Color2>,<Object2>,<Direction>,<Quantity>

Table 4.1: Usage of Tags for Each Action

tag in the Pour task would be counted in Color and Object Metric respectively. We also add two more Metrics Object & Color and ALL, the Object and Color metric would be calculating if the predicted sentence has both object and color correct while the ALL metric would be calculating if the predicted sentence has all the tags correct or not. For example, for the Twist task the ALL metric would calculate if the predicted sentence has all the tags involved to make that sentence i.e <Action>,<Agent>,<Object> correct or not. In short, the individual metrics would be looking for individual attributes/tags in the predicted sentence while the ALL metric would be looking if the predicted sentence has every attributes/tags correct or not, its a much stricter condition. Along with our testing metric we also evaluate our model on BLEU, ROUGE, METEOR and CIDEr. Evaluation on BLEU has some drawbacks as the metric evaluates the sentences on n-grams where it looks for word to word similarity, this doesn't help if the sentences are different in structure or if they are using words which are synonymous. It also doesn't help us to know what aspect of attributes/tags the model is failing to predict as compared to our metric which looks different attributes/tags in the predicted sentence for the defined robot task.

	Model	Features		Described Attributes in Predicted Sentence							
		Ctrl.	Img.	Action	Agent	Dir.	Obj.	Clr.	Quant.	Obj./Clr.	All
1	M^{2}T Cornia *et al.* (2020)		✓	**100.0**	**100.0**	86.42	95.94	**98.83**	99.33	92.75	87.00
2	M^{2}T Cornia *et al.* (2020)	✓	✓	**100.0**	**100.0**	87.55	95.89	98.08	99.66	92.00	87.14
3	Ours	✓		**100.0**	99.80	92.13	50.08	19.42	**100.0**	7.320	47.10
4	Ours		✓	98.60	96.20	76.87	**99.94**	97.80	98.66	**97.71**	83.41
5	**Ours**	✓	✓	**100.0**	98.60	**95.87**	99.88	96.20	98.00	96.11	**93.20**

Table 4.2: Results and Comparisons Against Baseline Methods

	Model	Described Attributes in Predicted Sentence							
		Action	Agent	Dir.	Obj.	Clr.	Quant.	Obj./Clr.	All
1	**Cogent Action**	100.0	100.0	96.00	93.33	92.56	92.00	87.16	81.00
2	**Lighting**	100.0	100.0	96.72	100.0	78.46	95.00	78.46	83.16
3	**Object Variation**	100.0	99.00	88.23	81.74	86.84	96.00	67.10	67.32
4	**Motion Variation**	98.66	100.0	72.85	98.99	68.08	87.75	66.66	52.66

Table 4.3: Generalization Results

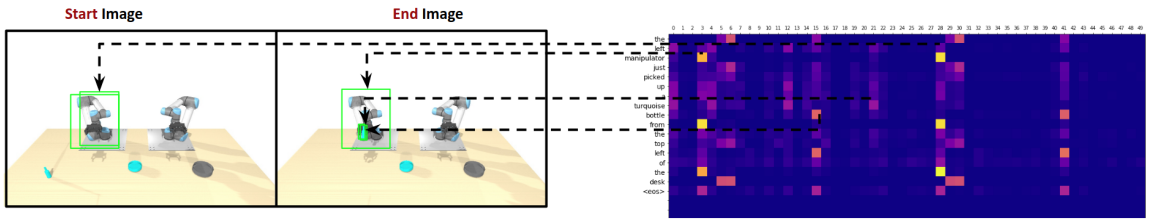

Figure 4.2: Visualization of Image Attention Weights

4.3 Ablations and Visualizations

The results of our model can be seen in Table 4.2. We studied the influence of each modality by creating separate models for each input modality. The model in row(3) takes in just the sequences of robot control from start to end of the robot task while the model in row(4) takes the pre and post conditioned image features of the task from FRCNN. The results show clear accuracy improvements over the single modalities when combining robot control and vision data in our model row(5) (93.20%). By analyzing the distributed attributes in the predicted sentence we see that the control modality is really good in predicting the spacial attribute (direction) (92.13%) as well as the action (100%), agent (99.80%) and quantity (100%) in the sentence. But it lacks in object and color category (7.32%), which make sense since there is no way the model can know both object and color. [Note the 50.08 % accuracy in model row(3) is slightly high because the open and twist task has only one object in them (saucepan lid for open and jar lid for twist), also the object & color accuracy is 7.32% whereas ALL metric shows 47.10% this is because for the task open and twist, the object & color accuracy doesn't apply since the color attribute is not present in describing these tasks]. On the other hand the Image Only model row(4) is really good in predicting both object and color (97.71%) for the robot task. Figure 4.2 shows the visualizations of the attention weights from the Image only model (row 4), where each word is attending to different image regions procured from FRCNN. The left image in Figure 4.2 shows the region of interest attended over the words, 'left', 'machine', 'turquoise' and 'bottle', whereas the right image shows the region of interest attended across all the words. Hence by combining the best of both the models we see a significant increase in the overall accuracy (All metric) (93.20%) of our model row(5) as compared to the control only model which has an

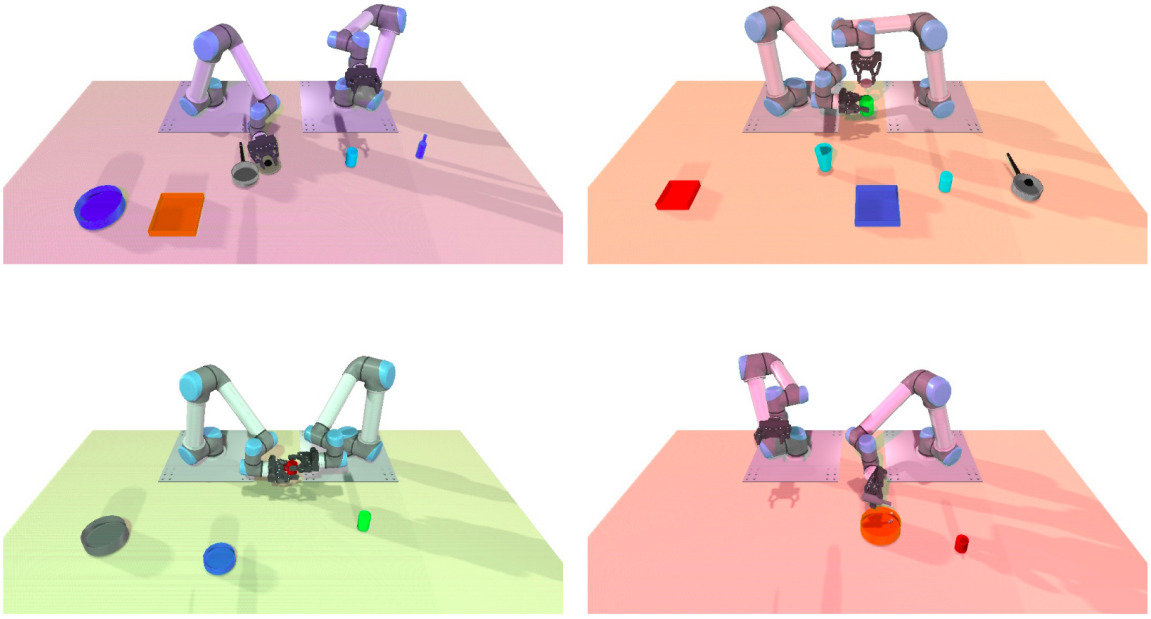

Figure 4.3: The Various Robot Tasks in Different Lighting Conditions

overall accuracy of 47.10% and the image only model which has an overall accuracy of 83.41% in describing the robot task. With this study we can say that our model is able to attend to relevant parts in each of the image and control modalities and combine them to get best results for describing the robot task.

4.4 Baselines

We compared our model with a state of the art image captioning model called "meshed-memory transformer" [25] Table 4.2 (row 1), we also compare a multimodal meshed memory transformer row(2) by adding the robot control modality to the model by concatenating robot control sequences with the FRCNN Image features that we get from the pre and post conditioned image of the robot task. Meshed-memory transformer works by learning multi-level representation of the relations between the image regions and by concatenating the robot control we expect it to also learn rela-

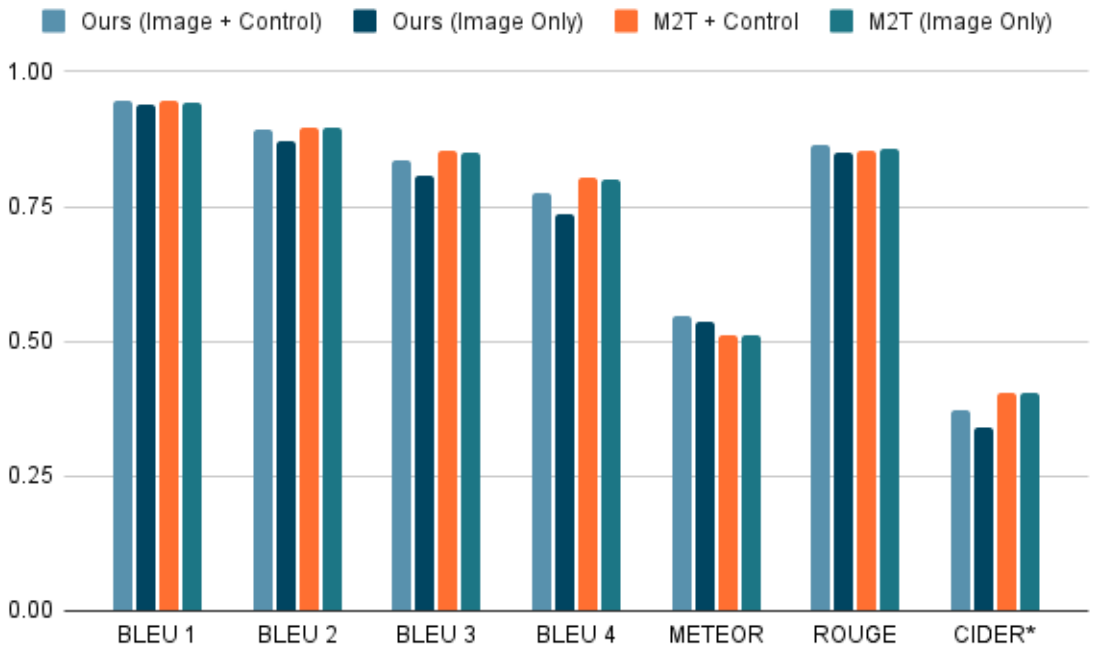

Figure 4.4: Results on Metrics Like Meteor, Bleu, Rouge and Cider

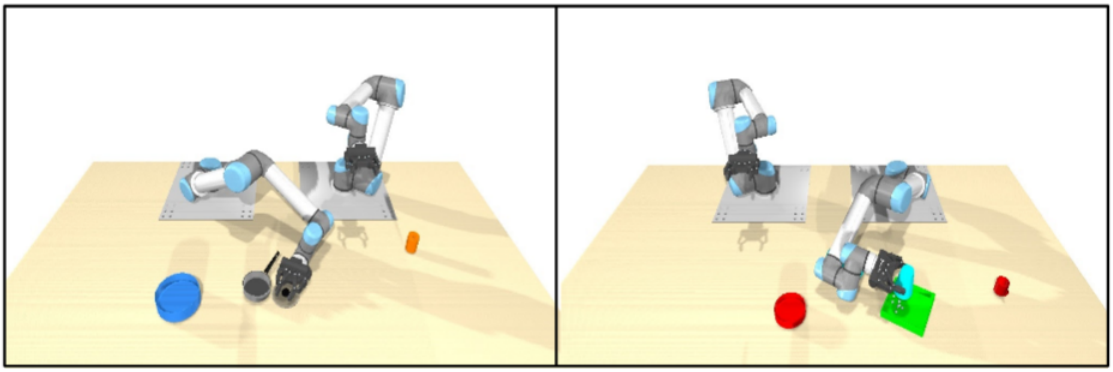

Figure 4.5: Robot Task Showing Change in Robot Motion

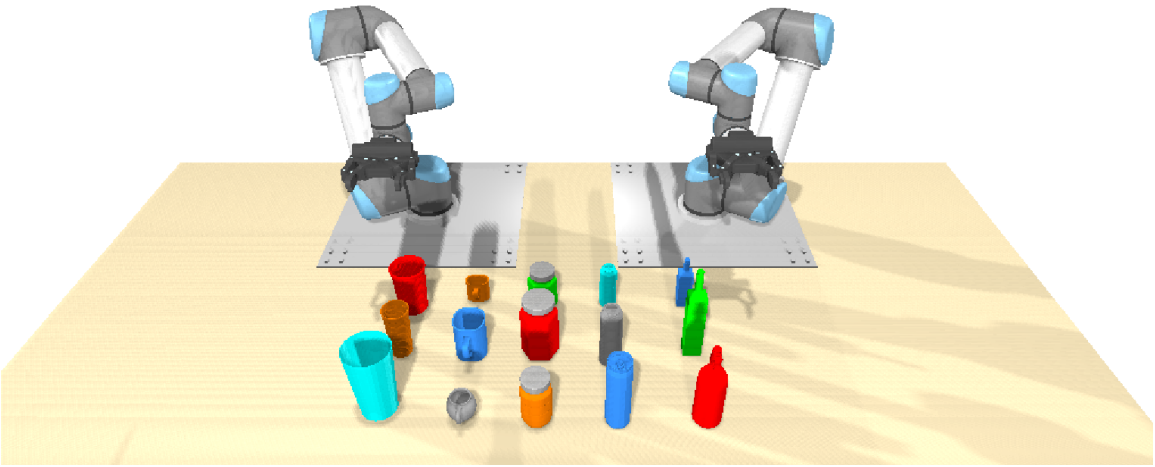

Figure 4.6: New Objects with Geometric Variations

tion between the image regions as well as robot control to produce better predictions. Table 4.2 shows that our multimodal model (row 5) significantly outperforms the two baselines which can be seen by comparing the All metric, 93.20% for our model row(5), 87.00% row(1) and 87.14% row(2) for M^2T and M^2T + control respectively. Furthermore we can see that between meshed memory image model (row 1) (87.02%) and meshed memory image and control model (row 2) (87.14%), there is not much improvement by adding the robot control modality. This suggests that we combine the image and robot control modalities effectively as compared to just simply concatenating them as inputs to a state of the art image captioning model. From the attributes predicted in sentence , we can confirm that the robot control modality helps in improving the spacial description(direction) in the predicted sentence. Apart from our evaluation metrics which looks at the described attributes in the sentence, we also evaluate on BLEU, METEOR, ROUGE and CIDEr scores, the result is shown in Figure 4.4. We see that BLUE and ROUGE have very similar scores and also the difference in scores between the models are not very much, this brings back to some of the drawbacks in BLUE as discussed before.

4.5 Generalizations

Subsequently, we evaluated our model's ability to generalize over unseen action object pairs, visual perturbations, object variations and motion variations. The results can be seen in Table 4.3. For Cogent Action we held out glass in exchange action and jar in pour action while training and test our model with this new object action pair. For visual perturbations we change the lighting conditions of the robot task to have random values across the three rgb channels, Figure 4.3 depicts the robot task in different lightings. We see the model performs really well in these different light conditions with an overall accuracy of 81.00% across 100 test cases. For the motion variation we tested our model by changing the robot's trajectory, this was achieved by rotating the object of interest -40 to +40 degrees and let the robot perform the task, figure 4.5 shows the change in robot motion. We tested the change in robot motion test across 150 test cases for Pick, Open and Pour tasks. Here we see our model was 52.66% accurate in describing the robot task. We also test our model on new object variations, these variations were achieved by varying the objects geometrically from their base forms, Figure 4.6 shows the new object variations. Here our model performs with an accuracy of 67.32% tested over 100 cases. For object and lighting variation we believe the decrease in overall accuracy was caused due to model's decreased accuracy in detecting both object and color attributes whereas for motion variation we see the decrease in overall accuracy was caused by the decrease in direction and color attributes in the predicted sentence.

4.6 Stochastic Forward Pass

We also assess our model's confidence in predicting sentences. This was achieved by performing a stochastic forward pass in the model by activating the dropout layer

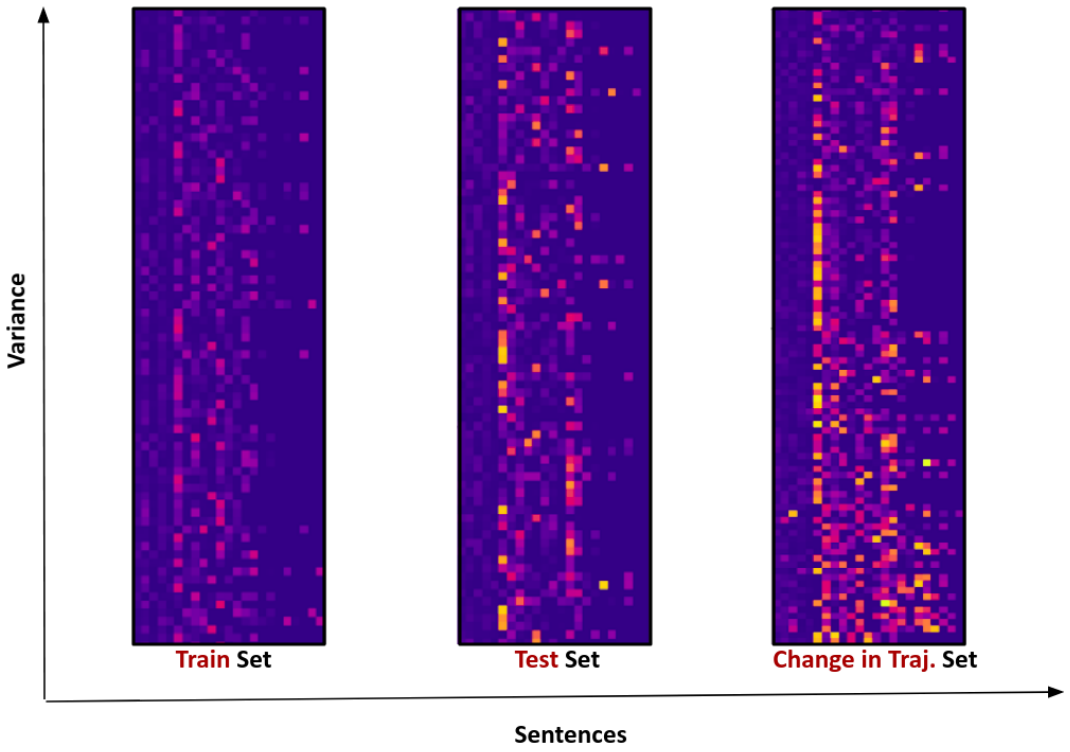

Figure 4.7: Stochastic Forward Pass

at inference time and running 500 passes. Figure 4.7 shows the stochastic forward pass analysis where the figure shows the results of the passes on 3 different sets of data: train set, test set, change in trajectory set. Here each row is a predicted sentence and the grids in each row show variance of each word in that sentence. The brightness of each grid is proportional to the predicted word, the higher the brightness, the higher the variance. We can clearly see here that the brightness decreases as we go from train set to change in trajectory set showing how the model was certain when it was predicting the train set , and how the confidence decreases as we test our model on edge cases. We can basically see the confidence of the model dropping as the input data goes from a known distribution to out of distribution.

Chapter 5

CONCLUSION

We used pre and post conditioned images and robot motion to describe a robot task. We found from experiments that adding the robot motion modality significantly increased the quality of our task descriptions. The robot control modality is really good in extracting the spatial features in the task environment which is otherwise very hard from images to capture. We also found that our way of fusing the modality is more efficient than just putting robot motion data in the existing state of the art image captioning model. We experimented with few generalizability tests, one of them included an unseen object and action pair i.e the objects 'jar' and 'glass' were missing in the training dataset for pour and exchange actions respectively and here our model was able to take parts of the available information and combine them to create a completely new sentence which it was never trained on. In future work we plan to generalize our task over multiple robots as well as describe failure cases.

REFERENCES

Chakraborti, T., A. Kulkarni, S. Sreedharan, D. E. Smith and S. Kambhampati, "Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior", Proceedings of the International Conference on Automated Planning and Scheduling **29**, 1, 86–96, URL https://ojs.aaai.org/index.php/ICAPS/article/view/3463 (2021).

Cornia, M., M. Stefanini, L. Baraldi and R. Cucchiara, "Meshed-memory transformer for image captioning", (2020).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pp. 4171–4186 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2019), URL http://aclweb.org/anthology/N19-1423.

Dillmann, R. and H. Friedrich, "Programming by demonstration: A machine learning approach to support skill acquision for robots", in "International Conference on Artificial Intelligence and Symbolic Mathematical Computing", pp. 87–108 (Springer, 1996).

Huang, G., B. Pang, Z. Zhu, C. Rivera and R. Soricut, "Multimodal pretraining for dense video captioning", in "AACL", (2020).

James, S., M. Freese and A. J. Davison, "Pyrep: Bringing v-rep to deep robot learning", arXiv preprint arXiv:1906.11176 (2019).

Javaid, M. and V. Estivill-Castro, "Explanations from a robotic partner build trust on the robot's decisions for collaborative human-humanoid interaction", Robotics **10**, 1, URL https://www.mdpi.com/2218-6581/10/1/51 (2021).

Johnson, J., A. Karpathy and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning", (2015).

Jurafsky, D. and J. H. Martin, *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition* (Pearson Prentice Hall, Upper Saddle River, N.J., 2009), URL http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y.

Kress-Gazit, H., G. E. Fainekos and G. J. Pappas, "Translating structured english to robot controllers", Advanced Robotics **22**, 12, 1343–1359 (2008).

Liu, R. and X. Zhang, "A review of methodologies for natural-language-facilitated human–robot cooperation", International Journal of Advanced Robotic Systems **16**, 3, 1729881419851402, URL https://doi.org/10.1177/1729881419851402 (2019).

Lynch, C. and P. Sermanet, "Language conditioned imitation learning over unstructured data", Robotics: Science and Systems URL https://arxiv.org/abs/2005.07648 (2021).

Martelaro, N., V. Nneji, a. Ju and a. Hinds, "Tell me more: Designing hri to encourage more trust, disclosure, and companionship", pp. 577–577 (2016).

Rohmer, E., S. P. N. Singh and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework", in "Proc. of The International Conference on Intelligent Robots and Systems (IROS)", (2013), www.coppeliarobotics.com.

Steels, L., "Evolving grounded communication for robots", Trends in cognitive sciences **7**, 7, 308–312 (2003).

Stepputtis, S., J. Campbell, M. Phielipp, S. Lee, C. Baral and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks", in "Advances in Neural Information Processing Systems", edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, vol. 33, pp. 13139–13150 (Curran Associates, Inc., 2020), URL https://proceedings.neurips.cc/paper/2020/file/9909794d52985cbc5d95c26e31125d1a-Paper.pdf.

Tellex, S., N. Gopalan, H. Kress-Gazit and C. Matuszek, "Robots that use language", Annual Review of Control, Robotics, and Autonomous Systems **3**, 1, 25–55, URL https://doi.org/10.1146/annurev-control-101119-071628 (2020).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is all you need", in "Advances in Neural Information Processing Systems", edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, vol. 30 (Curran Associates, Inc., 2017), URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vinyals, O., A. Toshev, S. Bengio and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge", IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 4, 652–663, URL http://dx.doi.org/10.1109/TPAMI.2016.2587640 (2017).

Winograd, T., *Understanding Natural Language* (Academic Press, Inc., USA, 1972).

Wu, Y., A. Kirillov, F. Massa, W.-Y. Lo and R. Girshick, "Detectron2", https://github.com/facebookresearch/detectron2 (2019).

Yoshino, K., K. Wakimoto, Y. Nishimura and S. Nakamura, *Caption Generation of Robot Behaviors Based on Unsupervised Learning of Action Segments*, pp. 227–241 (2021).

APPENDIX A

CODE REPOSITORY

https://github.com/kamaleshrathinam/Thesis.git