Review Paper: RT-2:Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

Chandra Mohan Jagadeesan
RWTH Aachen University
chandra.mohan.jagadeesan@rwth-aachen.de

***Abstract*—This paper reviews a group of models called by the authors as RT-2 (Robotic Transformer - 2) for robotic control. It is the next iteration of the Robotic Transformer-1 (RT-1) by the same authors. RT-1 introduced a way to learn robot policies for end-to-end control using transformers. RT-1 did not have any generalization capabilities and Chain-of-Thought reasoning capabilities. Getting large amounts of robotic training data to make the robots acquire those capabilities is costly and impossible soon. Since we know that large Vision Language Models and Large Language Models have those capabilities, the central theme of this paper is to combine Vision Language Models with RT-1 using action tokenization and study whether the robots get a boost in their generalization and semantic reasoning capabilities. Other papers tried to solve the issue only on high-level semantic reasoning; this paper uses a single model for end-to-end control. This approach of action tokenization and combining VLMs into RT1 leads to a new family of models called Vision Language Action (VLA) models.VLA models significantly improve the robot's generalization, emergent, and Chain-of-Thought reasoning capabilities, and they can also take any natural language command as input to the robot because of training with action tokens.**

***Index Terms*—Robotic task, Vision Language Action models, VLA, Action tokenization, end-to-end robotic control with VLMs.**

I. Introduction

Transformers [9] is a specific type of model architecture that is used widely for natural language processing and vision and text-based tasks. Transformers combine multi-head attention in an encoder-decoder architecture. Multi-head attention mechanisms capture the dependencies between the different inputs in the input sequence, making the transformers more efficient than LSTMs and eliminating the problem of vanishing gradients. RT-1 [2] (Robotics Transformer -1) learned policies to do end-to-end robotic tasks on seen objects, backgrounds, and environments by training the transformers to learn policies using behavioural cloning and action tokenization. However, the policies trained by this process could not generalize well over unseen objects, environments, and backgrounds. Furthermore, they did not have Chain of Thought reasoning and emergent capabilities. These capabilities are beneficial for a generalist robot.

How can the generalist robot acquire emergent capabilities and chain of Thought reasoning and use them for the tasks? One of the approaches could be to collect a lot of robotic data with robots doing various tasks in different settings and environments and try to train the model. This approach needs enormous amounts of data, which can take time. The second problem is that we do not know how to use that data to train the robots to do the tasks, as the data would have a lot of semantics, labels, texts, and images as inputs, which should be converted into Cartesian coordinates for the robot to work. We knew from other papers like GPT-3 that Large Language Models and Vision Language Models have Chain of Thought and emergent properties. In this paper, the authors have explored the possibilities of transferring the knowledge from those Large Vision Language Models to do low-level robotic control.

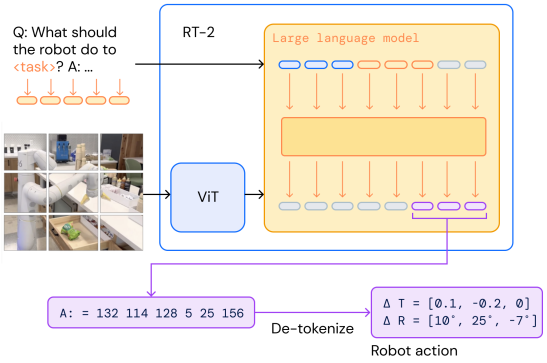

Fig. 1. RT-2 tokenization and detokenization [1]

There are a large number of vision language model types that are specific for many tasks and also for general tasks. In this paper, the authors have framed the problem as a Visual Question Answering with the images and questions as inputs for the Vision Language Model, and the robotic trajectory to follow would be the output from the model. In this regard, the authors have introduced tokenization for converting the actions into text tokens and giving them as inputs for to train the model. This model is then used for inference proves text tokens as output, which would be detokenized to get the action the robot. For Large Vision Language models, the existing trained models were taken and trained with the robotic data instead of training the whole model from the beginning. This method is called fine tuning and it was previously used by many natural language processing models to improve performance.

The family of models trained using this method is Robotic Transformer -2, and the policies from these models have shown significant improvements in doing the same tasks with unseen objects, environments, and backgrounds. They have also shown some emergent and chain-of-thought reasoning capabilities. Although the policies did not acquire new skills, the model could utilize them in new ways. the basic ides of RT-2 is shown in the fig1. By this way the knowledge from vision language models are transferred to the policy and also it used to make actions which is low level control of the robot.

II. Background

Behavioural cloning [10], a type of imitation learning, is used to train and get the policies in RT-1 and RT-2. This is a supervised learning method where the expert data for completing tasks are taken as the best way to perform the given task, and these demonstrations are used to train the model in a supervised machine learning method. In this way, the policy learned can be used for other similar tasks.

Let us formalize the behavioural cloning problem; first, we must know the basics of reinforcement learning. For reinforcement learning, we have an agent and an environment. The agent performs a set of actions that change the environment's state. This state change is given to the agent along with the reward. The agent tries to learn a policy to increase the expected cumulative reward function that it gets. Trajectory of actions, states and rewards from the start till the agent reaches the goal.

For many problems, reward shaping and defining a goal for the agent to accomplish are hard to define in reinforcement learning. The reinforcement learning problem was made into a behaviour cloning problem, and supervised learning was used to learn the policies. Two main steps are taken to make this transition.

1) The reward is kept as simple [0,1]. (i.e.) The reward for the agent is given as one when the agent accomplishes the task and zero otherwise. 2) The supervisor gives a set of demonstrations for every set of starting and end goals. These demonstrations are considered the best actions for the agent to take when the agent is in a particular state. In this way, the end goal of the agent would be to mimic the expert demonstrations, hence the term behavioural cloning.

In RT-1, transformers were used to learn the policy for performing pick and place tasks in a robot using behavioural cloning methods. In RT-2, the same method is used, but instead of using different models for processing vision, language and policy, one single large vision language model is used. By combining RT-1 and large vision language models, the authors developed Vision-language-action models for robotic control called RT-2.

III. Vision-Language Action Models

The following questions have to be answered to train Vision-Language-Action models,

- Which of the Pre-trained Visual Language Models to use?
- Which data must be used for fine-tuning?

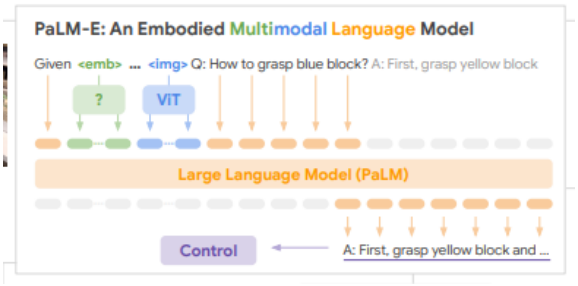

Fig. 2. PaLM-E Architecture [6]

- How to Perform fine-tuning or training?

A. Pre Trained Vision Language Models

There are mainly two types of Vision Language Models. 1) Representation learning models [12], which learn common embeddings between vision and text inputs, and 2) vision, text -¿ text models. The authors of the paper decided to use the second-type models for Vision Language Action models. This category of models is generally trained on many tasks like vision-question answering and image captioning tasks. For this paper, the authors chose PaLI-X [4] and PaLM-E [6] models as their pretrained-Vision Language Model. The resulting versions of RT-2 from the models are called RT-2-PaLI-X and RT2-PaLM-E.

1) PaLI-X: PaLI-X consists of ViT-22B for image processing. PaLI-X model can take a sequence of n images and convert them into n*k tokens. These tokens and text tokens are given as input for the encoder-decoder backbone with 32B parameters and 50 layers. This architecture processes to give the output texts.

2) PaLM-E: PaLM-E is the embodied version of PaLM (Pathways Language Model). PaLM is a Large Language model for performing high-level planning tasks in robots. In PaLM-E, the PaLM model is combined with the visual model made of ViT-4B. The vision and language instructions are then concatenated into textual space. Combining continuous variables with textual inputs makes it very useful for combining data from many different modalities, including sensor data. The architecture of PaLM-E is shown in fig**??**.

B. Data to be used for training

The datasets for RT-2 are a combination of the WebLI dataset and robotic data from RT-1. WebLi dataset consists of 10B image-text pairs in 109 languages. There are many vision question-answering tasks in it. These were the same datasets used for PaLI-X and PaLM-E. Respective datasets were used for the respective versions of RT-2. For training RT-2-PaLI-X, all the episodic data from the PaLI-X paper were removed.

The robotic dataset was taken from the previous paper RT-!. It consists of demonstration episodes collected from robots on the following tasks. 1) Pick Object 2) Move Object Near Object 3)Place Object Upright 4)Knock Object Over 5)Open Drawer 6)Close Drawer 7) Place Object into Receptacle 8)Pick object from Receptacle 9) Place on the counter.

For RT-2-PaLI-X training dataset consists of 50% robotics data and 50% webLI dataset whereas for RT-2-PaLM-E the amount of robotics data is 66%.

C. Fine tuning or training

The PaLI-X and PaLM-E models take images and text as inputs and give out text as outputs. Since the robotic controllers require the Cartesian coordinates of the end effector as inputs, Action tokenization is done.

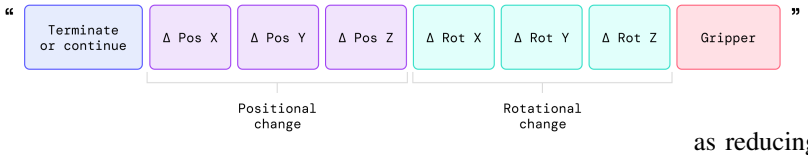

Fig. 3. RT-2 Action Encoder [1]

a) Action Tokenization: The action space of the robot consists of 6DoF positional and rotational displacement of the robot end effector plus the level of extension of the gripper. There is another command for terminating the episode. The continuous action spaces are discretized in 256 uniform bins.

The manipulable variables of the robot end effector are as follows

- Terminate or continue (zero indicates the episode is not over, and one indicates the episode is completed.)
- ΔPos_X - Change in X translation
- ΔPos_Y - Change in Y translation
- ΔPos_Z - Change in Z translation
- ΔRot_X - Change in X rotation
- ΔRot_Y - Change in Y rotation
- ΔRot_Z - Change in Z rotation
- gripper extension - gripper position

These above values are then concatenated and converted into a single string with space characters in between the values. For example, if the robot end effector can move from 0 to 20cm in X translation. 20/256 = 0.078125cm corresponds to the size of one bin. If the robot has to move anywhere between 0 and 0.078125cm, the model should output the token corresponding to the first bin. These tokens from the model are then detokenized to convert them as inputs for the robot. In this way, the movement of the robot can be represented by eight integer numbers. Each integer can vary between 0 and 256 action tokens. In this way, the action space ”terminate ΔPos_X ΔPos_Y ΔPos_Z ΔRot_X ΔRot_Y ΔRot_Z gripperExtension” can be converted into a target like ”1 128 91 241 5 101 127”. The concatenated action tokens are shown in fig3.

In the next step, the output tokens from the PaLI-X and PaLM-E are associated with the numbers from 0 to 256. In PaLI-X, the integers up to 1000 have unique tokens, so the numbers are associated with the bin tokens; for PaLM-E model 256, the least used tokens are assigned with the action tokens. The questions to the models are now posed in VQA format as ”What should the robot do to [Task instruction]? The action will be the answer from the policy, and once the task is done, the policy sends the signal 1 to indicate the end of the episode.

b) Robot training as Behavioural Cloning problem: The robot completing the different tasks can be formalized as a behavioural cloning problem with the expert demonstrations from the RT-1 dataset. Each robot training data consists of frames from time t=0 to t=T. x_j is the camera input at time j, a_j is the action from the robot at time j. i is the text instruction given to the robot in VQA format. The full interaction for the robot would be as $i, (x_j, a_j)_{j=0}^{T}$. For N training examples the robotic dataset will look like $D = (i^{(n)}, (x_t^{(n)}, a_t^{(n)})_{t=0}^{T^{(n)}})_{n=0}^{N}$ [2] since the data from the RT-1 dataset is considered to be the optimal action for the given instruction and starting image. The problem can be formulated as reducing the training loss function as given below.

Loss function for training [2].

$$Min \sum_{task\ n} \sum_{(x,a)\sim D^n} -log\pi(a^n \mid x_0^n, i^n) \tag{1}$$

The following hyperparameters are chosen during training

Parameter	PaLI-X 5B	PaLI-X 55B
Learning Rate	$1e^{-3}$	$1e^{-3}$
Batch Size	2048	2048
No Of Gradient Steps	270K	80K
Robot Data Percentage	50	50

Parameter	PaLI 3B	PaLM-E 12B
Learning Rate	$1e^{-3}$	$4e^{-4}$
Batch Size	128	512
No Of Gradient Steps	300K	1M
Robot Data Percentage	50	66

Table 1 Hyperparameters for training

The loss function is reduced for several prediction tasks: 1) Predict the action, given consecutive image frames and a text instruction 2) Predict the instruction given the image frames 3) Predict the robot arm position given the image frames 4) Predict the number of time steps given the image frames 5)predict whether the task was successful given the image frames and language instruction. The original PaLI-X and PaLM-E vision language models have ViT for processing vision. The weights of the ViT are kept frozen throughout the fine-tuning process.

The underlying VLM models are trained with a large amount of text data from the web, each word having a separate token. Since the same models are used to provide inputs for a robot with only 256 tokens, sometimes the model can output an invalid token. Only the valid tokens are sampled from the output of the Vision Language Model.

After training the models, there are two versions of RT-2 based on the underlying Vision Language Models. The model with PaLM-E is built with 5 billion parameters and is named RT-2-PaLM-E. Two versions of PaLI-X, one with 5 billion parameters and the other with 55 billion parameters are trained and are named RT-2-PaLI-X5B and RT-2-PaLI-X55B, respectively. This group of models which combine vision,

language and action are called Vision Language Action models (VLA).

IV. Experiments

Experiments of the model are based on the paper's goal and focus. The paper aimed to see if RT-2 can have any generalization capabilities and Chain of Thought reasoning properties. The experiments also focus on whether the models have acquired those properties and how well they perform in comparison with similar methods in the tests. The following tests are conducted to answer whether the model has emergent capabilities and generalization properties.

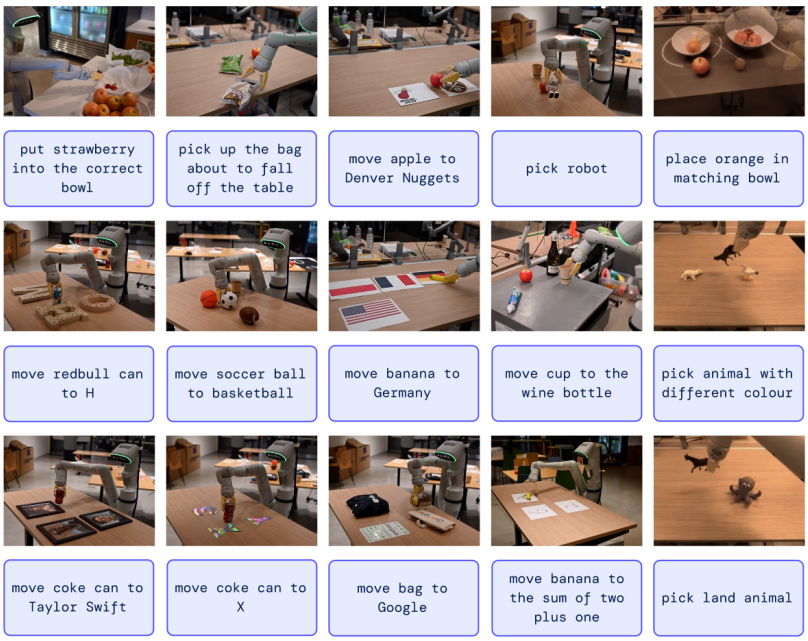

Fig. 5. Example test for emergent capabilities in RT-2 [1]

- How does the RT2 model perform on seen tasks with unseen objects, backgrounds and environments?
- Can we observe and measure any emergent capabilities?
- How does generalization vary with the variation in parameter numbers?
- Can RT-2 exhibit Chain of Thought reasoning similar to vision language models?

The following state-of-the-art models are chosen and trained using RT-1 architecture to compare with RT-2 since they cannot be used directly for giving robot actions. The dataset from RT-1 is used again to train the models.

the models used for comparison are

- RT-1 for state-of-the-art policy.
- VC-1 and R3M for state-of-the-art pre trained representations.
- MOO for comparing against another architecture.

A. How does RT-2 perform on seen tasks with unseen objects, backgrounds and environments?

For this comparison, the tasks are split into seen and unseen. The unseen task category is split into unseen objects, unseen backgrounds and unseen environments. These unseen tasks in each category are sub-divided into easy and hard tasks.

For the seen category, 200 tasks from the RT-1 dataset are selected: 36 for picking objects, 35 for knocking objects, 35 for placing things upright, 48 for moving objects, 18 for opening and closing various drawers, and 36 for picking out of and placing objects into drawers.

Two hundred eighty tasks with mostly pick and place operations are chosen for unseen tasks. For unseen objects, hard cases included objects that are hard to grasp. For unseen backgrounds, hard cases included varied backgrounds and new objects; for hard cases in unseen environments, there were visually distinct office desk environments. An example of unseen objects, environments and backgrounds can be seen in the fig4.

The results show that RT-1 and RT-2 performed well in seen tasks with seen environments, objects, and backgrounds. However, the other models performed significantly lower. In the unseen categories, the RT-2 performed well above all the other models. On average, both instantiations of RT-2 performed equally well, 2x greater than RT-1 and MOO and 6* greater than other baselines. One interesting outcome was that the PaLM-E version performed better than the PaLI-X version for harder cases, whereas the PaLI-X model performed better for easy tasks. The results can be seen in the figure fig6.

B. Can we observe and Measure any emergent capabilities of RT-2?

According to the authors of the paper, emergent capabilities are the capabilities which were transferred from Vision language models to RT-2. Since the Vision language models did not transfer new motions, The authors of the paper tried to measure the transfer of semantic and visual concepts from VLMs qualitatively and quantitatively. An example of the emergent capability can be seen in fig 5.

Qualitative tests: Qualitative tests check whether the models can understand the semantic relationships in vision and language and complete the action. One example is when the robot was instructed, "Put strawberry into the correct bowl", even though the robot was not told which bowl to put the strawberries in. The robot was able to identify that strawberries belonged to a category of fruits and put them into the bowl with all the fruits. Since the model understood what strawberries are and which bowl they should go into without being explicitly instructed, we can conclude that there are some emergent capabilities observed in RT-2 Models.

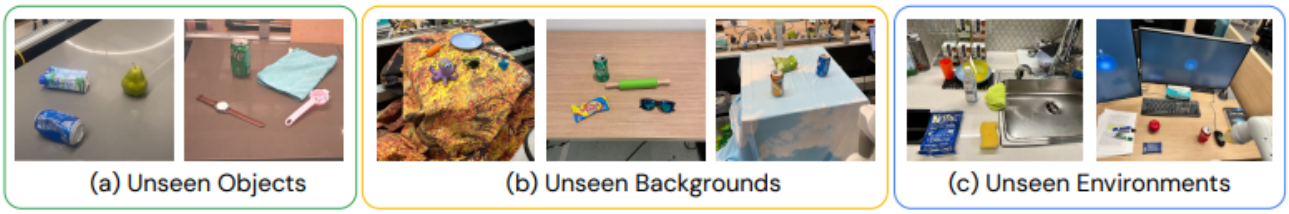

Fig. 4. Examples for unseen backgrounds, objects and environments [1]

Model	Seen Tasks	Unseen Objects		Unseen Backgrounds		Unseen Environments		Unseen Average
		Easy	Hard	Easy	Hard	Easy	Hard	
R3M (Nair et al., 2022b)	45	32	14	13	9	0	2	12
VC-1 (Majumdar et al., 2023a)	63	34	10	13	3	0	0	10
RT-1 (Brohan et al., 2022)	**92**	31	43	71	9	26	14	32
MOO (Stone et al., 2023)	75	58	48	38	41	19	3	35
RT-2-PaLI-X-55B (ours)	**91**	**70**	**62**	**96**	**48**	**63**	**35**	**62**
RT-2-PaLM-E-12B[1] (ours)	**93**	**84**	**76**	**75**	**71**	**36**	**33**	**62**

Fig. 6. Results for unseen objects, environments and backgrounds [1]

Quantitative tests: For this, the tests are split into three categories: 1)Symbol understanding, 2)Reasoning, and 3)Human recognition. All the tests are done with the same environment using the A/B testing framework so that we can compare the results easily. Only RT-1 and VC-1 are used as baselines.

It was observed from the results that RT-2 models performed significantly better in all the categories. RT-2-PaLI-X performs 3x better than others. One significant outcome of the result is that RT2-PaLI-X performs better in symbol understanding, while RT-2-PaLM-E performs better in math reasoning tasks. The results are shown in figure 7.

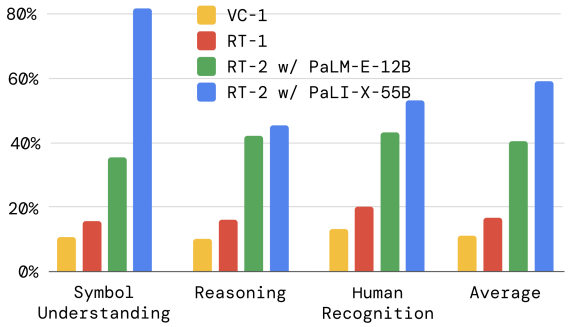

Fig. 7. Results for emergent capabilities [1]

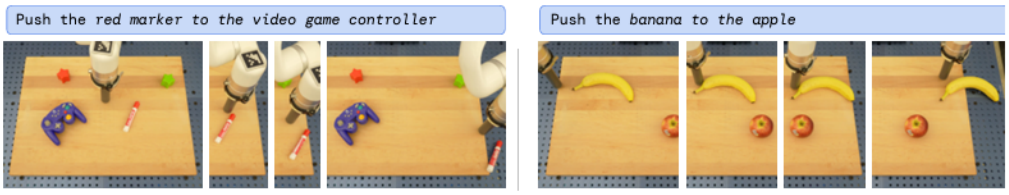

Fig. 9. Example for a failure case in RT-2 [1]

C. How does generalization vary with parameter count and other design decisions?

For this study, the RT-2-PaLI-X 5B parameter and RT-2-PaLI-X55B parameter are taken, trained from scratch and fine-tuned, and another version is co-fine-tuned. The performance of all five different models is compared to unseen objects, unseen backgrounds and unseen environments.

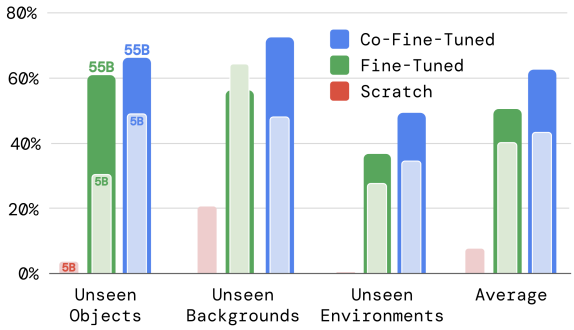

Fig. 8. Results for ablation studies [1]

It is observed that the models trained from scratch perform badly compared to the other versions, like fine-tuned and co-fine-tuned. Regardless of the size, the co-fine-tuned models performed better than the versions that were only fine-tuned with robotic data. The ablation study results are

shown in the figure7.

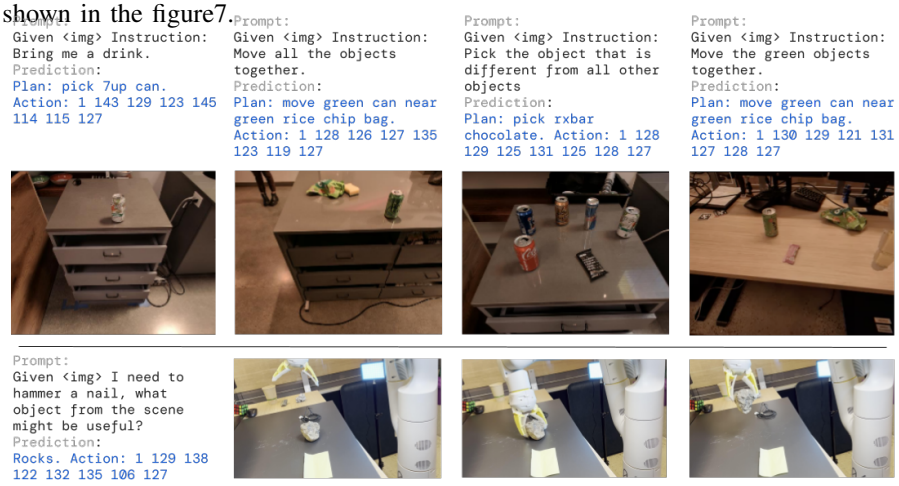

Fig. 10. Example for CoT Reasoning in RT-2 [1]

D. Can RT-2 exhibit Chain-Of-Thought Reasoning similar to vision language models?

To observe the Chain of Thought Reasoning capabilities of RT-2, the RT-2 model is trained again with the reason for doing the tasks in an intermediate plan step. In this way, the RT-2 model can first plan the action in natural language in the plan step and then use this plan to take action in the action step. Only the PaLM-E version of the RT-2 model is trained in this way. After the training, when we look at the output for the planning step from the RT-2 model, we can see that the model has some Chain of Thought Reasoning capabilities. To show this, the authors have provided an example where the instruction was "I am hungry". In the plan step, the RT-2 Model's output was to pick rebar chocolate. By this and other examples, we can conclude that the RT-2 model has acquired the chain of Thought reasoning capabilities from Vision Language Models. The figure10 shows an example.

V. Limitations

Although the Vision Language Action models could get generalization, emergent and chain of Thought reasoning capabilities, the models could not acquire new motions. This might be because of a small number of robotic datasets. In all the evaluations the authors have carefully selected only seen tasks and have not even evaluated their model on unseen tasks. There are some failure cases in the paper where even though the robot was able to move near the objects, it was not capable of grabbing the object due to the challenging dynamics of the object. This failure case can be seen clearly in the figure9.

RT-2 was also seen not having a good performance at:

- Grasping objects by specific parts.
- Motions other than what was seen in the dataset.
- Precise motions like folding a towel.
- Extended reasoning, which requires many steps of reasoning.

The other limitation of the paper is that the model was able to give inference only at a speed of 5Hz, which is not even close to what is needed in real-time applications. So, we must work on quantization techniques before the model can be used for any result time applications.

Other than that, there is no general framework for testing general-purpose robots, and all the models that the authors selected had to be trained using the RT-1 framework again by the authors with datasets from RT-1 from scratch.

VI. Conclusions

This paper has shown us that we could directly fine-tune the vision language models for robotic data, which would transfer some of the capabilities of vision language models to robotic actions. As the vision language models increase in capabilities in future they can be directly used for doing robotic actions. Furthermore, it also highlights the problems in this approach for robotics as the robot did not learn any new motions by itself and there needs to be a general framework for comparison of different machine learning models for robotic actions like in vision or natural language tasks.

The paper mentions that RT-2 is direct end-to-end control for robot actions and is an advantage of RT-2. From one point, this model is not dependent on the robotic parameters, so they can be used for different robotic models with very few changes, which is an advantage, but since the control strategies of the modern robots are highly sophisticated and faster already, if the model concentrates only on high-level planning and uses the robotic controller for doing the control, the model could have been faster and used for real-time inference. This needs to be clarified in the paper.

There are few opportunities for further research to reduce the gaps in the paper.

- Can we use Behaviour cloning along with human videos as in bc-z [8] paper?
- Can we use DAGGER or other methods to introduce exploration?
- Can we use Inverse Reinforcement Learning to learn the Reward function with available data and frame the problem with exploration?

Acknowledgments

I would like to thank Prof.Sebastian Trimpe, Antoine Moncho and Andres Posada of Data Science in Machine Learning in RWTH Aachen University for giving me the opportunity to participate in the Seminar : Learning Based Control, and providing me with valuable feed back throughout the course in order for me to complete the course and write this review paper. I would also like to thank my classmates for their values feedback.

REFERENCES

[1] 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, Brianna Zitkovich. **RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control**. arXiv:2307.15818, 2023

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman,A. Herzog, J. Hsu, et al. **Rt-1: Robotics transformer for real-world control at scale**. arXiv preprint arXiv:2212.06817, 2022.

[3] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. **Palm 2 technical report**. arXiv preprint arXiv:2305.10403, 2023.

[4] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay, S. Shakeri, M. Dehghani, D. Salz, M. Lucic, M. Tschannen, A. Nagrani, H. Hu, M. Joshi, B. Pang, C. Montgomery, P. Pietrzyk, M. Ritter, A. Piergiovanni, M.Minderer, F. Pavetic, A. Waters, G. Li, I. Alabdulmohsin, L. Beyer, J. Amelot, K. Lee, A. P.Steiner, Y. Li, D. Keysers, A. Arnab, Y. Xu, K. Rong, A. Kolesnikov, M. Seyedhosseini, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. **Pali-x: On scaling up a multilingual vision and language model**, 2023a.

[5] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, A. Kolesnikov, J. Puigcerver, N. Ding, K. Rong, H.Akbari, G. Mishra, L. Xue, A. Thapliyal, J. Bradbury, W. Kuo, M. Seyedhosseini, C. Jia, B. K.Ayan, C. Riquelme, A. Steiner, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. **Pali: A jointly-scaled multilingual language-image model**, 2023b.

[6] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. **Palm-e: An embodied multimodal language model**. arXiv preprint arXiv:2303.03378, 2023.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, Noah Fiedel. **PaLM: Scaling Language Modeling with Pathways**, arXiv:2204.02311, 2022.

[8] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. **Bc-z: Zero- shot task generalization with robotic imitation learning**. In Conference on Robot Learning, pages 991–1002. PMLR, 2021.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, **Attention Is All You Need**, arXiv:1706.03762, 2017.

[10] Dean A Pommerleau, **ALVINN:An autonomous land vehicle in a neural network**. Neural Information Processing systems, 1988

[11] Richard S.Sutton, Andrew G. Barto, **Reinforcement Learning An Introduction Second Edition**

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever, **Learning Transferable Visual Models From Natural Language Supervision**, arXiv:2103.00020, 2021.

[13] 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, Andy Zeng, **Do As I Can, Not As I Say: Grounding Language in Robotic Affordances**, arXiv:2204.01691, 2022.