What Matters in Employing Vision Language Models for Tokenizing Actions in Robot Control?

Nicolai Dorka[1*], Chenguang Huang[1*], Tim Welschehold[1], and Wolfram Burgard[2]

***Abstract*— Vision Language Models (VLMs) have demonstrated remarkable proficiency in comprehending images and text, as well as generating textual outputs based on such inputs, owing to their training on web-scale datasets. Their potential for robotics applications is particularly intriguing. One notable example is RT-2, a system capable of generating low-level actions represented in textual format from a given instruction alongside a sequence of historical actions and image observations. To stimulate further research in this domain, we introduce an open-source implementation tailored for utilizing VLMs in instruction-based robot control. This implementation supports a variety of VLM architectures and facilitates straightforward integration of new models. We use our framework to train multiple VLMs and evaluate them on a physical robot. The results validate the practical efficacy of our framework, thus paving the way for enhanced understanding and capabilities in instruction-based robot control systems. The code is available at: https://github.com/Nicolinho/RoboVLM.**

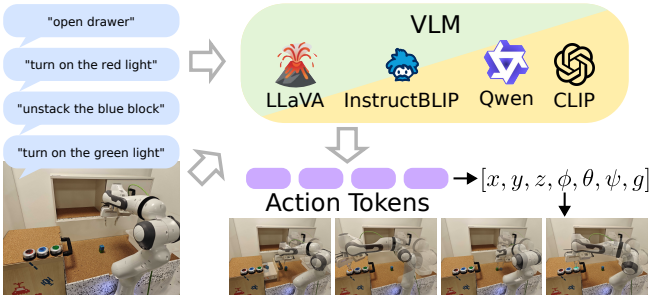

Fig. 1. We present a framework for learning robot manipulation policies as a token prediction problem using Vision Language Models, which can be tailored to various VLM architectures. Additionally, we delve into the critical aspects of applying VLM to the learning of action tokens.

I. Introduction

In recent years, significant progress has been made in integrating large language models (LLMs) and vision language models (VLMs), with widespread implications across various domains [1]–[5]. These models, renowned for their language comprehension abilities, offer intriguing prospects for instruction-based robotics applications. These include generating robot execution code [6], optimizing policy rewards [7], directly controlling robot low-level functions with language [8], and summarizing and correcting robot behavior [9]. Recent advancements in VLMs further expand these possibilities into new domains. The multimodal capabilities inherent in these foundational models offer valuable insights into robotics, such as applying VLMs to robot mapping [10], framing robot control as a Visual Question Answering problem [11], and developing foundation models trained on both multimodal and embodied data [12]–[14].

A notable endeavor in this domain is RT-2 [12], which employs a VLM to interpret an instruction, past actions and image observations, and generate subsequent actions in textual form. RT-2 has demonstrated the potential of creating instruction-based low-level robot control policies, showcasing remarkable performance and notable generalization capabilities. However, RT-2's unavailability to the public and its considerable scale, with 55 billion parameters, pose challenges for academic endeavors to engage with it effectively, hindering further exploration and refinement of this paradigm.

In this study, we present an open-source implementation for utilizing VLMs to predict actions in tokenized text format within the context of instruction-based low-level robot control. We train a variety of vision language action models based on different open-source VLMs and assess them on a Franka Emika Panda arm in real-world settings. Encouraging real-world outcomes underscore the potential of both the approach and our implementation for training such models. Furthermore, through the adoption of parameter-efficient fine-tuning techniques such as quantization [15] and low-rank adaptation (LoRA) [16], we demonstrate the feasibility of conducting training and inference with VLMs using relatively modest computational resources and hardware setups. Additionally, by exploring various models and training methodologies, we offer initial insights into the crucial factors influencing VLM training for robot control. Our objective is to equip the community with the necessary tools to delve into VLM approach for robot control systematically.

II. Technical Approach

A. Vision Language Models for low-level Control

Vision Language Models (VLMs) are capable of processing both images and natural language text to generate text output. We leverage their capabilities to generate actions, by treating actions as natural language sentences. Consequently, we represent observation-action sequences as coherent sentences. Specifically, the model's input comprises a task instruction in language, paired with a trajectory $(o_0, a_0, o_1, a_2, ...)$ of actions a_i and image observations o_i. To ensure compatibility with the VLM, we encode actions in string format.

* Equal contribution.
[1] Department of Computer Science, University of Freiburg, Germany.
[2] Department of Eng., University of Technology Nuremberg, Germany.

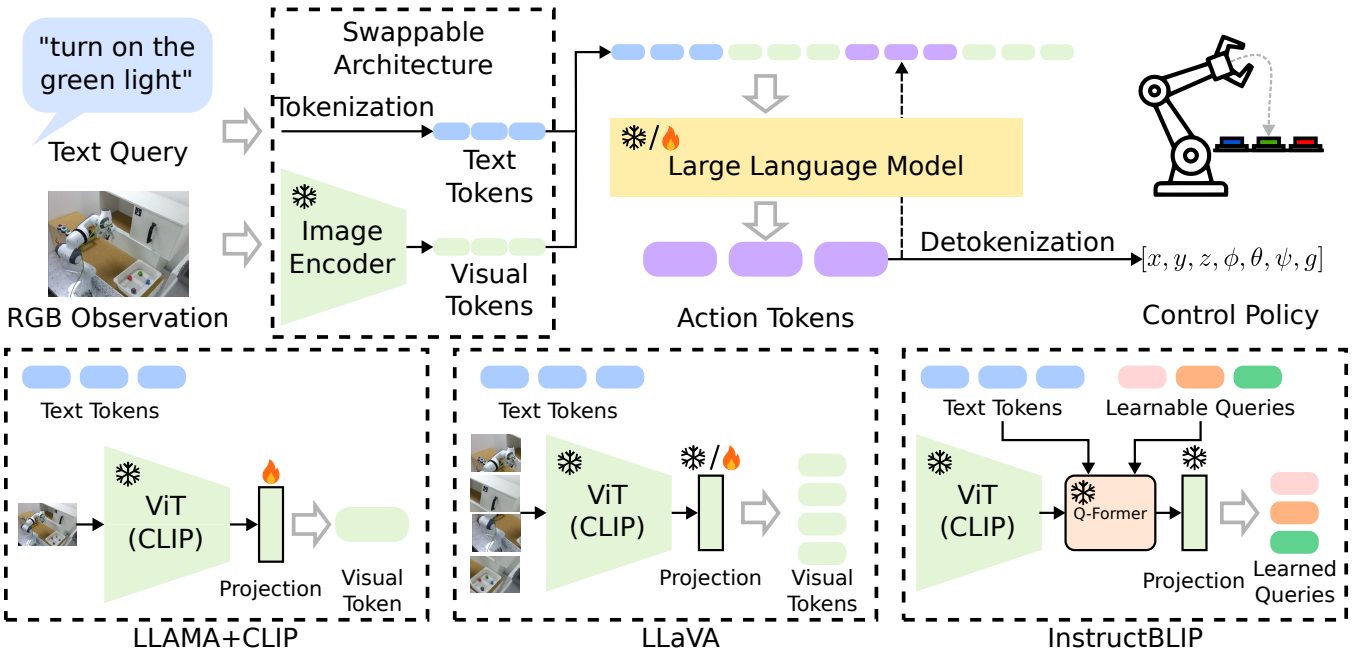

Fig. 2. Overview of the pipeline. In our pipeline, a swappable architecture comprising a text tokenizer and an image processing pipeline is employed, taking language instructions and static camera observations as input and yielding a list of text and visual tokens. Subsequently, a Large Language Model receives these tokens, along with tokenized actions from the history, and generates action tokens for the current step. With detokenization, the action tokens are converted into a control policy, which is then utilized by the robot manipulator to execute the task. In this work, we investigate several VLMs as the swappable architectures in the pipeline including CLIP [17], LLaVA [18], InstructBLIP [19], and QWen [20].

For this encoding, we employ a 6-dimensional representation for positional and rotational coordinates, with an additional dimension for the gripper extension. Each action dimension is discretized into 100 bins, represented by ordinal numbers ranging from 0 to 99. Individual action dimensions are separated by a single space in the string representation, and we append "[ea]" to signify the end of the action sequence. Thus, an action is encoded as "pos_x pos_y pos_z rot_x rot_y rot_z gripper_extension [ea]", with an example instantiation being "54 6 89 11 69 77 99 [ea]".

The VLM operates on these instruction and trajectory sentences to predict actions in textual form. Consequently, we can fine-tune the pretrained VLM akin to how models are fine-tuned for language tasks via next token prediction. During loss calculation, predictions corresponding to the positions of image and instruction encodings are disregarded. Hence, the model is exclusively trained to forecast actions. An overview of this approach is depicted in Figure 2.

B. Integrated Models

We integrated various state of the art open source VLMs in our framework. All of the VLMs share that they use some vision encoder like CLIP [17], a projection network that maps the vision features into the token embedding space of the language model (LM), and finally, the language model which generates the output from the vision and text embeddings. We test InstructBLIP [19], LLava1.5 [18], and Qwen-VL [20]. These models have been specifically finetuned for visual-language tasks like visual question answering.

InstructBLIP: InstructBLIP employs a Query Transformer, abbreviated as Q-Former, to derive visual features from a static image encoder. For the visual features it uses the patch features from ViT-g [21]. Within the Q-Former, a collection of K adaptable query embeddings interacts with the output of the image encoder via cross-attention. This interaction yields K encoded visual vectors, each corresponding to a query embedding. These vectors undergo linear projection before being inputted into the frozen LLM.

LLava1.5: LLava uses CLIP-ViT-L-336px [17] as image encoder. The single patch features are fed through a two-layer network to project them into the dimension of the token embedding space of the language model.

Qwen-VL: The image encoder consists of a ViT [22]. The projection network employs a set of trainable embedding vectors as query vectors and utilizes the patch features from ViT as keys for performing cross-attention. This process effectively condenses the visual feature sequence to a consistent length of 256.

Further, we implement a VLM we term **LLama+CLIP** that uses CLIP (clip-large-patch14) [17] as a vision encoder and LLama-v2 [4] as a language model. It projects the 1024-dimensional global feature vector of CLIP into the 4096-dimensional token embedding space of LLama via a single fully connected layer.

We freeze the vision encoder during training for all VLMs . In the case of LLama+CLIP, we train the language model and the projection layer. For all other models, we only train the language model as the projection networks have already been trained on visual-language tasks. We also conduct an ablation experiment for LLava where we

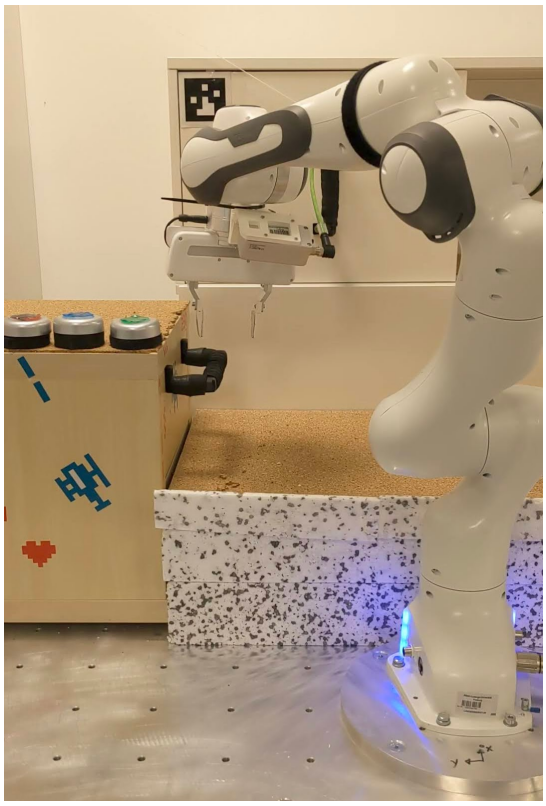

Fig. 3. Visualization of our experimental setup. A Franka Emika Panda arm interacts with a table top environment.

additionally train the projection network or only the projection network. Further, instead of full finetuning we employ low-rank adaptation (LoRA) for all model trainings. This reduces the computational and hardware requirements by a large factor.

C. Framework

Our codebase is meticulously crafted to prioritize clarity and seamless adaptability to diverse use cases. Users can effortlessly select from the available models via command line inputs, as well as adjust training and evaluation hyperparameters.

Integrating new models and datasets is straightforward within our framework. We furnish a convenient wrapper class housing the VLM, which streamlines both training and inference procedures. Incorporating a new VLM primarily entails loading the model itself and its associated data preprocessor. Depending on the input format of the VLM, slight modifications may be required in the existing data preprocessing functions.

Similarly, adding a new dataset is facilitated by leveraging existing methods to transform actions into the natural language format compatible with the VLM. Additionally, we supply code for training on datasets from the Open-X dataset, further enhancing the versatility of our codebase.

Model	Avg.	open drawer	unstack block	red light	green light
InstructBLIP	37.5	100.0	33.3	0.0	20.0
LLaVA-LM	35.0	40.0	30.0	30.0	40.0
LLaVA-LM-Proj	20.0	0.0	50.0	0.0	33.3
LLaVA-Proj	0.0	0.0	0.0	0.0	0.0
Qwen-VL	45.0	30.0	50.0	50.0	50.0
LLaMA+CLIP	65.0	100.0	10.0	50.0	100.0

TABLE I

THE PER TASK AND AVERAGE SUCCESS RATE FOR THE DIFFERENT VLMS TESTED ON A FRANKA EMIKA PANDA ARM. THE FULL TASK DESCRIPTIONS ARE "TURN ON THE GREEN LIGHT," "TURN ON THE RED LIGHT", "OPEN THE DRAWER", AND "UNSTACK THE BLUE BLOCK".

III. EXPERIMENTAL EVALUATION

Dataset. To create the training dataset, we gathered 514 expert demonstration episodes for 9 tasks by teleoperating a Franka Emika Panda robot arm using a VR controller. RGB images were captured from a fixed Azure Kinect camera to serve as visual observations. Concurrently, we recorded the relative position and orientation (Euler Angle) displacement of the end effector in the robot base coordinate frame, along with the gripper state (open or closed), as actions.

Training. In our training procedure, we divided the demonstration data into non-overlapping segments of 10 steps each. The models underwent training for 5 epochs. Leveraging LoRA for all models, we completed training on 500 episodes in less than 12 hours using four A40 GPUs for all the VLMs. Remarkably, LLaMA+CLIP can be trained even on a 12GB GPU with the aid of quantization techniques. For the training, we used a learning rate of 0.0003, batch size 128, cosine annealing with a warmup of 20 steps, weight decay 0.1, LoRA rank 32, LoRA alpha 64, and LoRA dropout 0.05.

Results. Evaluation of model performance was conducted on four distinct tasks: "turn on the green light," "turn on the red light", "open the drawer", and "unstack the blue block". Each task was evaluated across 10 trajectories with varying initial positions. We maintained a history length of 5 for all models.

Our findings are detailed in Table I. Remarkably, the simplest LLama+Clip model demonstrates the highest average success rate at 65%. It exhibits robust capability in opening the drawer and activating the green light. Notably, turning on the red light proves to be the most successful task across all models. However, this task presents challenges for all models due to its position at the periphery of the environment, sometimes leading to occlusion by the robot arm in the static camera image. The comparatively lower success rate in unstacking the blue block may stem from the use of global features from the image encoder instead of patch features. Encoding the precise position in a single vector could pose difficulty for the model, particularly considering the absence of training for the vision encoder. In contrast, other evaluated VLMs leverage patch features, comprising several hundred vectors, which potentially aids in locating the block. Nonetheless, their average success rates

fall short of LLama+CLIP. The second-best performer in our assessment is Qwen-VL, achieving an overall success rate of 45%, followed by InstructBLIP at 37.5%, and the most proficient LLava version, LLava-LM, at 35%. We speculate that the superior performance of LLaMA+CLIP may be attributed to the over-specialization of other VLMs during fine-tuning on instruction-based visual-language tasks, potentially diminishing the generality of their model weights and hindering further finetuning to downstream tasks.

Further, our results underscore the importance of fine-tuning the language model in LLaVA experiments. Interestingly, we observed that tuning both the projection and language model led to inferior results compared to solely fine-tuning the language model. This outcome is likely due to the fact that the projection model in LLaVA is already adept at handling visual-language tasks and produces generally beneficial features.

IV. Conclusion

In this study, we have introduced a codebase tailored for Visual Language Models (VLMs) aimed at instruction-based robot control. Through experiments conducted on a real Franka Emika Panda arm, we have demonstrated the framework's capability to facilitate the training of effective policies.

Our experimental results provide additional insights into the key considerations for employing VLMs in this context. Despite the notable performance of VLMs such as InstructBLIP, LLava, and Qwen-VL in vision-language tasks like visual-question answering, they are surpassed by the relatively straightforward LLama+CLIP model, which had not undergone vision-language fine-tuning prior to the training on the robot data. This underscores the potential efficacy of utilizing generally pretrained models over instruction-finetuned models for subsequent fine-tuning with robot control data. Furthermore, our findings regarding the training of different components of the LLava model highlight the paramount importance of training the language model component.

Although our initial work is based on a relatively modest dataset, primarily designed to assess the approach's potential, we envisage expanding this endeavor by leveraging larger datasets such as the Open-X dataset [14] in future iterations. This prospect is encouraging given the robust scalability demonstrated by VLMs when confronted with larger datasets. We further plan to integrate our codebase with a simulator to allow for faster experimentation.

Our ultimate goal is to build an open-source foundational model for robot control. The resulting model will be made publicly available, fostering a collaborative environment where others can readily fine-tune it to suit their specific use cases. We believe that already in its current form, our codebase holds value for researchers and practitioners engaged in developing VLMs for robot control applications.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] OpenAI, "Gpt-4v(ision) system card," 2023. [Online]. Available: https://cdn.openai.com/papers/GPTV_System_Card.pdf

[3] GeminiTeam, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[5] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[6] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *arXiv preprint arXiv:2209.07753*, 2022.

[7] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[8] 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.

[9] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," *arXiv preprint arXiv:2306.15724*, 2023.

[10] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.

[11] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, *et al.*, "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," *arXiv preprint arXiv:2402.07872*, 2024.

[12] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[13] 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.

[14] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[15] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023.

[16] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[18] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," *arXiv preprint arXiv:2310.03744*, 2023.

[19] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, 2023.

[21] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 358–19 369.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.