EdgeVLA: Efficient Vision-Language-Action Models

Paweł Budzianowski, Wesley Maa, Matthew Freed, Jingxiang Mo, Aaron Xie, Viraj Tipnis, Benjamin Bolte

***Abstract*— Vision-Language Models (VLMs) have emerged as a promising approach to address the data scarcity challenge in robotics, enabling the development of generalizable visuomotor control policies. While models like OpenVLA showcase the potential of this paradigm, deploying large-scale VLMs on resource-constrained mobile manipulation systems remains a significant hurdle. This paper introduces Edge VLA (EVLA), a novel approach designed to significantly enhance the inference speed of Vision-Language-Action (VLA) models while maintaining their representational power and enabling real-time performance on edge devices. We achieve this through two key innovations: 1) Eliminating the autoregressive requirement for end-effector position prediction, leading to a 6x speedup in inference, and 2) Leveraging the efficiency of Small Language Models (SLMs), demonstrating comparable training performance to larger models with significantly reduced computational demands. Our early results demonstrate that EVLA achieves comparable training characteristics to OpenVLA while offering substantial gains in inference speed and memory efficiency. We release our model checkpoints and training codebase to foster further research on mobile deployments.**

I. INTRODUCTION

The development of robust and generalizable manipulation policies has long been hampered by the limited availability of large-scale, diverse embodied datasets. Recent advancements in Vision-Language Models (VLMs) [11], [8] offer a compelling solution to this challenge. By leveraging the vast amount of readily available image-text data, VLMs can learn rich representations of the world and be adapted for visuomotor control tasks. Open-source models like OpenVLA [9] have demonstrated the effectiveness of this approach, showcasing impressive performance in various robotic manipulation tasks. However, deploying these large-scale VLMs, often exceeding billions of parameters, on resource-constrained mobile platforms with edge devices like the Jetson Nano presents significant challenges. Their high computational and memory requirements hinder real-time performance and limit accessibility for researchers and practitioners.

The progress in the mobile manipulation can be effective only if the systems we design are inexpensive and easily deployable without putting too much strain on compute requirements. That is why, this paper introduces Edge VLA (EVLA), a novel VLA architecture designed to address above mentioned challenges. EVLA offers potential significant improvements in inference speed and efficiency without compromising foundation models' representational power. Our approach centers around two key innovations. Our work focuses on architectural modifications to achieve significant speedups while maintaining model performance, specifically by eliminating the autoregressive requirement for end-effector prediction and leveraging the efficiency of SLMs. We challenge the conventional autoregressive approach for predicting end-effector positions, demonstrating that joint control, where the entire position is predicted simultaneously, does not diminish the model's encoding capabilities. This modification yields a 7 times increase in inference speed, crucial for real-time robotic control on edge devices.

Secondly, we explore the potential of recently developed Small Large Language Models (SMLs), such as Phi [1] and Gemma [13], which achieve comparable performance to larger counterparts thanks to scaling laws with significantly reduced computational footprints. Our proposed architecture EVLA comprises of a pretrained language model Qwen2-0.5B fused with two visual encoders SigLIP [18] and DINOv2 [12] adding to 1B parameters. EVLA maintains training performance comparable to 7 times largers models while significantly reducing hardware requirements.

Paweł Budzianowski, Matthew Freed, Wesley Maa, Aaron Xie, Viraj Tipnis, Benjamin Bolte are with K-Scale Labs. Email correspondence to pawel@kscale.dev

Jingxiang Mo is with McGill University.

II. RELATED WORK

Learning-based approaches to mobile manipulation starts to reaching or exceeding the performance of classical model-based control system. We can divide them roughly into systems trained from scratch or fine-tuned on top of the foundation models.

The former approach has been based on behavioral cloning where typically vision observation are mapped either to the end effector position and orientation or joint positions [3], [10]. These models can be enhanced with regularization, planning, multi-task learning pushing the limits of the performance. A second line of work on mobile manipulation enables easy deployment with relative cheap hardware but does not take advantage of foundation model[6]. These systems typically train model from scratch with the model size in ranges of 10-100M parameters preventing from seeing generalization capabilities to novel environments [19], [3].

The line of work that relies on foundation models incorporates all aboe-mentioned techniques while seeking more powerful generalization capabilities. The most heavily explored approached relies on vision-language models [11], [8]. The vision component is typically adapted to operate on the same token space as the LLM allowing to re-use different blocks Combined with large manipulation dataset like OpenX [4], they have showcased the promise of this paradigm generalizing to new environments [2], [9]. Although these work have highlighted the potential of leverag-

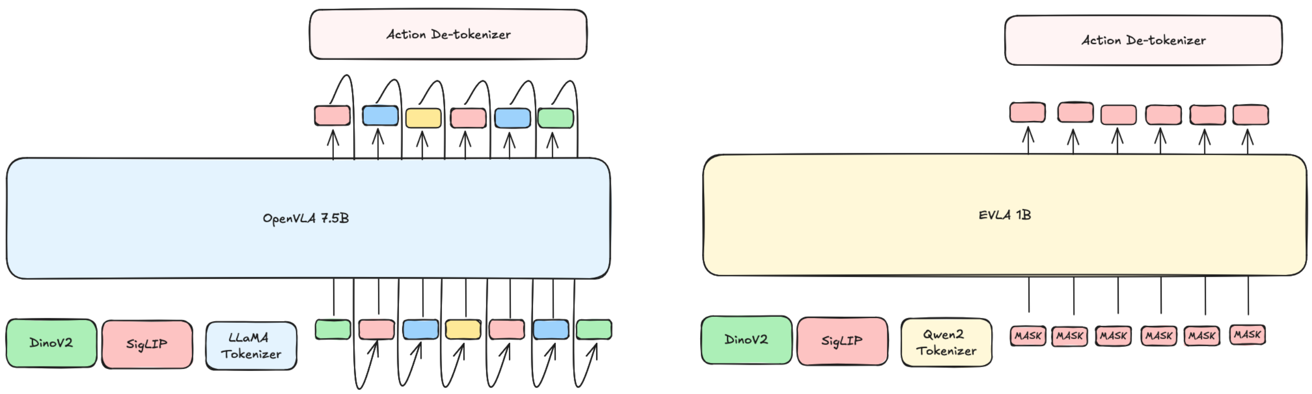


Fig. 1: The comparison of generation between OpenVLA and EVLA. The pretraining phase is identical. In the phase two of EVLA, the LLM is being retrained to generate in autoregressive fashion.

ing large language models (LLMs) they rely on large LLMs, resulting in substantial computational demands. Efforts to improve efficiency include quantization techniques [16] and hardware-specific kernels [14]. Nevertheless, these system achieves speed of only 5 to 10Hz with stationary compute systems preventing from the deployment on edge even in the laboratory setting.

III. METHOD

A. Phase 1: VLM Pretraining

EVLA's foundation is a VLM trained using a combination of image-text pairs sourced from diverse captioning datasets and synthetically generated multimodal instruction-tuning examples. We adopt a two-part visual encoder, employing pretrained SigLIP [18] and DinoV2 [12] models, following the architecture of OpenVLA.

For language processing, we utilize Qwen2 [17] with 0.5B parameters that demostrates the effectiveness of SLMs in achieving comparable performance to larger models. A projection layer maps the visual representation to the language model's token space. The pre-training dataset comprises 1.2M text-image pairs, facilitating the learning of robust visual and language representations following the recipe of PrismaticVLM family of models [8].

B. Phase 2: Joint Control for End-Effector Prediction

Traditional VLAs often employ an autoregressive approach for predicting end-effector positions, mimicking the causal nature of language generation. However, we hypothesize that for robotic control this restriction is not inherently necessary. We propose that predicting the entire end-effector position jointly, rather than sequentially, does not compromise the model's encoding capabilities while significantly improving inference speed.

By removing the causal mask in the LLM and training the model to output the complete end-effector position at once, we bypass autoregressive requirements achieving by definition 7 times speedup in inference, a crucial advancement for real-time applications on edge devices. See Figure 1 for the overall layout of the final model.

IV. EARLY RESULTS

In order to evaluate EVLA's capabilities of adapting to non-autoregressive loss while being substantially smaller we used on BridgeData V2 and OpenX datasets. We hypothesize that the early training results will shed some light on model characteristics.

A. BridgeData V2 training characteristics

Initial experiments on the BridgeData V2 dataset [15] conducted on a single node with 8 A100-80GB GPUs, validate that EVLA can achieve similar performance to the 7.5B parameters equivalent. Figure **??** illustrates the training progress, showcasing the comparable performance of the two models.

It is worth pointing out that the training efficiency is distinguishably slower for EVLA.

B. OpenX training characteristics

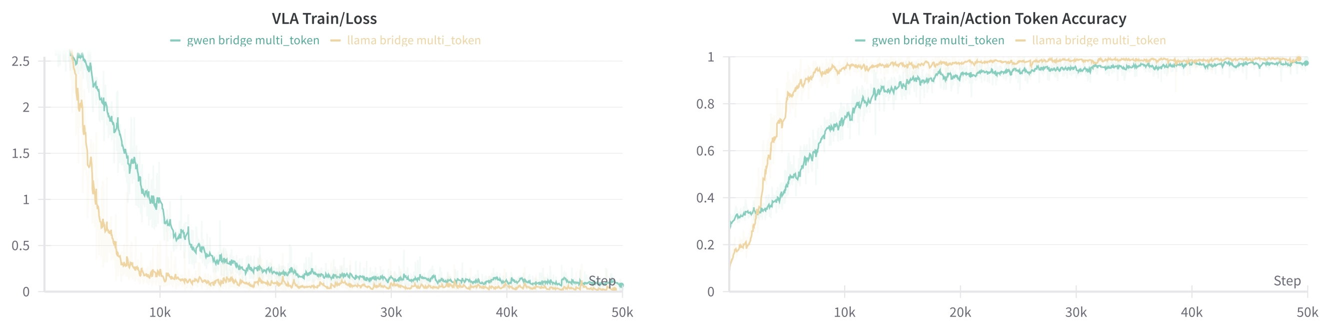
We further evaluate EVLA on the full OpenX dataset, utilizing 80 A100-40GB GPUs for a 5 days. While EVLA trains slower than OpenVLA due to the smaller representational power, the training iteration is 7 times faster and allows for larger batch sizes, effectively mitigating the difference in training efficiency. Figure **??** shows the training progress on the OpenX dataset.

Due to computational constraints, we were not able to reproduce full two weeks training as in the original implementation. However, the training curves behaves similarly to the BridgeV2 case.

C. Efficiency Gains

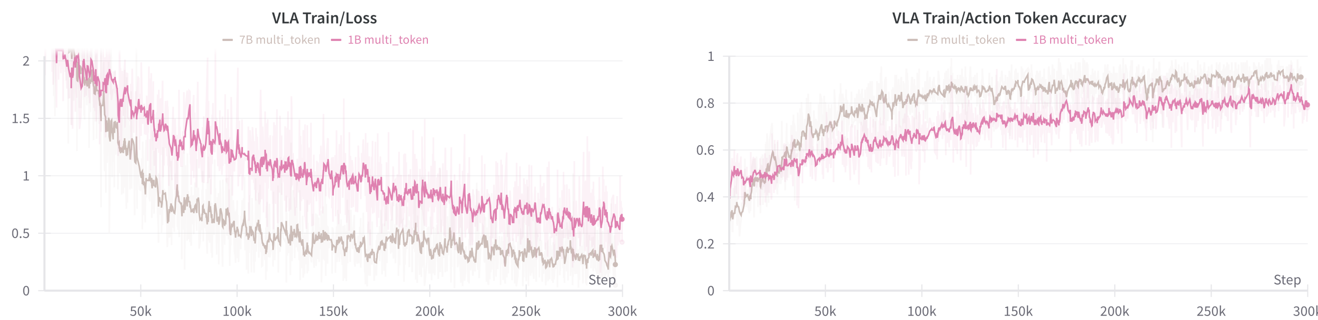
EVLA's architectural modifications result in substantial improvements in the inference speed and memory consumption, enabling deployment on resource-constrained edge devices. Table I compares the inference time and memory requirements of EVLA and OpenVLA on an A100-40GB GPU.

By using a smaller VLM and optimizing our architecture, we can achieve significant inference speed and memory improvements. These speedups will only increase with addition



(a) (b) Accuracy

Fig. 2: The loss (left) and action token accuracy (right) training curves for both OpenVLA and EVLA models during training on the BridgeData V2 dataset.



(a) loss (b) accuracy

Fig. 3: The loss (left) and action token accuracy (right) training curves for both OpenVLA and EVLA models during training on the OpenX dataset.

TABLE I: Efficiency Comparison of EVLA and OpenVLA.

Model	Inference Time (ms)	Memory Usage (GB)
OpenVLA	20	16
EVLA	5	4

of more degrees of freedom. It is worth noting that OpenVLA uses flash_attention2 [5] kernels, while EVLA is evaluated in the eager mode. Advances in flexible and efficient attention mechanisms, such as FlexAttention [7], are expected to push these numbers even lower. These results shows the path for the deployment on mobile manipulation systems on CPU architectures.

V. CONCLUSIONS

This paper presents Edge VLA (EVLA), a novel VLA architecture designed for efficient deployment on mobile manipulators or humanoids. By eliminating the autoregressive requirement for end-effector prediction and leveraging the efficiency of SLMs, EVLA achieves significant speedups in inference time and reductions in memory footprint without compromising model performance. Our results suggest EVLA's effectiveness on diverse robotic embodiments, paving the way for real-time VLA applications on resource-constrained platforms.

We release our model checkpoints and training codebase to facilitate further research. We believe that EVLA's efficiency and accessibility will empower researchers and practitioners to explore the full potential of VLAs for mobile manipulation. Future work will focus on further optimizing EVLA's architecture and exploring its deployment on a wider range of edge devices, including CPU-based platforms.

REFERENCES

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital

Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

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

[3] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024.

[4] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell

Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[5] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023.

[6] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024.

[7] Yanbo Liang Horace He, Driss Guessous and Joy Dong. Flexattention: The flexibility of pytorch with the performance of flashattention, 2024.

[8] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models, 2024.

[9] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. 2024.

[10] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions, 2024.

[11] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.

[12] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[13] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Bal-

dini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.

[14] Philippe Tillet, H. T. Kung, and David Cox. Triton: An intermediate language and compiler for tiled neural network computations. 2018.

[15] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024.

[16] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models, 2023.

[17] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[18] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[19] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. 2023.