Robotic Assistant: Completing Collaborative Tasks with Dexterous Vision-Language-Action Models

Boshi An, Chenyu Yang, Robert Katzschmann[1]

[1]Work done at Soft Robotics Lab, ETHz, Switzerland

1 Introduction

Robots are autonomous mechanical systems designed to assist humans with complex tasks. In many scenarios, effective human-robot collaboration is essential, whether through physical assistance or more nuanced forms of interaction. However, building robots that can collaborate naturally and intuitively with humans remains a significant challenge. Directly training control policies on specific tasks often leads to overfitting and fails to capture high-level task semantics or human intent.

Large language models (LLMs) have recently demonstrated impressive capabilities in reasoning, generalization, and multimodal understanding, making them promising candidates for enabling more flexible robotic behaviors. Yet, directly applying LLMs to real-world collaborative robotics remains impractical for two key reasons: (1) LLMs lack the mechanisms to bridge the gap between abstract reasoning and low-level control, and (2) they rely heavily on explicit language prompting, which introduces latency and inefficiency in real-time interactions.

To enable smoother and more intuitive collaboration, we envision a policy that minimizes the need for language-based prompting and instead infers human intent directly from motion cues — enabling a robot to act through **tacit understanding**.

In this project, we propose a novel approach that fine-tunes pre-trained vision-language-action (VLA) models for collaborative tasks. We introduce several key modifications to improve adaptation to real-world human-robot interaction: leveraging pre-trained visual encoders, incorporating human pose priors, and re-designing the model's action space. Our approach enhances the robot's ability to perceive, interpret, and respond to human behaviors in a context-aware and data-efficient manner. Real-world evaluations demonstrate the effectiveness of the proposed method.

2 Related Work

2.1 Human-Robot Interaction

Human-robot interaction (HRI) is a longstanding area of research aimed at improving the ways in which robots assist and collaborate with humans. Prior work has explored a variety of methods to enhance robot responsiveness, intention understanding, and physical cooperation. For example, Roveda et al. Roveda et al. [2019] employed fuzzy controllers to support humans in industrial settings. Yan et al. Yan et al. [2019] used long short-term memory (LSTM) networks for intention recognition in human-robot interaction. Similarly, Zhang et al. Zhang et al. [2020] applied recurrent models to predict human motion during assembly tasks to facilitate handovers. More recently, Wojtak et al. Wojtak et al. [2021] proposed using neural fields for learning object handover behaviors, while Ji et al. Ji et al. [2024] and Wang et al. Wang et al. [2024] explored foundation model-based approaches for collaborative assembly and tabletop interaction, respectively. However, these methods often depend on handcrafted robotic APIs and suffer from high inference latency, limiting their real-time applicability.

In contrast, we propose a system that enables real-time, smooth human-robot collaboration by directly generating robot actions from multimodal observations, without relying on predefined action schemas.

2.2 Learning from Demonstrations

Learning from demonstrations (LfD), also known as imitation learning, is a widely adopted paradigm in robotic learning Zare et al. [2024]. By mimicking human behavior, robots can acquire complex skills without requiring manually designed reward functions. Classical approaches include Behavior Cloning (BC), which maximizes the likelihood of expert actions given observed states, and Inverse Reinforcement Learning (IRL) Arora and Doshi [2021], which infers the underlying reward function from demonstrations. DAgger Ross et al. [2011] addresses distributional shift by iteratively querying the expert in an online setting.

To improve data efficiency and handle imperfect demonstrations, more recent methods incorporate probabilistic and generative modeling. Huang et al. Huang et al. [2018] proposed a Gaussian Mixture Model (GMM)-based framework for few-shot learning in long-horizon tasks, while Bütepage et al. Bütepage et al. [2020] used generative models for imitation in human-robot interaction scenarios.

The emergence of large-scale robotic datasets Brohan et al. [2022], O'Neill et al. [2024] has enabled the development of generalist policies trained with simple imitation objectives. These datasets support scaling imitation learning to diverse tasks and environments.

2.3 Vision-Language-Action (VLA) Models

Recent advances in large language models (LLMs) Brown et al. [2020], Achiam et al. [2023] have demonstrated strong capabilities in reasoning, abstraction, and multimodal alignment. This has motivated efforts to apply LLMs to robotics, where they could bridge perception and action through natural language.

Preliminary works such as Text2Motion Lin et al. [2023] and VoxPoser Huang et al. [2023] have explored this direction. Building on large-scale multimodal datasets and vision-language pretraining Anil et al. [2023], Liu et al. [2023, 2024], researchers have introduced VLA models that process visual and linguistic inputs to directly generate tokenized robot actions Driess et al. [2023], Kim et al. [2024], Team et al. [2024]. These models are trained using next-token prediction over sequences of multimodal inputs and demonstrations.

VLA models exhibit strong generalization and compositionality, allowing them to handle open-ended, unstructured tasks. They can also be adapted to specific domains via fine-tuning, making them a promising foundation for learning collaborative robot behaviors from modest data.

3 Method

3.1 Robotic System

We use a Mimic hand mounted on a Franka Panda robotic arm. Two Mimic hand cameras and two external cameras are used to capture visual input. The full system is shown in fig. 1.

A teleoperation system is also part of the robotic system. We use Rokoko mocap gloves to capture the absolute position, rotation and finger pose of human hands to allow teleoperation of the robot. The mocap data is mapped to the mimic hand and Franka Panda arm via a motion retargeting system.

3.2 Data Collection Pipeline

3.2.1 Collection

To collect training data for model training, we designed a collaborative data collection pipeline involving two human participants: the **teleoperator** and the **collaborator**. The teleoperator controls the robot by wearing Rokoko motion capture gloves, which record the absolute position, orientation, and finger articulation of their hands. This motion data is then retargeted onto the robotic system to enable teleoperation.

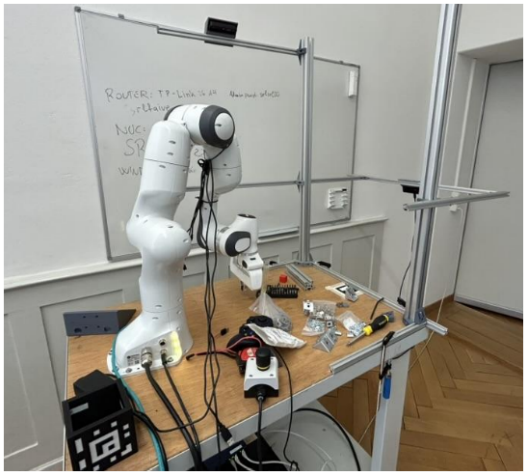
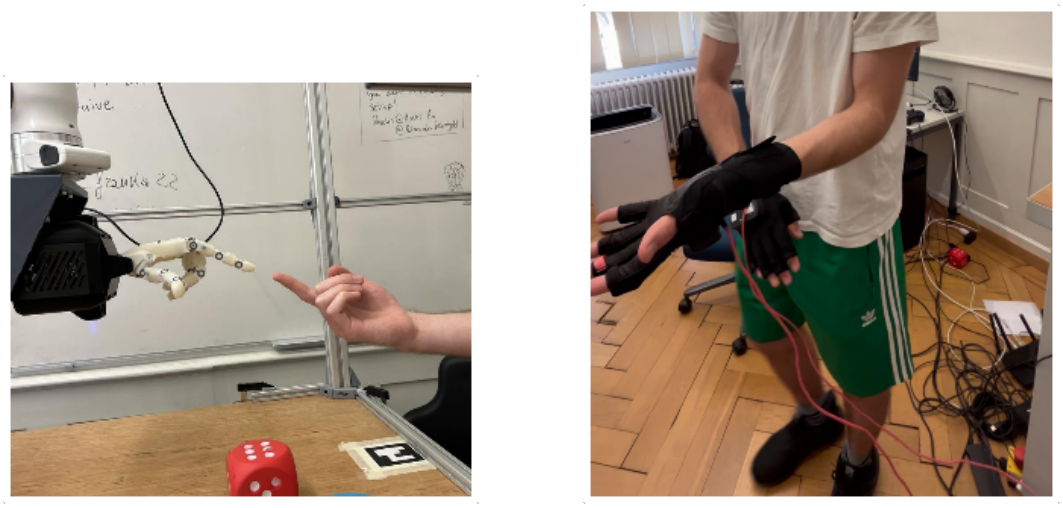


Figure 1: Robotic system

The collaborator interacts with the robot in a shared workspace, enabling natural human-robot interaction to unfold. The two roles are illustrated in fig. 2.



(a) Collaborator (b) Teleoperator

Figure 2: The individuals involved in data collection.

During each trial, the teleoperator controls the robot to perform collaborative tasks with the human collaborator. Throughout the trial, raw robot states, camera inputs, and action commands are streamed and recorded in HDF5 format.

3.2.2 Post-processing

After data collection, the raw streams are post-processed to construct structured datasets suitable for model training. This includes synchronizing all sensory and control data and extracting snapshot frames at a fixed frequency. For this project, we used a sampling rate of 10 Hz.

In addition to the synchronized sensory data, we augment each frame with a text prompt and several auxiliary labels. The text prompt encodes the intended command for the robot during the task (e.g., "pick up the red cube"). The auxiliary labels provide extra supervision to guide the model's understanding of human intent. Specifically, we include:

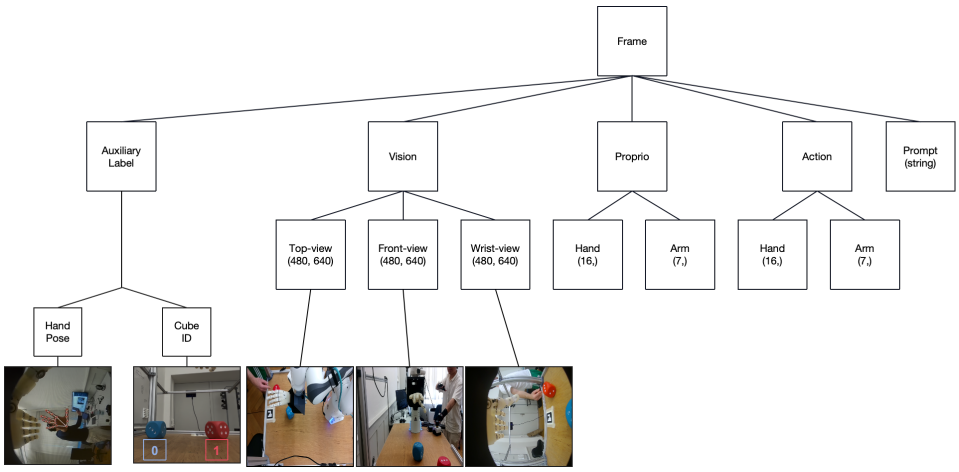


Figure 3: The final composition of the synchronized dataset.

- The 3D hand pose of the human collaborator, estimated using the Mediapipe hand pose detector Google AI Edge [2024]
- The index of the target object the human intends to interact with

These labels serve to improve the model's ability to interpret human motion and disambiguate collaborative goals.

The final structure of the synchronized dataset is illustrated in fig. 3.

3.3 Task Design

We design two toy tasks for this project: "*pick up cube*" and "*pass cube*". These tasks were carefully selected because they are illustrative of core capabilities required for human-robot collaboration. Specifically:

1. They demonstrate the robot's ability to assist a human physically, through object manipulation and transfer.
2. They can be composed into a longer sequence — first picking up an object indicated by the human, then passing it back — showcasing the model's ability to execute long-horizon, goal-directed behavior.
3. They require the robot to interpret human body language rather than relying on explicit natural language instructions, aligning with our goal of enabling tacit understanding.

The "*pick up cube*" task involves two cubes placed on a table — one red and one blue. The human collaborator points to one cube, and the robot must infer the intention and pick up the designated object.

The "*pass cube*" task begins with the robot already holding a cube. The robot is required to pass the object to the human collaborator and release it appropriately.

We collected 60 trajectories for each cube in the pick-up task (120 total), and 200 trajectories for the red cube and 60 for the blue cube in the pass task.

3.4 Vision-Language-Action Model

Our approach builds upon Open-VLA Kim et al. [2024], a recently proposed vision-language-action model. For visual perception, Open-VLA incorporates pre-trained encoders from SigLIP Zhai et al. [2023] and DINOv2 Oquab et al. [2023]. Language inputs are processed using a pre-trained LLaMA2-7B model Touvron et al. [2023]. These components are integrated into a unified multimodal transformer that fuses visual, linguistic, and proprioceptive inputs to generate robot actions.

To better adapt Open-VLA to collaborative settings, we introduce several key modifications, illustrated in fig. 4, and analyzed in subsequent sections:

1. **FiLM conditioning** Perez et al. [2018]: We insert FiLM layers into both vision encoders to improve cross-modal conditioning from text.
2. **Auxiliary intention loss**: We add an auxiliary prediction head to explicitly learn human intention by regressing collaborator hand pose.
3. **Action post-processing**: We constrain action predictions to a more compact and structured subspace, improving stability and learning efficiency.
4. **Directional loss**: We apply a directional loss on end-effector pose that emphasizes directional alignment while downweighting magnitude.

These modifications collectively improve the model's ability to interpret human cues and generate responsive, context-aware robot behavior in collaborative tasks.

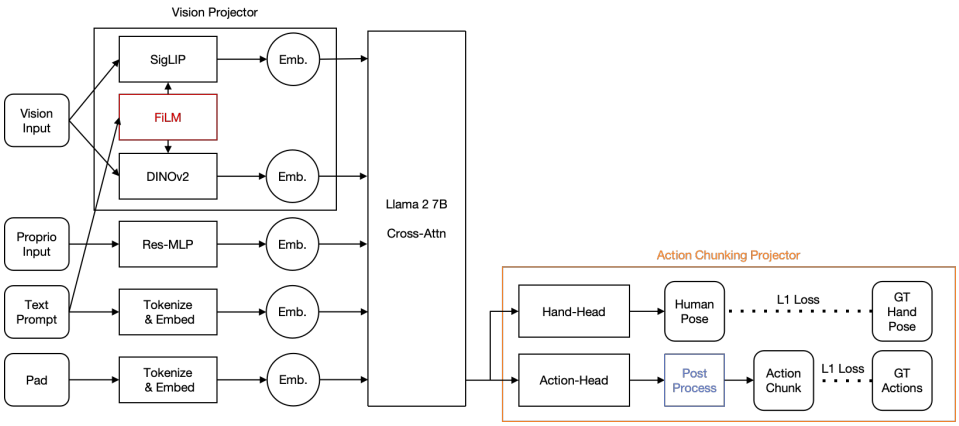


Figure 4: The modified model structure. Red block represents the FiLM layers added to vision encoders, orange block represents the modified action chunking projector, blue block represents the modified action post-processing module.

3.5 FiLM Conditioning

Feature-wise Linear Modulation (FiLM) Perez et al. [2018] is a technique for conditioning a vision encoder on additional inputs, typically text. FiLM layers apply affine transformations to feature maps, where the scale and bias are functions of the conditioning input. This enables the model to dynamically adjust visual representations based on linguistic context.

In the context of VLA models, FiLM layers allow the vision backbone to better align visual perception with task-specific language prompts. We incorporate FiLM conditioning into both vision encoders (SigLIP and DINOv2) and evaluate its impact on task performance in collaborative settings.

3.6 Auxiliary Loss

To enhance the model's understanding of human intent, we introduce human pose priors into training. A straightforward approach would be to extract pose-related features and feed them into the model via cross-attention. However, this method does not scale well: as the number of tasks and priors (e.g., grasp points, object bounding boxes) increases, it would require designing and maintaining multiple feature extractors.

Instead, we adopt an auxiliary loss formulation that encourages the model to implicitly learn human intention cues. Specifically, we add an auxiliary prediction head—referred to as the *hand head*—in parallel with the *action head*. This head receives the same model input and is trained to predict: (1) the 2D hand pose of the collaborator in each camera view, and (2) the color of the target cube. Hand pose annotations are extracted using MediaPipe Google AI Edge [2024], and the target object label is derived from task metadata.

The auxiliary loss is defined as the L2 distance between the predicted and ground-truth labels. During inference, the *hand head* is disabled, as it does not contribute to action generation.

3.7 Action Post-processing

The original action space—comprising 3D position, 4D rotation (quaternion), and 16 joint positions—totals 23 dimensions. However, the underlying structure of valid actions likely lies on a lower-dimensional manifold, making it difficult for the model to learn effectively in the raw space.

To address this, we reformulate the action space so that the model predicts actions in a compact, transformed space, which are then mapped back to the original representation via post-processing. This process consists of three stages:

- **Position**: Let p denote the current end-effector position, and a'_p the model's predicted delta. The final position command is computed as $a_p = p + a'_p$.
- **Rotation**: Let q be the current end-effector rotation in quaternion form, and $d = (\omega, x, y, z)$ be the a delta quaternion. The output rotation is computed as $a_r = q \cdot d$. The model predicts in the rotation vector form: $a'_r = \frac{(x,y,z)}{\sqrt{1-\omega^2}}$.
- **Hand joints**: We apply PCA to the 16-dimensional hand joint states in the training data and retain the top principal components. During inference, the model predicts in this low-dimensional PCA space, and the full joint configuration is reconstructed via inverse PCA.

3.8 Directional Loss

To improve the stability and relevance of end-effector motion, we design a **directional loss** that emphasizes the direction of movement rather than its magnitude. Let x denote the predicted delta pose, and y the ground-truth delta pose. We decompose x into two orthogonal components: $x_\parallel$ (parallel to y) and $x_\perp$ (orthogonal to y).

The directional loss is defined as:

$$\mathcal{L}_{\text{dir}} = r \cdot \frac{\|x_\parallel\|_2}{\|y\|_2 + r} + \|x_\perp\|_2,$$

where $r < 1$ is a scaling factor. When $\|y\|_2 \gg r$, the loss emphasizes directional alignment; when $\|y\|_2 \ll r$, it reduces to a standard L2 loss.

We evaluate this loss function's effect on model performance in later sections.

3.9 Training Pipeline

Our training pipeline is based on OpenVLA-OFT Kim et al. [2025], with significant modifications to support collaborative learning. Upon model initialization, the entire network is cast to `bfloat16` to reduce memory usage, and LoRA adapters are injected into all linear layers to enable efficient fine-tuning. FiLM adapters are injected to both vision encoders depending on whether the option is enabled.

The training data is collected using the procedure described in sections 3.2.1 and 3.2.2. Each dataset file contains a single demonstration trajectory. During training, frames are sampled randomly from these trajectories by the data loader. Each sampled frame is pre-processed into model-ready tensors in the training processor, then assembled into mini-batches by the training collator.

The model receives the processed inputs and computes the predicted outputs. A composite loss—consisting of action loss and auxiliary loss—is computed and optimized using the Adam optimizer Adam et al. [2014].

The full training pipeline is illustrated in fig. 5.

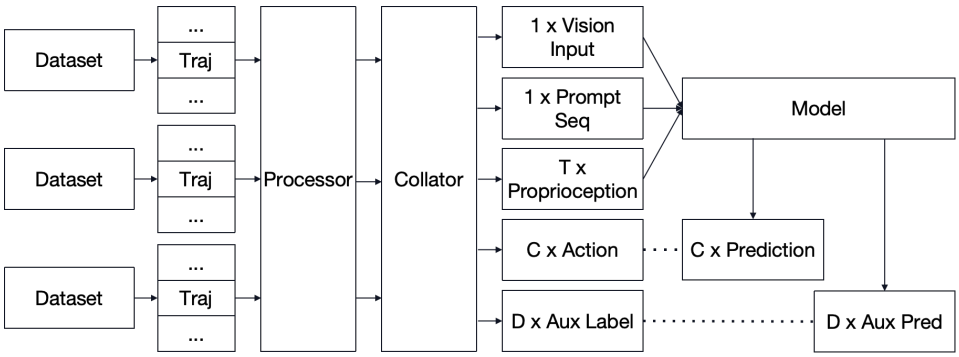


Figure 5: Overview of the training pipeline. The training is carried out with data distribution on a 4 GPU computation node.

We carefully tuned the hyperparameters to ensure stable and efficient data-distributed training on a 4×H100 GPU cluster. The final configuration is summarized below:

- **Batch size**: 6 samples/GPU × 4 GPUs = 24 samples/iteration
- **Epochs**: 20
- **Learning rate**: 3e-4
- **LoRA rank**: 32
- **Action chunk size**: 16
- **Proprioception history length**: 2
- **Vision history length**: 1
- **Head hidden size**: 1024
- **LLM hidden size**: 4096
- **Average training runtime**: 12 hours

3.10 Inference Pipeline

During inference, the system must stream real-time observations from the robot and cameras to the model, which in turn outputs action predictions with minimal latency.

We designed a dedicated robot interface to manage low-level hardware communication and forward sensory data to a model host. The model host runs the fine-tuned collaborative VLA model in inference mode, processes the incoming data, and returns the predicted robot actions. To support long-horizon tasks, we integrate a rule-based high-level planner that dynamically generates text prompts, allowing the model to chain multiple primitive behaviors into goal-directed sequences.

In our demonstration, we combined the "pick up cube" and "pass cube" tasks into a single long-horizon pipeline. The high-level planner monitors the vertical position of the robotic hand, and once a threshold height is exceeded—indicating that the pickup is complete—it automatically switches the text prompt from a pickup command to a passing command. This is aligned with the way demonstrations were collected: the teleoperator always lifts the robot hand after picking up a cube, providing a reliable signal for transition.

The overall inference pipeline is illustrated in fig. 6. When executed on a laptop equipped with an NVIDIA RTX 4090 GPU, the end-to-end latency (including model inference and action mapping) was approximately 0.3 seconds. While this was found to be marginally acceptable by human collaborators, further latency reductions remain an important direction for future improvement.

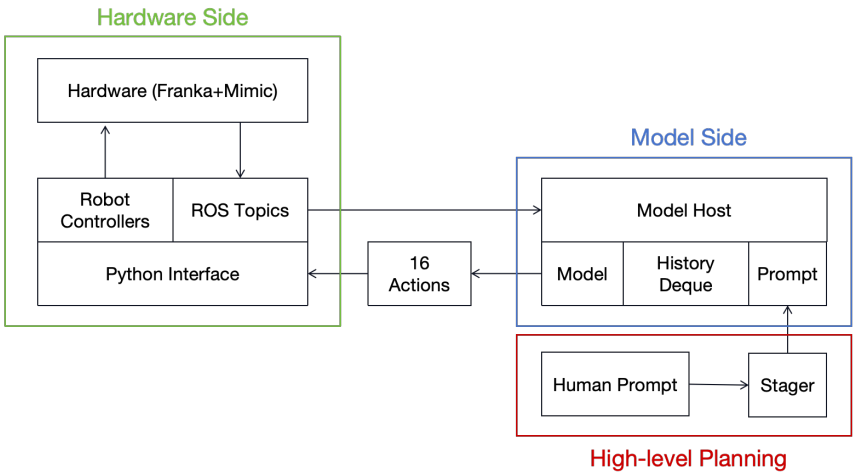


Figure 6: Inference pipeline. The full pipeline consists of a hardware side interface (green), a model side interface (blue) and a high-level planner (red).

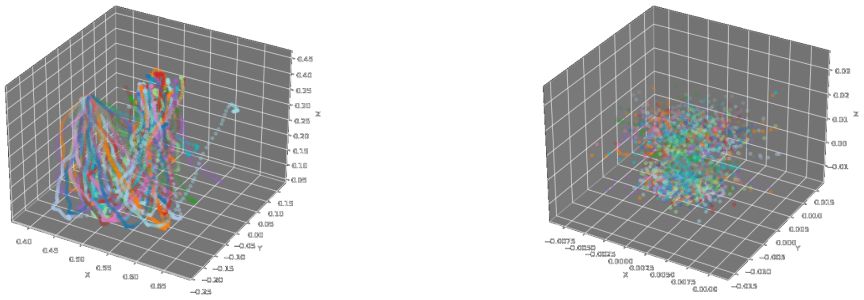
4 Analysis

4.1 Action Space Analysis

The motivation behind action post-processing is based on the assumption that the true action space lies on a low-dimensional manifold embedded in a high-dimensional space. Without explicitly modeling this structure, the model may struggle to learn meaningful mappings. By reducing the dimensionality, the action manifold can be transformed into a more compact and convex representation, facilitating more efficient learning.

We first analyze the action subspace related to end-effector pose. As shown in fig. 7a, the distribution of the raw xyz position components across trajectories is highly non-convex. However, when we differentiate the action sequence—i.e., consider relative rather than absolute motion—the resulting delta poses exhibit a much smoother and near-Gaussian distribution fig. 7b.

Next, we examine the hand joint subspace, which has 16 dimensions. We hypothesize that despite this high dimensionality, the actual configuration space is low-dimensional. To test this, we perform principal component analysis (PCA) on all hand joint states in the training set. The results, shown in fig. 8, reveal that just four principal components account for 96% of the total variance. This suggests that PCA-reduced components can be effectively used as the action representation, replacing the original high-dimensional hand joint space.



(a) Original action distribution

(b) Delta action distribution

Figure 7: Differentiating the action sequence results in a smoother and more normally distributed action space.

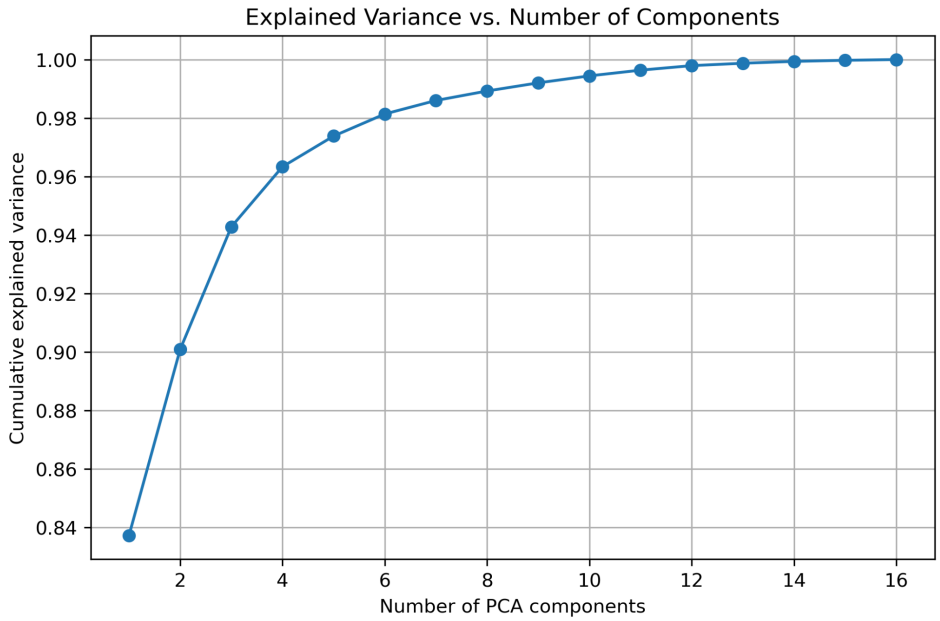


Figure 8: PCA analysis of hand joint states. Four principal components explain 96% of the variance.

4.2 Ablation Studies

We conduct ablation studies to evaluate the contributions of individual design components in the collaborative VLA model. The results are presented in fig. 9.

Several insights can be drawn from the experiments:

- **Action post-processing** is the most critical component, yielding the largest improvement across all metrics.
- **Auxiliary loss** on hand pose provides consistent, though modest, gains—highlighting the benefit of including human-pose priors.
- **Directional loss** consistently reduces performance across metrics, suggesting it may overly constrain the learning dynamics.
- **FiLM conditioning** improves performance on low-dimensional objectives (e.g., L2 and PCA losses) but appears detrimental for other loss types.

4.3 Auxiliary Predictions and Trainer Overfitting

When we evaluate the model in real world set ups, we discovered an interesting fact: when trained on data collected from one specific collaborator, the model accurately interprets their intentions during inference. However, when interacting with a different person, it fails to adapt and instead reverts to a fixed routine—behavior as if no meaningful commands were received.

We named this phenomenon as **trainer overfitting**: the model becomes overly specialized to the behavior of a single demonstrator. This overfitting is also common in intelligent creatures, for example, dogs only follow the commands of their owners Merola et al. [2012]. To further analyze this phenomenon, we conducted an experiment that uses auxiliary loss to quantify **trainer overfitting**.

We trained the model with human collaborator A and tested the model on data from both collaborator A and collaborator B, and plotted the loss curve in fig. 10. The elevated loss confirms that the model fails to generalize across different collaborators.

This finding opens new possible research directions on how to reduce **trainer overfitting** in collaborative robots, or maybe, service dogs.

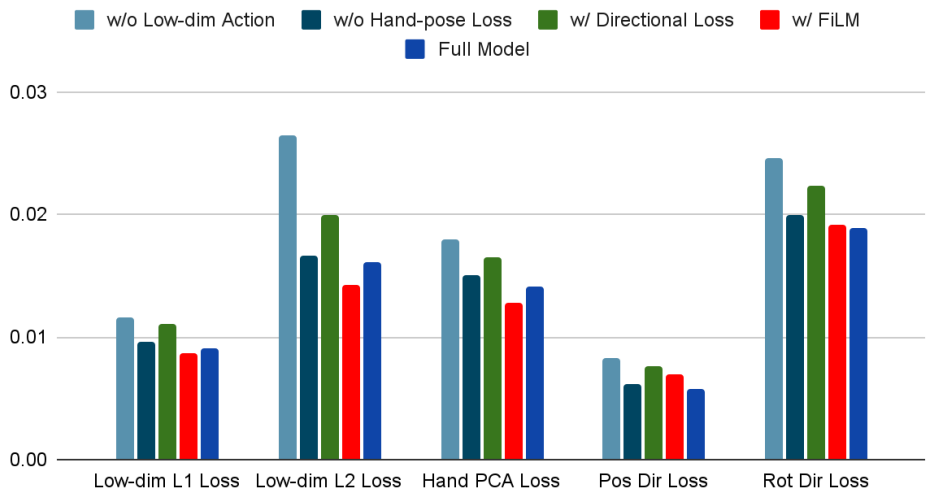


Figure 9: Ablation study results. The "Full Model" includes action post-processing and hand-pose auxiliary loss, but excludes directional loss and FiLM conditioning.

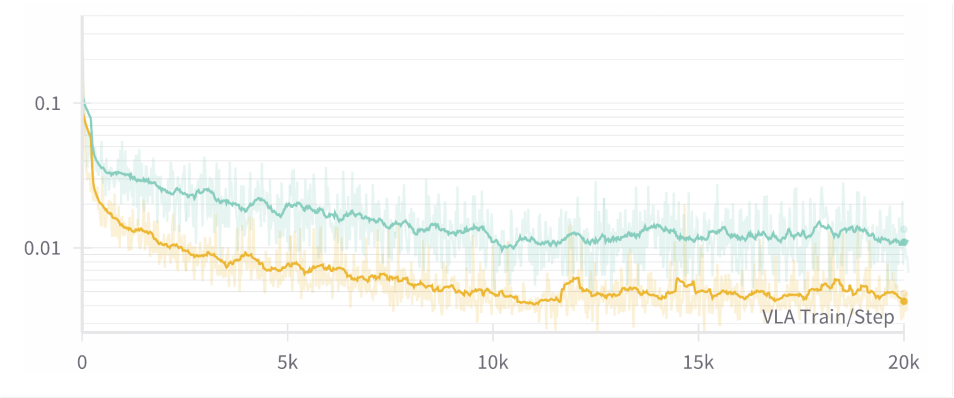


Figure 10: Auxiliary loss when evaluated on same-hand vs. different-hand collaborators. Orange curve: same hand as training. Blue curve: different hand.

4.4 Real World Evaluations

We carried out a real-world evaluation on the model. The snap shots of real-world executions are included in fig. 11.

With the inference pipeline introduced in section 3.10, we are able to test the model in real-time interactively. Due to time constraints, we are not able to roll out enough trials for all the tasks to report success rate for each ablation on each task. In addition, during testing, the human collaborator is different from the trained data collected, thus influencing the overall performance of the real world evaluation due to the **trainer overfitting** phenomenon we found in section 4.3.

The model completed the combined long-horizon task (pick up then pass the cube) successfully once, in a total number of 10 trials. In all the other 9 cases, the model failed to recognize the cube that the human collaborator is pointing to, thus could not pick up the cube.

We believe that with more diverse training data collected, the model can do better in real-world scenarios.

Figure 11: The snapshots of the robot carrying out successful real-world inference. The rows from top to bottom are the front and top view of task *pass cube*, front and top view of task *pick up cube* and front and top view of the combined long-horizon task. Each sequence is executed from left to right.

5 Conclusion

In this project, we present a method for enabling robots to collaborate with humans through tacit understanding instead of relying solely on language-based prompting. We adapt a pre-trained Vision-Language-Action (VLA) model to collaborative tasks by introducing several architectural modifications, including FiLM conditioning, auxiliary hand-pose prediction, and action-space post-processing. These modifications improve the model's ability to perceive, interpret, and respond to human intentions.

To demonstrate the effectiveness of our approach, we constructed a real-world collaborative dataset and designed two representative tasks: *pick up cube* and *pass cube*. By combining these tasks, we further demonstrated the model's ability to handle long-horizon interactions. Our analysis shows that the proposed architectural changes are effective, and that the model can be trained efficiently on modest-scale hardware.

Our findings suggest that large VLA models can be effectively adapted to physical collaboration tasks when equipped with the appropriate inductive biases. This approach opens the door to more intuitive and efficient forms of human-robot interaction.

6 Challenges and Future Work

While the collaborative VLA framework demonstrates a promising direction for enabling intuitive human-robot interaction, several challenges remain.

A key issue is *trainer overfitting* (see section 4.3), where the model becomes overly specialized to a single human demonstrator. This limits generalization across different users. Although this issue may diminish with larger-scale training and more diverse collaborators, it remains a significant limitation in the current system.

Another critical challenge is latency. Collaborative interaction is highly sensitive to response time, requiring the robot to react rapidly to human motions. Our current inference pipeline exhibits a latency of approximately 0.3 seconds—marginally acceptable in real-world settings. Reducing this latency, potentially through techniques such as temporal ensembling Zhao et al. [2023], is essential for improving the fluidity of interaction.

Finally, the system's high-level planning is currently rule-based, limiting adaptability in dynamic environments. More flexible approaches—such as embodied chain-of-thought reasoning Zawalski et al. [2024]—may offer improved performance in task sequencing and long-horizon planning.

Addressing these challenges will be crucial for deploying collaborative VLA systems in more complex, open-ended scenarios.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6), 2014.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Judith Bütepage, Ali Ghadirzadeh, Özge Öztimur Karadağ, Mårten Björkman, and Danica Kragic. Imitating by generating: Deep generative models for imitation of interactive tasks. *Frontiers in Robotics and AI*, 7:47, 2020.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Google AI Edge. Mediapipe. `https://github.com/google-ai-edge/mediapipe`, 2024. Accessed: 2025-07-13.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Yanlong Huang, Joao Silvério, Leonel Rozo, and Darwin G Caldwell. Generalized task-parameterized skill learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5667–5474. IEEE, 2018.

Yuchen Ji, Zequn Zhang, Dunbing Tang, Yi Zheng, Changchun Liu, Zhen Zhao, and Xinghui Li. Foundation models assist in human–robot collaboration assembly. *Scientific Reports*, 14(1):24828, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

Isabella Merola, Emanuela Prato-Previde, and Sarah Marshall-Pescini. Dogs' social referencing towards owners and strangers. *PloS one*, 7(10):e47653, 2012.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Loris Roveda, Shaghayegh Haghshenas, Marco Caimmi, Nicola Pedrocchi, and Lorenzo Molinari Tosatti. Assisting operators in heavy industrial tasks: On the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules. *Frontiers in Robotics and AI*, 6:75, 2019.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Chao Wang, Stephan Hasler, Daniel Tanneberg, Felix Ocker, Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, and Michael Gienger. Lami: Large language models for multi-modal human-robot interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2024.

Weronika Wojtak, Flora Ferreira, Paulo Vicente, Luís Louro, Estela Bicho, and Wolfram Erlhagen. A neural integrator model for planning and value-based decision making of a robotics assistant. *Neural Computing and Applications*, 33(8):3737–3756, 2021.

Liang Yan, Xiaoshan Gao, Xiongjie Zhang, and Suokui Chang. Human-robot collaboration by intention recognition using deep lstm neural network. In *2019 IEEE 8th International Conference on Fluid Power and Mechatronics (FPM)*, pages 1390–1396. IEEE, 2019.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

Jianjing Zhang, Hongyi Liu, Qing Chang, Lihui Wang, and Robert X Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP annals*, 69(1):9–12, 2020.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.