CVLAMP: Compliant Vision Language Action Movement Primitives

Vamsi Krishna Origanti[1,2], Alexander Fabisch[1], and Frank Kirchner[1,2]

***Abstract*— We present Compliant Vision-Language-Action Movement Primitives (CVLAMP), a closed-loop framework for contact-rich manipulation. CVLAMP conditions a VLA backbone on images, language, Cartesian state, and force history, decoding a compact Cartesian motion primitive in place of a dense action chunk. Each primitive couples a Dynamic Movement Primitive (DMP) pose rollout with shared-phase wrench and gripper profiles, executed by a phase-governed controller that can run, slow, hold, or abort from real-time feedback. The architecture extends a pretrained $\pi_{0.5}$ flow-matching VLA with cross-attention-fused state and force encoders, trained with a composite regression, rollout-consistency, and mode-classification objective, via single-GPU frozen-backbone adaptation of the released checkpoint (LoRA staged for later). In simulation, the model localizes the goal object from vision and reliably approaches and grasps the target on held-out, unseen object positions; task completion and real-robot deployment remain future work.**

I. Introduction

Contact-rich manipulation is difficult for learned policies because raw action trajectories are high-dimensional and poorly suited to closed-loop execution under contact uncertainty. CVLAMP instead maps multimodal observations to structured Cartesian motion primitives: a pretrained vision-language backbone processes images and instructions, and a lightweight decoder outputs a primitive packet of DMP-style pose parameters, wrench/gripper coefficients, a duration, and a mode label over a shared phase. A closed-loop executor rolls out the primitive online, adjusting from phase, pose, and wrench feedback.

We contribute (1) a hybrid VLA-primitive architecture fusing a PaliGemma/Gemma backbone with Cartesian-state and force-history encoders; (2) a synchronized pose/wrench/gripper primitive with an explicit phase governor; (3) a composite training objective and staged adaptation recipe (frozen-backbone Stage 1 reported; LoRA planned) for a released $\pi_{0.5}$ checkpoint; and (4) a complete ROS 2 implementation, demonstrated in simulation and designed for direct transfer to real hardware.

II. Related Work

Vision-language-action policies: VLA policies such as $\pi_{0.5}$ [1] use a flow-matching action expert on a PaliGemma-style backbone [2]. CVLAMP adopts the $\pi_{0.5}$ backbone but replaces its action-chunk head with a primitive decoder, since these regress low-level actions with no explicit contact representation. More recently, ForceVLA [3], FD-VLA [4], and FM-VLA [5] add force-awareness directly into a VLA backbone (a force-aware MoE, a distilled force token, and force-history memory).

Movement primitives and compliant control: DMPs [6] and their Cartesian/orientation extensions [7] allow compact, compliant motion representations suited to force-modulated execution. We use `movement_primitives` [8] for Cartesian DMP fitting and rollout. However, DMPs lack the semantic grounding of modern VLA models.

Multimodal contact-rich manipulation: Lee et al. [9] learn joint vision-touch representations for contact-rich insertion via self-supervision; CVLAMP instead predicts an explicit, structured primitive with a rule-based governor, keeping closed-loop safety interpretable rather than latent.

Where CVLAMP sits: PrimitiveVLA [10] sequences pre-extracted primitives via a separate VLM planner, and Language Movement Primitives [11] uses a VLM to set DMP parameters; CVLAMP instead regresses continuous primitive parameters directly from perception, end-to-end, with an explicit, rule-based phase governor rather than a dense force-conditioned action chunk.

III. System Architecture

A. Primitive representation

Fig. 1 summarizes the pipeline. CVLAMP represents a short-horizon action as a primitive

$$p_t = \{\theta_{\text{pose}}, g_{\text{pose}}, \theta_{\text{wrench}}, \theta_{\text{gripper}}, \tau, u_{\text{goal}}, m\}, \qquad (1)$$

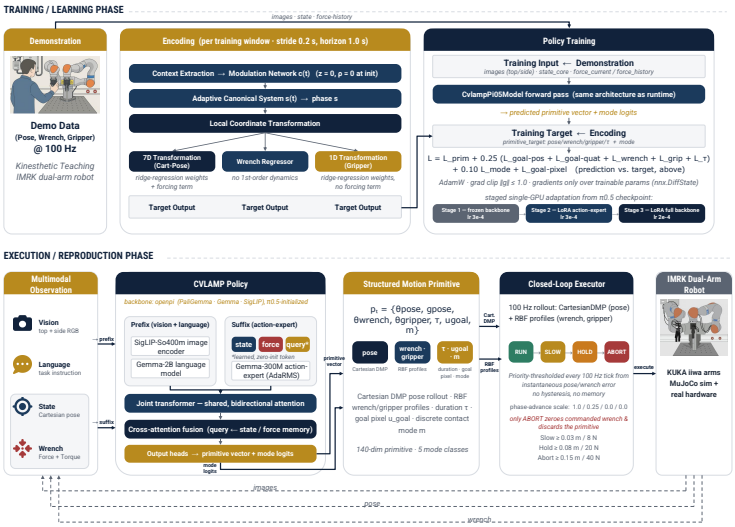
where θ_{pose} are Cartesian DMP basis weights, g_{pose} is the goal pose, θ_{wrench}/θ_{gripper} are RBF coefficients over a shared phase $\phi \in [0, 1]$, τ is the primitive duration, u_{goal} is a predicted goal-pixel location (lifted to 3D via depth), and m is a discrete mode label – 10 RBF bases each for pose/wrench/gripper, a 7-dim pose goal, 6-dim wrench, 1-dim gripper, and 2-dim goal pixel give a 140-dimensional primitive plus a 5-way mode logit.

B. Model

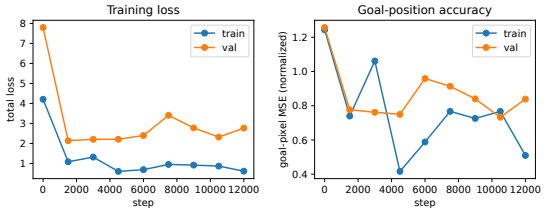
The backbone is the released $\pi_{0.5}$ checkpoint: SigLIP-So400m and Gemma-2B process the prefix; a Gemma-300M, AdaRMS-conditioned action-expert processes a suffix of state/force tokens with a learned query token that attends over state/force memory via sigmoid-gated cross-attention. We replace $\pi_{0.5}$'s action-chunk head with a deterministic forward pass: a linear head regresses the primitive and mode logits, while a soft-argmax head over SigLIP patch tokens predicts the goal pixel u_{goal}.

[1]Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI GmbH), Bremen, Germany

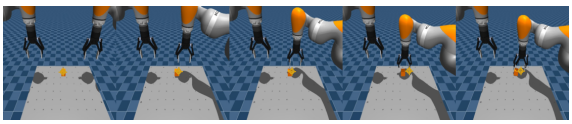
[2]Robotics Research Group, University of Bremen, Bremen, Germany



(a) Methodology: training (top), phase-governed execution (bottom).



(b) Training/validation curves.



(c) Closed-loop rollout: goal-pixel tracking.

Fig. 1. CVLAMP methodology and preliminary results (prompt: *"pick the orange cube and lift it"*): training curves and a closed-loop rollout.

C. Closed-loop executor

At runtime, the pose block and goal are rolled out with a Cartesian DMP [6], [7] via `movement_primitives` [8]; wrench/gripper trajectories are reconstructed from the RBF coefficients at the same phase, with the gripper clamped open by a hysteresis gate until near the target. A phase governor compares pose/wrench error at 100 Hz against fixed thresholds (SLOW $\geq$0.03 m/8 N, HOLD $\geq$0.08 m/20 N, ABORT $\geq$0.15 m/40 N), scaling phase-advance and zeroing wrench on ABORT; wrench commanding is currently disabled, so the governor acts on pose error alone.

IV. Training and System Implementation

The training loss combines primitive regression with auxiliary rollout-consistency terms:

$$\mathcal{L} = \mathcal{L}_{\text{prim}} + 0.25\left(\mathcal{L}_{\text{goal-pos}} + \mathcal{L}_{\text{goal-quat}} + \mathcal{L}_{\text{wrench}} + \mathcal{L}_{\text{grip}} + \mathcal{L}_{\tau}\right) + 0.10\,\mathcal{L}_{\text{mode}} + \mathcal{L}_{\text{goal-pixel}}, \tag{2}$$

where $\mathcal{L}_{\text{prim}}$ is the primitive MSE, $\mathcal{L}_{\text{goal-quat}}$ penalizes quaternion misalignment, $\mathcal{L}_{\text{wrench}}/\mathcal{L}_{\text{grip}}$ are MSE over resampled RBF rollouts, $\mathcal{L}_{\text{mode}}$ is cross-entropy, and $\mathcal{L}_{\text{goal-pixel}}$ is the unweighted MSE on u_{goal} (the largest term). Training initializes from the released $\pi_{0.5}$ checkpoint [1] and proceeds in stages on a single 24 GB GPU: (1) freeze the backbone, training only CVLAMP-specific layers; (2) add LoRA [12] to the action-expert once Stage 1 saturates; (3) optionally extend LoRA across the backbone for multi-task scaling. Demonstrations are sharded into 0.2 s-stride, 1.0 s-horizon windows targeting the runtime primitive representation.

A. Preliminary results

Stage 1 (frozen backbone) trains only 6.9M of 3.36B parameters (0.21%) for 12,000 steps on 8 demonstrations (Fig. 1b); we report the step-4,500 checkpoint, before the primitive-regression loss overfits. Evaluated offline on raw, un-normalized u_{goal} (distinct from Fig. 1b's normalized-space curve), goal-pixel MSE (normalized units^2) is 0.00207 (train), 0.00246 (val), 0.00111 held-out on this single-observation, open-loop metric; held-out matching train may simply reflect an easier position distribution, not generalization. In closed-loop simulation (Fig. 1c) the system localizes, approaches, and grasps the target from vision alone; task completion and real-robot deployment remain future work.

V. Conclusion

CVLAMP pairs a foundation VLA backbone with a compact, physically structured motion primitive and an explicit phase governor without sacrificing closed-loop safety or interpretability. The architecture, training recipe, and ROS 2 system are demonstrated in simulation. Real-robot deployment is planned.

References

[1] Physical Intelligence, K. Black *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[2] L. Beyer *et al.*, "PaliGemma: A versatile 3B VLM for transfer," *arXiv preprint arXiv:2407.07726*, 2024.

[3] J. Yu *et al.*, "ForceVLA: Enhancing VLA models with a force-aware MoE for contact-rich manipulation," *arXiv preprint arXiv:2505.22159*, 2025.

[4] R. Zhao *et al.*, "FD-VLA: Force-distilled vision-language-action model for contact-rich manipulation," *arXiv preprint arXiv:2602.02142*, 2026.

[5] R. Li *et al.*, "FM-VLA: Force-based memory for vision-language-action models in contact-rich manipulation," *arXiv preprint arXiv:2607.18231*, 2026.

[6] A. J. Ijspeert *et al.*, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[7] A. Ude *et al.*, "Orientation in Cartesian space dynamic movement primitives," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2997–3004.

[8] A. Fabisch, "movement_primitives: Imitation learning of Cartesian motion with movement primitives," *Journal of Open Source Software*, vol. 9, no. 97, p. 6695, 2024.

[9] M. A. Lee *et al.*, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," *arXiv preprint arXiv:1810.10191*, 2019.

[10] Y. Li *et al.*, "PrimitiveVLA: Learning reusable motion primitives for efficient and generalizable robotic manipulation," *arXiv preprint arXiv:2605.28634*, 2026.

[11] Y. Dai *et al.*, "Language movement primitives: Grounding language models in robot motion," *arXiv preprint arXiv:2602.02839*, 2026.

[12] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.