Language-Grounded Decoupled Action Representation for Robotic Manipulation

Supplementary Material

6. Training and Implementation Details

LaDA is implemented in PyTorch and trained in mixed precision on 4 NVIDIA RTX 5090 GPUs. All experiments use a single third-person RGB observation without temporal stacking, depth input, or wrist-mounted sensing.

Pretraining. We pretrain LaDA for 100 epochs with a global batch size of 256. Optimization uses AdamW (learning rate 5×10^{-6}, cosine decay, weight decay 0.1), and all components—including the CLIP visual and language encoders, FiLM layers, MLP adapters, and primitive classifiers—are trained jointly.

The soft-label similarity matrix S (defined in Eq. (1)) is constructed using primitive affinity weights $(w_t, w_r, w_g) = (0.65, 0.25, 0.10)$ for translation, rotation, and gripper primitives. The dual-path contrastive losses in Eqs. (2)–(3) use a temperature of $\tau = 0.07$, and the contrastive–imitation balance coefficient in Eq. (4) is set to $\lambda = 1.0$.

To stabilize training, we compute an exponential moving-average estimate $\mathrm{MA}(\cdot)$ for both $\mathcal{L}_{\mathrm{CL}}$ and $\mathcal{L}_{\mathrm{IL}}$ using a smoothing factor of 0.99 over the most recent $K = 50$ iterations. These smoothed losses determine the adaptive weights w_{CL} and w_{IL}.

Fine-tuning. All parameters of LaDA are fine-tuned end-to-end using the same optimizer configuration as in pretraining. Depending on task complexity, we fine-tune for 10–30 epochs. An $\mathcal{L}_1$ trajectory regression loss over continuous 7-DoF actions is used to refine low-level control accuracy while preserving the semantic alignment learned during pretraining.

7. Language-Grounded Action Decomposition

This section provides additional details on how continuous 7-DoF actions are converted into the language-grounded motion primitives used during pretraining in Sec. 3.2.

For each action, the 3D translation and 3D rotation components are first normalized into metric and angular units. The dominant motion axis is then identified for both translation and rotation, and each component is discretized into a small set of bins that encode both direction and approximate magnitude. This yields a compact symbolic representation of end-effector motion between consecutive frames. The gripper state is handled independently and categorized as opening, closing, or no change.

Each primitive is subsequently mapped to a structured natural-language description following the templates introduced in the main paper. Translation primitives follow *"Move [dist] meters along [axis]"*, rotation primitives follow *"Rotate [angle] degrees around [axis]"*, and gripper changes are described as *"Open the gripper"*, *"Close the gripper"*, or *"Do not change the gripper"*. Positive and negative values indicate movement along the positive or negative direction of the corresponding axis. When the translation or rotation magnitude falls below a negligible threshold, an explicit no-op description (e.g., *"do not move the arm"*, *"do not rotate the arm"*) is generated to ensure that stationary behavior is encoded explicitly.

The translation, rotation, and gripper templates are then concatenated into a single structured action description, which serves as the linguistic supervision used in the semantic-guided contrastive learning objective. To enhance linguistic diversity and avoid overfitting to fixed templates, each description is further paraphrased into several semantically equivalent variants using GPT-4o and DeepSeek.

8. Additional Rollout Visualizations

This section provides supplementary qualitative rollouts illustrating LaDA's execution behavior on a subset of tasks from the LIBERO and MimicGen benchmarks (Figs. 8 and 9). The presented examples are selected as representative cases covering different categories of manipulation behaviors. All visualizations are shown from a third-person RGB viewpoint. In addition, we include real-world rollout visualizations showing the 3D end-effector trajectories (Fig. 10).

9. Limitations and Discussions

LaDA is evaluated under a single third-person RGB viewpoint without depth or wrist-mounted sensing. While this setup is sufficient for the tasks considered, relying solely on an external RGB camera may limit performance in scenes with severe occlusions or fine-grained contact interactions where additional modalities (e.g., depth or tactile cues) could provide complementary information. In addition, the action decomposition is based on a fixed set of translation, rotation, and gripper primitives designed for semantic interpretability. Although effective for general manipulation, such predefined primitives may be less expressive for highly dexterous behaviors or continuous in-hand manipulation. These factors outline the current scope of LaDA and suggest avenues for expanding its applicability in more complex real-world settings.

Figure 8. Representative rollouts from LIBERO, covering spatial, object-centric, goal-conditioned, and long-horizon tasks. All examples use a single third-person RGB view.

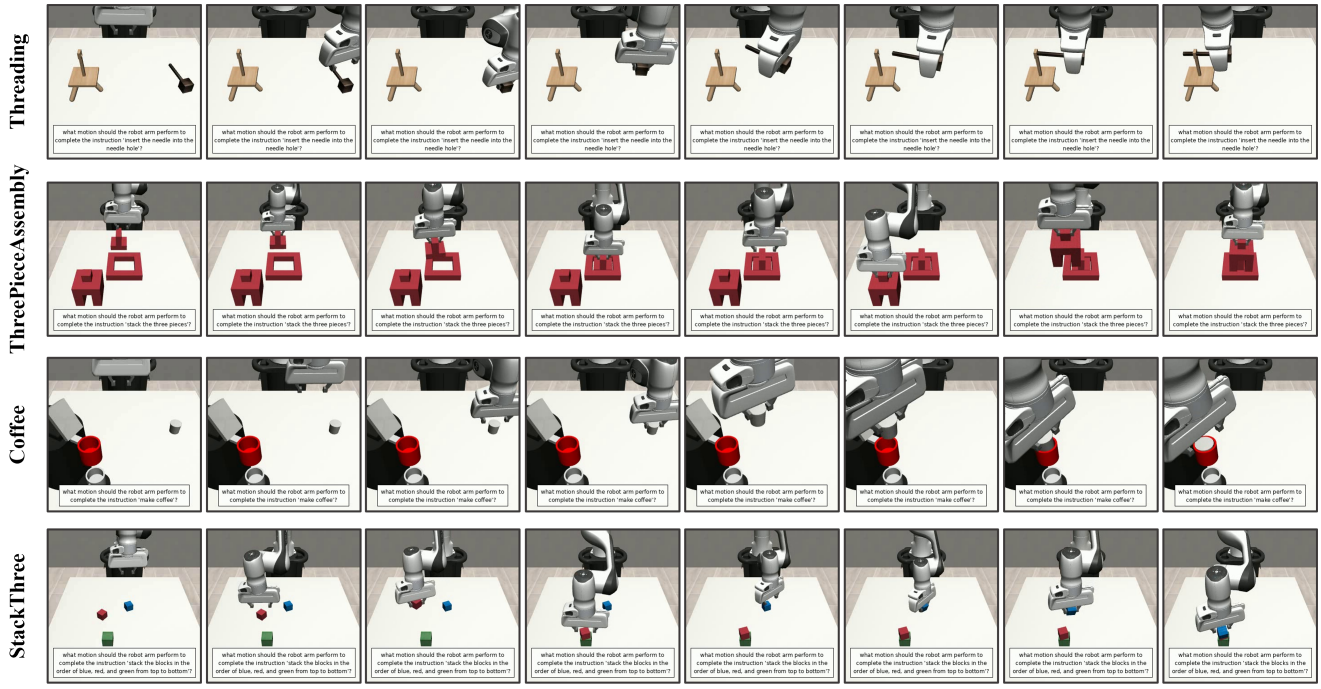

Figure 9. Additional rollouts from MimicGen tasks, including Threading, ThreePieceAssembly, Coffee, and StackThree.

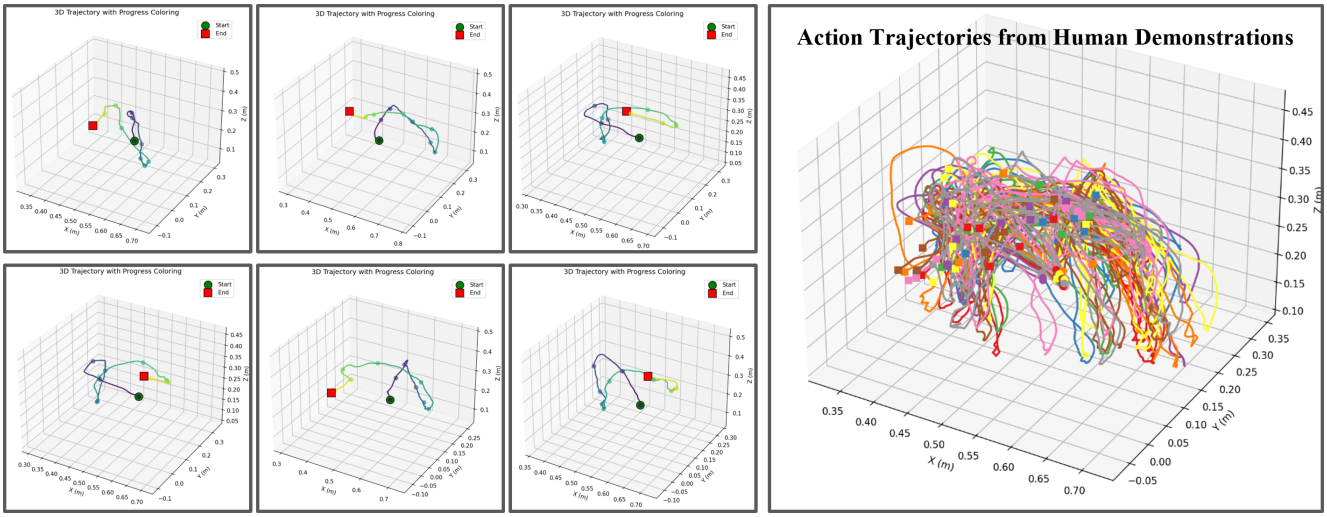

Figure 10. Real-world 3D end-effector trajectories executed by LaDA. Left: Representative trajectories visualized with progress coloring. Right: 3D trajectories extracted from the training dataset.