Supplementary Material for TDSR-VLA: Transition-aware Denoising Sequence Representations for Vision-Language-Action

Dongwoo Kim[1*], Keunho Song[1*], Seungmin Lee[1*], Hwanhee Ju[1], Eun Sang Cha[1†], and Daekyum Kim[1,2†]

[1] Department of Mechanical Engineering, Korea University, Seoul, Republic of Korea
[2] School of Smart Mobility, Korea University, Seoul, Republic of Korea

kimdongwoo@korea.ac.kr, bill1869@korea.ac.kr, albert31115@gmail.com, jhh0415@korea.ac.kr, jessecha@korea.ac.kr, daekyum@korea.ac.kr

[*]Equal contribution. [†]Corresponding authors.

A Training Details and Dimension Reducer

A.1 Training Details

Specifically, we constructed the Vision Planner based on the 2B-parameter Stable Diffusion 3 [3] Medium, integrating a LoRA [5] adapter trained using a mixed-precision [9] approach with FP16 for the base architecture and FP32 for the LoRA [5] weights. Subsequent to the offline precomputation phase, we optimized the Action Expert alongside the Dimension Reducer, which collectively comprise approximately 600M parameters. The comprehensive hyperparameter configurations employed for both modules are detailed in Table 1. Regarding computational resources, the Vision Planner was trained over a period of 2 days using 4×NVIDIA RTX 4090 GPUs, whereas the training process for the Action Expert and MLP Dimension Reducer required 3 days.

A.2 Dimension Reducer

Building on the enhanced multimodal capabilities demonstrated by Multi-Layer Perceptrons (MLPs) in recent large multimodal models [6], we use the Dimension Reducer, which functions as an MLP Projector, to transform high-dimensional features into a unified embedding space within the Action Expert. Designed as a feed-forward architecture, this module comprises an input LayerNorm [1], a linear expansion layer that projects features into a higher-dimensional space with GELU [4] activation, followed by a linear compression layer and a final output LayerNorm [1]. It projects the multimodal context embeddings, action-useful planner-side transition cues (SR_t, USR_t), and SigLIP [11] tokens representing all camera views as well as the future subgoal image into embeddings compatible with the Action Expert, which are then injected as the prefix Key-Value (KV) conditioning.

Table 1: Hyperparameters for the Vision Planner and Action Expert.

	Vision Planner	Action Expert & Reducer
Global batch size	256	240
Learning rate	1.25×10^{-4}	1.0×10^{-4}
LR scheduler [7]	cosine	cosine
LR warm up ratio	0.1	0.03
Epochs	10	10
Optimizer	AdamW [8] (β_1=0.9, β_2=0.81)	AdamW [8] (β_1=0.9, β_2=0.95)
LoRA [5] rank	256	–
Precision	FP16 (LoRA [5] weights: FP32)	FP32
Gradient clipping [10] (L2)	1.0	1.0

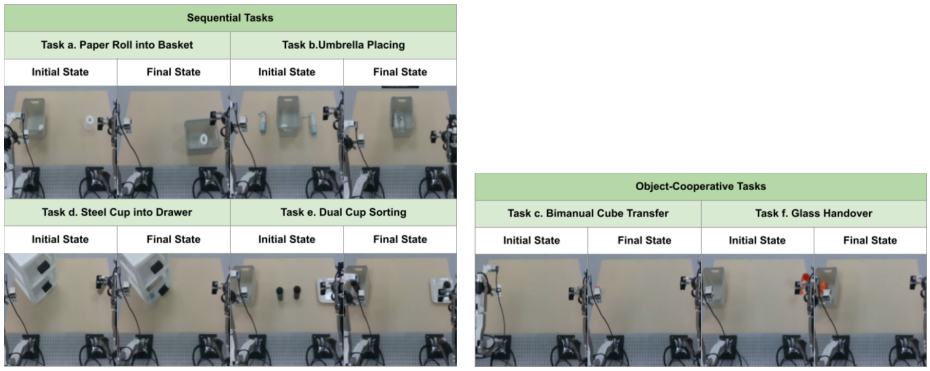
B Real-World Task Details

As illustrated in Fig. 1(a), Tasks a, b, d, and e are organized as sequential tasks in which the two arms execute temporally ordered sub-actions with distinct role assignments. In contrast, Fig. 1(b) shows object-cooperative tasks (Tasks c and f), where both arms coordinate around a shared object through relay or handover.

B.1 Detailed Task Descriptions for Real-World Evaluation

Overview. We group the six real-world OpenArm [2] tasks into two categories. Tasks a, b, d, and e are sequential tasks, in which the two arms execute temporally ordered sub-actions with distinct role assignments. In contrast, Tasks c and f are object-cooperative tasks, in which both arms must coordinate around the same object through relay or handover. This categorization highlights whether task success depends primarily on step-by-step temporal ordering or on tightly coordinated bimanual object transfer.

Task a. Paper Roll into Basket (Sequential Task). This task evaluates sequential bimanual coordination with asymmetric roles. The left arm first grasps the basket, while the right arm grasps the paper roll. The left arm then moves the basket toward the right side and positions it in front of the right arm. After



(a) Sequential tasks **(b)** Object-cooperative tasks

Fig. 1: Real-world evaluation task details. (a) Sequential tasks, where the two arms perform temporally ordered sub-actions with distinct role assignments. (b) Object-cooperative tasks, where both arms coordinate around a shared object through relay or handover.

the basket is prepared at the target location, the right arm places the paper roll into the basket. The task emphasizes temporally ordered execution, since successful completion depends on the left arm preparing the placement target before the right arm performs the final insertion.

Task b. Umbrella Placing (Sequential Task). This task evaluates sequential execution after simultaneous object acquisition. Both arms first grasp their respective umbrellas. The left arm places its umbrella into the basket first, after which the right arm places the second umbrella into the same basket. The task emphasizes ordered multi-step execution, since the two arms must follow a predefined temporal sequence rather than acting as a cooperative pair on the same object.

Task c. Bimanual Cube Transfer (Object-Cooperative Task). This task evaluates single-object cooperative manipulation across the shared workspace. The right arm first grasps the Rubik's Cube on the right side and transfers it to the center area between the two arms. The left arm then grasps the cube from that shared intermediate location and moves it to the left side. Unlike the sequential tasks, this task requires both arms to coordinate around the same object through an inter-arm relay, thereby testing object-level cooperation and effective workspace extension.

Task d. Steel Cup into Drawer (Sequential Task). This task evaluates long-horizon sequential coordination involving both object manipulation and articulated environment interaction. The left arm grasps the handle of the lower drawer, while the right arm grasps the steel cup. The left arm opens the drawer, enabling the right arm to place the steel cup inside. After the placement is

completed, the right arm retreats and the left arm closes the drawer. The task emphasizes temporally dependent sub-actions, since each later step becomes feasible only after the preceding environmental manipulation has been completed.

Task e. Dual Cup Sorting (Sequential Task). This task evaluates sequential bimanual sorting with object discrimination. The left arm grasps the brown cup and places it into the basket on the left side, while the right arm grasps the green cup and places it onto the plate on the right side. The task emphasizes role-specific execution across two ordered placement objectives rather than cooperative manipulation of a single shared object. It also evaluates whether the policy can distinguish object identity and map each object to its designated target location.

Task f. Glass Handover (Object-Cooperative Task). This task evaluates direct inter-arm handover of a single object. The right arm first grasps the orange wine glass and moves it to the shared workspace between the right and left arms. The object is then handed over to the left arm, which places it into the basket on the left side. Unlike the sequential tasks, this task requires tightly coordinated timing between the two arms around the same object, making stable object handover in the shared inter-arm workspace the central challenge.

Summary of Task Categories. Overall, Tasks a, b, d, and e evaluate whether the policy can decompose long-horizon bimanual behavior into temporally ordered sub-actions with separate arm roles. In contrast, Tasks c and f evaluate whether the policy can maintain object-centric coordination when both arms must manipulate the same object through relay or handover.

C Real-World Robot Experimental Setup

We provide additional details of the real-world robotic setup used in our experiments, including the hardware embodiment, low-level controller, visual observation interface, and action interface. These details are intended to clarify the practical deployment setting and improve reproducibility. The main specifications are summarized in Table 2.

Table 2: Real-World Robot Experimental Setup Spec.

Category	Item	Content
Robot embodiment / hardware configuration	Platform configuration	Bimanual robot platform with 8 actuated axes per arm (rev1–rev8).
	Arm DOFs / joint type / actuator	rev1–rev7 are revolute arm joints; rev8 is a revolute gripper motor axis.
	Control mode	MIT control: hybrid position-velocity-torque command mode.
	Actuator type	DM4340 motors are used mainly for rev1–rev4, and DM4310 motors mainly for rev5–rev8.
	Gripper command type	Position command on rev8.
Low-level Control	Low-level controller type	Pinocchio-based gravity compensation with joint-limit spring terms and torque clamping.
Camera	Network input resolution	256×256.
	RGB or RGB-D	RGB is used in the current confirmed pipeline.
	State vector definition	16-dimensional state: left arm (7 joints + gripper) + right arm (7 joints + gripper).
Policy-control interface	Policy action dimension	16-dimensional.
	Action semantics	$([L_{1..7}, L_g, R_{1..7}, R_g])$.
	Absolute or delta command	Absolute joint targets are published by the policy.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. In: arXiv preprint arXiv:1607.06450 (2016), https://arxiv.org/abs/1607.06450, accessed 30 June 2026
2. Enactic: OpenArm: an open-source humanoid arm for physical AI research and deployment in contact-rich environments. https://github.com/enactic/openarm (2025), gitHub repository. Accessed 30 June 2026
3. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML. Proceedings of Machine Learning Research, vol. 235, pp. 12606–12633. PMLR (2024), https://proceedings.mlr.press/v235/esser24a.html, accessed 30 June 2026
4. Hendrycks, D., Gimpel, K.: Gaussian error linear units (GELUs). In: arXiv preprint arXiv:1606.08415 (2016), https://arxiv.org/abs/1606.08415, accessed 30 June 2026
5. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: ICLR (2022), https://openreview.net/forum?id=nZeVKeeFYf9, accessed 30 June 2026
6. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR. pp. 26296–26306 (2024). https://doi.org/10.1109/CVPR52733.2024.02484, https://openaccess.thecvf.com/content/CVPR2024/html/Liu_Improved_Baselines_with_Visual_Instruction_Tuning_CVPR_2024_paper.html, accessed 30 June 2026
7. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017), https://openreview.net/forum?id=Skq89Scxx, accessed 30 June 2026
8. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019), https://openreview.net/forum?id=Bkg6RiCqY7, accessed 30 June 2026
9. Micikevicius, P., Narang, S., Alben, J., Diamos, G.F., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H.: Mixed precision training. In: ICLR (2018), https://openreview.net/forum?id=r1gs9JgRZ, accessed 30 June 2026
10. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: ICML. pp. 1310–1318 (2013), https://proceedings.mlr.press/v28/pascanu13.html, accessed 30 June 2026
11. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: ICCV. pp. 11975–11986 (2023). https://doi.org/10.1109/ICCV51070.2023.01100, https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html, accessed 30 June 2026