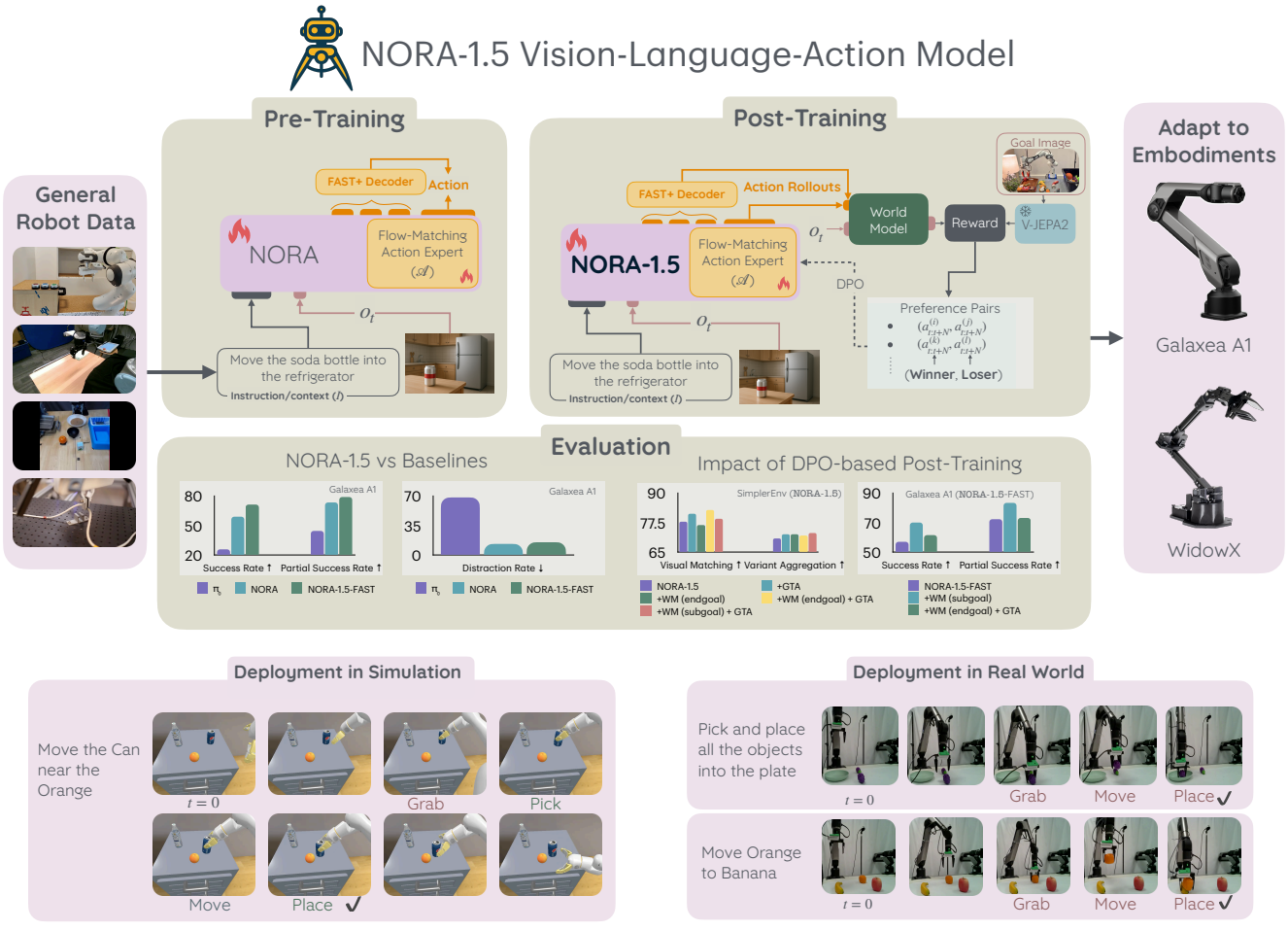
NORA-1.5: A Vision-Language-Action Model Trained using World Model- and Action-based Preference Rewards

Chia-Yu Hung[1] Navonil Majumder[1] Haoyuan Deng[1] Liu Renhang[1] Yankang Ang[1]
Amir Zadeh[2] Chuan Li[2] Dorien Herremans[3] Ziwei Wang[1] Soujanya Poria[1]

[1]Nanyang Technological University
[2]Lambda Labs
[3]Singapore University of Technology and Design



Project Page: https://declare-lab.github.io/nora-1.5
Code: https://github.com/declare-lab/nora-1.5

Abstract

Vision–language–action (VLA) models have recently shown promising performance on a variety of embodied tasks, yet they still fall short in reliability and generalization, especially when deployed across different embodiments or real-world environments. In this work, we introduce `NORA-1.5`*, a VLA model built from the pre-trained NORA backbone by adding to it a flow-matching–based action expert. This architectural enhancement alone yields substantial performance gains, enabling* `NORA-1.5` *to outperform NORA and several state-of-the-art VLA models across both simulated and real-world benchmarks. To further improve robustness and task success, we develop a set of reward models for post-training VLA policies. Our rewards combine (i) an action-conditioned world model (WM) that evaluates whether generated actions lead toward the desired goal, and (ii) a deviation-from–ground-truth heuristic that distinguishes good actions from poor ones. Using these reward signals, we construct preference datasets and adapt* `NORA-1.5` *to target embodiments through direct preference optimization (DPO). Extensive evaluations show that reward-driven post-training consistently improves performance in both simulation and real-robot settings, demonstrating significant VLA model-reliability gains through simple yet effective reward models. Our findings highlight* `NORA-1.5` *and reward-guided post-training as a viable path toward more dependable embodied agents suitable for real-world deployment.*

1. Introduction

Recent advancements in Vision–Language–Action Models (VLAs) have demonstrated remarkable performance across a variety of simple embodied tasks, such as picking and placing objects [5, 16, 19, 35, 43]. Despite this progress, most existing approaches rely heavily on large-scale imitation learning [33] from expert-collected cross-embodiment action trajectories [11], followed by supervised fine-tuning (SFT) on embodiment-specific data for downstream tasks. However, SFT-based adaptation inherits a strong bias from limited manually curated demonstrations, restricting the model's ability to fully generalize or improve beyond the quality of expert data.

To enable more calibrated and scalable post-training, we explore the use of direct preference optimization (DPO) [37] by generating preference datasets from reward models capable of ranking the quality of actions produced by the VLA policy.

We introduce `NORA-1.5`, constructed by coupling a flow-matching–based action expert with the pre-trained autoregressive VLA model NORA [16] through layer-wise self-attention. We choose this architecture due to its promise in achieving impressive performance at a better inference speed as first proposed by Intelligence et al. [18]. While prior work suggested that flow-matching primarily improves inference speed, its impact on policy performance was not investigated. In contrast, we conduct a detailed study and find that flow-matching–based action generation consistently improves performance across multiple benchmarks. We attribute this gain to a strong architectural synergy: the flow-matching expert leverages rich representations encoded by the autoregressive VLA, while the VLA receives informative gradients from the expert, encouraging it to plan coherent multi-step trajectories that the expert can effectively realize. However, we also observe that the flow-matching expert may underperform in low-data settings, likely due to insufficient joint training with the VLA backbone. Overall, `NORA-1.5` achieves state-of-the-art performance on simulated benchmarks such as SimplerEnv and LIBERO, and its capabilities transfer well to real-world robot experiments on a novel embodiment.

Recent robotics approaches [23] have attempted to obtain reward signals by simulating action rollouts. However, such pipelines are computationally expensive, slow to train, and difficult to scale. To address this limitation, we explore reward-driven post-training using lightweight yet effective reward signals derived from compact action-conditioned world models. In this formulation, rewards are estimated by rolling out candidate action sequences through the world model and assessing their ability to reach the goal. Since reward modeling in robotics typically requires estimating how well an action sequence achieves a desired outcome, world models offer a natural mechanism: they directly predict future frames or their latent embeddings conditioned on actions.

Motivated by this, we employ a 1.3B-parameter action-conditioned world model, V-JEPA2-AC [2], as a goal-based reward estimator. Yet because V-JEPA2-AC is adapted with limited data, its predictions can be noisy. To mitigate this, we incorporate a complementary heuristic reward that measures the distance between sampled actions and ground-truth actions in the training data. These two reward components serve distinct roles: the goal-based world model captures diverse feasible trajectories, while the distance-based heuristic helps counteract noise and provides a stable reference. Across benchmarks, we find that combining these lightweight reward formulations with DPO-based preference tuning consistently improves downstream performance. Once the reward mechanism is in place, it can be applied in multiple ways—including preference optimization (as done here) or reinforcement learning—providing a scalable and data-efficient path for post-training VLAs.

This post-training paradigm defines an economical to scale policy refinement of large Vision–Language–Action (VLA) models. Rather than relying on manually-annotated

labels or extensive on-robot rollout execution, NORA-1.5 constructs learned evaluators—a world-model based predictor combined with geometric/heuristic checks—that serve as reward proxies to rank model-generated trajectories to form preference pairs; these ranked pairs are then consumed by Direct Preference Optimization (DPO). This approach has three interlinked advantages. Firstly, it converts policy improvement into a compute-bound process: synthetic rollouts sampled from the VLA can be assessed en masse by the learned evaluator, thereby post-training throughput scales with available compute rather than significantly longer physical robot time. Secondly, since DPO optimizes from pairwise preferences rather than exact likelihoods or calibrated densities, the pipeline is naturally compatible with flow-matching or diffusion-based action heads that lack tractable or well-calibrated likelihoods; thus, preference-based objectives avoid a key optimization bottleneck for contemporary VLA architectures. Thirdly, the learned evaluator provides a unifying rating function that can harmonize heterogeneous corpora: when applied to a corpus such as Open X-Embodiment (OXE), the evaluator may consistently rank trajectories originating from dozens of embodiments, sensors, and task specifications, enabling the entire action-rich structure of OXE to be converted into a massive preference dataset for DPO; the trained world model could be a source of noise this approach. Overall, this enables a single, automated post-training stage that leverages billions of diverse trajectories to produce reward-aligned refinements that generalize across embodiments and deployment conditions—i.e., a scope that meaningfully exceeds mere cross-embodiment adaptation.

In summary, our work makes the following key contributions:

• **Introducing NORA-1.5.** We present NORA-1.5, a VLA model built on a strong pre-trained autoregressive VLA (NORA) by integrating a trainable flow-matching action expert and jointly training them on the Open X-Embodiment dataset. NORA-1.5 significantly outperforms NORA and achieves state-of-the-art results across diverse simulated benchmarks (SimplerEnv, LIBERO) and real-world embodiments (Galaxea A1).

• **Action-rewarding mechanisms through multiple strategies.** We propose a reward framework composed of (i) goal-based rollouts using an action-conditioned world model (V-JEPA2-AC), (ii) distance-based rewards measuring deviation from ground-truth actions, and (iii) subgoal-based scoring. These complementary signals provide robust and scalable criteria for ranking VLA-generated actions and support DPO-based post-training.

• **Comprehensive architectural analysis.** We conduct a detailed investigation of coupling a flow-matching expert with an autoregressive VLA backbone. Our analysis reveals strong mutual benefits: the expert leverages rich VLA encodings, while the VLA improves its trajectory-level planning through feedback from the expert. We also identify data-regime–dependent behaviors.

• **Advancing scalable post-training of VLAs.** We demonstrate that simple reward models combined with DPO-based preference optimization yield consistent performance gains across both simulation and real robots, establishing a scalable and data-efficient direction for post-training VLA models.

2. Preliminaries

2.1. NORA

NORA [15] is a 3B-parameter auto-regressive Vision-Language-Action (VLA) model obtained by fine-tuning a strong vision language model (VLM) backbone Qwen-2.5-VL-3B [3] on the Open X-Embodiment dataset [11] to predict action tokens. Using such a strong VLM backbone imbues NORA with robust world knowledge with multimodal reasoning, representation learning, and instruction-following capabilities, paramount for natural language- and visuals-driven robotic operations. On the other hand, FAST+ tokenizer [31] is used for action representation, owing to its efficient discretization of action sequences and proven efficacy [17] across a wide range of action spaces involving single-arm, bi-manual, and mobile robot tasks.

2.2. V-JEPA-2-AC

V-JEPA-2-AC [1] is based on pre-trained V-JEPA-2—a joint embedding architecture model pre-trained by predicting embeddings of the sequence of visual frames from their masked versions. V-JEPA-2-AC uses V-JEPA2 as vision encoder and adds an additional predictor network on top to predict the future frame embeddings, given current frame(s) and a sequence of actions. This model serves as the action-conditioned world model to post-train the VLA models.

3. NORA-1.5

NORA-1.5 uses NORA [15] as its VLA/VLM backbone due to its solid vision-language and instruction following capabilities derived from its VLM backbone and imitation learning-based training on a large volume and variety of action trajectories (see Sec. 2.1). However, given the performance considerations and efficacy of flow-matching action heads [17], we add a flow-matching-based dedicated action expert that accepts input from the NORA backbone to generate the action sequences directly. On the other hand, to guide the action generation with a world model—V-JEPA-2 [1] in our experiments—, we align the action expert outputs with direct preference optimization (DPO) [37] using the difference between the action-conditioned world model output and the ground-truth frames as a proxy reward. The alignment approach is depicted in Fig. 1.

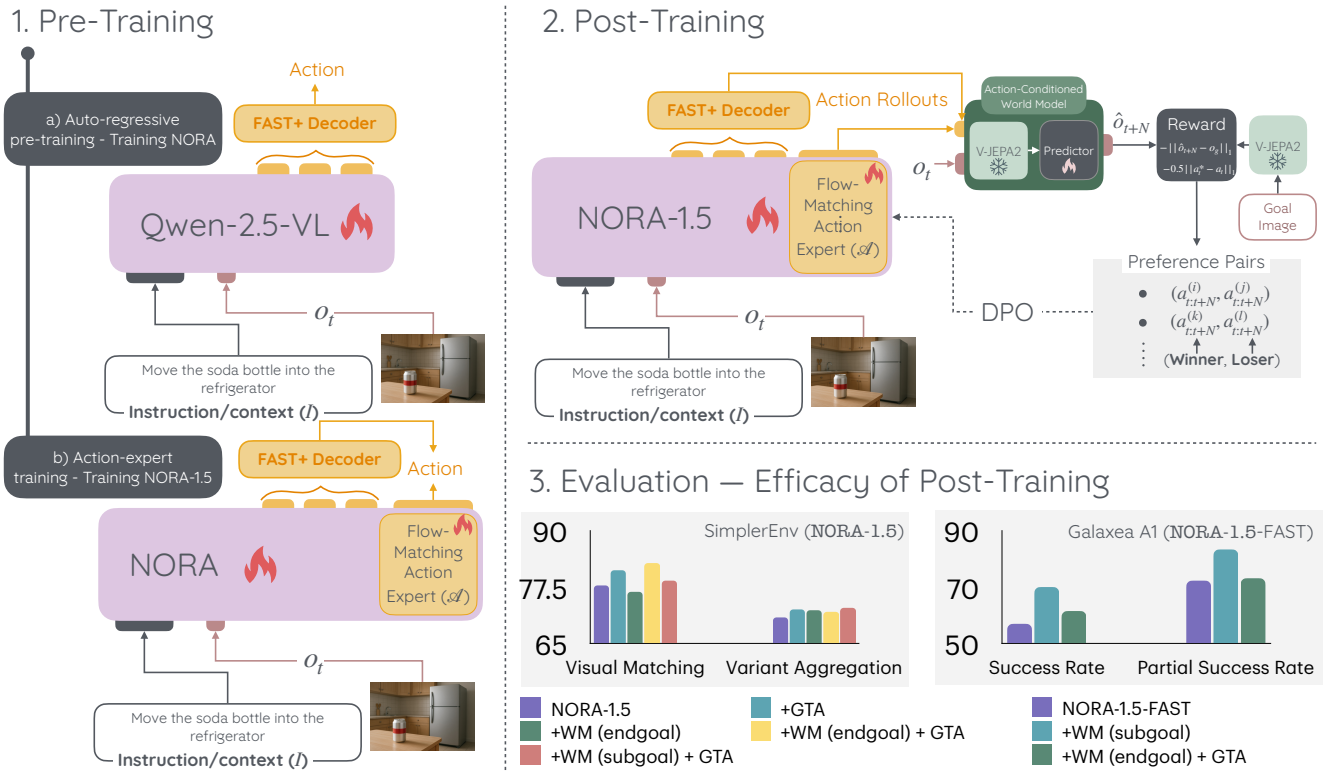


Figure 1. Training pipeline of `NORA-1.5` where firstly a VLA model is pre-trained through imitation learning and subsequently a preference dataset of the actions is created for preference optimization. `WM` stands for WM-guided goal-based reward (Eq. (6)) and `GTA` stands for the reward based on ground-truth action (Eq. (7)).

3.1. Architecture

To circumvent to the often slower auto-regressive action decoding of NORA, we use a separate action expert $\mathcal{A}$ that directly regresses the action sequence $a_{t:t+N}$ within a horizon of length N, based on the joint natural language instruction I and visual observation o_t encoding from NORA ($\mathcal{VL}$):

$$K_{\mathcal{VL},t}, V_{\mathcal{VL},t} = \mathcal{VL}_\theta(o_t, I), \quad (1)$$

$$a_{t:t+N} = \mathcal{A}_\theta(K_{\mathcal{VL},t}, V_{\mathcal{VL},t}), \quad (2)$$

where $K_{\mathcal{VL},t}$ and $V_{\mathcal{VL},t}$ are the keys and values from the transformer layers of $\mathcal{VL}$. In `NORA-1.5`, we use the exact same horizon length as NORA-Long, i.e $N = 5$.

Input Encoding with NORA ($\mathcal{VL}$). Being based on a strong VLM Qwen-2.5-VL-3B allows NORA to have a strong foundation in joint visual-linguistic understanding. Simultaneously, its imitation learning phase on a large volume of diverse trajectories imbued NORA with action generation abilities for a large variety of robots. The latter being an advantage over the typical VLMs makes NORA a good choice for robotics-relevant vision-language encoding to jointly encode natural language instruction and visual observation. To this end, the key-value pairs (see Eq. (1)) of the constituent transformer layers of NORA are used to condition the action expert.

Action Expert ($\mathcal{A}$). The action expert is defined as a flow-matching head that regresses the action sequence of horizon N, conditioned on key-value pairs of $\mathcal{VL}$. Given the action sequence $a_{t:t+N}$, the noisy action sequence is defined as $a^\tau_{t:t+N} = (1-\tau)a_{t:t+N} + \tau a_0$ where τ is the flow matching timestep and $a_0 \sim \mathcal{N}(0, 1)$. The action expert $\mathcal{A}$ directly regresses the ground-truth velocity $v = a_0 - a_{t:t+N}$ against the predicted velocity by minimizing the flow matching loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{v, a^\tau_{t:t+N}} \parallel \mathcal{A}(a^\tau_{t:t+N}, K_{\mathcal{VL},t}, V_{\mathcal{VL},t}) - v \parallel^2 \quad (3)$$

The vector field regressor $\mathcal{A}(a^\tau_{t:t+N}, K_{\mathcal{VL},t}, V_{\mathcal{VL},t})$ is parameterized as a stacked transformer network, architecturally identical to NORA:

$$x^{(l+1)} = Tr^{(l)}(Q = W^{(l)}_Q x^{(l)}, K = K^{(l)}_{\mathcal{VL}} \oplus W^{(l)}_K x^{(l)}, V = V^{(l)}_{\mathcal{VL}} \oplus W^{(l)}_V x^{(l)}), \quad (4)$$

where l is the layer index, $x^{(0)} = a_\tau$, Tr is a transformer layer, and Q, K, and V are the query, key, and value inputs

to the multi-headed attention therein; the head indices are omitted.

3.2. Reward Modeling for Post-training VLAs

In LLM research, significant gains in System-II level intelligence and task performance have been achieved through extensive post-training using reinforcement learning. The key idea is that the model explores the solution space by generating multiple rollouts. A reward model then evaluates these rollouts based on criteria such as task completion, efficiency, and optimality. The reward signals are used to update the policy, enabling the model to gradually improve its action selection and favor strategies that achieve higher rewards. This process effectively combines exploration of possible actions with guided learning from feedback, allowing the model to discover increasingly effective behaviors. Extending this paradigm to Visual-Language-Action (VLA) models faces a fundamental challenge: how can we define and provide reward signals for these models? Training a reward model requires data where each action is evaluated based on its affinity to complete the goal successfully.

A naïve strategy would be to sample N action sequences from a VLA model, execute them either in simulation or on a physical robot, and then construct hand-crafted reward signals based on the observed outcomes. These collected trajectories could then be used to fit reward or value functions capable of evaluating newly generated rollouts and assigning corresponding scores, thereby forming a conventional Reinforcement Learning (RL) pipeline. In practice, however, this approach presupposes access to highly accurate, fast, and embodiment-specific simulators—or alternatively, to substantial real-robot infrastructure—both of which are costly and often infeasible at scale. As a simpler surrogate, one might instead define rewards by measuring the distance between model-generated actions and their corresponding ground-truth actions; however, such heuristics inherit the limitations of the underlying demonstrations. In tasks for which multiple valid trajectories exist, distance-based rewards can bias the learner toward a single demonstration path, thereby creating local optima and discouraging exploration of alternative successful behaviors. Moreover, because these rewards provide no guidance once the policy deviates from the demonstration manifold, they may lead to poor failure recovery and can cause the policy to collapse in off-distribution states encountered during evaluation.

Recent advances in world models and video generative models offer a promising alternative. These models can serve as implicit reward estimators by predicting the consequences of actions and evaluating whether desired subgoals are achieved. Leveraging such learned models as reward functions could enable scalable post-training of VLA policies without the need for fully engineered simulators, providing a practical path forward for reinforcement learning in embodied settings.

Improving the Action Expert through Rewards. Given N rollouts from the action expert, we leverage several techniques to compute rewards as explained in the following section.

Once the reward model is trained, preference optimization techniques, such as Direct Preference Optimization (DPO) [37], RL, and GRPO (Group Reward Preference Optimization) [39], can be adopted for improving the action expert. In our case, we use DPO.

Reward Designs. Our reward model has two components: (i) WM-guided goal-based reward and (ii) action-based reward. *WM-guided goal-based reward* is designed to quantify the alignment of the generated actions to the specified goal. For this, action-conditioned world models can be used to predict the resulting future states. These states can then be compared to the ground-truth goal states—we experimented with both final goal, denoted as `WM` (endgoal) reward, and immediate subgoal states, denoted as `WM` (subgoal) reward (see Sec. 4.5)—using a suitable metric to obtain the reward signal for the action expert. The immediate subgoal states could guide the model toward immediate short-term goals, as opposed to the end-goal state that could guide toward the final long-term goal. Following this hypothesis, to estimate the quality of the actions in terms of achieving the end-goal or subgoal, we train an action conditioned world dynamics model $\mathcal{W}$ that is based on pre-trained V-JEPA2[1]—trained to encode images and sequence of images. Inspired by Assran et al. [1], we train a predictor transformer model (P_θ) that accepts the current observation o_t encoded by V-JEPA2 ($\mathcal{J}$) and an action sequence $a_{t:t+N}$ as input, to regress the embedding of the next observation $\hat{o}_{t+N}$, as defined in Eq. (5).

$$\mathcal{J}(o_{t+N}) = \mathcal{W}_\theta(o_t, a_{t:t+N}) \coloneqq \mathcal{P}_\theta(\mathcal{J}(o_t), a_{t:t+N}), \quad (5)$$

$$\mathcal{R}_g(a_{t:t+N}, o_t) \coloneqq -||\mathcal{J}(o_g) - \mathcal{W}_\theta(o_t, a_{t:t+N})||_1, \quad (6)$$

$$g \in \{\text{endgoal}, \text{subgoal-}t\},$$

$$\mathcal{R}_a(a_{t:t+N}) \coloneqq -||a^*_{t:t+N} - a_{t:t+N}||_1, \quad (7)$$

$$\mathcal{R}_{\text{tot}}(a_{t:t+N}, o_t) \coloneqq \mathcal{R}_g(a_{t:t+N}, o_t) + 0.5\mathcal{R}_a(a_{t:t+N}) \quad (8)$$

The WM-guided goal-based reward, as defined in Eq. (6), is the difference between the final goal image o_{endgoal} or the immediate subgoal image $o_{\text{subgoal-}t}$ and the world model-estimated resultant image of the candidate action $a_{t:t+N}$. This difference could indicate how close an action a_t is to take the task to the end-goal or the immediate subgoal. The ground-truth subgoal image $o_{\text{subgoal-}t}$ at time t is chosen as the $t+N$-th available frame o_{t+N}.

[1] https://dl.fbaipublicfiles.com/vjepa2/vitg.pt

On the other hand, *action-based reward* [20] (referred to as `GTA` in the experimental results), as defined in Eq. (7), quantifies how close an action $a_{t:t+N}$ is to the gold action $a^*_{t:t+N}$. The total reward $\mathcal{R}_{\text{tot}}$ combines these two components, where the action-based reward is given half as much weight as the WM-guided goal-based reward. This combination could mitigate the noisiness of the WM-guided goal-based reward inherited from the action-conditioned world model $\mathcal{W}$ that is trained on limited data and may not generalize well to all scenarios. On the other hand, action-based reward can be too constrained, as the ground-truth trajectory may not be unique, and in such cases, goal-driven reward may work well.

The reward model used in this work provides dense, stepwise evaluations which permit the model to rank sampled candidate actions at each timestep. Concretely, given a fixed task specification and observation s_t, the model assigns comparative scores to different candidate actions $\{a^{(1)}_{t:t+N}, \ldots, a^{(N)}_{t:t+N}\}$, enabling the VLA to discriminate the relative quality of these actions and thereby encouraging deeper step-level exploration during Direct Preference Optimization (DPO). Because the ranking is performed at the action level, the policy can explore diverse local decision branches and propagate preference information that is localized in time. This reward model can also be integrated directly into conventional RL objectives (e.g., as per-step rewards r_t or as an auxiliary critic), enabling hybrid training regimes.

By contrast, an alternative is to collect data where we use sparse per-step rewards derived from the final trajectory outcome, use that to train a value function, and finally to perform RL through learned value functions; while repeated trajectory-level rollouts also promote exploration, they generally yield shallower exploration because credit is assigned over whole trajectories rather than to individual time steps.

Preference Dataset Construction. We construct preference datasets $\mathcal{D}_{\text{goal}}$ and $\mathcal{D}_{\text{act}}$ of (`winner`, `loser`) action preference pairs $(a^W_{t:t+N}, a^L_{t:t+N})$ based on rewards defined in Eqs. (7) and (8), respectively, where $a^{W,L}_{t:t+N} \sim \text{VLA}_\theta(o_t, I)$, $\text{VLA} := \mathcal{A}_\theta \circ \mathcal{VL}_\theta$, and $\mathcal{R}(a^W_{t:t+N}, \cdot) > \mathcal{R}(a^L_{t:t+N}, \cdot)$. Given the current state, instruction, and these pairs, we use Eq. (6), Eq. (7), and Eq. (8) to rank the actions given an observation and construct the preference pairs accordingly.

3.3. Training

There are two major training stages:

i. **Action-Expert Training.** The action expert parameters are randomly initialized and subsequently jointly trained with the VLA-backbone (NORA) parameters with a combined flow-matching loss on the action expert output and cross-entropy loss on the FAST+ output tokens of NORA.

ii. **Reward-guided Post-Training.** We align the action expert-generated action sequences with DPO objective

$$
\begin{aligned}
L_{\text{DPO-FM}} = -\,\mathbb{E}_{\tau\sim\mathcal{U}(0,1),(a^W_{t:t+N},a^L_{t:t+N},o_t,I)\sim\mathcal{D}.} \\
\log\sigma\Big(-\beta\Big[\underbrace{\|\mathcal{A}(a^W_{t:t+N},o_t,I,\tau;\theta)-v^W_\tau\|_2^2}_{\text{Winning loss}} \\
-\underbrace{\|\mathcal{A}(a^L_{t:t+N},o_t,I,\tau;\theta)-v^L_\tau\|_2^2}_{\text{Losing loss}} \\
-\underbrace{\|\mathcal{A}(a^W_{t:t+N},o_t,I,\tau;\theta_{\text{r}})-v^W_\tau\|_2^2}_{\text{Winning reference loss}} \\
+\underbrace{\|\mathcal{A}(a^L_{t:t+N},o_t,I,\tau;\theta_{\text{r}})-v^L_\tau\|_2^2}_{\text{Losing reference loss}}\Big]\Big). \qquad (9)
\end{aligned}
$$

On the other hand, we also align the FAST+ action outputs from the VLA decoder head with the DPO objective by Rafailov et al. [37]. The evaluations of FAST+ outputs are indicated with a '-FAST' suffix. The DPO-based post-training is applied to the SFT models i.e., after fine-tuning the VLA on target embodiment's supervised data.

4. Experiments

4.1. Baselines

As baselines, we use existing well known VLA models including autoregressive VLAs such as SpatialVLA [35], RT-1 [7], MolmoAct [21], Emma-X [41], NORA [15], and OpenVLA [19] and diffusion or flow-matching based such as π_0 [5] etc.

4.2. Benchmarks and Evaluation Settings

The VLA models are evaluated across both simulated and real-world settings using LIBERO, SimplerEnv, and a Galaxea A1 robotic arm. The LIBERO benchmark comprises four subsets—Spatial, Object, Goal, and Long—each evaluated over 500 episodes, with results averaged across three runs using different seeds; fine-tuning is performed by combining data from all four subsets after removing no-op actions. SimplerEnv focuses on closing the simulation–reality gap with optimized PD parameters and evaluates four tasks (pick coke can, move object near object, open drawer, close drawer) under two protocols: visual matching and variant aggregation, covering over 1,000 episodes in total; results are averaged over two runs. For real-world cross-embodiment evaluation, we use the Galaxea A1 robot—absent from the pretraining dataset—and collect 1,000 teleoperated pick-and-place episodes with randomized object placement across nine unique tasks (e.g., "Put apple on the plate"). Evaluation is conducted on nine tasks grouped into three categories (seen tasks, unseen-object–seen-distractor tasks, and unseen-instruction–seen-distractor tasks), each

repeated for 10 trials using fixed starting positions consistent across baselines. Simulation benchmarks report binary success rates (1 if the task is completed, else 0), while real-robot evaluations report both success rate (*Succ.*↑) and partial success rate (*Part. Succ.*↑) to capture finer-grained differences in performance i.e., if the robot successfully grasps the correct object, we reward it with one point. Additionally, we also report the occurrences of grasping the distractors (Dist.↓) in the environment where a lower score is preferred.

4.3. Performance of NORA-1.5

As evident in Tabs. 1 to 3, NORA-1.5 generally outperforms all the baselines as analyzed below.

SimplerEnv. The results in Tab. 1 clearly show a superiority of NORA-1.5 in visual matching evaluation. Particularly on *pick coke can* and *move near* tasks, zero-shot NORA-1.5 outperform all the baseline zero-shot models by a wide margin of 4.6% and 10.7%, respectively. The performance advantage on these two tasks still holds for the fine-tuned variant of NORA-1.5 by 6.8% and 0.8%, respectively, against fine-tuned SpatialVLA. However, for *open/close drawer* task this performance gain is not present. Magma far outperforms all the models in this regard, but its overall performance is far below even zero-shot NORA-1.5. This could be attributed to its limited adaptation to such dragging and pushing actions that are relatively less prevalent in the pre-training dataset than pick and place actions—based on a keyword search on the task descriptions of Open X Embodiment pre-training dataset. However, the performance of fine-tuned NORA-1.5 on this task is still better by 4.8% than the generally next-most capable fine-tuned model of SpatialVLA. For visual matching, overall zero-shot and fine-tuned variants of NORA-1.5 surpass the other equivalent next-best models by a wide 6.4% and 4.2%, respectively.

For variant aggregation setting, post-DPO NORA-1.5 performs comparably to the best model MolmoAct. However, the minute performance advantage of MolmoAct comes from a performance large performance advantage on *drawer open/close* tasks. Whereas, it has huge underperformance on *pick coke* and *move next* tasks. Thus, NORA-1.5 could be considered more robust across tasks and variable visual settings.

NORA vs. NORA-1.5. We observe that NORA-1.5 consistently outperforms NORA across all benchmarks. According to Intelligence et al. [17], flow matching in $\pi_{0.5}$ was primarily introduced to improve inference speed rather than performance. In contrast, in NORA-1.5, coupling a flow-matching-based action expert with a pre-trained VLM-based autoregressive VLA leads to noticeable performance

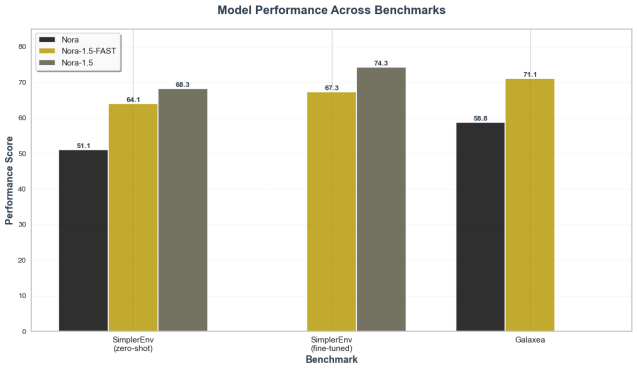


Figure 2. Comparing FAST+ with flow-matching.

gains over the latter. We attribute this to the architectural design, where the flow-matching-based action expert and the autoregressive pre-trained VLA mutually benefit from each other. The flow-matching expert leverages the VLA's rich representations—such as encoded observations, instructions, and overall plans—required for generating coherent actions. At the same time, the autoregressive VLA benefits from gradient feedback propagated through the action expert, enabling more effective learning such as improving the overall abstract plan for action expert to generate actions. In this way, the VLA is encouraged to plan the entire action trajectory that the expert subsequently leverages to generate actions. As shown in Fig. 2, NORA-1.5-FAST further surpasses NORA in both zero-shot and fine-tuned evaluations.

LIBERO. Tab. 2 shows that the performance gain by DPO over the NORA-1.5 baselines is generally quite consistent and wide, except on LIBERO-Object evaluation tasks. This could be attributed to the limited variability of the object dimensions in these tasks, while the spatial setting and goal remain fixed. This lack of variability, as compared to the other tasks that differ in goals and spatial relationships, may have made these tasks easier. Thus, significant improvements could be harder to achieve. In fact, all the remaining models are not far behind the top models on these tasks, as compared to the remaining tasks. Overall, NORA-1.5 outperforms recent state-of-the-art models such as π_0.

Performance with Limited Real-life Robot Training Data. Tab. 3 shows the performance of π_0, NORA, and NORA-1.5 on the Galaxea A1 robotic arm. Across all experimental settings, NORA-1.5 outperforms these strong baselines by 13% to 46%. The improvement is larger for tasks with unseen distractors, suggesting the robustness of NORA-1.5. Although we jointly optimize the flow-matching and autoregressive losses when fine-tuning the

Table 1. Performance comparison across models on SimplerEnv evaluation. The baseline results are taken from Lee et al. [21]. VM:=Visual Matching, VA:=Variant Aggregation, PCC:=Pick Coke Can, MN:=Move Near, DR:=Open/Close Drawer.

Model	Visual Matching				Variant Aggregation			
	Pick Coke Can	Move Near	Open/Close Drawer	Avg	Pick Coke Can	Move Near	Open/Close Drawer	Avg
HPT	56.0%	60.0%	24.0%	46.0%	—	—	—	—
TraceVLA	28.0%	53.7%	57.0%	42.0%	60.0%	56.4%	31.0%	45.0%
RT-1-X	56.7%	31.7%	59.7%	53.4%	49.0%	32.3%	29.4%	39.6%
RT-2-X	78.7%	77.9%	25.0%	60.7%	82.3%	79.2%	35.3%	64.3%
Octo-Base	17.0%	4.2%	22.7%	16.8%	0.6%	3.1%	1.1%	1.1%
OpenVLA	16.3%	46.2%	35.6%	27.7%	54.5%	47.7%	17.7%	39.8%
RoboVLM (zero-shot)	72.7%	66.3%	26.8%	56.3%	68.3%	56.0%	8.5%	46.3%
RoboVLM (fine-tuned)	77.3%	61.7%	43.5%	63.4%	75.6%	60.0%	10.6%	51.3%
Emma-X	2.3%	3.3%	18.3%	8.0%	5.3%	7.3%	20.5%	11.0%
Magma	56.0%	65.4%	**83.7%**	68.4%	53.4%	65.7%	68.8%	62.6%
π_0 (fine-tuned)	72.7%	65.3%	38.3%	58.7%	75.2%	63.7%	25.6%	54.8%
π_0-FAST (fine-tuned)	75.3%	67.5%	42.9%	61.9%	77.6%	68.2%	31.3%	59.0%
GR00T N1.5 (fine-tuned)	69.3%	68.7%	35.8%	52.4%	46.7%	62.9%	17.5%	43.7%
SpatialVLA (zero-shot)	81.0%	69.6%	59.3%	70.0%	89.5%	71.7%	36.2%	65.8%
SpatialVLA (fine-tuned)	86.0%	77.9%	57.4%	73.7%	88.0%	72.7%	41.8%	67.5%
MolmoAct (zero-shot)	71.3%	73.8%	66.5%	70.5%	57.8%	43.8%	76.7%	59.3%
MolmoAct (fine-tuned)	77.7%	77.1%	60.0%	71.6%	76.1%	61.3%	**78.8%**	**72.1%**
NORA-Long (zero-shot)	74.2%	75.0%	31.7%	60.3%	36.0%	73.0%	16.9%	42.0%
`NORA-1.5`-FAST (zero-shot)	79.5%	**90.9%**	51.5%	74.0%	67.3%	71.6%	24.0%	54.3%
`NORA-1.5`-FAST (fine-tuned)	88.6%	86.4%	41.2%	72.1%	85.2%	**85.2%**	31.7%	67.4%
`NORA-1.5` (zero-shot)	85.6%	88.6%	56.7%	76.9%	73.0%	80.1%	26.2%	59.7%
`NORA-1.5` (fine-tuned)	92.8%	78.7%	62.2%	77.9%	**95.0%**	75.7%	41.5%	70.7%
`NORA-1.5` (DPO)	**94.0%**	88.0%	66.4%	**82.8%**	92.6%	79.0%	44.1%	71.9%
Δ from DPO	1.2%	9.3%	4.2%	4.9%	-2.4%	3.3%	2.6%	1.2%

Table 2. Comparison of different baselines on spatial, object, goal, and long-horizon evaluation in LIBERO. The baseline results are taken from Lee et al. [21]. Each subtask is evaluated across three random seed.

Baseline	Spatial	Object	Goal	Long	Avg
TraceVLA	84.6%	85.2%	75.1%	54.1%	74.8%
Octo-Base	78.9%	85.7%	84.6%	51.1%	75.1%
OpenVLA	84.7%	88.4%	79.2%	53.7%	76.5%
SpatialVLA	88.2%	89.9%	78.6%	55.5%	78.1%
CoT-VLA	87.5%	91.6%	87.6%	69.0%	83.9%
WorldVLA	87.6%	96.2%	83.4%	60.0%	79.1%
π_0-FAST	96.4%	96.8%	88.6%	60.2%	85.5%
π_0	96.8%	**98.8%**	**95.8%**	85.2%	94.2%
ThinkAct	88.3%	91.4%	87.1%	70.9%	84.4%
MolmoAct-7B-D	87.0%	95.4%	87.6%	77.2%	86.6%
NORA	85.6%	89.4%	80.0%	63.0%	79.5%
NORA-Long	92.2%	95.4%	89.4%	74.6%	87.9%
`NORA-1.5`	97.3%	96.4%	94.5%	89.6%	94.5%
`NORA-1.5` (DPO)	**98.0%**	96.0%	95.4%	**90.5%**	**95.0%**
Δ from DPO	0.7%	-0.4%	0.9%	1.0%	0.6%

model on Galaxea A1 data, we observe that flow-matching-based action generation performs worse than autoregressive decoding. This differs from our observations of SimplerEnv and LIBERO (Fig. 2), where flow-matching-based generation performs substantially better. We believe this difference arises from the smaller real-robot dataset (50K frames) as compared to SimplerEnv (4M frames). Since unlike π_0 our action expert lacks extensive flow-matching pre-training, it would likely require more data to effectively adapt than the autoregressive VLM backbone. This could explain the performance advantage of flow-matching-based generation on SimplerEnv and LIBERO, where larger fine-tuning datasets are available.

4.4. Impact of DPO-based Post-training

We study the impact of three reward formulations: an action-based reward (Eq. (7)), WM-guided goal-based reward (Eq. (6)), and a linear combination of these two (Eq. (8)). We perform these experiments primarily on the SimplerEnv and LIBERO simulation tasks.

SimplerEnv. The results in Tab. 4 suggest that different reward strategies utilized for DPO generally outperform the SFT baseline for both Visual Matching and Variant Aggregation. The WM-guided goal-based reward proved to be an exception, leading to a performance degradation on the "Move Near" task for both visual matching and variant aggregation. This is likely because a purely goal-directed guidance does not account for the implicit safety constraints of the task. Based on the success criteria of "Move Near" in SimplerEnv, the robot is required to avoid obstacles while completing the task. Notably, in Tab. 4, the hybrid `WM + GTA` model achieves the best overall performance of 82.8% on Visual matching and a significant increase on the "Move Near" subtask, suggesting that by adding ground truth action to the goal-based reward,

Table 3. Experimental results of NORA-1.5 and baselines on nine real-world tasks with Galaxea A1 robotic arm. **Task Format** indicates the types of the physical objects—seen (S) vs unseen (U)—and how they are related in the task. The red-inked objects are distractors (Dist.) in the setup. Succ. := % success rate$_\tau$

Task Format Target(s) [Distractor]	**Task** (τ)	π_0 (3.3B)			NORA (3B)			NORA-1.5-FAST (3.3B)		
		Part. Succ.↑	Dist. ↓	Succ. ↑	Part. Succ.↑	Dist. ↓	Succ. ↑	Part. Succ.↑	Dist. ↓	Succ. ↑
Put U in U	τ_1 : Put eggplant in bowl	90%	-	80%	90%	-	90%	100%	-	**100%**
	τ_2 : Put apple in plate	70%	-	30%	100%	-	80%	100%	-	**90%**
	τ_3 : Put mango in basket	90%	-	80%	80%	-	70%	90%	-	**90%**
Put U in S [S]	τ_4 : Put strawberry in plate [apple]	0%	90%	0%	70%	0%	**70%**	70%	10%	**70%**
	τ_5 : Put grape in plate [eggplant]	0%	90%	0%	70%	20%	50%	80%	20%	**80%**
	τ_6 : Put orange in plate [banana]	0%	100%	0%	30%	20%	40%	70%	30%	**60%**
Move U to U [S]	τ_7 : Move strawberry to banana [apple]	50%	50%	10%	60%	20%	20%	60%	10%	**40%**
	τ_8 : Move orange to banana [apple]	50%	30%	10%	80%	0%	50%	70%	20%	**60%**
	τ_9 : Move cube to orange [banana]	50%	50%	20%	80%	20%	60%	70%	0%	50%
	Average	44.44%	68.33%	25.55%	73.3%	**13.3%**	58.88%	**78.88%**	15.00%	**71.11%**

Table 4. Ablation study on the proxy reward for DPO of NORA-1.5 through SimplerEnv evaluation. WM stands for world model-guided goal-based reward (Eq. (6)) and GTA stands for the reward based on ground-truth action (Eq. (7)). VM:=Visual Matching, VA:=Variant Aggregation, PCC:=Pick Coke Can, MN:=Move Near, DR:=Open/Close Drawer.

Reward	**VM**				**VA**			
	PCC	MN	DR	Avg	PCC	MN	DR	Avg
SFT (no reward)	92.8%	78.7%	62.2%	77.9%	95.0%	75.7%	41.5%	70.7%
Reward Techniques for DPO								
WM (endgoal)	93.6%	73.9%	61.6%	76.4%	95.1%	74.6%	**47.7%**	72.5%
WM (subgoal)	**95.5%**	81.8%	58.8%	78.7%	**95.3%**	**80.7%**	40.5%	72.2%
GTA	92.8%	86.0%	64.4%	81.2%	94.6%	80.1%	42.9 %	72.5%
WM (endgoal) + GTA	94.0%	**88.0%**	**66.4%**	**82.8%**	92.6%	79.0%	44.1%	71.9%
WM (subgoal) + GTA	92.4%	83.0%	61.6%	79.0%	93.4%	80.1%	45.4%	**73.0%**

LIBERO. The performance on LIBERO also improves with DPO, although the gains are smaller as compared to SimplerEnv. This is likely because the SFT model already performs strongly on LIBERO, leaving limited room for further improvement. Among the LIBERO benchmarks, the most challenging task is LIBERO-Long, where we observe consistent performance gains of 1% to 1.7% across all reward modeling techniques.

Galaxea A1 Robotic Arm. After observing marginal to substantial improvements from our action-rewarding strategies and DPO across simulation benchmarks such as LIBERO and SimplerEnv, we next evaluate whether these gains carry over to real-robot settings. Real-world experiments also allow us to understand more concretely how DPO-based post-training benefits VLA models. To this end, we extend the "Put U in S [S]" task suite by adding four more variants with additional distractors to increase task difficulty.

As shown in Tab. 5, across all thirteen tasks, the DPO-trained NORA-1.5 achieves a notable 13% performance improvement. Specifically, correct-object grasping accuracy increases by 11%, while unintended grasps of distractors decrease by 4%. While DPO marginally improves the performance for in-domain or seen objects/tasks, we see a larger performance improvement of 15%-16% on unseen tasks and objects. These trends indicate that DPO contributes primarily in two ways: (1) enhancing the affordance and reliability of the grasping action, and (2) improving the model's ability to focus on the intended target object.

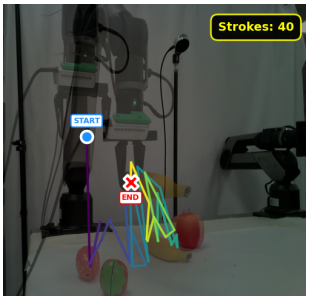
Fig. 3 further illustrates these effects. With reward-driven DPO post-training, the gripper trajectory becomes more consistent and well-formed, whereas the non-DPO model exhibits more fixations and zig-zag motions. Consequently, while NORA-1.5 with DPO requires only 7.0 action chunks on average to grasp the target, NORA-1.5 without DPO takes 9.7 action chunks, indicating that the latter struggles to execute smooth and efficient grasps—an issue mitigated by DPO post-training. The results also show that DPO helps reduce the likelihood of the robot accidentally picking up distractor objects.

4.5. Ablations

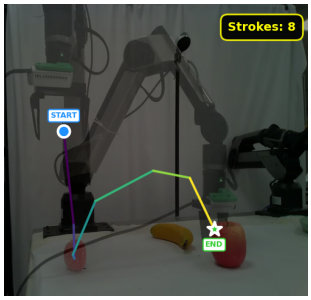
To determine the individual contribution of the various reward elements—WM-guided goal-based reward (Eq. (6)) and ground-truth action-based reward (Eq. (7))—, we also curate preference datasets for SimplerEnv and LIBERO with the individual elementary rewards and apply DPO to the pre-trained NORA-1.5. Evaluating the elementary rewards in SimplerEnv, as shown in Tab. 4, reveals a general superiority of the combined reward (Eq. (8)) for visual matching. Interestingly, the end-goal-based reward WM (endgoal) causes the performance on *move near* and *drawer open/close* tasks to drop below SFT. On the other hand, the subgoal-based reward WM (subgoal) is overall 1.7% better than WM (endgoal) and overall beats the SFT baseline by a small margin. Specifically, WM (subgoal) surpasses the SFT baseline on *move near* task by 3.1%, whereas WM (endgoal)

Table 5. Comparing NORA-1.5-FAST w/ and w/o DPO. Metrics shown separately as Correct/Dist./Succ. The newly introduced tasks are : τ_{10} : Put mango in plate [apple, grape], τ_{11} : Put mango in plate [apple, grape, orange], τ_{12} : Put orange in plate [apple, grape], τ_{13} : Put orange in plate [apple, grape, mango]. Remaining tasks can be found in Tab. 3.

Format		Reward Method for DPO											
Target(s) [Dist.]	Task (τ)	NORA-1.5-FAST			w/ WM (subgoal)+GTA			w/ WM (endgoal)+GTA			w/ GTA		
		Part. Succ.↑	Dist.↓	Succ.↑	Part. Succ.↑	Dist.↓	Succ.↑	Part. Succ.↑	Dist.↓	Succ.↑	Part. Succ.↑	Dist.↓	Succ.↑
Put U in U	τ_1	100	-	100	100	-	100	90	-	90	100	-	100
	τ_2	100	-	90	100	-	100	100	-	100	90	-	90
	τ_3	90	-	90	90	-	90	90	-	90	90	-	90
Put U in S [S]	τ_4	70	10	70	70	0	70	70	0	70	70	0	70
	τ_5	80	20	80	80	10	80	90	0	70	80	10	60
	τ_6	70	30	60	90	0	80	90	0	80	70	0	70
	τ_{10}	80	10	70	90	10	90	70	10	70	90	0	70
	τ_{11}	40	20	10	50	50	30	40	40	40	50	40	10
	τ_{12}	60	10	10	80	10	40	50	0	10	40	30	0
	τ_{13}	50	20	10	80	20	30	30	20	10	50	0	0
Move U to U [S]	τ_7	60	0	40	80	10	70	80	10	60	80	10	70
	τ_8	70	20	50	100	0	70	90	0	60	70	20	60
	τ_9	70	20	60	80	10	60	60	20	50	80	20	60
Average		72.30	16.00	56.92	**83.84**	12.00	**70.00**	73.07	**10.00**	61.53	73.84	13	57.69
(Improvement)					11.54	4.00	13.08	0.77	6.00	4.61	1.54	3.00	0.77
Average (Unseen)		65.00	16.00	46.00	**80.00**	12.00	**62.00**	67.00	**10.00**	**52.00**	68.00	13.00	47.00
(Improvement)					15.00	4.00	16.08	2.00	6.00	6.00	3.00	3.00	1.00



(a) NORA-1.5 without DPO. The gripper trajectory exhibits frequent fixations and zig-zag motions, often resulting in failed grasps and grasp attempts toward distractor objects.



(b) NORA-1.5 with reward-driven DPO post-training. The gripper trajectory becomes smoother and more consistent, with fewer corrective strokes and more reliable target grasps.

Figure 3. Effect of DPO post-training on real-robot gripper trajectories for the Galaxea A1 arm. Compared to the non-DPO baseline (a), the DPO-trained NORA-1.5 (b) executes smoother trajectories with fewer strokes, aligning with the reduced number of action chunks and improved grasp success reported in Tab. 5.

lags behind by 4.8%. This could be indicative of the noisiness of the signal from the world model, where the guidance of the final goal image is noisier than the immediate subgoal images due to shaky long-term dependency modeling.

The ground-truth action-based reward (GTA) is generally superior to all other elementary rewards for visual matching. This reward might teach the model to follow the most straightforward trajectory to achieve the goal, achieving superior results. However, for *pick coke can* task, this reward fails to surpass the SFT baseline and falls behind the other two elementary goal-based rewards. This may indicate the drawback of such a straightforward approach, which may induce certain biases in the model that may not work out in very specific cases.

Table 6. Ablation study on the proxy reward for DPO of NORA-1.5 through spatial, object, goal, and long-horizon evaluation in LIBERO. WM stands for world model-guided goal-based reward (Eq. (6)) and GTA stands for the reward based on ground-truth action (Eq. (7)).

Reward	Spatial	Object	Goal	Long	Avg
SFT (no reward)	97.3%	**96.4%**	94.5%	89.6%	94.5%
Reward Techniques for DPO					
WM (endgoal)	98.0%	96.0%	**95.4%**	90.5%	**95.0%**
GTA	**98.3%**	95.9%	94.7%	90.7%	94.9%
WM (endgoal) + GTA	97.9%	95.9%	94.1%	**91.3%**	94.8%

For the evaluation with variant aggregation, the overall performance of all the elementary rewards are in the same ballpark. WM (subgoal) beats the other two elementary rewards on *pick coke can* and *move next* tasks by a small margin. In fact, all these elementary rewards beat the combined reward WM (endgoal) + GTA across all the tasks, but the overall performance is very comparable. Interestingly,

the WM (subgoal) + GTA combination shows the most stable performance under this setting by outperforming all on average despite not being the best at any individual task. This may underscore the robustness of short-term subgoal modeling to the changing visuals.

For real-world Galaxea A1-based experiments (Tab. 5), both the WM (subgoal)+GTA and WM (endgoal)+GTA reward methods improve performance over the SFT model. However, WM (subgoal)+GTA reward consistently yields stronger gains across all metrics—including grasping the correct object, avoiding distractors, and overall task completion. A plausible explanation is the presence of unseen objects and varying environmental conditions in the real-robot setup. In such settings, localized guidance from subgoal rewards may offer a more reliable and less noisy training signal than endgoal-based rewards. We also observe that GTA rewards offer limited benefits for real robots. In fact, for most tasks in the "Put U in S" category—including all newly introduced challenging tasks—its performance is worse than the baseline. This reinforces our assumption that, in real-world environments, multiple trajectories can successfully accomplish the same task. As a result, forcing the model to consistently follow a single labeled trajectory may introduce unnecessary noise into the robot's behavior when operating in unseen scenarios. By combining subgoal/goal information with GTA to construct the training dataset, we provide additional contextual signals that help guide the robot toward selecting an appropriate trajectory to complete the task.

For LIBERO, Tab. 6 shows that the overall performance gain over SFT by both elementary and combined reward is quite minute. This could be ascribed to the diminished gain potential for LIBERO due to already high performance of the SFT baseline.

5. Conclusion

Increasing research and commercial interest in VLA models call for effective adaptation of these models to a wide-range of embodiments/robots. This work shows that preference optimization-based post-training improves adaptation of NORA-1.5 for both real world and simulation, given appropriate reward modeling. The experiments and analyses substantiate that our world model-driven goal- and action-based reward is quite potent proxy reward for DPO of our NORA-1.5, resulting in significant performance gains. Our real-world evaluation also highlight the performance advantage of NORA-1.5 over the state-of-the-art open VLA model π_0. We hope this paper provides a sturdy foundation for the future research on post-training VLA models and for embodied AI as a whole.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. 3, 5

[2] Mido Assran, Adrien Bardes, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint*, 2025. 2

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 3

[4] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 3

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π0: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*, 2024. 2, 6

[6] Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 3

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022. 6

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta,

Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. 3

[9] Anthony Brohan, Noah Brown, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint*, 2023. 1

[10] Jun Cen, Chaohui Yu, Hangjie Yuan, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. 4

[11] Open X-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. https://arxiv.org/abs/2310.08864, 2023. 2, 3, 1

[12] Open X-Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 1

[13] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025. 2

[14] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, et al. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025. 4

[15] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks, 2025. 3, 6

[16] Chia-Yu Hung, Qi Sun, Pengfei Hong, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025. 2, 4

[17] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian

Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. 3, 7

[18] Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint*, 2025. 2

[19] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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. In *8th Annual Conference on Robot Learning*, 2024. 2, 6, 1, 3

[20] Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025. 6, 2

[21] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space, 2025. 6, 8

[22] Jason Lee, Jiafei Duan, Haoquan Fang, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025. 2, 3

[23] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 2

[24] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 1

[25] Xuanlin Li, Kyle Hsu, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 2

[26] Xinghang Li, Peiyan Li, Minghuan Liu, et al. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 3

[27] Bo Liu, Yifeng Zhu, et al. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. 1, 2

[28] NVIDIA Isaac Robotics Team. Gr00t n1.5: An upgraded foundation model for humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_5/, 2025. 3

[29] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 3

[30] 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. 3

[31] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. 3

[32] Karl Pertsch, Kyle Stachowicz, et al. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint*, 2025. 2

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 2

[34] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, Maoqing Yao, Haoran Yang, Jiacheng Bao, Bin Zhao, and Dong Wang. Eo-1: Interleaved vision-text-action pretraining for general robot control, 2025. 2

[35] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 2, 6, 3

[36] Delin Qu, Haoming Song, Qizhi Chen, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 2

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. 2, 3, 5, 6

[38] Rafael Rafailov, Archit Sharma, et al. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. 2

[39] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. 5

[40] Zhihong Shao, Peiyi Wang, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint*, 2024. 2

[41] Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning, 2024. 6

[42] Qi Sun, Pengfei Hong, et al. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv preprint*, 2024. 2

[43] Qi Sun, Pengfei Hong, Tej Deep Pala, et al. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv preprint arXiv:2412.11974*, 2025. 2, 3

[44] 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. 3

[45] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in Neural Information Processing Systems*, 2024. 3

[46] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. In *NeurIPS*, 2024. 2

[47] Jianwei Yang, Junjie Cao, Xiyang Chen, et al. Magma: A foundation model for multimodal ai agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3

[48] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952. IEEE, 2023. 3

[49] Qingqing Zhao, Yao Lu, Moo Jin Kim, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025. 3

[50] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 3

[51] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, et al. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 2

[52] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025. 2

[53] Gaoyue Zhou, Hengkai Pan, et al. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint*, 2025. 2

A. Evaluation Settings and Metrics

The comparative evaluation of our VLA model is performed under both real-world and simulated settings. On one hand, Galaxea A1 robotic arm is chosen as the embodiment for the real-world evaluation. On the other hand, LIBERO [27] and SimplerEnv [24] simulated benchmarks are used to evaluate the VLA models under a diverse range of settings.

LIBERO [27] simulated benchmark comprising four subsets to test generalization across spatial layouts (LIBERO-Spatial), objects (LIBERO-Object), task goals (LIBERO-Goal), and long horizon tasks (LIBERO-Long). We followed the approaches of Kim et al. [19] and purged all the no-op actions during fine-tuning. For fine-tuning, we combined the data from four distinct subsets to train a single model. For evaluation, each of the four corresponding tasks was evaluated across 500 episodes. We report the average performance over three runs for each task, using three different random seeds.

SimplerEnv [24] simulated benchmark was aimed at minimizing the gap between reality and simulation by optimizing the PD parameters with simulated annealing to minimize the gap between real and simulated end-effector trajectories. The evaluation is focused on four built-in tasks: pick coke can, move object near object, open drawer, and close drawer. Further, SimplerEnv allows two types of evaluation: (i) *visual matching*, where the success of a task is determined by superimposing the real-life images on the simulation background and (ii) *variant aggregation*, where the success rate of a task is averaged over the many variants of evaluation environment that differ in lighting, background, textures, distractor objects, etc. The full evaluation suite contains more than 1,000 episodes across these built-in tasks. We report the average performance over two runs for each task.

Cross-Embodiment Evaluation. To evaluate our model in the real-world, we assessed it on a Galaxea A1 robotic arm. This embodiment was deliberately chosen due to its absence from the large-scale pretraining dataset [11]. To adapt to this embodiment, we first collected 1,000 episodes of Pick-and-Place tasks via teleoperation. During data collection, we randomize the location of the objects in each episodes to enforce spatial generalization. We collected nine unique tasks, such as, *"Put apple on the plate", "Put mango in the basket"*, and *"Move the banana next to the plate"*. This set of tasks was designed to cover a variety of common objects.

To validate the performance of our models, we designed nine tasks to perform evaluation. Each task is repeated for 10 trials, adhering to Kim et al. [19]. To ensure a rigorous and fair comparison, these trials used 10 different fixed starting positions, which were kept consistent across all baselines. We divided our nine evaluation tasks into three categories, with three tasks per category. The first category consists of "seen" tasks, which are tasks that were also included in our fine-tuning dataset. This aims to validate the performance of our model and other baselines by cross-embodiment transfer.

The second category, "Unseen Object with Seen Distractor", features tasks like "Put X in plate". Here, the target 'X' is an unseen object absent from the fine-tuning dataset, while a familiar "seen" object 'Y' present in the fine-tuning dataset is simultaneously placed in the environment as a distractor. This setup aims to evaluate the models' ability to generalize and their instruction following capabilities.

The final category, "Unseen Instruction with Seen Distractor" features tasks like "Move X to Z". These tasks consist of simple instructions absent from the fine-tuning set and require the model to manipulate a novel object 'X' (unseen in fine-tuning) and place it relative to a "seen" destination object 'Z'. Crucially, a separate "seen" object 'Y' is also present in the scene as a distractor. This setup aims to evaluate the models' ability to generalize to out-of-distribution instructions and their robustness to the presence of distractors.

Metric. In the LIBERO and SimplerEnv simulations, if the robot successfully completes the task specified by the prompt, then the trial is counted as a success, receiving a score of 1; otherwise, a score of 0 is assigned:

$$\%\ \text{success rate}_\tau := (100\mathbb{E}_{\text{trial}\sim\{1,2,\cdots,10\}}\mathbf{1}[\text{task } \tau \text{ is successfully completed}])\%.$$

For the Cross-Embodiment Evaluation, we report both success rate and partial success rate. The partial success metric is crucial for this real-world setting, as it allows us to differentiate between models that fail completely and those that make significant progress, thereby providing a more comprehensive breakdown of performance and failure modes.

B. Related Works

Vision–Language–Action Models. Large-scale vision–language–action (VLA) models learn general robot policies by training transformer policies on diverse demonstration datasets. RT-1 [9] and the RT-X family trained on the Open X-Embodiment dataset [12] demonstrated that scaling real-world robot data and model capacity yields strong generalization across tasks and embodiments. Subsequent open VLA models follow this recipe while

incorporating stronger vision–language backbones, including OpenVLA [19], SpatialVLA [36], TraceVLA [51], NORA [16], Emma-X [42], EO-1[34], and MolmoAct [22]. These approaches primarily rely on supervised imitation learning on large cross-embodiment datasets, sometimes with additional embodiment-specific fine-tuning, but they do not study reward-based post-training of VLA policies.

Orthogonally, flow-matching-based action models such as π_0 [6] and $\pi_{0.5}$ [18] attach a continuous-time flow-matching action head to a pre-trained vision–language backbone to generate smooth, real-time continuous action trajectories. In parallel, discretized action tokenization methods such as FAST [32] focus on compressing continuous action sequences into short sequences of discrete tokens for efficient autoregressive decoding, and can be combined with π_0 to obtain the π_0-FAST variant. Our NORA-1.5 architecture similarly augments a pre-trained VLA backbone [16] with a flow-matching-based action expert; unlike prior work, we find that this coupling not only improves inference speed [6, 18] but also yields consistent accuracy gains across simulated and real-world benchmarks.

World Models for Visual Robot Control. Self-supervised video and world models aim to predict future observations conditioned on current observations and actions, and have been used for planning and model-based control [13, 46, 52]. V-JEPA2 [2] learns a latent video prediction objective that can be extended to action-conditioned dynamics, while DINO-WM [53] performs planning in the latent space of a pre-trained visual encoder. These methods typically use the world model online for planning or trajectory optimization. In contrast, we repurpose an action-conditioned V-JEPA2 variant as a reward model that scores full action sequences. This enables scalable synthetic preference generation for VLA post-training without task-specific reward engineering or high-fidelity robot simulators. Sim2real evaluation frameworks such as SimplerEnv [25] focus on accurately matching real robot trajectories in simulation to provide reliable evaluation for manipulation policies; we adopt such benchmarks to assess the gains from our world-model-based rewards.

Preference-based Post-Training and Reward Design. Preference optimization has become a standard tool for aligning large language models with human intent. Direct Preference Optimization (DPO) [38] optimizes a policy directly from pairwise preferences, and Group Relative Preference Optimization (GRPO) [40] extends this idea to group-wise comparisons. While VLA models are usually trained purely with supervised imitation learning, recent work such as RoboMonkey [20] explores synthetic reward signals for test-time sampling and verification of robot actions, without updating the underlying policy. Our work brings preference-based post-training to the VLA setting by constructing synthetic preferences from two complementary reward signals: a world-model-based goal reward and a distance-to-expert-action heuristic. We show that combining these rewards with DPO yields consistent performance improvements over purely supervised training on LIBERO [27] and SimplerEnv [25] benchmarks.

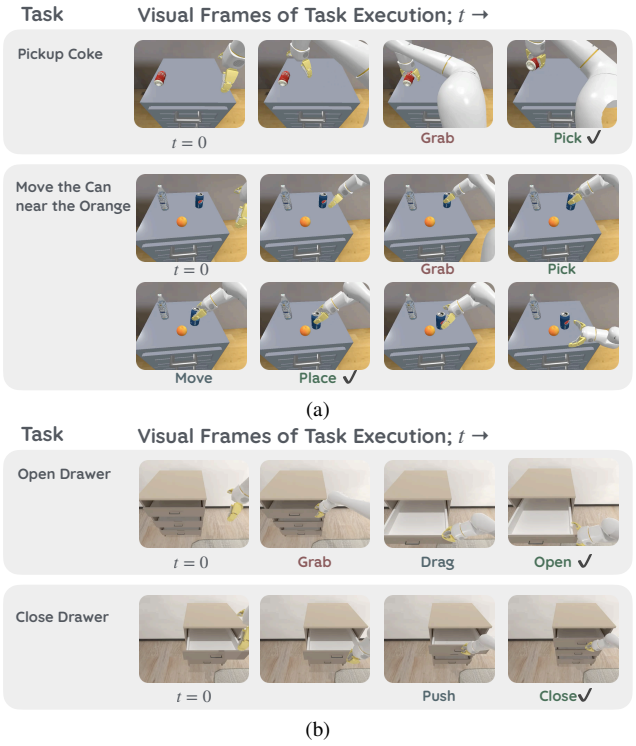


Figure 4. Examples of `NORA-1.5` executing evaluation tasks in SimplerEnv: (a) pickup coke and move object near another object and (b) open and close drawer.

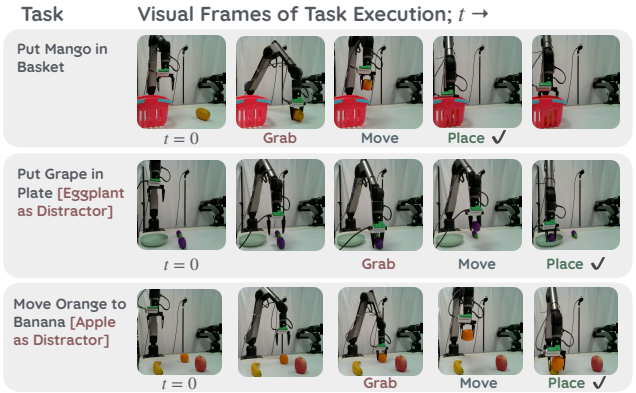


Figure 5. Examples of `NORA-1.5` executing evaluation tasks with Galaxea A1 robotic arm in the real world.

C. Model Architecture and Training Details

`NORA-1.5` is adapted from NORA-Long by initializing a new action expert that has approximately 400 million parameters. Embeddings of Qwen 2.5 VL and the newly initialized action expert interact through self-attention only. To prevent action representation of FAST token leaking to the action expert, the action expert is only allowed to attend embeddings corresponding to the language instruction and the image. During training, we optimize a joint cross entropy loss on FAST tokens as well as flow matching loss on the action expert: $\mathcal{L}_{\text{Loss}} = \mathcal{L}_{\text{CE}} + \alpha\mathcal{L}_{\text{FM}}$ where $\mathcal{L}_{\text{CE}}$ denotes the cross entropy loss on FAST tokens, $\mathcal{L}_{\text{FM}}$ is same as Equation 3 and α is a scaling term. We set $\alpha = 10$ during our training.

`NORA-1.5` is trained on the exact same subset of the Open-X-Embodiment dataset as NORA-Long. Previously, NORA-Long was trained for 900k gradient steps with a global batch size of 256. After initializing the action expert, we jointly optimize $\mathcal{L}_{\text{Loss}}$ for another 150k gradient steps with a global batch size of 512. We use a maximum learning rate of $5e-5$, a linear warm up of 5000 steps, and a cosine decay to 0. We used a single node of H100 GPU to train this for 5 days, approximately using 960 H100 hours.

D. Galaxea Data Collection

We collected 1000 episodes of simple pick and place tasks via teleportation for fine-tuning π_0, `NORA-1.5` and NORA-Long on Galaxea A1 robot. During the collection of data, we randomly place objects on the table and do not follow any order. We collected a total of 9 different tasks where each task has about $\tilde{1}00$+ episodes.

E. Baselines

We use the following baselines for a comparative evaluation of our approaches.

OpenVLA [19]: A VLA model is built upon a Llama 2 language model[44] combined with a visual encoder that integrates pretrained features from DINOv2 [30] and SigLIP[48]. It is pretrained on the Open-X-Embodiment dataset [11], which comprises 970k real-world robot demonstrations.

SpatialVLA [35]: A VLA model focused on spatial understanding for robot manipulation, incorporating 3D information such as spatial movement. It learns a generalist policy for spatial manipulation across diverse robots and tasks. SpatialVLA predicts four actions at a time.

TraceVLA [50]: A VLA model enhancing spatial-temporal reasoning via visual trace prompting. Built by fine-tuning OpenVLA on robot manipulation trajectories, it encodes state-action history as visual prompts to improve manipulation performance in interactive tasks.

RT-1 [8]: A scalable Robotics Transformer model designed to transfer knowledge from large task-agnostic datasets. Trained on diverse robotic data, RT-1 achieves a high level of generalization and task-specific performance across a variety of robotic tasks, demonstrating the value of open-ended task-agnostic training of high-capacity models.

HPT [45]. Heterogeneous Pre-trained Transformers (HPT) pretrain a shared transformer trunk on a large mixture of heterogeneous robot and video datasets, aligning proprioceptive and visual inputs into a unified token sequence. The resulting policy improves generalization across embodiments and tasks, and we use the released HPT policies as SimplerEnv baselines.

Octo-Base [29]. Octo is a transformer-based diffusion policy trained on ∼800k trajectories from Open X-Embodiment. We use the Octo-Base variant, a ViT-B–sized model that supports flexible action and observation spaces and can be fine-tuned efficiently for new robot setups.

RoboVLM [26]. RoboVLM is a framework for systematically studying design choices in VLAs and building generalist policies from diverse VLM backbones, architectures, and cross-embodiment data. We adopt their best-performing RoboVLM policy as a strong generalist VLA baseline.

π_0 and π_0-FAST [6]. π_0 is a vision-language-action model that attaches a flow-matching action expert to a pre-trained VLM and is trained on a large cross-embodiment dataset for high-frequency, dexterous control. π_0-FAST tokenize actions as discrete token using the FAST tokenizer. This enables faster convergence with lesser training compute. Both models serve as powerful generalist baselines.

MolmoAct / MolmoAct-7B-D [22]. MolmoAct is an action reasoning VLA that factors control into three stages: depth-aware perception tokens, mid-level spatial trajectory traces, and low-level actions. We use the 7B-D variant, MolmoAct-7B-D, which achieves strong zero-shot and fine-tuned performance on SimplerEnv and LIBERO.

Emma-X [43]. Emma-X is a 7B VLA obtained by fine-tuning OpenVLA on a hierarchical dataset derived from BridgeV2, with grounded chain-of-thought reasoning and look-ahead spatial guidance.

Magma [47]. Magma is a multimodal agentic foundation model that unifies vision, language, and action for both digital UI navigation and physical robot manipulation. It introduces visual planning traces and serves as a large-scale generalist baseline in our real-robot comparisons.

GR00T N1.5 [4, 28]. GR00T N1 is an open VLA foundation model for humanoid robots with a dual-system design: a vision-language backbone and a diffusion-based action policy. GR00T N1.5 is an improved release with architectural and data updates; we use the 3B N1.5 policy as a strong generalist baseline.

CoT-VLA [49]. CoT-VLA augments VLAs with *visual*

chain-of-thought reasoning: it first predicts subgoal images as visual plans and then generates short action sequences to reach those subgoals, improving performance on long-horizon and multi-step manipulation.

WorldVLA [10]. WorldVLA unifies a VLA policy and an image world model in a single autoregressive transformer, jointly modeling images, language, and actions. The world model predicts future images conditioned on actions, and the action head benefits from world-model feedback for better planning.

ThinkAct [14]. ThinkAct is a dual-system VLA that separates high-level reasoning from low-level action. A multimodal LLM produces structured embodied plans which are compressed into a visual latent, conditioning a downstream action policy for few-shot adaptation and long-horizon control.

NORA and NORA-Long [16]. NORA is a 3B VLA built on Qwen2.5-VL-3B and trained on Open X-Embodiment data with FAST tokenizer, designed to provide strong performance under tight compute budgets. NORA-Long is a variant with an extended action horizon and the original NORA VLA.