PRTS: A Primitive Reasoning and Tasking System via Contrastive Representations

Yang Zhang[1,2,*,†], Jiangyuan Zhao[1,3,*], Chenyou Fan[1,4], Fangzheng Yan[1], Tian Li[1], Haitong Tang[1], Xuan'er Wu[1], Qizhen Weng[1], Weinan Zhang[3], Xiu Li[2], Chi Zhang[1], Chenjia Bai[1,‡] and Xuelong Li[1]
[1]Institute of Artificial Intelligence (TeleAI),China Telecom, [2]Tsinghua University,
[3]Shanghai Jiaotong University, [4]Fudan University
[*]Equal Contributions, [†]Project Lead, [‡] Corresponding Author

Abstract*—Vision-Language-Action (VLA) models advance robotic control via strong visual-linguistic priors. However, existing VLAs predominantly frame pretraining as supervised behavior cloning, overlooking the fundamental nature of robot learning as a goal-reaching process that requires understanding temporal task progress. We present PRTS (Primitive Reasoning and Tasking System), a VLA foundation model that reformulates pretraining through Goal-Conditioned Reinforcement Learning. By treating language instructions as goals and employing contrastive reinforcement learning, PRTS learns a unified embedding space where the inner product of state-action and goal embeddings approximates the log-discounted goal occupancy, the probability of reaching the language-specified goal from the current state-action, quantitatively assessing physical feasibility beyond static semantic matching. PRTS draws this dense goal-reachability supervision directly from offline trajectories without reward annotations, and folds it into the VLM backbone via a role-aware causal mask, incurring negligible overhead over vanilla behavior cloning. This paradigm endows the high-level reasoning system with intrinsic goal reachability awareness, bridging semantic reasoning and temporal task progress, and further benefits goal-conditioned action prediction. Pretrained on 167B tokens of diverse manipulation and embodied-reasoning data, PRTS reaches state-of-the-art performance on LIBERO, LIBERO-Pro, LIBERO-Plus, SimplerEnv, and a real-world suite of 14 complex tasks, with particularly substantial gains on long-horizon, contact-rich, and zero-shot novel-instruction settings, confirming that injecting goal-reachability awareness significantly improves both execution success and long-horizon planning of general-purpose robotic foundation policies.

I. Introduction

Representation learning lies at the heart of robot learning, where the core challenge centers on extracting actionable knowledge from high-dimensional sensory inputs to enable goal-conditioned decision making. Vision-Language-Action (VLA) models have made remarkable strides by leveraging Vision-Language Models (VLMs) as powerful perceptual backbones, adopting a *dual-system* or *Mixture-of-Transformers* (MoT) architecture: System 2 (the VLM) handles high-level semantic understanding and task reasoning, while System 1 (the action expert) executes low-level continuous control [4, 5, 3]. However, this paradigm overlooks a fundamental characteristic of embodied interaction: *robotic trajectory is inherently a goal-reaching process over time*. While VLMs excel at encoding static visual-linguistic correlations [22, 29], they intrinsically lack any notion of such *goal reachability*, a quantitative estimate of how likely a given state-action pair is to eventually achieve the language-specified objective. Existing VLAs acquire visuomotor priors almost exclusively through behavior cloning during pretraining [24, 70, 4, 5, 3, 62, 52, 57]. Even prior works augmented with VQA data [5, 62] restrict System 2 to qualitative semantic reasoning ("*what to do*"), failing to encapsulate the quantitative goal-reachability awareness ("*how close it is to being done*").

To address this deficiency, we reformulate VLA pretraining through the lens of *Goal-Conditioned Reinforcement Learning*. Specifically, we adopt Contrastive Reinforcement Learning (CRL) [17], which learns paired state-action and goal representations whose similarity approximates the log-discounted goal occupancy. This formulation intrinsically endows the VLM with temporal awareness of goal reachability directly within its representation space.

In this paper, we present **PRTS** (**P**rimitive **R**easoning and **T**asking **S**ystem), a novel VLA foundation model that, for the first time, scales *Contrastive RL* directly into VLA pretraining for representation learning. PRTS introduces three key technical contributions. **(i) Language-conditioned contrastive RL formulation.** Standard CRL geometrically samples future state within a single trajectory as the positive. However, in our setting, the language instruction serves as a fixed goal shared across all timesteps of a trajectory. PRTS reformulates this as a *multi-positive* contrastive problem. By introducing a *temporal-weighting* scheme that mathematically recovers geometric sampling, we derive bidirectional InfoNCE objectives that jointly capture *task identity* $(s, a \to l)$ and *task progress* $(l \to s, a)$. This enables the model to extract dense goal-reachability supervision entirely from offline trajectory structures, eliminating the need for manual reward annotations. **(ii) Single-forward-pass architectural integration.** A naive integration would require an additional forward pass to extract the contrastive representations. PRTS circumvents this by appending two small auxiliary token blocks (`<CRL_action>`, `<CRL_goal>`) to the sequence and enforcing strict information isolation via a *role-aware causal mask* implemented as a custom CuTe-fused FlashAttention kernel [61]. This guarantees that the behavior-cloning loss remains identical to a vanilla setup while matching the per-gradient-step training cost of a pure-BC run on the Qwen3-VL backbone. Consequently, we seamlessly scale CRL pre-training to a 167B-token corpus across 64 H100 GPUs in one week. **(iii) Substan-**

tially stronger out-of-distribution robustness. We evaluate PRTS across four simulation suites (LIBERO [34], LIBERO-Plus [19], LIBERO-Pro [69], SimplerEnv WidowX [30]) and a real-world 14-task suite spanning dual-arm (RealMan) and single-arm (Flexiv) platforms. Under strictly comparable post-training budgets, PRTS not only matches or exceeds the strongest prior VLAs in-distribution but, more importantly, exhibits performance gaps that *widen with task difficulty*. These advantages are most pronounced in long-horizon execution, zero-shot novel-instruction following, and recovery under heavy human interventions, precisely validating the goal-reachability prior brought by our pre-training.

II. Preliminaries

Goal-Conditioned Reinforcement Learning. We consider the standard goal-conditioned reinforcement learning (GCRL) formulation, where an agent interacts with a Markov Decision Process (MDP) to reach a desired goal state $s_g \in \mathcal{S}$, where $\mathcal{S}$ is the state space. The environment is defined by state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $p(s' \mid s, a)$, and initial state distribution $p_0(s)$.

Following Eysenbach et al. [17], we focus on the *goal-reaching reward* function, which quantifies the immediate probability of reaching the goal:

$$r_g(s_t, a_t) \triangleq (1-\gamma)\, p(s_{t+1} = s_g \mid s_t, a_t),$$

where $\gamma \in (0, 1)$ is the discount factor. Unlike sparse 0/1 rewards, this formulation assigns $(1-\gamma)$ scaled by the transition probability to s_g, implicitly capturing the likelihood of goal achievement through the dynamics rather than hard thresholding.

Given a goal-conditioned policy $\pi(a \mid s, s_g)$, the Q-function represents the expected cumulative discounted return:

$$Q^{\pi}_{s_g}(s, a) \triangleq \mathbb{E}_{\pi}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_g(s_{t'}, a_{t'}) \,\middle|\, s_t = s, a_t = a\right].$$

Discounted State Occupancy Measure. To establish the connection between value estimation and probability, we introduce the *discounted state occupancy measure* [49]. For a policy π conditioned on goal s_g, the occupancy measure of state s is defined as:

$$p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s) \triangleq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p_t^{\pi(\cdot|\cdot,s_g)}(s_t = s),$$

where $p_t^{\pi}(s_t = s)$ denotes the probability density of visiting state s after executing π for exactly t steps starting from $p_0(s)$. This measure admits an intuitive sampling procedure: one first samples a time offset $t \sim \mathrm{Geom}(1-\gamma)$ from a geometric distribution, then rolls out policy π for t steps. The resulting state follows the occupancy distribution $p^{\pi}(s_{t+})$. This property forms the theoretical foundation for constructing positive samples in standard contrastive RL.

Q-Function as Occupancy Probability. A critical insight from goal-conditional RL is that the Q-function for goal-reaching rewards corresponds exactly to the conditional occupancy probability:

Lemma 1 (Eysenbach et al. [17])**.** *For the goal-reaching reward r_g, the Q-function equals the probability of reaching goal s_g under the discounted state occupancy measure starting from (s, a):*

$$Q^{\pi}_{s_g}(s, a) = p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s_g \mid s, a). \tag{1}$$

This equivalence transforms the problem of value estimation into one of probability estimation: estimating how likely the agent is to visit s_g in the future when following π from (s, a).

Contrastive RL Framework. Building upon this equivalence, Contrastive RL estimates the Q-function via contrastive representation learning, avoiding the bootstrapping errors inherent in Temporal-Difference (TD) learning. The critic $f(s, a, s_g)$ measures the correlation between a state-action pair (s, a) and a future goal s_g. Following standard practice, we parameterize it as the inner product between encoded representations:

$$f(s, a, s_g) = \phi(s, a)^{\top} \psi(s_g),$$

where $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is the state-action encoder and $\psi : \mathcal{S} \to \mathbb{R}^d$ is the goal encoder.

Contrastive RL objective learns f by discriminating between *achievable future states* (positive samples) and *random states* (negative samples), both denoted as s_g but drawn from different distributions:

- **Positive:** A goal state sampled from the discounted occupancy measure, $s_g^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+} \mid s, a)$, representing a state actually reachable by π from (s, a);
- **Negative:** A goal state sampled from the marginal data distribution, $s_g^- \sim p(s_g)$, representing a random state unrelated to the current policy.

The critic is trained via the CRL objective based on the infoNCE objective [53]:

$$\mathcal{L}_{\text{crl}} = \max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a),\\ s_g^+ \sim p^{\pi}(s_{t+}|s,a),\\ s_g^- \sim p(s_g)}} \Big[\log \exp(f(s, a, s_g^+)) - \log(\exp(f(s, a, s_g^+)) + \sum \exp(f(s, a, s_g^-)))\Big], \tag{2}$$

The optimal critic satisfies: $f^*(s, a, s_g) = \log p^{\pi}(s_{t+} = s_g \mid s, a) - \log p(s_g) + c(s, a)$, where $c(s, a)$ is a function independent of s_g. Leveraging Lemma 1 where $Q^{\pi}_{s_g}(s, a) = p^{\pi}(s_{t+} = s_g \mid s, a)$, this yields the direct correspondence:

$$f^*(s, a, s_g) = \log Q^{\pi}_{s_g}(s, a) - \log p(s_g) + c(s, a), \tag{3}$$

implying $\exp(f^*(s, a, s_g)) \propto Q^{\pi}_{s_g}(s, a)$ up to a state-action-dependent normalization, without any explicit reward labeling.

III. Methodology

A. Problem Setup

We consider a language-conditioned robotic manipulation task under the imitation learning paradigm. Let $\mathcal{D}_{\text{pre}} =$

$\{\tau_1, \tau_2, \ldots, \tau_n\}$ denote a large-scale pre-training dataset comprising n expert demonstration trajectories collected from diverse robotic tasks. Each trajectory $\tau = (s_{1:T}, l, a_{1:T})$ consists of a natural language instruction l specifying the task objective, a sequence of observations $s_{1:T}$, and corresponding actions $a_{1:T}$. At timestep t, the observation $s_t = (\mathbf{I}_t^1, \ldots, \mathbf{I}_t^V, \mathbf{q}_t)$ comprises V RGB camera views and the robot's proprioceptive state $\mathbf{q}_t$. The action a_t represents the low-level continuous control commands.

Our objective is to learn a unified representation space during pre-training that encodes both visual-linguistic semantics and the temporal structure of goal-reaching. Formally, we aim to extract a goal-conditioned action-value function $Q_l^\pi(s, a)$, i.e., the discounted goal occupancy measure, directly from the representation geometry of the VLM backbone, entirely bypassing explicit reward annotations or auxiliary networks. We realize this via Contrastive RL, whose InfoNCE classification loss aligns natively with the cross-entropy objective of next-token prediction and scales stably to over 1000-layer networks [55]. These properties make it uniquely well-suited to the scale of VLA backbones.

B. Language-Conditioned Contrastive Reinforcement Learning

Standard CRL draws a single positive future state via geometric sampling $\Delta t \sim \text{Geom}(1-\gamma)$ within the same trajectory. In language-conditioned manipulation, however, the goal is a fixed textual instruction l shared across all timesteps of a trajectory. This structure transforms the standard formulation into a *multi-positive* contrastive problem.

Temporal Weighting as Geometric Sampling. Consider a mini-batch $\mathcal{B} = \{(s_j, a_j, l_j, \tau_j)\}_{j=1}^B$ sampled from the pre-training dataset $\mathcal{D}_{\text{pre}}$, where τ_j is the task identifier. For a given anchor sample i, the positive set comprises all intra-batch samples sharing the same task: $\mathcal{S}(i) \triangleq \{j \in \mathcal{B} : \tau_j = \tau_i\}$. Because language goals cannot be sampled geometrically from a future timeline, we emulate geometric state sampling by explicitly assigning temporal weights over the multiple positive state-action pairs. For each sample $j \in \mathcal{S}(i)$ at trajectory timestep t_j (with total episode length T_j), the discounted probability of reaching the goal can be approximated by $\gamma^{T_j - t_j}$ via Monte-Carlo estimation. Thus, we define the normalized temporal weight as:

$$q_{ij} = \frac{\gamma^{T_j - t_j}}{\sum_{j' \in \mathcal{S}(i)} \gamma^{T_{j'} - t_{j'}}}. \tag{4}$$

This weighting scheme assigns exponentially larger weights to states closer to task completion and mirrors the decay of the geometric distribution.

Bidirectional Contrastive Objectives. To jointly capture task identity and temporal progress, we optimize two complementary cross-entropy objectives, where $\phi_i \triangleq \phi(s_i, a_i)$ and $\psi_i \triangleq \psi(l_i)$.

(i) State-Action to Language ($s, a \to l$): For each state-action anchor, we optimize:

$$\mathcal{L}^{\text{sa}\to l} = \mathbb{E}_{i\sim\mathcal{B}} \left[-\gamma^{T_i - t_i} \log \frac{\exp(\phi_i^\top \psi_i)}{\exp(\phi_i^\top \psi_i) + \sum_{k\in\mathcal{B},\, \tau_k \neq \tau_i} \exp(\phi_i^\top \psi_k)} \right]. \tag{5}$$

Since each (s_i, a_i) has exactly *one* positive goal l_i (its own task's instruction), the formulation is reduced to the standard CRL problem. Thus, to present fixed goals under the geometric sampling distribution, we weight each state-action pair by $\gamma^{T_i - t_i}$. This objective primarily enforces *task-level discrimination*, effectively aligning state-action pairs with their semantic instructions while remaining agnostic to the precise time-to-completion.

(ii) Language to State-Action ($l \to s, a$): To explicitly inject the temporal signal, we reverse the direction using the multi-positive soft targets q_{ij}:

$$\mathcal{L}^{l\to\text{sa}} = \mathbb{E}_{i\sim\mathcal{B}} \left[-\sum_{j\in\mathcal{S}(i)} q_{ij} \log \frac{\exp(\psi_i^\top \phi_j)}{\sum_{k\in\mathcal{B}} \exp(\psi_i^\top \phi_k)} \right]. \tag{6}$$

Unlike $sa \to l$ direction, l_i now has *multiple* positives. This objective forces the predicted similarity distribution to match the temporal decay of the trajectory, scaling as $\gamma^{T_j - t_j}$ and thereby encoding the *temporal distance to task completion* within the representation space. We formally prove in Theorem 1 (see Appendix C) that the optimal representations under $\mathcal{L}^{l\to\text{sa}}$ satisfy

$$\psi^*(l)^\top \phi^*(s, a) = \log Q_l^{\pi^*}(s, a) + C(l), \tag{7}$$

recovering the log-discounted occupancy of standard CRL (Eq. (3)) up to a goal-dependent constant.

The combination of **(i)** and **(ii)** ensures that the learned representations capture both *task identity* and *task progress*. $\mathcal{L}^{\text{sa}\to l}$ prevents inter-task interference, while $\mathcal{L}^{l\to\text{sa}}$ arranges intra-task states by their proximity to the goal, creating a structured representation space where the inner product $\psi(l)^\top \phi(s, a)$ serves as a valid estimate of the log-discounted language goal occupancy measure. Finally, we emphasize that although the derivation in this section uses single-step actions, the formulation naturally extends to action chunks used in VLAs without changing the objectives or theoretical results.

C. PRTS Architecture

We instantiate PRTS upon the Qwen3-VL backbone [2], extending it to jointly optimize discrete action token modeling and contrastive representation learning. As illustrated in Fig. 1, the input sequence comprises multi-view visual tokens, textual-format discretized proprioception [5, 15], language instructions, and auto-regressive (AR) action tokens tokenized via FAST [42]. Crucially, to extract contrastive embeddings within the identical forward used for next-token prediction, we append two auxiliary token blocks at the sequence tail: `<CRL_action>` and `<CRL_goal>`. The `<CRL_action>` block duplicates the action tokens and concludes with a learnable `<|action_end|>` token, whose final hidden state encodes $\phi(s_t, \mathbf{a}_t)$. Similarly, the `<CRL_goal>` block duplicates the

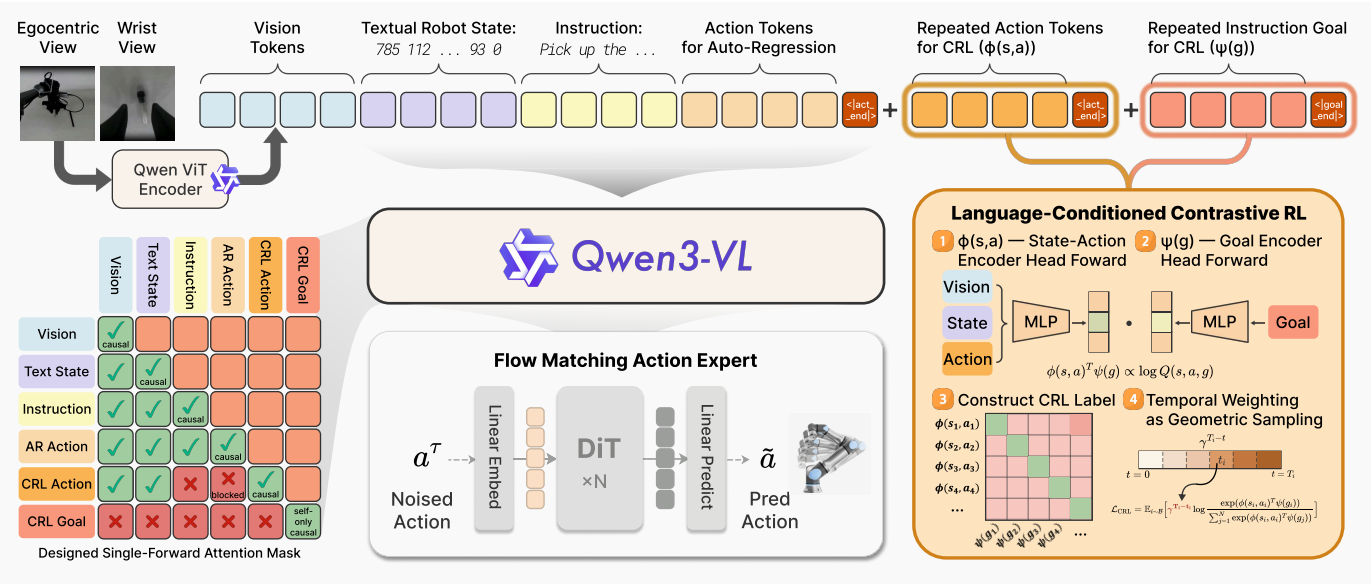


Fig. 1: **Overview of the PRTS Architecture.** PRTS is built on Qwen3-VL and involves two training stages. (i) **Pre-training**: The VLM backbone processes a unified sequence of vision, proprioception, language, and auto-regressive (AR) action tokens. We append two auxiliary blocks (<CRL_action> and <CRL_goal>) to extract representations $\phi(s_t, \mathbf{a}_t)$ and $\psi(l)$. A *role-aware causal mask* prevents information leakage, enabling joint optimization of next-token prediction and bidirectional CRL within a single forward pass. (ii) **Post-training**: A DiT-based flow-matching action expert is attached and fine-tuned to predict continuous action chunks.

instruction tokens and terminates with <|goal_end|> token, encoding $\psi(l)$.

Role-Aware Causal Mask for Disentangled Representations. To prevent information leakage between $\phi(s_t, \mathbf{a}_t)$ and $\psi(l)$ while preserving the efficiency of standard next-token training, we introduce a role-aware causal mask that partitions tokens into five roles, vision/state, instruction, AR action, CRL_action, and CRL_goal, under three attention constraints: (i) AR action tokens follow standard causal attention over all preceding tokens, keeping the BC loss *identical* to a non-CRL run; (ii) CRL_action tokens attend only to vision/proprioception (blocked from language and AR action tokens), ensuring that $\phi(s, a)$ extracts a pure state-action representation independent of the language goal; (iii) CRL_goal tokens employ self-only causal attention within their own block, forcing $\psi(l)$ to be a compact task-specific encoding of the instruction alone, serving as the goal anchor in the contrastive objective. This design enables computing both discrete action token predictions and contrastive representations $\phi(s_t, \mathbf{a}_t), \psi(l)$ within a single forward pass, preserving training efficiency.

Practical Implementations. We implement the role-aware mask on top of FlashAttention [61] via a custom CuTe kernel that supports flexible attention computation acceleration. This brings the joint forward pass to the throughput as a pure behavior cloning run with FlashAttention-2/3 as backend [13, 47]. The design further remains compatible with the sequence packing trick, which we also involve to accelerate our large-scale pre-training. PRTS therefore produces the discrete-action logits, $\{\phi_i\}$, and $\{\psi_i\}$ in a single forward pass whose per-step cost closely matches that of a pure-BC run on the same Qwen3-VL backbone (quantified in Sec. IV), a property we rely on to scale contrastive-RL pre-training to over 167B tokens.

Action Expert. For real-time continuous control, we attach a flow-matching action expert [32, 36, 33] built on a Diffusion Transformer (DiT) [41], following recent VLA designs [4, 5]. It outputs the action chunk $\tilde{\mathbf{a}}_t$ with a chunk horizon H over 5 denoising steps, conditioned on the VLM backbone hidden states, enabling high-frequency continuous action generation while the backbone focuses on high-level semantic reasoning. The action expert f_θ is trained via conditional flow matching [33]:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau\sim[0,1],\,\epsilon\sim\mathcal{N}(0,I)} \left\| f_\theta(\mathbf{a}_t^\tau,\, s_t,\, l) - (\mathbf{a}_t - \epsilon) \right\|_2^2, \quad (8)$$

where $\mathbf{a}_t^\tau = \tau\,\mathbf{a}_t + (1-\tau)\,\epsilon$ is the noised action chunk, and $\tau \in [0, 1]$ denotes the flow-matching time index.

Pre-training and Post-training Objectives. PRTS adopts a two-stage scheme that separates representation learning from continuous-action adaptation, with gradients flowing through the VLM backbone in both. In *pre-training*, the backbone is trained jointly on the bidirectional contrastive losses and the standard next-token cross-entropy, which covers the BC loss on AR action tokens and the auxiliary robot-QA and sub-task prediction losses; the action expert is not involved. For each mini-batch $\mathcal{B}$, the representations $\phi(s_{t_j}, \mathbf{a}_{t_j})$ and $\psi(l_i)$ are extracted from the <CRL_action> and <CRL_goal> hidden states via lightweight MLP heads, utilizing ℓ_2-normalization. Together with the temporal weights q_{ij} (Eq. (4)) and a learnable temperature τ_{crl}, the dot products $\phi_j^\top \psi_i$ of these representations drive the two contrastive objectives $\mathcal{L}^{\text{sa}\to l}$ and $\mathcal{L}^{l\to\text{sa}}$. Since the InfoNCE objective benefits from a large

negative pool [53], we shard the similarity computation across GPUs as in CLIP [44], so each device aggregates negatives globally while back-propagating only its local batch. The pre-training objective is

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{RobotQA}} + \mathcal{L}_{\text{Subtask}} + \lambda_{\text{crl}} \left(\mathcal{L}^{\text{sa}\to l} + \mathcal{L}^{l\to\text{sa}} \right), \tag{9}$$

where $\mathcal{L}_{\text{BC}}$ is the cross-entropy on AR action tokens and $\mathcal{L}_{\text{RobotQA}}, \mathcal{L}_{\text{Subtask}}$ are the auxiliary robot-QA and sub-task losses. In *post-training*, we attach the action expert and fine-tune on downstream demonstrations with the flow-matching loss alone, $\mathcal{L}_{\text{post-train}} = \mathcal{L}_{\text{FM}}$; the pre-trained representations already encode task identity and goal reachability, so this stage focuses entirely on continuous-action regression.

Pre-training Data. Our pre-training corpus comprises **404M** samples, totaling **167.8B** tokens, strategically partitioned into two primary streams. To establish robust cross-embodiment execution priors, the *Action-Labeled Data* incorporates AgiBotWorld, RoboMind, Open X-Embodiment, and our self-collected RealMan and Agilex trajectories. To foster complex cognitive capabilities, the *Visual-Reasoning Data* spans spatial grounding (RefCOCO, Pixmo-Point, RoboPoint), affordance perception (RoboAfford), long-horizon planning (EgoPlanIT, Cosmos-Reason1), and general vision-language competence (LLaVA-Instruct). Detailed source breakdowns and dataset distributions (Fig. 7) are provided in Appendix D.

IV. Experiments

We evaluate PRTS across simulation and real-world environments to answer six questions: **(Q1) Overall Performance:** How does PRTS compare against state-of-the-art (SOTA) VLAs across varied domains? (Sec. IV-B); **(Q2) Effect of Contrastive RL:** How does CRL pre-training drive downstream performance under strictly bounded fine-tuning budgets? (Sec. IV-D); **(Q3) Generalization Capability:** Can PRTS achieve robust generalization across physical (e.g., spatial, lighting, object appearance) and semantic (novel instructions) distribution shifts? (Sec. IV-C); **(Q4) Robustness:** Can the policy dynamically recover from frequent human interventions during execution? (Sec. IV-C); **(Q5) Value Representation:** Do the learned representations faithfully encode goal-conditioned action values? (Sec. IV-D); and **(Q6) Pre-Training Efficiency:** How computationally efficient is our optimized infrastructure for scaling CRL? (Sec. IV-D).

A. Experimental Setup

Pre-training recipe. We initialize PRTS from Qwen3-VL-4B-Instruct [2] and optimize Eq. (9) ($\lambda_{\text{crl}} = 1.0$, $\gamma = 0.995$) over the corpus (Appendix D). By packing short sequences to length 4096, we complete one pre-training epoch on 64 H100 GPUs in about 1 week (global batch size 256, about 220K gradient steps). A sharded similarity strategy provides about 2K in-batch negatives to effectively scale the CRL objective.

Post-training recipe. We append a randomly initialized 675M-parameter flow-matching DiT action expert (5 denoising steps) and optimize Eq. (8). Crucially, our post-training budgets (Table I) are strictly bounded, which is often much smaller than baselines (e.g., $1/8$ the batch size of $\pi_{0.5}$ on LIBERO), ensuring performance gains stem from our pre-trained visuomotor prior rather than heavy downstream fine-tuning.

Benchmark	BS	Steps	Action Chunk
LIBERO	32	30K	20
SimplerEnv (WidowX)	1024	20K	16
RealMan (all-task fine-tuning)	64	100K	20
Flexiv (single-task fine-tuning)	32	40K	20

TABLE I: **Post-training budgets.**

Method	Spat.	Obj.	Goal	Long	Avg.
OpenVLA [24]	84.7	88.4	79.2	53.7	76.5
GR00T-N1.6 [3]	97.7	98.5	97.5	94.4	97.0
π_0 [4] (bs=32, 30K)	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$ [5] (bs=256, 30K)	**98.8**	98.2	98.0	92.4	96.9
ABot-M0 [59] (bs=32, 30K)	98.6	99.6	98.6	94.8	97.9
ABot-M0 [59] (bs=64, 40K)	**98.8**	**99.8**	**99.0**	**96.6**	**98.6**
PRTS (Ours) (bs=32, 30K)	**98.8**	**99.8**	98.4	**96.6**	98.4

TABLE II: **LIBERO Evaluation.** Post-training budgets in gray. PRTS matches SOTA under bounded budgets.

Benchmarks & Protocols. We evaluate across five settings spanning simulation and two hardware platforms (a single-arm Flexiv and a stationary 14-DoF dual-arm RealMan platform; Appendix F): (1) **LIBERO** [34] for in-distribution accuracy; (2) **LIBERO-Plus** [19] (10,030 tasks, 7 perturbation axes) for robustness; (3) **LIBERO-Pro** [69] (5 axes) for anti-memorization instruction adherence; (4) **SimplerEnv** [30] (WidowX) for real-to-sim evaluation (BridgeData V2); and (5) **Real-world setups** (3 Flexiv, 11 RealMan tasks) under controlled physical perturbations. We report Success Rates (SR) against state-of-the-art VLAs in simulation. Real-world evaluations directly benchmark against π_0 [4] and $\pi_{0.5}$ [5] over 20 trials per task, with strict failure conditions (exceeding 5 grasp attempts, or 2/3-minute limits for Flexiv/RealMan). Throughout evaluations, baseline post-training budgets are annotated in gray to explicitly highlight PRTS's superior efficiency under comparable or minimal constraints.

B. Overall Performance

We evaluate PRTS across simulation and real-world settings. For clarity, main-text tables highlight comparisons against SOTA baselines and primary competitors ($\pi_0, \pi_{0.5}$); complete baseline results are deferred to Appendix E.

LIBERO. Under a strict post-training budget (bs$=32$, 30K steps), PRTS reaches 98.4% average SR, matching SOTA while $\pi_{0.5}$ uses an $8\times$ larger batch for only 96.9%. Restricting the primary baseline ABot-M0 to our identical budget degrades its average SR to 97.9% with a notable drop from 96.6% to 94.8% on the *Long* suite. PRTS instead sustains 96.6% on *Long*, indicating its temporal reasoning capabilities stem from the pre-trained prior rather than downstream SFT.

Zero-Shot Robustness (LIBERO-Plus & Pro). On LIBERO-Plus (zero-shot, Table III), PRTS achieves the highest

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Average
OpenVLA [24]	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
π_0 [4] (bs=32, 30K)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
ABot-M0 [59] (bs=32, 30K)	**65.7**	53.1	81.2	**96.8**	95.0	**87.5**	81.3	78.7
$\pi_{0.5}$ [5] (bs=256, 30K)	64.0	58.0	88.5	96.6	81.4	**87.5**	**85.9**	80.7
PRTS (Ours) (bs=32, 30K)	59.5	**72.4**	**89.6**	96.5	**96.7**	81.3	83.3	**81.4**

TABLE III: **Evaluation on LIBERO-Plus.** PRTS exhibits superior zero-shot robustness across seven perturbation axes.

Method	Sem.	Obj.	Pos.	Task	Avg.
OpenVLA [24]	**97.2**	93.0	0.0	0.0	47.6
ABot-M0 [59]	97.1	82.5	7.1	22.3	52.2
π_0 [4]	90.5	90.5	0.0	0.0	45.3
$\pi_{0.5}$ [5]	95.8	**96.0**	20.8	0.8	53.3
PRTS (Ours)	97.0	82.3	**24.3**	**31.5**	**58.8**

TABLE IV: **Evaluation on LIBERO-Pro.**

Method	Spoon	Carrot	Block	Eggpl.	Avg.
Qwen3-VL-PI [48]	78.1	46.9	30.2	88.5	60.9
GR00T-N1.5 [3]	75.3	54.3	57.0	61.3	61.9
π_0 [4]	29.1	0.0	16.6	62.5	27.1
$\pi_{0.5}$ [5]	29.2	41.7	0.0	41.7	28.2
PRTS (Ours)	**83.3**	**66.6**	**58.3**	**100.0**	**77.1**

TABLE V: **SimplerEnv (WidowX) visual matching.**

average SR (81.4%), outperforming $\pi_{0.5}$ (80.7%) despite an $8\times$ smaller post-training batch, with peak gains on *Robot* (+14.4%) and *Background* (+15.3%). LIBERO-Pro additionally tests instruction adherence against memorization (e.g., position swaps, novel tasks). PRTS reaches 58.8% average (+5.5% over $\pi_{0.5}$, Table IV), with 31.5% on the most challenging *Task* axis, a 9.2-point margin over the second-best, demonstrating zero-shot generalization.

Real2Sim Evaluation (SimplerEnv). Post-trained on BridgeData V2 with a budget matching GR00T-N1.5, PRTS averages 77.1% SR (Table V), outperforming GR00T-N1.5 (+15.2%) and Qwen3-VL-PI (+16.2%) and reaching 100% on *Eggplant*.

Real-World Evaluation. Fig. 3a reports SRs on the 11-task bimanual RealMan and 3-task single-arm Flexiv suites. On RealMan, we evaluate a single shared checkpoint across all tasks to stress-test PRTS under cross-task interference settings. PRTS averages 95.9% SR vs. $\pi_{0.5}$'s 85.5%; on the two-minute *Office Long Term* task, PRTS maintains 95% while $\pi_{0.5}$ collapses to 40%. On the contact-rich Flexiv platform (per-task fine-tuning), PRTS averages 90.0% SR (+15.0% over $\pi_{0.5}$). It exhibits exact spatial grounding on *Block Combination* (100% vs. 90%) and mitigates compounding errors on the long-horizon *Flower Arrangement* (70% vs. 35%). These results indicate the CRL prior supports long-horizon execution and tight spatial constraints across diverse hardware setups.

C. Real-World Generalization and Robustness

Controlled Real-World Generalization (Q3). Serving as the physical counterpart to the LIBERO-Plus/Pro distribution shifts, we evaluate real-world robustness by perturbing four representative bimanual tasks along four axes (Fig. 2 and Fig. 4): *Light* (illumination), *Pos* (out-of-distribution spatial layouts), *Obj* (unseen instances), and *Task* (instruction recombination). The *Task* axis tests compositional generalization by demanding the retrieval and recomposition of previously seen skills under novel semantic instructions (Appendix G).

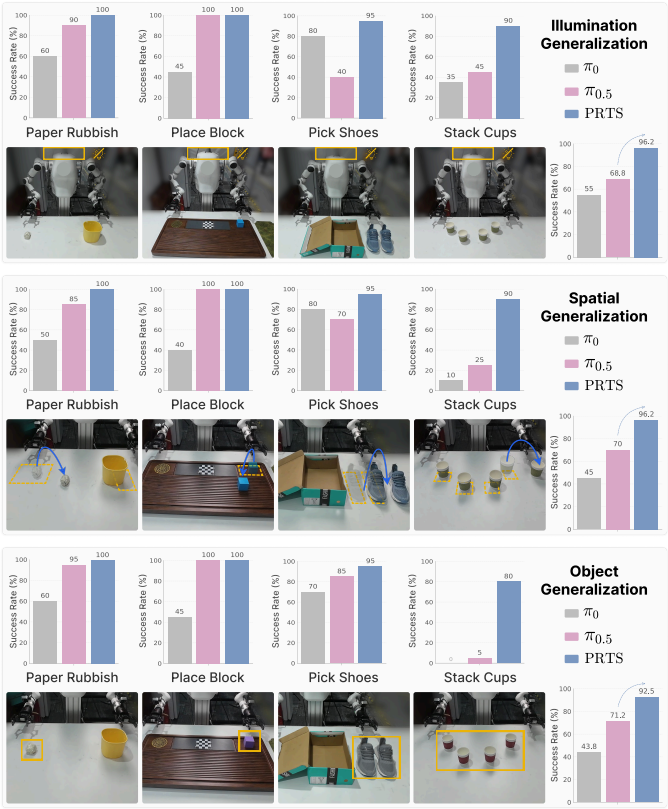
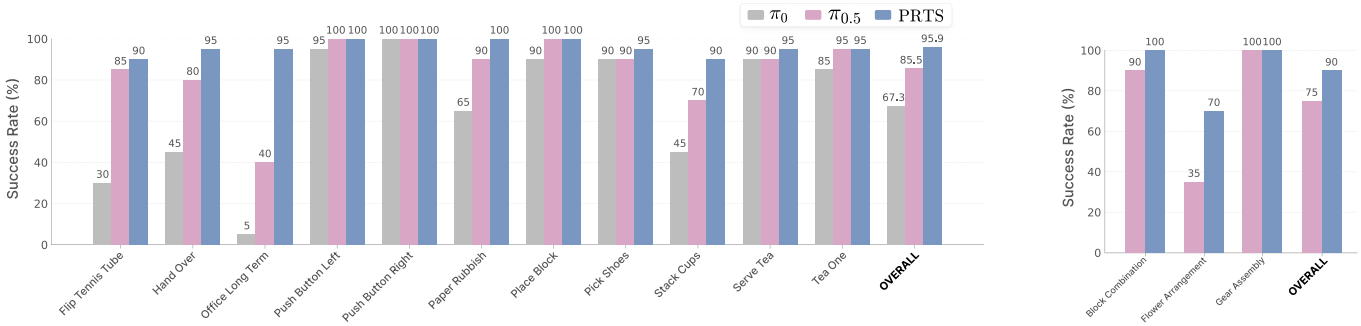


Fig. 2: **Lower-level generalization axes on RealMan.** We visualize three experimental settings testing visual and spatial robustness: *Illumination* (alters global lighting), *Spatial* (moves objects outside the post-training layout distribution), and *Object* (substitutes visual instances while preserving semantic categories).

Across physical shifts (*Light, Pos, Obj*), PRTS sustains 100% SR on *Paper Rubbish* and *Place Block*, and $\geq$ 80% on the remaining tasks (Table XII). The *Task* axis poses a stricter semantic challenge, where PRTS averages 73.8% SR compared to $\pi_{0.5}$ (35.0%) and π_0 (13.8%). For example, when substituting a crumpled paper target with a flat wireless mouse, baselines fail to adapt their grip. In contrast, PRTS seamlessly retrieves a planar grasp from a different context (80% SR), indicating that CRL pre-training aligns language goals with feasible state-action outcomes rather than memorizing trajectories.



(a) RealMan task suite.

(b) Flexiv task suite.

Fig. 3: **Real-world evaluation.** Success rates averaged over 20 trials per task. **(a) RealMan dual-arm suite:** PRTS reaches $\geq 90\%$ on all 11 tasks, outperforming baselines under an identical multi-task protocol. **(b) Flexiv single-arm suite:** PRTS excels across all three contact-rich and spatially-aligned tasks.

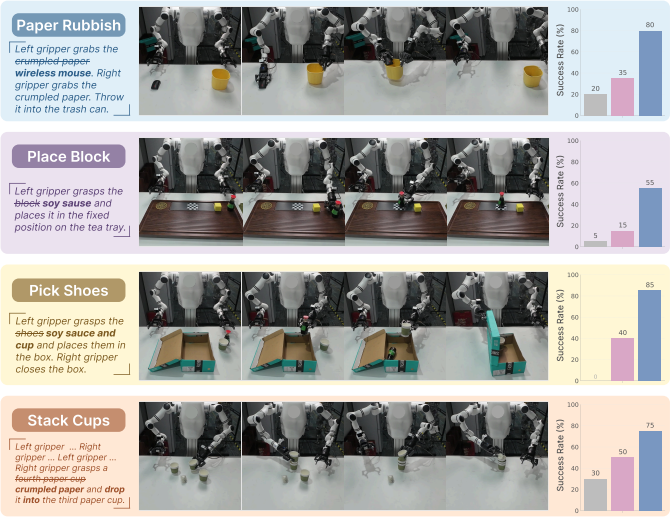


Fig. 4: **Task-level generalization on RealMan.** This axis fundamentally alters the language goal by recombining components of seen tasks to test semantic instruction binding. It evaluates whether the policy can bind a novel instruction to feasible manipulation steps. Quantitative results are in Table XII.

Robustness under Human Interventions (Q4). To evaluate dynamic recovery during execution, we conduct human intervention stress tests across four representative tasks: *Paper Rubbish*, *Pick Shoes*, *Place Block* (RealMan), and *Gear Assembly* (Flexiv). Unlike initial-state perturbations, mid-rollout interventions invalidate ongoing actions, reset object states, or regress multi-stage progress. The policy must therefore continuously re-estimate the state and dynamically replan toward the language goal, rather than executing open-loop trajectories. We evaluate qualitative recovery behaviors across $\sim$5 trials per task (Fig. 5).

PRTS demonstrates robust recovery across diverse dynamic disruptions. In *Place Block*, it dynamically tracks the moving target, halts invalidated actions when the block is removed mid-transport, and successfully re-enters the task logic following a post-completion reset. Similarly, in *Paper Rubbish* and *Pick Shoes*, PRTS successfully re-localizes manipulated items, infers unfinished subgoals, and selectively re-executes affected stages. Under severe interventions in the contact-rich *Gear Assembly*—where a fully assembled gear is forcefully dismantled—PRTS accurately detects the regressed state and repeats the necessary insertion sequence, validating that its recovery remains strictly goal-conditioned.

Conversely, baselines struggle with dynamic state invalidation. π_0 defaults to open-loop replays, failing under all *Place Block* perturbations. While $\pi_{0.5}$ exhibits reactivity to simple pre-grasp object movements, it struggles to re-enter earlier subgoals following severe post-completion resets, stalling due to erroneous progress estimation. These failures demonstrate that PRTS's robustness stems directly from its goal-reachability-aware representations, which continuously evaluate feasible state-action outcomes even when execution histories are invalidated.

D. Ablation and System Analysis

Effect of CRL (Q2). To isolate CRL's contribution, we evaluate a baseline ($\lambda_{\text{crl}} = 0$) under identical budgets. While yielding modest in-distribution gains ($+0.6\%$ SR), CRL unlocks substantial zero-shot robustness: $+4.9\%$ on LIBERO-Plus (e.g., $+16.3\%$ on *Robot* shifts) and $+5.0\%$ on LIBERO-Pro. Furthermore, t-SNE visualizations indicate CRL structures embeddings into distinct, reusable primitives compared to the less structured representations of pure BC, suggesting a composable goal-conditioned prior (Appendix H).

Value Representation (Q5). Theorem 1 establishes that the contrastive inner product $\phi(s_t, \mathbf{a}_t)^\top \psi(l)$ tracks the log-transformed goal-conditioned action value $\log Q_l^\pi(s_t, \mathbf{a}_t)$. Consequently, this score should increase as a trajectory approaches completion under the correct instruction, while remaining persistently low under a mismatched instruction. To verify this, we evaluate the pre-trained CRL encoder heads on a held-out *Pick Shoes* trajectory *prior* to any RealMan post-training. Because these demonstrations are strictly out-of-distribution, this tests whether the learned representations capture zero-shot goal-reachability semantics. We score the identical state-action sequence $(s_t, \mathbf{a}_t)$ under two conditions: a correct instruction describing the actual task (*grasp two shoes and place them into the shoebox*), and a deliberately mismatched instruction referencing absent objects (*soy sauce*

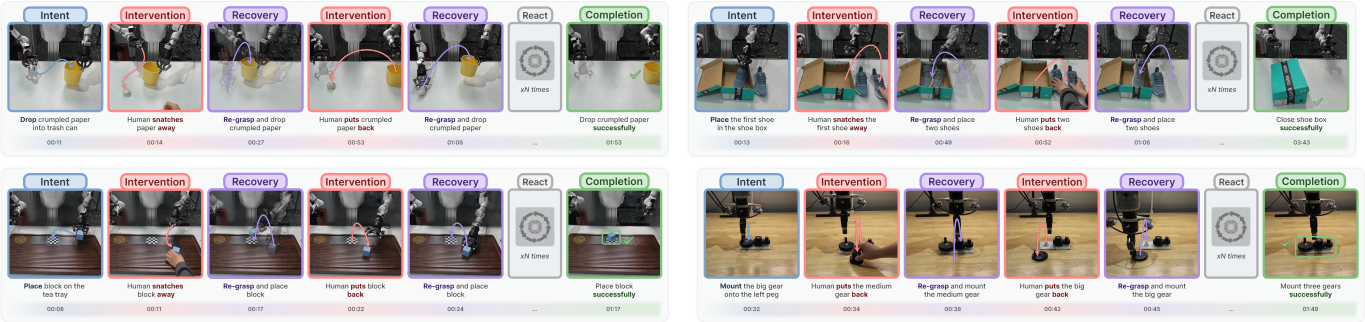


Fig. 5: **Recovery under human interventions.** Rollouts under dynamic physical disruptions. PRTS continuously re-estimates task progress to fulfill the pending language goal despite invalidated execution histories.

and *green paper cup*), mirroring the semantic shift from Sec. IV-C.

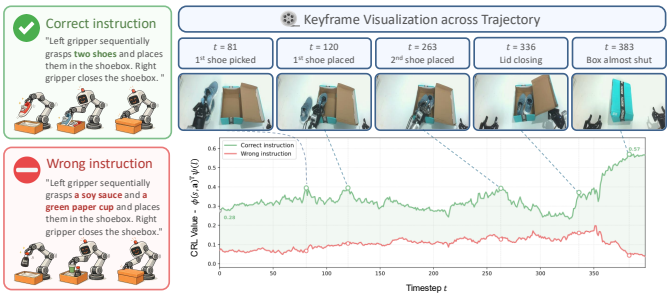


Fig. 6: **Zero-shot CRL value assessment on a held-out *Pick Shoes* trajectory.** The trace visualizes: **(Left)** The identical state-action sequence scored under correct (green) and mismatched (red) instructions (soy sauce, paper cup swapped to absent objects). **(Top Right)** Timestep indices marking structural task milestones. **(Bottom Right)** The contrastive value $\phi(s_t, \mathbf{a}_t)^\top \psi(l)$ as a function of timestep t, displaying progress-sensitive local peaks. Dashed lines link keyframes to the correct (green) instruction's markers. A persistent margin between the correct (green) and mismatched (red) curves validates the robust goal-conditioned discriminability.

The empirical scores confirm these theoretical properties (Fig. 6). Under the correct instruction, the value exhibits a progressive upward trend. Local peaks align precisely with key manipulation milestones: grasping the first shoe (t=81), placing it (t=120), placing the second shoe (t=263), and closing the lid (t=336 and t=383). The score rises from 0.28 at initialization to 0.57 as the shoebox closes, with major increases tightly coupled to semantic task progress rather than superficial visual changes or arbitrary arm motion. Conversely, the value under the mismatched instruction remains persistently low ($0.04 \sim 0.20$) and drifts downward, accurately reflecting that the executed shoe-grasping actions do not advance toward a soy-sauce or paper-cup goal at any frame. Notably, the mismatched score never intersects the correct instruction's curve. This persistent margin validates that CRL pre-training establishes a robust, zero-shot goal-reachability prior characterized by progress-sensitive value estimation and strict goal discriminability.

Pre-Training Efficiency (Q6). To prevent CRL token blocks and role-aware masking from creating computational bottlenecks, we implement the mask via a fused CuTe kernel. This incurs minimal overhead ($1.18\times$ standard FlashAttention-3 forward time), while end-to-end throughput maintains a near-linear 85.1% scaling efficiency across 64 H100 GPUs, making large-scale contrastive pre-training highly practical (Appendix I).

V. Conclusion and Limitations

We presented **PRTS**, a VLA foundation model that, for the first time, scales reward-label-free contrastive reinforcement learning into VLA pre-training itself. A language-conditioned contrastive objective shapes the backbone so that the pre-trained VLA encodes goal-reachability-aware representations, drawing supervision entirely from offline trajectory structure, without manual reward annotations or a separate value network (Sec. III-B). A set of system-level optimizations—most notably a role-aware causal mask fused into FlashAttention via a custom CuTe kernel, together with sequence packing and sharded contrastive similarity computation—folds the contrastive head into the same forward pass as behavior cloning at near-BC throughput, making CRL pre-training feasible at the parameter scale of current VLA backbones (Sec. IV-D).

Across LIBERO, LIBERO-Plus, LIBERO-Pro, SimplerEnv (WidowX), and real-world bimanual RealMan and single-arm Flexiv platforms, PRTS matches or exceeds the strongest prior VLAs at substantially smaller post-training compute, with the gap to baselines widening as evaluation moves further off the post-training distribution—novel instruction following, long-horizon execution, and human-intervention recovery (Sec. IV-B–IV-C). A direct examination of the value prediction further confirms that the same backbone produces a goal-reachability-aware and goal-discriminative score on unseen rollouts without any post-training adaptation (Sec. IV-D).

Returning to the premise that motivated this work—that robotic trajectory learning is inherently a goal-reaching process and that goal-conditioned RL has been shown to scale gracefully in pure decision-making domains [55]—these results extend that scaling story to VLA pre-training itself: scaling reward-free, language-conditioned CRL inside the VLA backbone delivers the goal-reachability awareness that not only substantially lifts overall task performance, but also gives rise to remarkably strong out-of-distribution generalization and robustness, most notably in zero-shot instruction following

(highlighted in Fig. 4) and recovery under human interventions (highlighted in Fig. 5). We view this kind of goal-reachability-aware behavior as an important milestone signaling that VLA pre-training is no longer just a more-capable imitator, but an emerging foundation for general-purpose, goal-directed robotic intelligence.

REFERENCES

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=vlhoswksBO.

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

[7] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[8] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[9] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

[10] Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning, 2024. URL https://arxiv.org/abs/2312.06722.

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[12] AgiBot World Colosseum contributors. Agibot world colosseum. https://github.com/OpenDriveLab/AgiBot-World, 2024.

[13] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

[14] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models, 2024. URL https://arxiv.org/abs/2409.17146.

[15] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

[16] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tc5qisoB-C.

[17] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

[18] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Train-

ing value functions via classification for scalable deep RL. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=dVpFKfqF3R.

[19] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[20] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[21] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

[22] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1724–1734, 2025.

[23] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL https://aclanthology.org/D14-1086.

[24] 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. URL https://openreview.net/forum?id=ZMnD6QZAE6.

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

[26] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

[27] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

[28] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[29] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

[30] 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.

[31] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

[32] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[33] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[34] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[35] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[36] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[37] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vl-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 976–983. IEEE, 2023.

[38] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

[39] NVIDIA, Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna

Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, Jacob Huffman, Jingyi Jin, Brendan Johnson, Rizwan Khan, George Kurian, Elena Lantz, Nayeon Lee, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Yen-Chen Lin, Ming-Yu Liu, Andrew Mathau, Yun Ni, Lindsey Pavao, Wei Ping, David W. Romero, Misha Smelyanskiy, Shuran Song, Lyne Tchapmi, Andrew Z. Wang, Boxin Wang, Haoxiang Wang, Fangyin Wei, Jiashu Xu, Yao Xu, Xiaodong Yang, Zhuolin Yang, Xiaohui Zeng, and Zhe Zhang. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

[40] Abby O'Neill, Abdul Rehman, 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, 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, Freek Stulp, and et al. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[42] 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. *arXiv preprint arXiv:2501.09747*, 2025.

[43] 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.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[45] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pages 1–16. IEEE, 2020.

[46] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mirchandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak Shafran, Brian Ichter, and Yuan Cao. Robovqa: Multimodal long-horizon reasoning for robotics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 645–652, 2024. doi: 10.1109/ICRA57147.2024.10610216.

[47] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37: 68658–68685, 2024.

[48] starVLA Contributors. Starvla: A lego-like codebase for vision-language-action model developing. GitHub repository, 1 2025. URL https://github.com/starVLA/starVLA.

[49] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

[50] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

[51] Yingbo Tang, Lingfeng Zhang, Shuyi Zhang, Yinuo Zhao, and Xiaoshuai Hao. Roboafford: A dataset and benchmark for enhancing object and spatial affordance learning in robot manipulation. *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025. URL https://api.semanticscholar.org/CorpusID:282395061.

[52] Spirit AI Team. Spirit-v1.5: Clean data is the enemy of great robot foundation models. *Spirit AI Blog*, 2026. https://www.spirit-ai.com/en/blog/spirit-v1-5.

[53] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[54] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[55] Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised rl: Scaling depth can enable new goal-reaching capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[56] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*,

2024.

[57] Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

[58] Siyuan Yang, Yang Zhang, Haoran He, Ling Pan, Xiu Li, Chenjia Bai, and Xuelong Li. Steering vision-language-action models as anti-exploration: A test-time scaling approach. *arXiv preprint arXiv:2512.02834*, 2025.

[59] Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

[60] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[61] Ted Zadouri, Markus Hoehnerbach, Jay Shah, Timmy Liu, Vijay Thakkar, and Tri Dao. Flashattention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. *arXiv preprint arXiv:2603.05451*, 2026.

[62] Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting vlms toward the embodied space. *arXiv preprint arXiv:2509.11766*, 2025.

[63] Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*, 2025.

[64] Yang Zhang, Chenwei Wang, Ouyang Lu, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, and Xuelong Li. Align-then-steer: Adapting the vision-language action models through unified latent guidance. *arXiv preprint arXiv:2509.02055*, 2025.

[65] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025. URL https://api.semanticscholar.org/CorpusID:277435005.

[66] Chongyi Zheng, Benjamin Eysenbach, Homer Rich Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive RL: Techniques for robotic goal reaching from offline data. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Xkf2EBj4w3.

[67] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[68] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.

[69] Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[70] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

APPENDIX

A. Contributions

B. Related Work

Visual-Language-Action Models. Early VLAs such as RT-1 [6], RT-2 [70], and OpenVLA [24] formulate policy learning as next-token prediction over discretized action tokens. While effective for action prediction, this objective provides only weak, indirect supervision for learning rich visual-semantic representations beyond behavior cloning. Therefore, current VLA training pipeline typically consists of large-scale pre-training over the mixture of action-labeled data and visual-reasoning data [5, 21, 3, 62], followed by task- or embodiment-specific post-training [25, 64, 58]. PRTS specifically tackles the representation bottleneck during the pre-training phase. To enrich the pre-training representations and further improve action generation, recent systems incorporate auxiliary VLM-side training objectives: $\pi_{0.5}$ [5] and GR00T N1 [3] add VQA, sub-task prediction, and discrete-action classification to the backbone, whereas RoboBrain [22], RoboVLM [29], and CoT-VLA [65] inject chain-of-thought reasoning and task decomposition. However, while these auxiliary objectives improve static visual-semantic grounding, they fail to encode the *temporal structure of goal-reaching*—how a task unfolds along a trajectory toward its language goal. Most closely related to our approach are value-augmented VLAs such as $\pi^*_{0.6}$ [21], VLAC [63], which attach separate value networks to affect action label supervision. A fundamental limitation of these approaches is that their value modules are architecturally decoupled from the VLM backbone. Consequently, gradients from value learning cannot directly shape the shared vision-language representation. In contrast, PRTS integrates contrastive reinforcement learning into VLM pre-training itself, allowing goal-conditioned value supervision to co-train the same representation used for both semantic reasoning and action generation. Moreover, existing value-augmented methods also rely on Monte Carlo (MC) returns or temporal-difference (TD) bootstrapping. MC estimation suffers from high variance in long-horizon, sparse-reward tasks, while TD learning accumulates function-approximation error [26, 27]. PRTS instead formulates value learning as a robust contrastive classification problem with temporally weighted positives, enabling stable and efficient learning of temporal-aware, goal-conditioned representations.

Contrastive Reinforcement Learning. Goal-conditioned reinforcement learning (GCRL) aims to learn policies that reach arbitrary target states. Classical approaches such as Hindsight Experience Replay [1] and C-Learning [16] address the sparse-reward challenge of GCRL through goal relabeling or classifier-based value estimation, yet still inherit the instability of TD bootstrapping under function approximation [49, 26]. Contrastive RL (CRL) [17] closes this gap by reformulating value learning as an InfoNCE classification problem [53]: the goal-conditioned value function under geometric discounting is recoverable purely by cross-entropy classification, without Bellman regression. Zheng et al. [66] further develops offline-data techniques that make CRL practical for robotic goal reaching. The central practical point is that CRL replaces the regression-based TD/MC loss with a cross-entropy classification loss, a swap that matters for scale. Wang et al. [55] shows that CRL trains stably over one thousand layers, a depth at which TD-regression losses provide no meaningful learning signal, and classification-based value estimation has been shown to scale better than regression across deep RL more broadly [18]. These two properties make CRL a particularly natural fit for VLA pre-training. (i) VLMs are natively optimized with a cross-entropy token-prediction objective, so CRL's classification loss composes with the backbone without introducing a regression head or auxiliary optimization regime. (ii) A VLA is itself a goal-conditioned RL problem: the language instruction specifies the goal, and the policy must reach states that satisfy it. PRTS builds on this alignment, integrating contrastive goal-reaching objectives directly into VLM pre-training so that semantic reasoning and value-aware planning are learned within a single cross-entropy framework.

C. Theorem 1 and Its Proof

1) Theoretical Equivalence to CRL.: We now establish that the $l \to s, a$ objective yields representations equivalent to standard CRL's discounted occupancy estimation.

Theorem 1 (Temporal Weighting Implements Geometric Sampling)**.** *Let π^* be the deterministic expert policy generating the demonstrations. For any state-action pair (s, a) in the dataset belonging to task with goal l, the optimal representations (ϕ^*, ψ^*) minimizing $\mathcal{L}^{l\to sa}$ satisfy:*

$$\psi^*(l)^\top \phi^*(s, a) = \log Q_l^{\pi^*}(s, a) + C(l),$$

where $Q_l^{\pi^*}(s, a) = p_\gamma^{\pi^*}(s_{t+} = s_g \mid s, a)$ is the discounted state occupancy measure (i.e., the probability of reaching goal s_g under π^* starting from (s, a)), and $C(l)$ is a function depending only on the goal l.

Theorem 1 reveals that while the weights q_{ij} are computed using temporal distances $(T - t)$, the learned representations do not merely encode 'how many steps remain.' Instead, they encode the *logarithm of the discounted occupancy measure* $\log Q_l^\pi(s, a)$, which is the fundamental quantity in goal-conditioned RL. Finally, we emphasize that although the derivation of the whole section is presented using single-step actions, the formulation naturally extends to action chunks used in VLAs, without changing the objectives or theoretical results.

2) Proof of Theorem 1.: We first formalize the $l \to$ sa objective with explicit expectation notation. Let $\mathcal{D}$ denote the data distribution over trajectories τ, and let $\mathcal{B} \sim \mathcal{D}^B$ denote a batch of B samples. For a given language goal l, let $\mathcal{S}(l; \mathcal{B}) = \{(s_j, a_j, t_j, T_j) \in \mathcal{B} : \text{task}(s_j) = l\}$ be the set of state-action pairs in the batch belonging to task l, where t_j is the timestep and T_j is the trajectory length.

The $l \to$ sa loss is:

$$\mathcal{L}^{l\to\text{sa}}(\phi,\psi) = \mathbb{E}_{\mathcal{B}\sim\mathcal{D}^B}\left[\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left(-\sum_{j\in\mathcal{S}(l_i;\mathcal{B})} q_{ij}\log\frac{\exp(\psi(l_i)^\top\phi(s_j,a_j))}{\sum_{k\in\mathcal{B}}\exp(\psi(l_i)^\top\phi(s_k,a_k))}\right)\right], \tag{10}$$

where the weights $q_{ij} = \frac{\gamma^{T_j-t_j}}{\sum_{j'\in\mathcal{S}(l_i;\mathcal{B})}\gamma^{T_{j'}-t_{j'}}}$ satisfy $\sum_{j\in\mathcal{S}(l_i;\mathcal{B})} q_{ij} = 1$.

Proof: Consider a fixed goal l and the subset of the batch $\mathcal{S}(l;\mathcal{B}) = \{j_1,\ldots,j_m\}$ containing m samples from this task. For this fixed l, the loss contribution is:

$$\mathcal{L}_l = -\sum_{r=1}^{m} q_r \log p_r, \tag{11}$$

where $p_r = \frac{\exp(\psi(l)^\top\phi(s_{j_r},a_{j_r}))}{\sum_{k\in\mathcal{B}}\exp(\psi(l)^\top\phi(s_k,a_k))}$ and $q_r = \frac{\gamma^{T_{j_r}-t_{j_r}}}{\sum_{r'}\gamma^{T_{j_{r'}}-t_{j_{r'}}}}$.

The cross-entropy $\mathcal{L}_l$ is minimized if and only if $p_r = q_r$ for all $r \in \{1,\ldots,m\}$. Taking logarithms:

$$\psi(l)^\top\phi(s_{j_r},a_{j_r}) - \log Z = \log q_r = (T_{j_r}-t_{j_r})\log\gamma - \log\left(\sum_{r'}\gamma^{T_{j_{r'}}-t_{j_{r'}}}\right), \tag{12}$$

where $Z = \sum_{k\in\mathcal{B}}\exp(\psi(l)^\top\phi(s_k,a_k))$ is the partition function (constant with respect to r).

Rearranging:

$$\psi(l)^\top\phi(s_{j_r},a_{j_r}) = (T_{j_r}-t_{j_r})\log\gamma + \underbrace{\left(\log Z - \log\sum_{r'}\gamma^{T_{j_{r'}}-t_{j_{r'}}}\right)}_{\triangleq C(l,\mathcal{B})}. \tag{13}$$

Now we establish the connection to the Q-function. In deterministic expert demonstrations, the policy π^* deterministically transitions toward the goal. The discounted occupancy measure (Definition in Section Preliminaries) becomes:

$$Q_l^{\pi^*}(s_t,a_t) = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k\mathbb{I}[s_{t+k}=s_g] \tag{14}$$

$$= (1-\gamma)\gamma^{T-t}, \tag{15}$$

where the second equality holds because under deterministic demonstrations, the goal is reached exactly at step T (so $\mathbb{I}[s_{t+(T-t)}=s_g]=1$ and all other terms are 0).

Therefore:

$$(T-t)\log\gamma = \log Q_l^{\pi^*}(s_t,a_t) - \log(1-\gamma). \tag{16}$$

Substituting back:

$$\psi(l)^\top\phi(s,a) = \log Q_l^{\pi^*}(s,a) + \underbrace{(C(l,\mathcal{B}) - \log(1-\gamma))}_{\triangleq C(l)}. \tag{17}$$

Taking expectation over the batch sampling $\mathcal{B}$, the batch-dependent constant $C(l,\mathcal{B})$ converges to a deterministic function $C(l)$ of the goal l alone, yielding the desired result. ■

Comparison with Standard CRL. Standard CRL [17] achieves $f^*(s,a,g) = \log\frac{p_\gamma^\pi(s_{t+}=g|s,a)}{p(g)} + c(s,a)$ by geometrically sampling future states as goals. This creates an *implicit* weighting: states k steps ahead are sampled with probability $(1-\gamma)\gamma^{k-1}$, naturally emphasizing near-term futures.

Our temporal weighting scheme achieves the *same* optimal solution but through *explicit* soft labels $q_{ij} \propto \gamma^{T-t}$. The key differences are:

- **Sampling vs. Weighting:** CRL samples one positive per anchor (geometrically); we weight all available positives in the batch.
- **Goal space:** CRL requires a continuous, sampleable goal space (future states); ours works with discrete, fixed goals (language instructions).
- **Variance:** Our multi-positive approach reduces variance by utilizing all positive samples in the batch, rather than a single geometric sample.

Both approaches drive the representation inner product toward $\log p_\gamma^\pi(s_{t+}=g|s,a)$, confirming that temporal weighting is the appropriate adaptation of CRL to the language-goal setting.

D. Pre-Training Dataset Details

To construct the PRTS visuomotor prior, we aggregate a large-scale multi-modal pre-training dataset comprising **404 M samples** totaling **167.8 B tokens**. As illustrated in Fig. 7, the dataset is partitioned into two complementary domains: *Action-Labeled Data* for cross-embodiment physical execution priors, and *Visual-Reasoning Data* for semantic and spatial grounding.

1) Action-Labeled Data: To translate visual understanding into robust physical execution, we build an extensive collection of embodied action-labeled trajectories. Characterized by heterogeneous hardware and long-horizon tasks, the action-labeled subset is primarily aggregated from four sources:

- **AgiBotWorld** [12]. Comprising over one million trajectories across 217 tasks in five deployment scenarios, AgiBotWorld covers contact-rich manipulation, long-horizon planning, and dual-arm collaboration equipped with end-effectors from grippers to dexterous hands. It also provides fine-grained sub-task annotations.
- **RoboMind** [56]. As a unified teleoperation dataset, RoboMind provides demonstration trajectories across 479 diverse tasks involving 96 object classes. It enriches the physical priors by providing standardized, high-quality trajectories across diverse scenes and complementing AgiBotWorld in task breadth.
- **Open X-Embodiment** [40]. The Open X-Embodiment Dataset encompasses 22 robotic embodiments across distinct laboratory and real-world environments, serving as the primary source for cross-embodiment generalization priors.
- **PRTS Self-Collected Dataset.** Collected on the dual-arm RealMan and Agilex platforms, this subset covers daily manipulation tasks. It introduces multi-scene scenarios that require precise, contact-rich operations, further diversifying the kinematic training distribution.

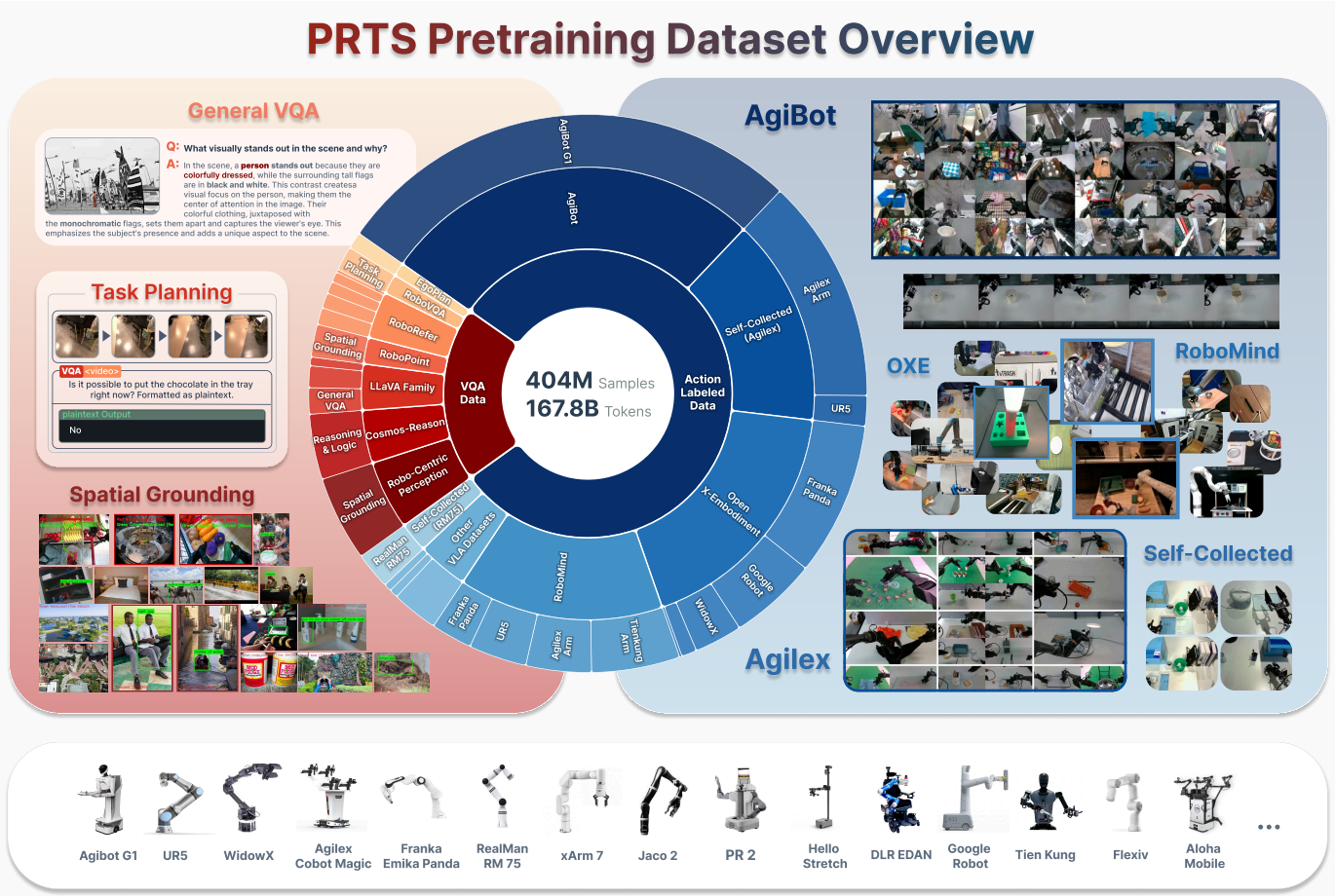


Fig. 7: **Pre-training dataset overview of PRTS.** Our pre-training dataset is partitioned into two primary domains: *Action-Labeled Data* provides physical execution priors for cross-embodiment manipulation, while *Visual-Reasoning Data* provides complementary supervision for spatial grounding, affordance perception, and task planning. The area of each segment is scaled proportionally to the square root of its total tokens to reflect the structural distribution of the data sources.

2) Visual-Reasoning Data: The visual-reasoning subset supplies grounding, planning, and general vision–language competence that is often under-represented in action-labeled demonstrations. We organize this reasoning corpus into three hierarchical capability tiers:

Spatial Grounding and Affordance Perception. To support pixel-level localization and operable region identification, we utilize **RefCOCO** [23] for fundamental referring expression comprehension, pairing natural language descriptions with precise 2D bounding boxes. To further enhance fine-grained localization, we incorporate **Pixmo-Point** [14], which employs a Point-QA paradigm, requiring the model to output precise 2D pixel coordinates rather than abstract text. This localization enables fine-grained visual grounding, such as pointing-based counting and visual explanation.

To bridge the domain gap to embodied workspaces, we incorporate **RoboPoint** [60] for spatial-relation grounding alongside standard object detection. We further include **RoboRefIt** [37] (VL-Grasp) to resolve referential ambiguity in cluttered environments, encouraging the model to track correct grasping targets under complex relative spatial relations. We also incorporate **RefSpatial**[1] [68] to improve 3D scene understanding and enable dynamic reasoning about instruction-specified locations for interaction. Moreover, dexterous manipulation inherently requires understanding functional object parts. To this end, we incorporate **RoboAfford** [51] to endow the model with physical affordance perception, enabling it to distinguish operable regions (e.g., the handle of a knife versus the blade).

Task Planning and High-level Reasoning. Long-horizon execution requires temporal foresight and physical common sense. Therefore, we incorporate **Cosmos-Reason1** [39] to provide dense reasoning chains (Chain-of-Thought) regarding physical causality and scene dynamics. To support task decomposition, we leverage **EgoPlanIT** [10] to train the model in decomposing abstract human instructions into sequential, actionable sub-goals from an egocentric perspective. Additionally, **RoboVQA** [46] is included to enhance step-by-step temporal reasoning and context awareness over continuous embodied video streams.

General Semantic Understanding. To retain general

[1] Introduced in RoboRefer.

vision–language competence during large-scale embodied pre-training, we include a curated, high-quality subset of **LLaVA-Instruct** [35]. This subset acts as a semantic regularizer, preventing catastrophic forgetting and preserving zero-shot language-following abilities within the VLA backbone.

E. Complete Simulation Results and Baseline Details

This appendix provides the comprehensive evaluation results across all simulated benchmarks, fully restoring the baseline comparisons and detailed textual analyses that were condensed in Sec. IV-B of the main text.

1) Complete LIBERO Benchmark Suite: Table VI reports the complete results across the four standard LIBERO suites (*Spatial*, *Object*, *Goal*, and *Long*) against a total of 16 prior baselines, expanding upon the abridged summary in the main text.

As established in the main text's controlled-budget analysis, decoupling representation quality from downstream post-training compute is vital for assessing true visuomotor priors. Prior large-scale VLAs often require extensive fine-tuning or large optimization windows to achieve robust temporal and spatial reasoning. For instance, $\pi_{0.5}$ relies on an $8\times$ larger batch size to reach 96.9% average success rate (SR). When we restrict the primary baseline, ABot-M0, to our identical minimal budget (global batch size of 32 and 30K gradient steps), it reaches 97.9% but experiences a severe performance degradation on the temporally extended *Long* suite (dropping to 94.8%). In contrast, PRTS natively maintains a SOTA success rate of 96.6% on this suite under the same minimal budget. This indicates that the goal-reachability-aware representation learned through contrastive RL pre-training is especially critical for long-horizon manipulation, reducing the reliance on large-scale downstream optimization for temporal reasoning.

2) Complete LIBERO-Plus Robustness Evaluation: LIBERO-Plus [19] serves as a robustness-oriented framework incorporating $10{,}030$ test tasks to evaluate policy execution stability under explicit distribution shifts. The benchmark introduces seven controlled perturbation axes: shifted camera viewpoints (*Camera*), robot initial state variations (*Robot*), LLM-rewritten instructions (*Language*), lighting intensity fluctuations (*Light*), background texture modifications (*Background*), camera sensor noise (*Noise*), and macro object spatial layouts (*Layout*).

Table VII presents the complete baseline results. Evaluating policies zero-shot on these axes directly probes whether the fine-tuned visuomotor prior anchors task semantics to invariant features or overfits to nominal training distributions. PRTS consistently outperforms large-scale baselines, suggesting that contrastive RL pre-training provides a visually and semantically robust goal prior.

3) Complete LIBERO-Pro Generalization Evaluation: LIBERO-Pro [69] is specifically structured to expose mechanical trajectory memorization. By swapping the relative operational coordinates of pick-and-place items (*Position*) or changing the linguistic task target entirely within a static workspace (*Task*), it prevents models from relying on open-loop demonstration replays.

Table VIII reports the comprehensive baseline comparison. PRTS's strong absolute margins across these semantic axes validate its capability to synthesize goal-reachability-aware actions under strict task-compositional variations.

4) Complete SimplerEnv (WidowX) Cross-Embodiment Evaluation: SimplerEnv [30] evaluates sim-to-real transfer via a WidowX robotic embodiment across diverse rigid-object manipulation behaviors, providing an essential metric for cross-embodiment adaptability. Table IX supplies the complete dataset, demonstrating PRTS's robust generalizability to novel physical hardware.

F. Real-World Evaluation: Task Descriptions and Setup

This appendix provides the natural-language task definitions and rollout illustrations for the real-world evaluation suite. The robot platform configurations and the trial-level evaluation protocol are described in Sec. IV-A; here we focus on the task content, object arrangements, and the manipulation behavior required by each instruction.

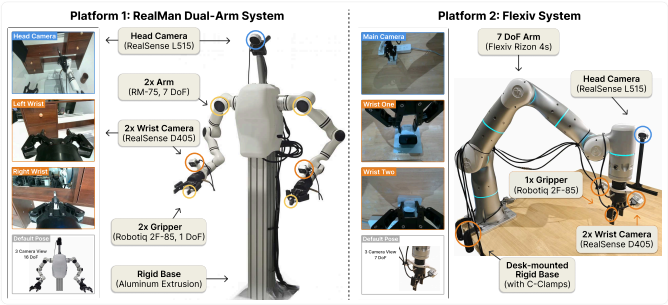


Fig. 8: **Overview of real-world robot platforms.** Left: dual-arm RealMan platform with two parallel-jaw grippers, one egocentric head camera, and two wrist cameras. Right: Flexiv single-arm platform equipped with a parallel-jaw gripper, a third-view head camera, and two wrist cameras.

1) RealMan dual-arm tasks: The RealMan suite is designed to test bimanual coordination, instruction following, fine-grained contact, object transfer, and long-horizon sequencing in tabletop and office-like scenes. Figure 9 visualizes representative rollouts.

Paper Rubbish. The scene contains a trash can and a piece of crumpled paper. The left gripper holds the trash can while the right gripper grasps the paper and drops it into the can.

Place Block. The robot grasps a block with the left gripper and places it at the specified location on the tea tray.

Serve Tea. The robot grasps a teacup with the left gripper and places it on the green coaster.

Flip Tennis Tube. The task begins with a tennis-ball tube on the workspace. The left gripper first grasps the tube, after which the right gripper takes the tube and places it at the designated position on the tray.

Hand Over. The task requires a bimanual handover of a pen holder. The left gripper grasps the pen holder, and the right

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy [11]	78.5	87.5	73.5	64.8	76.1
OpenVLA [24]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [43]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [65]	87.5	91.6	87.6	69.0	83.9
GR00T-N1 [3]	94.4	97.6	93.0	90.6	93.9
F1-VLA [38]	98.2	97.8	95.4	91.3	95.7
InternVLA-M1 [9]	98.0	99.0	93.8	92.6	95.9
Discrete Diffusion VLA [31]	97.2	98.6	97.4	92.0	96.3
GR00T-N1.6 [3]	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT [25]	97.6	98.4	97.9	94.5	97.1
X-VLA [67]	98.2	98.6	97.8	**97.6**	98.1
π_0-Fast [42] (bs=32, 30K steps)	96.4	96.8	88.6	60.2	85.5
π_0 [4] (bs=32, 30K steps)	96.8	98.8	95.8	85.2	94.2
Qwen3-VL-PI [48] (bs=32, 30K steps)	95.2	99.0	96.2	88.4	94.7
$\pi_{0.5}$ [5] (bs=256, 30K steps)	**98.8**	98.2	98.0	92.4	96.9
ABot-M0 [59] (bs=32, 30K steps)	98.6	99.6	98.6	94.8	97.9
ABot-M0 [59] (bs=64, 40K steps)	**98.8**	**99.8**	**99.0**	96.6	**98.6**
PRTS (Ours) (bs=32, 30K steps)	**98.8**	**99.8**	98.4	96.6	98.4

TABLE VI: **Complete LIBERO evaluation.** Post-training budgets are indicated in gray next to model names. Best results are in **bold**; second-best are underlined.

Method	Camera	Robot	Lang.	Light	Backg.	Noise	Layout	Avg.
OpenVLA [24]	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
WorldVLA [8]	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
NORA [20]	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA [7]	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
RIPT-VLA [50]	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT [25]	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
π_0 [4] (bs=32, 30K)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast [42] (bs=32, 30K)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
Qwen3-VL-PI [48] (bs=32, 30K)	43.6	46.6	81.8	88.4	89.4	70.1	74.7	68.9
ABot-M0 [59] (bs=32, 30K)	**65.7**	53.1	81.2	**96.8**	95.0	**87.5**	81.3	78.7
$\pi_{0.5}$ [5] (bs=256, 30K)	64.0	58.0	88.5	96.6	81.4	**87.5**	**85.9**	80.7
PRTS (Ours) (bs=32, 30K)	59.5	**72.4**	**89.6**	96.5	**96.7**	81.3	83.3	**81.4**

TABLE VII: **Complete LIBERO-Plus evaluation.** Zero-shot robustness across seven perturbation axes [19].

Method	Semantic	Object	Position	Task	Average
OpenVLA [24]	**97.2**	93.0	0.0	0.0	47.6
Qwen3-VL-PI [48] (bs=32, 30K steps)	95.5	70.4	4.3	3.9	43.5
ABot-M0 [59] (bs=32, 30K steps)	97.1	82.5	7.1	22.3	52.2
π_0 [4] (bs=32, 30K steps)	90.5	90.5	0.0	0.0	45.3
$\pi_{0.5}$ [5] (bs=256, 30K steps)	95.8	**96.0**	20.8	0.8	53.3
PRTS (Ours) (bs=32, 30K steps)	97.0	82.3	**24.3**	**31.5**	**58.8**

TABLE VIII: **Complete LIBERO-Pro evaluation.** Zero-shot generalization across semantic, object, position, and task variations [69].

Method	Put Spoon On Towel	Put Carrot On Plate	Stack Green Block On Yellow Block	Put Eggplant In Yellow Basket	Average
OpenVLA [24]	0.0	0.0	0.0	4.1	1.0
RT-1-X [40]	0.0	4.2	0.0	0.0	1.1
OpenVLA-OFT [25]	34.2	30.0	30.0	72.5	41.8
SpatialVLA [43]	16.7	25.0	29.2	**100.0**	42.7
CogACT [28]	71.7	50.8	15.0	67.5	51.3
F1-VLA [38]	50.0	**70.8**	50.0	66.7	59.4
Qwen3-VL-PI [48]	78.1	46.9	30.2	88.5	60.9
GR00T-N1.5 [3]	75.3	54.3	57.0	61.3	61.9
π_0 [4]	29.1	0.0	16.6	62.5	27.1
$\pi_{0.5}$ [5]	29.2	41.7	0.0	41.7	28.2
π_0-Fast [42]	29.1	21.9	10.8	66.6	48.3
PRTS (Ours)	**83.3**	66.6	**58.3**	**100.0**	**77.1**

TABLE IX: **Complete results on SimplerEnv WidowX benchmark (Visual Matching).** Real-to-sim evaluation [30].

Method	Spatial		Object		Goal		Long	
	Pos	Task	Pos	Task	Pos	Task	Pos	Task
OpenVLA [24]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen3-VL-PI [48] (bs=32, 30K)	14.4	0.0	0.2	0.0	1.2	8.8	1.2	6.8
ABot-M0 [59] (bs=32, 30K)	13.4	54.8	9.0	9.8	5.6	10.6	0.2	14.0
π_0 [4] (bs=32, 30K)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\pi_{0.5}$ [5] (bs=256, 30K)	20.0	1.0	17.0	1.0	**38.0**	0.0	8.0	1.0
PRTS (Ours) (bs=32, 30K)	**26.8**	**62.2**	**36.0**	8.8	19.0	**33.8**	**15.4**	**21.2**

TABLE X: **Detailed breakdown on the *Position* and *Task* variations of LIBERO-Pro.** Values report success rates (%).

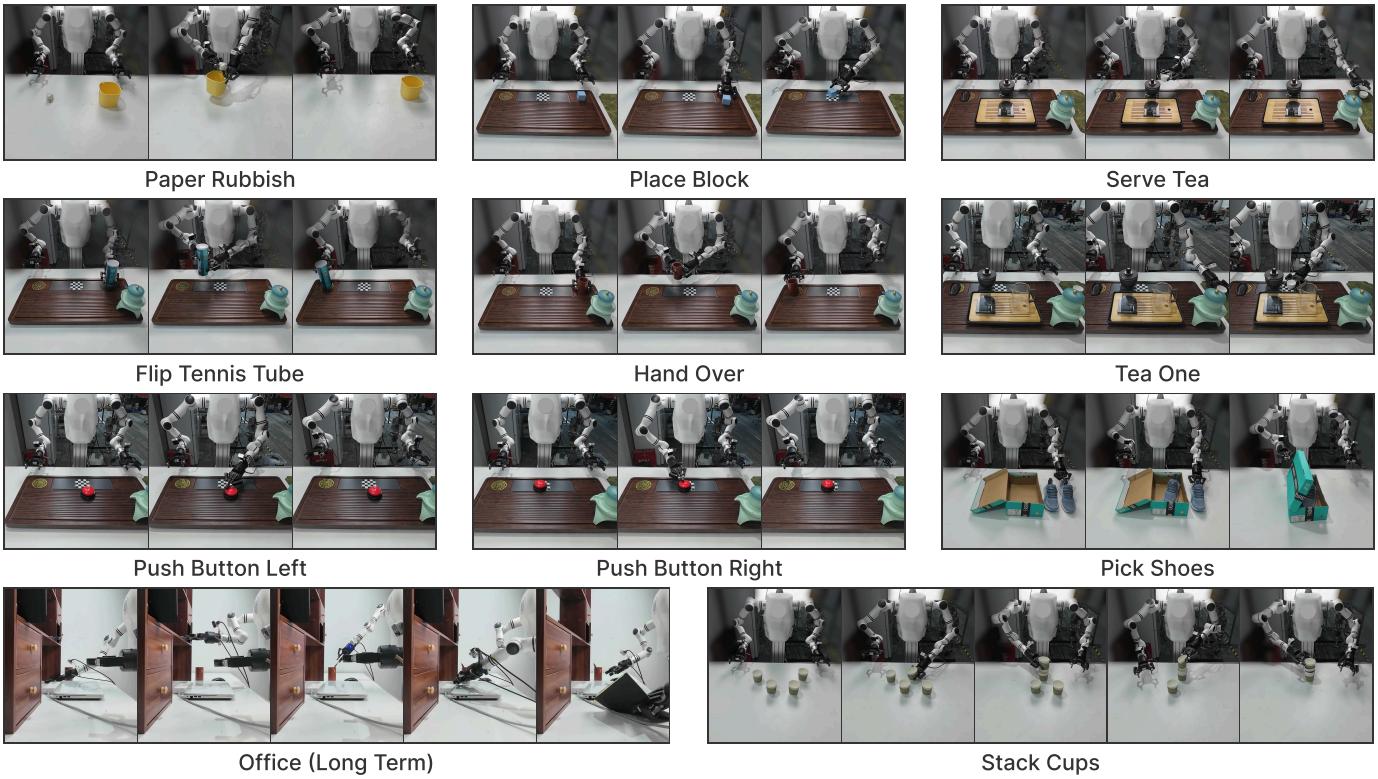


Fig. 9: **RealMan task suite.** We evaluate PRTS on eleven RealMan tasks, including ten short-horizon desktop manipulation tasks and one long-horizon office task. The montage visualizes representative rollouts for the task settings described in Appendix F.

Task	π_0	$\pi_{0.5}$	PRTS
Flip Tennis Tube	30.0	85.0	90.0
Hand Over	45.0	80.0	95.0
Office Long Term	5.0	40.0	95.0
Push Button Left	95.0	100.0	100.0
Push Button Right	100.0	100.0	100.0
Paper Rubbish	65.0	90.0	100.0
Place Block	90.0	100.0	100.0
Pick Shoes	90.0	90.0	95.0
Stack Cups	45.0	70.0	90.0
Serve Tea	90.0	90.0	95.0
Tea One	85.0	95.0	95.0
OVERALL	**67.3**	**85.5**	**95.9**

TABLE XI: **RealMan dual-arm evaluation.** Per-task success rates over 20 real-world trials.

gripper receives it and places it at the fixed target position on the tea tray.

Tea One. The robot grasps the white teacup on the coaster with the left gripper, places it at the fixed target position on the tea table, and then returns the left gripper to an initial-like pose.

Push Button Left. The robot uses the left gripper to press the target button.

Push Button Right. The robot uses the right gripper to press the target button.

Pick Shoes. The robot organizes a pair of shoes into a shoebox. The left gripper sequentially picks up two shoes and places them inside the box, and the right gripper then closes the shoebox.

Stack Cups. The robot stacks four paper cups into a vertical stack through alternating bimanual actions. The left gripper places the first cup on the table, the right gripper stacks the second cup on top, the left gripper stacks the third cup, and the right gripper completes the stack with the fourth cup.

Office (Long Term). This task is a long-horizon office organization sequence with multiple objects and interaction types. The robot first picks up a wireless mouse and places it on the desk, then picks up a pen from the bookshelf with the right gripper, inserts it vertically into the pen holder, and releases it. It then presses the power strip with the left gripper, presses the switch with the right gripper, takes a book from the bookshelf with the left gripper, and places the book on the left side of the desk.

2) Flexiv single-arm tasks: The Flexiv suite contains three tasks requiring sub-millimeter geometric placement and contact-rich manipulation (Fig. 10).

Block Combination. The robot picks up the black square block with four corner holes and places it onto the white four-post alignment fixture, requiring the plate holes to align with the fixture posts.

Gear Assembly. The robot sequentially mounts a small, medium, and big black gear onto the left, middle, and right posts of a white fixture, keeping all center holes perfectly aligned.

Flower Arrangement. The robot executes a sequential pick-and-place operation, retrieving the blue and red flowers from the work surface via their stems and depositing them into the glass receptacle.

G. Real-World Evaluation: Controlled Generalization Shifts

This section provides detailed qualitative observations for the *Task* generalization evaluations reported in Sec. IV-C (Table XII). The task-instruction axis presents a uniquely demanding challenge: it requires the policy to dynamically retrieve and recombine manipulation skills acquired during post-training, correctly grounding them within a novel language instruction and physical scene. Below, we dissect the model behaviors across the four representative task shifts:

Paper Rubbish (Target Shift): In this variation, the original trash-can target (crumpled paper) is substituted with a flat wireless mouse from a separate long-horizon office task. This substitution demands a qualitatively different, deeper, and highly planar grasp. Prior models ($\pi_0, \pi_{0.5}$) tend to default to the original crumpled-paper grasping pattern, failing to establish a stable grip on the flat mouse. In contrast, PRTS successfully transfers the planar mouse-grasping skill into the new task context, achieving an 80% success rate.

Place Block (Distractor & Affordance Shift): Here, the target object is changed to a soy-sauce bottle requiring a grasp at a narrow, stable region, while both the new bottle and the original block remain present in the workspace. This rigorously tests whether the policy reliably grounds its actions to the language-specified object rather than defaulting to the scene's habitual affordance (the block). PRTS achieves 55% success. Conversely, π_0 largely ignores the novel instruction and continues to grasp the original block, while $\pi_{0.5}$ often attends to the correct bottle but drops it due to kinematically unadapted grasp placements.

Pick Shoes (Morphological Shift): The two shoes are replaced by a soy-sauce bottle and a paper cup, yielding a visually distinct scene that necessitates completely different, object-specific grasp strategies before closing the box. Faced with this shifted context, π_0 exhibits rigid open-loop behavior, moving its end-effector to the original shoe locations and stalling. While $\pi_{0.5}$ can occasionally pick the new objects, it frequently fails the subsequent reasoning step of placing them into the box. PRTS, however, adaptively completes the recomposed sequence with an 85% success rate.

Stack Cups (Sequential Cue Deletion): In this temporal shift, the final paper-cup stacking step is replaced by an instruction to drop a crumpled paper ball into the third cup. Crucially, this removes a key visual cue (the lip of the fourth cup) typically relied upon for localizing the next stacking target, forcing the policy to precisely track its current execution stage in the absence of standard visual anchors. PRTS reaches 75% success, whereas prior baselines exhibit severe precision degradation and compounding errors even during the earlier, unperturbed cup-stacking stages.

H. Ablation Study: Effect of Contrastive RL

After establishing PRTS's overall performance advantages, we examine the precise contribution of the contrastive RL

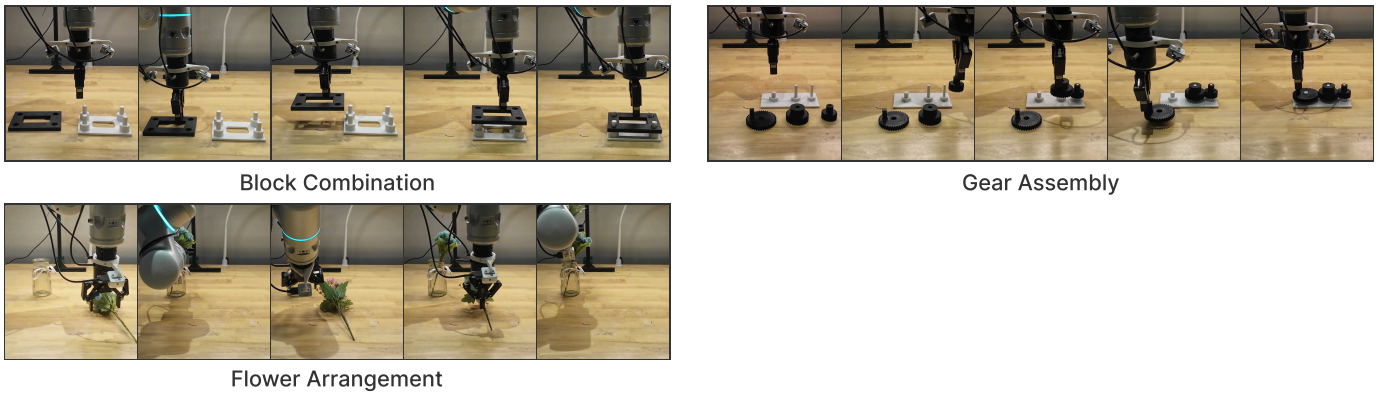


Fig. 10: **Flexiv task suite.** The Flexiv evaluation complements the RealMan suite with single-arm manipulation tasks that emphasize precise alignment, assembly, and insertion-like object placement.

Method	**Paper Rubbish**				**Place Block**				**Pick Shoes**				**Stack Cups**			
	Light	Pos	Obj	Task	Light	Pos	Obj	Task	Light	Pos	Obj	Task	Light	Pos	Obj	Task
π_0	60.0	50.0	60.0	20.0	45.0	40.0	45.0	5.0	80.0	80.0	70.0	0.0	35.0	10.0	0.0	30.0
$\pi_{0.5}$	90.0	85.0	95.0	35.0	100.0	100.0	100.0	15.0	40.0	70.0	85.0	40.0	45.0	25.0	5.0	50.0
PRTS (Ours)	100.0	100.0	100.0	80.0	100.0	100.0	100.0	55.0	95.0	95.0	95.0	85.0	90.0	90.0	80.0	75.0

TABLE XII: **Controlled RealMan generalization on the RealMan dual-arm platform.**

objective utilized during pre-training. To isolate this effect, we pre-train a matched variant of PRTS using an identical architecture, dataset, and optimization recipe (Sec. IV-A), except the CRL coefficient in Eq. (9) is zeroed out ($\lambda_{\text{crl}} = 0.0$). Following pre-training, both PRTS and its non-CRL variant are equipped with the identical flow-matching action expert and post-trained on standard LIBERO with a minimal budget (global batch size 32, 30K gradient steps). We evaluate these checkpoints on standard LIBERO and further deploy them zero-shot on LIBERO-Plus and LIBERO-Pro without any benchmark-specific adaptation. The aggregated results are reported in Table XIII.

On standard LIBERO (in-distribution), CRL yields a modest improvement in the average success rate ($97.8\% \rightarrow 98.4\%$). The most notable gains manifest on the *Spatial* and *Long* suites ($+1.0\%$ each), suggesting that CRL-shaped representations inherently benefit relational spatial grounding and temporally extended execution.

However, the architectural advantage of CRL crystallizes under zero-shot evaluation on LIBERO-Plus. Here, CRL improves the average SR by 4.9%, driving immense gains on *Robot* initialization shifts ($+16.3\%$) and sensor *Noise* ($+9.0\%$), alongside steady improvements on *Camera* ($+2.8\%$) and *Language* ($+2.3\%$) axes. Under these perturbations, successful execution cannot rely on open-loop trajectory replays; the policy must preserve the language-conditioned goal and dynamically re-ground it against a shifted perceptual stream. By explicitly aligning state representations with high-level task semantics during pre-training, CRL equips the policy with this exact capability. Consequently, PRTS w/ CRL exhibits profound robustness when the nominal observation-to-action mapping is disturbed, confirming our hypothesis that contrastive pre-training inherently strengthens the goal-conditioned prior leveraged by the downstream action expert.

The LIBERO-Pro evaluation further reinforces this paradigm by probing the model's capacity for zero-shot generalization under severe semantic and structural recombinations. Here, CRL yields a 5.0% overall increase, driven almost entirely by the highly demanding *Position* ($+10.5\%$) and *Task* ($+11.4\%$) axes. Unlike lower-level visual noise, these dimensions demand cognitive flexibility: they require the policy to actively reinterpret redefined instruction goals, identify novel target relations, and synthesize new action sequences. The suite-level breakdown in Table XIV provides a granular view of this mechanism. CRL effectively neutralizes the catastrophic degradation observed in the non-CRL variant under complex re-grounding scenarios. For instance, when evaluating novel spatial configurations (*Object-Position*), SR surges from 4.8% to 36.0%. Furthermore, CRL enables seamless adaptation to entirely new task logics, improving *Spatial-Task* and *Goal-Task* performance from 34.4% to 62.2% and 20.2% to 33.8%, respectively.

While the LIBERO-Plus results demonstrate that CRL provides robustness when underlying task semantics are fixed, the LIBERO-Pro evaluation reveals a deeper advantage: the capability to generalize when semantics are fundamentally altered. Because contrastive pre-training anchors state representations to goal reachability rather than specific visual trajectories, the policy avoids overfitting to nominal layouts. Consequently, PRTS w/ CRL can dynamically extrapolate learned behaviors

Benchmark	Suite / Perturbation	PRTS w/o CRL	PRTS w/ CRL	Δ
LIBERO	Spatial	97.8	98.8	+1.0
	Object	99.8	99.8	0.0
	Goal	98.0	98.4	+0.4
	Long	95.6	96.6	+1.0
	Average	**97.8**	**98.4**	**+0.6**
LIBERO-Plus	Camera	56.7	59.5	+2.8
	Robot	56.1	72.4	+16.3
	Language	87.3	89.6	+2.3
	Light	96.4	96.5	+0.1
	Background	95.8	96.7	+0.9
	Noise	72.3	81.3	+9.0
	Layout	82.9	83.3	+0.4
	Average	**76.5**	**81.4**	**+4.9**
LIBERO-Pro	Semantic	96.8	97.0	+0.2
	Object	84.5	82.3	-2.2
	Position	13.8	24.3	+10.5
	Task	20.1	31.5	+11.4
	Average	**53.8**	**58.8**	**+5.0**

TABLE XIII: **Effect of contrastive RL pre-training across the LIBERO benchmark family.** Both variants are post-trained identically on standard LIBERO and evaluated zero-shot on LIBERO-Plus and LIBERO-Pro. Success rates (%) demonstrate that while CRL offers modest gains in-distribution, it drives massive improvements under physical and semantic distribution shifts. (Δ: PRTS w/ CRL minus PRTS w/o CRL).

Method	LIBERO-Spatial		LIBERO-Object		LIBERO-Goal		LIBERO-Long	
	Pos (Avg.)	Task (Avg.)	Pos (Avg.)	Task (Avg.)	Pos (Avg.)	Task (Avg.)	Pos (Avg.)	Task (Avg.)
PRTS w/o CRL	26.8	34.4	4.8	10.0	13.0	20.2	10.4	15.6
PRTS w/ CRL	26.8	62.2	36.0	8.8	19.0	33.8	15.4	21.2
Δ	0.0	+27.8	+31.2	-1.2	+6.0	+13.6	+5.0	+5.6

TABLE XIV: **Suite-level performance breakdown on LIBERO-Pro.** Success rates (%) are decomposed by the original LIBERO suites to illustrate performance under *Position* (spatial adaptability) and *Task* (procedural generalization) distribution shifts. The massive Δ highlights CRL's capability to generalize when underlying task semantics are fundamentally altered.

to novel spatial and procedural configurations. This further validates that our approach equips the downstream action expert with a highly composable representation space, enabling true procedural generalization beyond mere robustness.

To further explore why CRL improves robustness and instruction-conditioned generalization, we visualize the state-action representations extracted from the pretrained backbone using t-SNE [54]. Specifically, we sample trajectories from ten representative manipulation tasks in the RoboMind dataset covering diverse skills and effector. From each episode, we extract non-overlapping temporal chunks. Then for each chunk, we feed the corresponding observation, proprioceptive state, language instruction, and action tokens into the pretrained backbone, and use the hidden state at the `<|action_end|>` token as the state-action representation since this token encodes the full multimodal context of the sequence. These representations are extracted separately from the PRTS w/ CRL and PRTS w/o CRL models and projected into a unified t-SNE space. The obtained representations are shown in Fig. 11.

The inclusion of CRL organizes the embedding space around reusable manipulation primitives. Samples corresponding to *Hand-Pick*, *Gripper-Pick*, *Hand-Open*, and *Hand-Close* form tightly clustered, well-separated neighborhoods. This indicates that CRL successfully pulls together state-action pairs that share similar goal-reaching semantics, while pushing apart trajectories associated with divergent outcomes. In contrast, the representations from the non-CRL variant remain highly entangled and diffuse. This suggests that pure behavioral cloning is prone to overfitting to trajectory-specific correlations, rather than capturing the shared, goal-reaching behavioral structure across demonstrations. This structural alignment explains the performance gains across the LIBERO benchmark family. Under the perceptual shifts of LIBERO-Plus, this

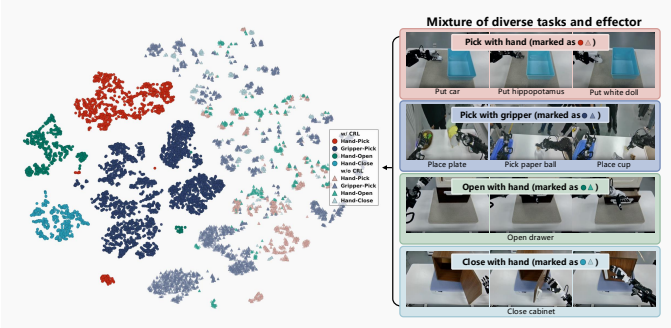


Fig. 11: **T-SNE visualizations of learned task representations.**

cleanly separated space allows the policy to robustly maintain the intended primitive. Furthermore, under the severe semantic shifts of LIBERO-Pro, this composable representation equips the policy to accurately ground novel instruction goals and dynamically synthesize required behaviors, entirely bypassing rote memorization.

I. System Analysis: Pre-Training Efficiency

A natural concern is whether adding CRL token blocks and role-aware masking makes large-scale VLA pre-training prohibitively expensive. We show that the answer is no: with the fused CuTe kernel, the additional cost stays close to the FlashAttention 3 (FA3) baseline at the attention level, and thus end-to-end training maintains near-linear throughput scaling up to 64 H100 GPUs.

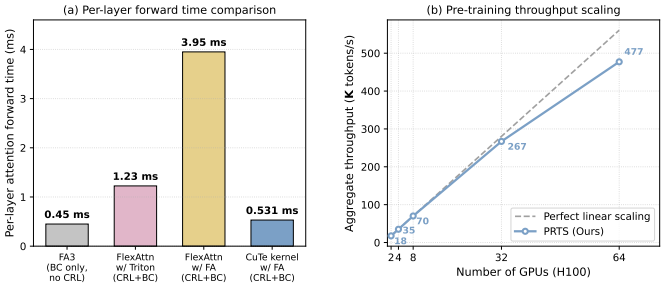


Fig. 12: **Pre-training efficiency of PRTS. (a)** Per-layer attention forward time on 2 H100 GPUs for BC-only FA3, two FlexAttention backends, and the fused CuTe kernel. **(b)** Aggregate token throughput vs. number of H100 GPUs; dashed line: linear scaling from the 2-GPU point.

Sec. III-B introduces the role-aware attention mask for the CRL token blocks and its CuTe-kernel implementation on top of FlashAttention [61], which applies the mask at block granularity. Here we quantify its efficiency from two perspectives: a kernel microbenchmark that isolates per-layer attention time under fixed packing on 1 H100 GPU, and a throughput study that measures aggregate training tokens/s when the same per-device packing configuration is scaled across $\{2, 4, 8, 32, 64\}$ H100 GPUs using DeepSpeed ZeRO-2 distributed strategy [45].

Kernel microbenchmark. All rows in Fig. 12a share the same setup: a local batch with size $bs = 4$ and packed sequence length 4096. We compare (i) regular BC forward using FA3 without the two CRL token blocks, (ii–iii) FlexAttention with Triton and FlashAttention as backends implementing the joint CRL + BC forward with the role-aware mask, and (iv) our fused CuTe kernel that feeds FlashAttention a block-sparse layout for the CRL blocks. Notably, the per-layer attention forward time of the fused kernel is only $1.18\times$ of the FA3 reference while enabling the full CRL objective optimization. The large gap between FlexAttention and CuTe mainly comes from how the five-role mask is implemented: expressing it through a general flexible-attention path introduces substantial overhead, while integrating the sparsity pattern directly into the fused kernel keeps the cost close to standard BC training.

Throughput scaling. Fig. 12b plots aggregate tokens/s against GPU count for the full training step, using 4 packed samples per GPU and DeepSpeed ZeRO-2 strategy as a controlled sweep of the distributed training run. Against a linear reference anchored at the 2-GPU point, scaling efficiency remains at almost 100% through 8 GPUs, 95.2% at 32, and 85.1% at 64. The drop at larger scale matches the expected communication overhead of ZeRO-2 rather than CRL-specific computation.

Together, these results show that CRL adds little overhead when implemented with the fused role-aware kernel: per-layer attention time remains close to the FA3 baseline, and end-to-end throughput scales nearly linearly up to 64 H100 GPUs. This makes large-scale contrastive pre-training practical without sacrificing pre-training efficiency.