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], Sen Fu[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*. This requires the representations inside a general-purpose robotic policy to not only encode static visual-linguistic semantic correlations, but also encapsulate the underlying *temporal progress toward ultimate goal*—specifically, how close it is to achieving the goal at the current state. While VLMs excel at encoding static semantic similarities between scenes and concepts [24, 31], they intrinsically lack any notion of *goal reachability*—a quantitative estimate of how likely a given state-action pair is to eventually achieve the language-specified objective.

Existing VLAs learn visuomotor priors almost exclusively through behavior cloning during pretraining [26, 73, 4, 5, 3, 65, 50, 60]. Although some works [5, 65] additionally incorporate VQA data to preserve semantic reasoning capabilities, these approaches still focus on static understanding and immediate action prediction, without explicitly injecting temporal awareness of goal reachability into the pretrained VLA. Consequently, System 2 operates with only qualitative semantic reasoning capabilities (determining "*what to do*") but without the quantitative awareness of goal reachability needed to evaluate execution difficulty or to distinguish difficult-yet-reachable states from easy-yet-erroneous ones.

To address this deficiency, we reformulate VLA pretraining through the lens of *Goal-Conditioned Reinforcement Learning*. In particular, we adopt Contrastive Reinforcement Learning (CRL) [19], recently shown to scale to over 1000-layer deep networks [58], which learns paired state-action and goal representations whose similarity approximates the reachability of goals under given state-action pairs. This formulation naturally equips the VLM backbone 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 leverages *Goal-Conditional RL* for representations learning in VLA pre-training phase. Our approach introduces three key technical designs. **(i) Language-conditioned contrastive RL for representation learning.** Unlike prior VLAs [4, 5, 3, 50, 60] that pre-train the policy purely by behavior cloning, PRTS reformulates policy learning as a goal-conditioned representation learning problem with language instructions as goals. Specifically, we construct positive pairs using state-action pairs from trajectories paired with their corresponding text instructions, and negative pairs using text goals from different tasks. However, in contrast to standard CRL where future goals are sampled from a geometric distribution within the same episode, the goal is a language instruction shared across

all timesteps in language-conditioned manipulation setup. To bridge this gap, we transform the geometric distribution sampling into a temporal weighting scheme over state-action pairs. This formulation drives the model to learn a representation space where the similarity between state-action embeddings and goal embeddings encodes the discounted goal occupancy measure, thereby capturing the temporal structure of goal-reaching behavior. **(ii) Implicit dense goal-reachability supervision.** The contrastive objective is built upon a goal-reaching reward assumption where the reward is defined as the transition probability of reaching the goal from the current timestep [19]. This enables the model to extract dense supervision and acquire a dense goal-conditioned action-value function directly from trajectory structure without manual reward annotations, quantifying the long-term utility of state-action pairs in purely offline pretraining. **(iii) Single-forward-pass design at negligible cost.** Adding two contrastive heads to a VLM typically requires a second forward pass or a separately trained value network [23]. PRTS instead appends two small token blocks, <CRL_action> and <CRL_goal>, to the standard input sequence and enforces information isolation through a role-aware causal mask integrated into a custom FlashAttention kernel [64]. The autoregressive action tokens retain standard causal attention, ensuring that the behavior-cloning loss remains identical with or without the CRL tokens. Meanwhile, the CRL tokens extract state-action embeddings and goal embeddings from isolated information streams within the same forward pass. As a result, end-to-end pretraining incurs negligible additional cost compared to vanilla behavior cloning on the same backbone.

Notably, PRTS unifies semantic reasoning, discrete action prediction, and goal-conditioned representation learning within a single VLM architecture, enjoying joint optimization and shared vision-language representations. It equips the VLM with a quantitative notion of goal reachability that goes well beyond static semantic understanding. Learning to estimate goal reachability requires the backbone to acquire task-relevant vision-language representations that encode the functional affordances required for task completion.

Built upon this architecture, we construct a large-scale pre-training dataset comprising diverse robotic trajectories with language annotations. We train the PRTS foundation model on over 167B tokens, employing infrastructure optimizations including a custom CuTe-FlashAttention kernel and sequence packing to achieve high-throughput training on 64 H100 GPUs for one week. We subsequently fine-tune and evaluate the model on diverse downstream tasks: in simulation, we validate foundational manipulation capabilities on standard benchmarks including LIBERO [36], LIBERO-Pro [72], LIBERO-Plus [21], and SimplerEnv (WidowX) [32]; in real-world deployment, we establish a comprehensive evaluation suite comprising 14 complex real-world manipulation tasks across a dual-arm RealMan platform and a single-arm Flexiv platform. Experimental results demonstrate that PRTS achieves state-of-the-art performance in both simulation and real-world environments, exhibiting substantial advantages in long-horizon execution, zero-shot novel-instruction generalization, and recovery under human interventions. These findings validate the efficacy of learning goal-reaching representations for enhancing planning robustness and execution success rates in VLA systems.

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. [19], we focus on the *goal-reaching reward* function, which quantifies the immediate probability of reaching the goal:

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

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:

$$Q^{\pi}_{s_g}(s, a) \triangleq \mathbb{E}_{\pi}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_g(s_{t'}, a_{t'}) \,\middle|\, s_t = s, a_t = a\right]. \tag{2}$$

Discounted State Occupancy Measure. To establish the connection between value estimation and probability, we introduce the *discounted state occupancy measure* [52]. For a policy π conditioned on goal s_g, the occupancy measure of state s is defined as:

$$p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s) \triangleq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p_t^{\pi(\cdot|\cdot,s_g)}(s_t = s), \tag{3}$$

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 $t \sim \text{Geom}(1-\gamma)$ 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. [19])**.** *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):*

$$Q^{\pi}_{s_g}(s, a) = p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s_g \mid s, a). \tag{4}$$

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:

$$f(s, a, s_g) = \phi(s, a)^\top \psi(s_g), \tag{5}$$

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 [55]:

$$\mathcal{L}_{\text{crl}} = \max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a),\\ s_g^+ \sim p^\pi(s_{t+}|s,a),\\ s_g^- \sim p(s_g)}} \Big[\log \exp(f(s, a, s_g^+)) - \log(\exp(f(s, a, s_g^+)) + \sum \exp(f(s, a, s_g^-))) \Big], \tag{6}$$

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:

$$f^*(s, a, s_g) = \log Q^\pi_{s_g}(s, a) - \log p(s_g) + c(s, a), \tag{7}$$

implying $\exp(f^*(s, a, s_g)) \propto Q^\pi_{s_g}(s, a)$ up to a state-action-dependent normalization. Thus, the critic provides a valid estimate of the relative Q-values for policy evaluation.

III. Method

A. Problem Setup

We consider a language-conditioned robotic manipulation task with visual observations 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 demonstration trajectories collected from diverse robotic tasks. Each trajectory $\tau = (s_{1:T}, l, a_{1:T})$ consists of a natural language instruction l describing 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 images and the robot's proprioceptive state $\mathbf{q}_t \in \mathbb{R}^{d_q}$. The action $a_t \in \mathbb{R}^{d_a}$ represents the low-level control commands.

The central objective of VLA pre-training is to learn representations that encode not only visual-linguistic semantics but also the *temporal structure of goal-reaching*—quantifying how likely a given observation-action pair is to eventually achieve the language-specified objective, i.e., *goal reachability*. While recent works such as $\pi^\star_{0.6}$ [23] and VLAC [66] have explored value-augmented VLA training, they share fundamental bottlenecks: (i) reliance on explicit reward or progress annotations, e.g., episode success/failure labels for $\pi^*_{0.6}$ and curated pairwise progress labels for VLAC; (ii) a separately trained value network that incurs substantial additional training cost or requires a multi-stage pipeline; and (iii) the categorical targets are either discretized scalar returns or hand-engineered progress fractions, none of which directly encode the underlying goal-reaching structure of the task.

In contrast, PRTS frames goal reachability estimation as *contrastive RL* [19]: the categorical structure arises naturally from the multi-task training batch—each language goal serves as a class, and the model learns to classify which goal a given (s, a) is progressing toward. This formulation eliminates the need for reward labels, return discretization, pairwise data curation, and auxiliary value networks. Instead, the resulting goal-conditioned action-value function $Q^\pi_l(s, a)$ emerges directly from the representation geometry of the shared VLM backbone within a single forward pass. Furthermore, recent advances demonstrate that contrastive objectives exhibit favorable scaling properties in architectures with thousands of layers [58], holding promise for the integration with large-scale VLA foundation models. Moreover, their cross-entropy formulation aligns naturally with the discrete token prediction paradigm of VLMs, enabling unified pre-training where goal-conditioned representation learning and language modeling objectives are optimized jointly.

B. Language-Conditioned Contrastive Reinforcement Learning

Adapting standard contrastive RL to the vision-language-action setting induces a shift in how task goals are parameterized. Standard CRL operates in state-based goal settings: for each state-action pair (s_t, a_t) sampled from a trajectory, a goal state s_g is drawn via geometric sampling $\Delta t \sim \text{Geom}(1 - \gamma)$ from future timesteps. Since $s_{t+\Delta t}$ is almost surely unique within a batch, this yields a *single-positive* contrastive pair in which every other sample's future state serves as a distinct negative goal. In PRTS, the goal is instead defined as the *language instruction* l, which specifies the task-completion condition and is shared across all timesteps of a trajectory. This shared-goal structure breaks the single-positive assumption and demands a reformulation of CRL.

Multi-Positive Problem. Since the goal l is identical across all timesteps of every trajectory belonging to the same task, a mini-batch drawn from the dataset typically contains many state-action pairs that share a single language goal. Formally, consider a mini-batch $\mathcal{B} = \{(s_j, a_j, l_j, \tau_j)\}_{j=1}^B$ of size B sampled from the pre-training dataset $\mathcal{D}_{\text{pre}}$, where each tuple contains:

- (s_j, a_j): the state-action pair at timestep t_j,
- l_j: the language instruction (goal) for the trajectory,
- τ_j: the task identifier (indicating which task the sample belongs to).

For a given anchor sample $i \in \{1, \dots, B\}$, we define the *positive set* as the indices of all samples in the batch belonging to the same task:

$$\mathcal{S}(i) \triangleq \{j \in \{1, \dots, B\} : \tau_j = \tau_i\}. \tag{8}$$

That is, $\mathcal{S}(i)$ collects all batch indices that share the anchor's task, including i itself. This structural property prevents direct application of standard CRL's single-positive objective (Eq. (6)), necessitating a reformulation that handles multiple positives while still preserving the temporal structure of goal-reaching.

Temporal Weighting as Geometric Sampling. In standard CRL, geometric sampling $\Delta t \sim \text{Geom}(1-\gamma)$ makes the probability of sampling a future state k steps ahead proportional to γ^k. Under this sampling, the optimal classifier satisfies

$$f^*(s, a, s_g) = \log \frac{p^\pi(s_{t+} = s_g \mid s, a)}{p(s_g)} + c(s, a), \tag{9}$$

where $p^\pi(s_{t+} = s_g \mid s, a) = (1-\gamma) \sum_{k=1}^{\infty} \gamma^{k-1} p(s_{t+k} = s_g \mid s_t, a_t)$ is the discounted goal occupancy measure and $c(s, a)$ is a goal-independent term that is absorbed into the learned bias of the network and leaves the relative temporal information intact.

However, language goals cannot be sampled geometrically from a trajectory—they are fixed per task. Fortunately, in expert demonstrations, the discounted probability of reaching the goal from timestep t can be approximated by γ^{T-t}, i.e., the discount factor raised to the power of remaining timesteps. This observation motivates a temporal weighting scheme that emulates geometric sampling by assigning weights to multiple positive samples according to their temporal distance to the goal. Formally, for each positive sample $j \in \mathcal{S}(i)$ at trajectory timestep t_j (with length T_j) sharing the same language goal l_i, we define the temporal weight:

$$q_{ij} = \frac{\gamma^{T_j - t_j}}{\sum_{j' \in \mathcal{S}(i)} \gamma^{T_{j'} - t_{j'}}}. \tag{10}$$

This weighting scheme assigns exponentially larger weights to states closer to task completion and mirrors the decay of the geometric distribution. As we prove below, optimizing with these soft targets yields representations whose inner product $\psi(l)^\top \phi(s, a)$ is proportional to the log-discounted occupancy, recovering the same temporal structure as standard CRL.

Bidirectional Contrastive Objectives. We optimize two complementary cross-entropy objectives that play distinct representational roles.

(i) State-Action to Language ($s, a \to l$): For each state-action anchor sampled from the batch, we compute:

$$\mathcal{L}^{\text{sa}\to l} = \mathbb{E}_{i \sim \mathcal{B}} \left[-\gamma^{T_i - t_i} \log \frac{\exp(\phi_i^\top \psi_i)}{\exp(\phi_i^\top \psi_i) + \sum_{k \in \mathcal{B},\, \tau_k \neq \tau_i} \exp(\phi_i^\top \psi_k)} \right] \tag{11}$$

where $\phi_i \triangleq \phi(s_i, a_i)$ and $\psi_i \triangleq \psi(l_i)$, and the denominator sums over all unique language goals in the batch (including negatives). In this direction, each (s_i, a_i) has exactly *one* positive goal l_i (its own task's instruction), reducing the formulation to the standard CRL problem. Thus, to present fixed goals under the geometric sampling distribution, we weight each state-action pair by $\gamma^{T_i - t_i}$.

However, sample-wise weight γ^{T-t} alone only rescales the loss magnitude without altering the gradient direction, pushing all states from the same task toward the same similarity with l_i and failing to encode the temporal progression within the trajectory. Consequently, this objective primarily promotes *task-level discrimination*: it aligns state-action pairs with their corresponding language goals, while remaining largely agnostic to temporal distance to task completion.

(ii) Language to State-Action ($l \to s, a$): In contrast, the temporal signal is introduced in the reverse direction. For a language anchor l_i with positive set $\mathcal{S}(i) = \{j \in \mathcal{B} : \tau_j = \tau_i\}$, we optimize:

$$\mathcal{L}^{l\to\text{sa}} = \mathbb{E}_{i \sim \mathcal{B}} \left[-\sum_{j \in \mathcal{S}(i)} q_{ij} \log \frac{\exp(\psi_i^\top \phi_j)}{\sum_{k \in \mathcal{B}} \exp(\psi_i^\top \phi_k)} \right], \quad \text{with} \quad q_{ij} = \frac{\gamma^{T_j - t_j}}{\sum_{j' \in \mathcal{S}(i)} \gamma^{T_{j'} - t_{j'}}}. \tag{12}$$

Unlike the $sa \to l$ direction, l_i now has *multiple* positives. The soft targets q_{ij} require the predicted probability that the state-action pair j belongs to task i to scale as $\gamma^{T_j - t_j}$. This forces the representation inner product to satisfy $\psi_i^\top \phi_j = (T_j - t_j) \log \gamma + C$ according to Eq.(9), thereby encoding the *temporal distance to task completion* within the representation space.

The combination of **(i)** and **(ii)** ensures that the learned representations capture both *task identity* (via $s, a \to l$) and *task progress* (via $l \to s, a$). While $s, a \to l$ prevents interference between different tasks, $l \to s, a$ arranges states along a trajectory in order of their value (proximity to 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 occupancy measure.

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:*

$$\psi^*(l)^\top \phi^*(s, a) = \log Q_l^{\pi^*}(s, a) + C(l), \tag{13}$$

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.

Proof: The proof is deferred to Appendix A. ∎

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.

C. PRTS Architecture

We instantiate PRTS upon the Qwen3-VL backbone [2], extending it to support unified pre-training for both discrete action prediction and contrastive representation learning. As illustrated in Fig. 1, the input sequence comprises visual tokens (encoded via a Qwen ViT Encoder from egocentric and wrist views $\mathbf{I}_t^1, \ldots, \mathbf{I}_t^V$), robot proprioception tokens for $\mathbf{q}_t$, language instruction tokens for l, and auto-regressive (AR) action tokens obtained by FAST [44] tokenization of action chunk $\mathbf{a}_t = a_{t:t+H}$. Here, we represent proprioceptive states as text after discretization like [5, 17]. Crucially, we append two auxiliary token blocks at the tail of the sequence: <CRL_action> and <CRL_goal>. The <CRL_action> block consists of repeated action tokens followed by a learnable token <|action_end|>, whose hidden state encodes $\phi(s_t, \mathbf{a}_t)$. Similarly, the <CRL_goal> block consists of repeated instruction tokens ended with a learnable token <|goal_end|>, whose hidden state encodes $\psi(l)$. These auxiliary tokens extract representations for contrastive learning within the same forward pass used for action generation.

Role-Aware Causal Mask for Disentangled Representations. It is non-trivial to ensure that $\phi(s_t, \mathbf{a}_t)$ and $\psi(l)$ correctly encode the desired modalities without information leakage while preserving the efficiency of standard next-token prediction training. To address this, we introduce a *role-aware* causal attention mask, depicted in the bottom-left of Fig. 1, which assigns each token one of five roles: vision/state, instruction, AR action, CRL_action, and CRL_goal. Based on these roles, we enforce three structured attention constraints:

(i) AR action tokens use standard causal attention over all preceding visual, proprioception, and instruction tokens. This preserves the original autoregressive action token prediction, ensuring that the behavior cloning loss on FAST-tokenized actions remains identical to a non-CRL run.

(ii) CRL_action tokens $\phi(s_t, \mathbf{a}_t)$ are masked to attend *only* to vision tokens, proprioception tokens, and the CRL action tokens themselves. They are blocked from attending to language instructions and AR action tokens. This ensures that $\phi(s,a)$ extracts a pure state-action representation independent of the language goal, which is essential for the validity of the contrastive similarity $\phi^\top \psi$.

(iii) CRL_goal tokens $\psi(l)$ employ self-only causal attention restricted to the <CRL_goal> token block, with all other tokens masked out. This forces $\psi(l)$ to be a compact, task-specific encoding of the language instruction alone, serving as the goal anchor in the contrastive objective.

This design enables computing both action 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 structurally sparse role-aware mask on top of the latest FlashAttention [64] combined with a custom CuTe kernel, which encodes the five-role rules at the block level so that only the non-masked blocks are computed. This brings the joint forward pass to the same throughput as a pure-causal next-token-prediction behavior cloning run with FlashAttention-2/3 as backend [15, 49]. The design further remains compatible with 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. VI), a property we rely on to scale contrastive-RL pre-training to a over 167B token corpus.

Action Expert. For real-time continuous control, we adopt a flow-matching action expert [34, 38, 35] based on a Diffusion Transformer (DiT) [43], following recent VLA paradigms [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 [35]:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau\sim[0,1],\,\epsilon\sim\mathcal{N}(0,I)} \left\| f_\theta(\mathbf{a}_t^\tau,\, s_t,\, l) - (\mathbf{a}_t - \epsilon) \right\|_2^2, \quad (14)$$

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.

Pretraining and Post-training Objectives. PRTS adopts a two-stage training scheme that cleanly separates representation learning from continuous-action adaptation. The VLM backbone is shaped during pre-training, and the flow-matching action expert is attached and adapted during post-training. Gradients flow through the VLM backbone in both stages.

In the *pre-training* stage, the backbone is trained jointly on the bidirectional contrastive losses, and the standard next-token prediction cross-entropy loss, which encompasses the BC loss on AR action tokens and the auxiliary cross-entropy losses for robot-QA and sub-task prediction. The action expert is not included at this stage. At each iteration, a mini-batch $\mathcal{B}$ yields the temporal weights q_{ij} of Eq. (10), and the representations $\phi(s_{t_j}, \mathbf{a}_{t_j})$ and $\psi(l_i)$ are respectively produced by lightweight MLP heads applied to the final hidden states in the <CRL_action> and <CRL_goal> blocks, followed by ℓ_2-normalization and scaling by a learnable temperature τ_{crl}. The resulting dot products $\phi_j^\top \psi_i$ are then used to compute the bidirectional contrastive objectives: the state-action to language loss $\mathcal{L}^{\text{sa}\to l}$ (Eq. (11)) and the language to state-action loss $\mathcal{L}^{l\to\text{sa}}$ (Eq. (12)). As demonstrated by Oord et al. [42], the performance of the InfoNCE objective relies on the size of the negative pool. To maximize this pool without exceeding

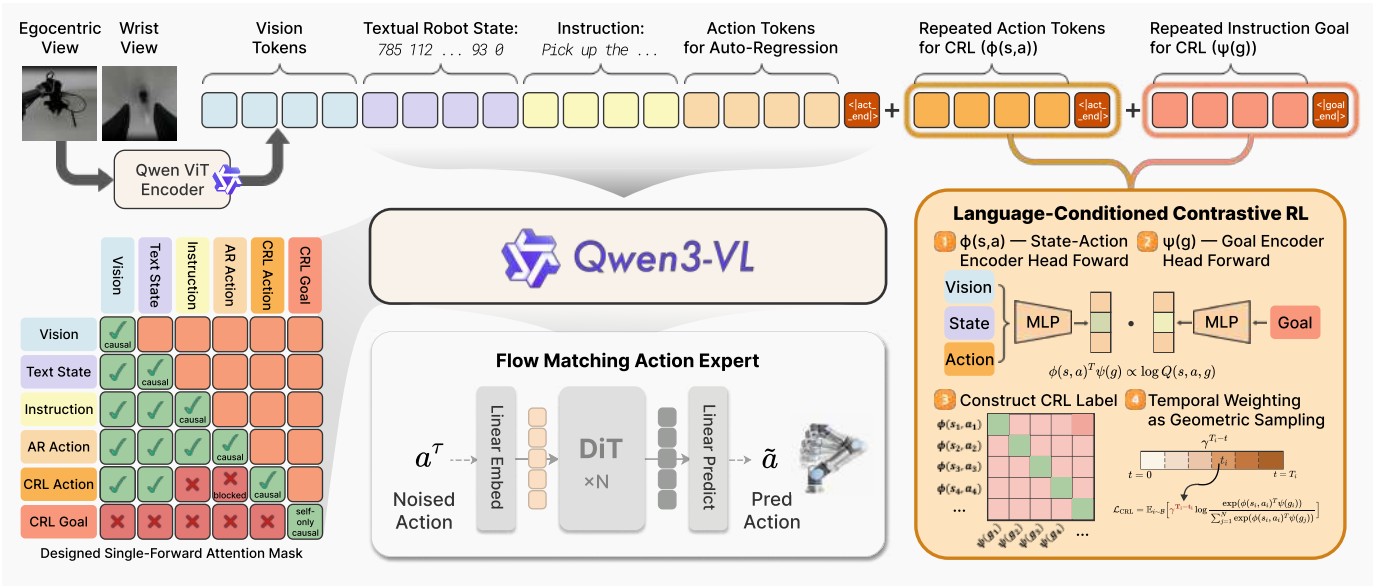


Fig. 1: **Overview of PRTS.** PRTS is built on Qwen3-VL-4B-Instruct and involves two training stages. (i) In the **pre-training** stage, the backbone consumes a unified token sequence of multi-view visual tokens (Qwen ViT), textual proprioceptive-state tokens, language-instruction tokens, and FAST-tokenized AR action tokens, followed by two auxiliary blocks <CRL_action> and <CRL_goal>, in which the last learnable tokens <|action_end|> and <|goal_end|> are passed through lightweight MLP heads (right) to produce the state-action representation $\phi(s_t, \mathbf{a}_t)$ and the goal representation $\psi(l)$. A *role-aware causal mask* (bottom-left) prevents information leakage across the five token roles: <CRL_action> tokens attend only to vision and proprioception; <CRL_goal> tokens attend only within their own block; AR action tokens keep standard causal attention. All three outputs—discrete-action logits, $\phi(s_t, \mathbf{a}_t)$, and $\psi(l)$—are produced in a single forward pass and trained jointly under the bidirectional contrastive (InfoNCE) losses with temporal weighting and the next-token cross-entropy loss, so that $\phi(s_t, \mathbf{a}_t)^\top \psi(l) \approx \log Q_l^\pi(s_t, \mathbf{a}_t)$. The action expert is absent at this stage. (ii) In the **post-training** stage, a freshly initialized DiT-based flow-matching action head is attached and fine-tuned via $\mathcal{L}_{\text{FM}}$ to produce continuous action chunks.

memory constraints, we employ a cross-device contrastive strategy akin to CLIP [46]. Specifically, we shard the similarity computation across GPUs, allowing each device to aggregate negative representations globally while only computing the gradient for its local batch. The pre-training objective is

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{RobotQA}} + \mathcal{L}_{\text{Subtask}} + \lambda_{\text{crl}} \left(\mathcal{L}^{\text{sa}\to l} + \mathcal{L}^{l\to \text{sa}} \right), \tag{15}$$

where $\mathcal{L}_{\text{BC}}$ is the cross-entropy loss on AR action tokens and $\mathcal{L}_{\text{RobotQA}}, \mathcal{L}_{\text{Subtask}}$ are the auxiliary robot-QA and subtask prediction losses. This unified objective enables the VLM backbone to learn goal-conditioned representations that simultaneously support (i) semantic grounding, (ii) discrete and continuous action generation over action chunks, and (iii) dense goal-reachability estimation expressed via the inner product of the contrastive embeddings.

In the *post-training* stage, we then attach the flow-matching action expert and fine-tune on downstream demonstration data using Eq. (14),

$$\mathcal{L}_{\text{post-train}} = \mathcal{L}_{\text{FM}}. \tag{16}$$

We drop the BC and auxiliary terms in this stage. The representations after pre-training already encode task identity and goal reachability, so post-training can focus entirely on continuous-action regression.

Benefits for Representation and Value Extraction. In Eq. (15), while $\mathcal{L}_{\text{BC}}$ focuses on temporally local, low-level action accuracy, the CRL objectives encourage the model to learn a *globally consistent* representation space whose geometry encodes the temporal structure of goal-reaching. Specifically, the contrastive signal aligns visual observations, robot states, and language instructions into a unified metric space in which distances correspond to task progress. This mitigates overfitting to spurious correlations in demonstration data and promotes generalization to long-horizon and fine-grained manipulation tasks.

Upon training completion, the model inherently provides a dense goal-conditioned action-value without requiring a separately trained value network. The learned representations satisfy:

$$Q_l^\pi(s, \mathbf{a}) \propto \exp\left(\phi(s, \mathbf{a})^\top \psi(l)\right), \tag{17}$$

or equivalently, $\log Q_l^\pi(s, \mathbf{a}) \approx \phi(s, \mathbf{a})^\top \psi(l) + \text{const}$. As established in Theorem 1, this inner product estimates the log-probability of reaching the language goal under the discounted state-occupancy measure, yielding a scalar signal that quantifies progress toward satisfying the instruction. Notably, this dense $Q_l^\pi(s, \mathbf{a})$ emerges as a by-product of the same backbone used for action generation, suggesting that value awareness is

implicitly integrated into the policy during pre-training and embedded directly into the learned representations.

IV. Datasets

To build a value-aware foundation VLA capable of both high-level reasoning and low-level control, we construct a large-scale multi-modal pre-training dataset of **404 M samples** totaling **167.8 B tokens**. As illustrated in Fig. 2, 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.

A. 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** [14]. 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. Besides, it provides detailed fine-grained sub-task annotations.
- **RoboMind** [59]. As a large-scale, unified teleoperation data dataset, RoboMind introduces a vast array of 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** [13]. Open X-Embodiment Dataset [13] encompasses 22 robotic embodiments across distinct laboratory and real-world environments, which is essential for endowing the model with strong cross-embodiment generalization capabilities.
- **PRTS Self-Collected Dataset.** Collected on dual-arm RealMan and Agilex platform, our dataset covers a wide range of daily manipulation tasks and complements the dataset with highly complex, multi-scene scenarios that demand precise, contact-rich operations, enhancing the diversity of the training distribution.

B. Visual-Reasoning Data

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

Spatial Grounding and Affordance Perception. To safely and accurately interact with the physical world, the model needs to achieve pixel-level localization and understand operable object regions. We utilize **RefCOCO** [25] for fundamental referring expression comprehension, pairing natural language descriptions with precise 2D bounding boxes. To further enhance fine-grained localization, we incorporate **Pixmo-Point** [16], which employs a Point-QA paradigm, forcing the model to output precise 2D pixel coordinates rather than abstract text. This precise 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** [63] for spatial-relation grounding alongside standard object detection. We further include **RoboRefIt** [39] to resolve referential ambiguity in cluttered environments, encouraging the model to track correct grasping targets under complex relative spatial relations. We also incorporate **RefSpatial** [71] 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** [54] 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. To this end, we inject **Cosmos-Reason1** [41] to provide dense reasoning chains (Chain-of-Thought) regarding physical causality and scene dynamics. For long-horizon execution, we leverage **EgoPlanIT** [10] to train the model in decomposing abstract human instructions into sequential, actionable subgoals from an egocentric perspective. Additionally, **RoboVQA** [48] is included to enhance step-by-step temporal reasoning and context awareness over continuous embodied video streams.

General Semantic Understanding. To retain general vision–language competence during large-scale embodied pre-training, we include a curated, high-quality subset of **LLaVA-Instruct** [37]. This subset acts as a semantic regularizer, preserving robust zero-shot language-following abilities, visual world knowledge, and conversational fluidity, ensuring the VLA backbone model remains a highly capable conversational agent.

V. Related Work

Visual-Language-Action Models. Early VLAs such as RT-1 [6], RT-2 [73], and OpenVLA [26] 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, 23, 3, 65], followed by task- or embodiment-specific post-training [27, 67, 61]. 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 [24], RoboVLM [31], and CoT-VLA [68] inject chain-of-thought reasoning and task decomposition. However, while these auxiliary objectives improve static visual-semantic grounding, they fail to encode

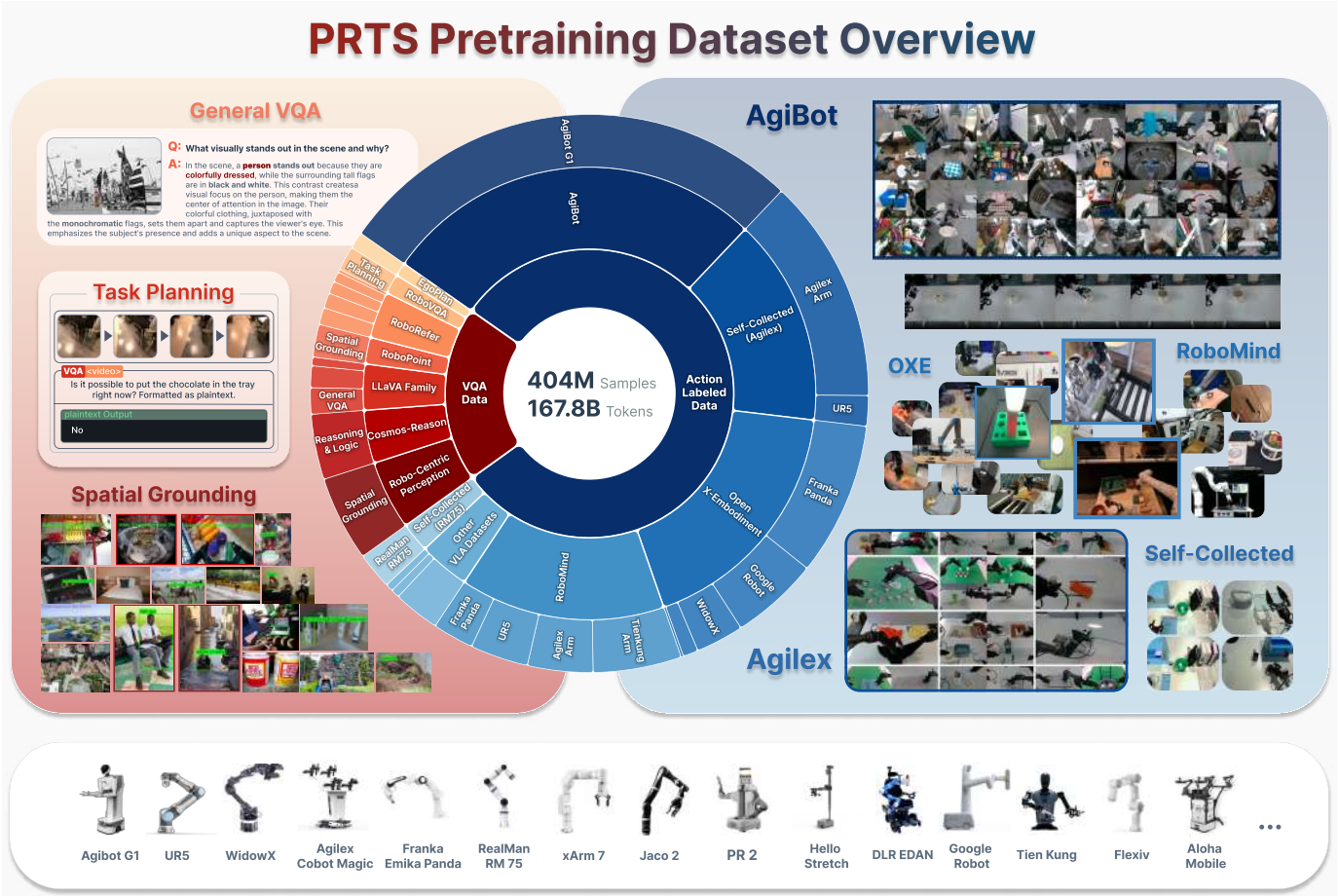


Fig. 2: **Pre-training dataset overview of PRTS.** Our pre-training dataset is partitioned into two primary domains: *Action-Labeled Data* provides physical execution priors for cross-embodiment manipulation, while *Visual-Reasoning Data* provides complementary supervision for spatial grounding, affordance perception, and task planning. The area of each segment is scaled proportionally to the square root of its total tokens to highlight the rich, structural diversity of our data sources.

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 $\pi^*_{0.6}$ [23], VLAC [66], 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 [28, 29]. 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 [18] 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 [52, 28]. Contrastive RL (CRL) [19] closes this gap by reformulating value learning as an InfoNCE classification problem [42]: the goal-conditioned value function under geometric discounting is recoverable purely by cross-entropy classification, without Bellman regression. Zheng et al. [69] 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. [58] 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 [20]. 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.

VI. Experiments

PRTS is designed to perform robustly and generalize effectively across a wide range of embodied tasks including long-horizon tasks and settings with distribution shifts. To this end, we evaluate PRTS through a comprehensive set of experiments spanning simulation and real-world deployment, robustness to distribution shifts, instruction-following generalization, value analysis, and pre-training efficiency. Through extensive experiments, we aim to answer the following key questions:

Q1: Overall Performance. How does PRTS compare with state-of-the-art VLAs in both simulation and real-world settings? (Sec. VI-B)

Q2: Effect of Contrastive RL. How does incorporating contrastive RL during pre-training contribute to the final performance of PRTS when the post-training recipe is held fixed? (Sec. VI-C)

Q3: Generalization Capability. How well does PRTS generalize under distribution shifts, including variations in spatial position, lighting conditions, object appearance, and even novel instructions, i.e., zero-shot instruction-following? (Sec. VI-D)

Q4: Robustness and Recovery under Human Interventions. Can PRTS maintain robust performance and recover from failures under frequent human interventions during execution? (Sec. VI-E)

Q5: Value Representation. Do the learned representations faithfully encode goal-conditioned action values? (Sec. VI-F)

Q6: Pre-Training Efficiency. How efficient are our optimized pre-training infrastructure and framework for scaling contrastive RL? (Sec. VI-G)

A. Experimental Setup

Pre-training recipe. We initialize PRTS from Qwen3-VL-4B-Instruct [2] and pre-train it on the corpus described in Sec. IV using the objective in Eq. (15). We set the contrastive-RL coefficient to $\lambda_{\text{crl}} = 1.0$ and the temporal discount in the contrastive weights to $\gamma = 0.995$. To improve training efficiency, we pack multiple short sequences into a single sequence of length 4096, reducing wasted computation on padding tokens. Pre-training runs for one epoch on $64 \times$ H100 GPUs with a global batch size of 256 packed sequences, totaling approximately 220K gradient steps and finishing in about one week. Finally, thanks to the sharded contrastive similarity computation strategy, we obtain roughly 2K in-batch negatives for the CRL objective, which is critical for scaling contrastive RL during pre-training.

Post-training recipe. For each downstream benchmark, we attach a randomly initialized flow-matching DiT action expert to the pre-trained backbone and optimize the post-training objective in Eq. (14). The action expert has 675M parameters and uses 5 flow-matching denoising steps at inference. Table I summarizes the post-training schedules. Across both simulation and real-robot evaluations, PRTS uses a post-training compute budget that is no larger than, and in most cases substantially smaller than, those used by strong representative VLAs such as $\pi_{0.5}$. For example, comparisons on LIBERO simulation benchmark use only $1/8$ of the post-training compute used by $\pi_{0.5}$. This design makes performance differences easier to attribute to how well the pre-trained VLA acts as a general visuomotor prior rather than to a larger downstream fine-tuning budget.

Real-world robot platforms. Our real-world evaluation uses the two platforms shown in Fig. 3: a Flexiv platform for three single-arm manipulation tasks and a dual-arm RealMan platform for eleven bimanual manipulation tasks. The RealMan platform uses a stationary 14-DoF bimanual configuration, with each arm equipped with a parallel-jaw gripper. It observes the workspace through three RGB cameras: one egocentric head camera and one wrist camera per arm. The Flexiv platform, shown in the right panel of Fig. 3, provides a complementary single-arm setting.

Benchmarks. We evaluate PRTS on five benchmarks:

- **LIBERO** [36], the four standard suites (*Spatial*, *Object*, *Goal*, *Long*) for measuring in-distribution visuomotor accuracy.
- **LIBERO-Plus** [21], a robustness-oriented extension with $10{,}030$ test tasks spanning seven perturbation dimensions: object layout, camera viewpoint, robot initial state, language instruction, lighting, background texture, and sensor noise.
- **LIBERO-Pro** [72], a perturbation-based extension of LIBERO that tests whether a policy follows the task specification rather than memorizing the original scene, covering five generalization dimensions: *Object*, *Position*, *Semantic*, *Task*, and *Environment*.
- **SimplerEnv (WidowX)** [32], the BridgeData V2/WidowX track of SIMPLER, which evaluates policies trained on real-world data in simulation through four rigid-object pick-place-stack tasks.
- **Real-world evaluation**, three fine-grained manipulation tasks on the Flexiv platform and eleven manipulation tasks on the dual-arm RealMan platform, together with deliberate perturbations that vary lighting, initial object position, object identity, and task instruction at deployment time. We visualize all these task in Figs. 11 and 12 and provide detailed descriptions in Appendix B.

Baselines. The baseline set differs across benchmarks because we use strong representative VLAs reported under each

Benchmark	Global batch size	Gradient steps	Action chunk H
LIBERO	32	30K	20
SimplerEnv (WidowX)	1024	20K	16
RealMan Dual-Arm Robot Platform (all-task fine-tuning)	64	100K	20
Flexiv Robot Platform (single-task fine-tuning)	32	40K	20

TABLE I: **Post-training recipes per benchmark.**

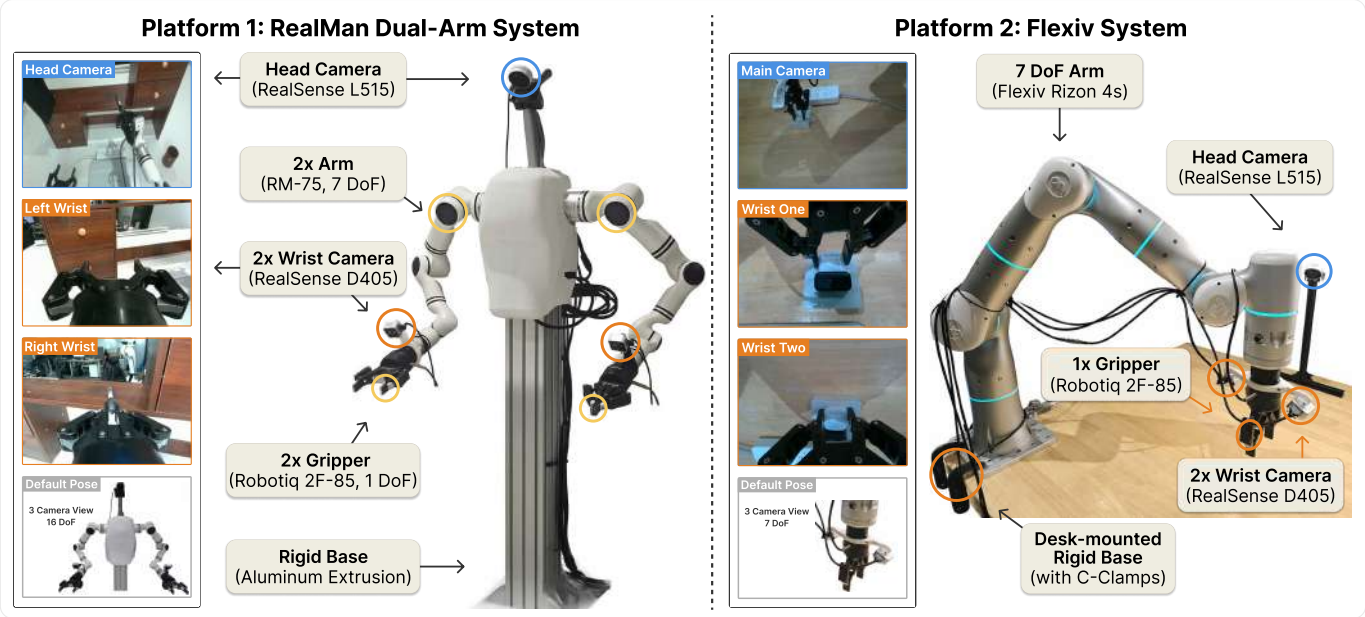


Fig. 3: **Overview of real-world robot platforms.** Left: dual-arm RealMan platform with two parallel-jaw grippers, one egocentric head camera, and two wrist cameras (one per arm). Right: Flexiv single-arm platform equipped with a parallel-jaw gripper, a third-view head camera, and two wrist cameras.

protocol, whereas our real-world experiments directly compare against π_0 [4] and $\pi_{0.5}$ [5] under the same evaluation protocol. Also, we particularly annotate the post-training budget of those competitive VLA baselines in gray next to the model name when available, to make the compute budget explicit. This annotation aims to clearly highlight how rich a general visuomotor prior PRTS provides compared to existing VLAs, particularly under the same or even smaller post-training budget.

Evaluation protocol. All simulated benchmarks report task success rate (SR) under the standard protocol of the corresponding benchmark. For real-world evaluation, each task cell is evaluated over 20 physical trials. On the Flexiv platform, a trial is marked as failed if the robot does not succeed within five grasp attempts or within two minutes. On the RealMan platform, grasping tasks similarly allow up to five feasible re-grasp attempts and are marked as failed if they exceed a three-minute time limit.

B. Overall Performance

We first evaluate PRTS on four simulation benchmarks that probe complementary aspects of policy competence: in-distribution visuomotor execution on LIBERO, robustness to distribution shifts on LIBERO-Plus and LIBERO-Pro, instruction-grounded generalization on LIBERO-Pro, and cross-embodiment adaptability to the WidowX robot in SimplerEnv. For LIBERO, PRTS is post-trained with global batch size 32 for 30K gradient steps. The resulting checkpoint is then evaluated *zero-shot* on LIBERO-Plus and LIBERO-Pro, without any additional training on those benchmark-specific perturbation datasets. For SimplerEnv, PRTS is post-trained with global batch size 1024 for 20K gradient steps, same as the compute budget of GR00T-N1.5 [3], the strongest prior results on that benchmark. The baseline set differs between benchmarks and is listed in each table.

1) LIBERO: Table II compares PRTS against strong VLA baselines on the four standard LIBERO suites. With only global batch size 32 and 30K post-training steps, PRTS reaches 98.4% average SR and matches the state-of-the-art performance of ABot-M0 while using much less post-training compute than several strong baselines. In particular, $\pi_{0.5}$ uses an $8\times$ larger batch (bs=256, 30K steps) yet reaches 96.9%, while the strongest ABot-M0 setting uses both a larger batch and more steps (bs=64, 40K steps) to reach 98.6%. To separate

representation quality from post-training compute, we also train ABot-M0 under the same bs=32 and 30K gradient steps budget as PRTS. Under this matched budget, ABot-M0 reaches 97.9% average SR, 0.5 points below PRTS, and suffers a dramatic drop from 96.6% to 94.8% on the *Long* suite. PRTS, in contrast, reaches 96.6% on *Long*, matching the best reported result with significantly smaller post-training cost. It suggests that the goal-reachability-aware representation learned through contrastive RL pre-training is especially critical for long-horizon manipulation tasks. Overall, the controlled-budget comparison indicates that PRTS benefits from a stronger pre-trained visuomotor prior, rather than relying on heavier downstream optimization to recover missing temporal reasoning ability.

2) LIBERO-Plus: Table III reports zero-shot evaluation on LIBERO-Plus using the same checkpoint post-trained only on standard LIBERO. LIBERO-Plus introduces seven controlled perturbation axes, so performance reflects whether a policy can preserve task execution under shifted camera views, robot states, language, lighting, backgrounds, sensor noise, and object layouts. PRTS achieves the best average SR (81.4%), outperforming $\pi_{0.5}$ (80.7%) despite using an $8\times$ smaller LIBERO post-training batch. More importantly, the gains are concentrated on quite challenging perturbations: *Robot* ($+14.4$), *Background* ($+15.3$), and *Language* ($+1.1$), while remaining competitive across all other axes. These results show that the CRL pre-trained backbone provides a robust goal-conditioned prior. After fine-tuning on standard LIBERO, the policy can still identify and execute the intended task under changes in robot state, scene appearance, and LLM-rewritten instructions.

3) LIBERO-Pro: LIBERO-Pro (Table IV) further tests whether the LIBERO-finetuned checkpoint follows the task specification rather than replaying memorized trajectories. We again perform zero-shot evaluation. This benchmark was designed to expose a key flaw of current VLAs: a high success rate on standard LIBERO can stem from mechanical memorization of training scenarios rather than transferable task-solving strategies [72]. To this end, LIBERO-Pro introduces perturbation axes that directly test whether a policy can solve the specified task rather than replay a familiar trajectory.[1] The *Semantic* axis paraphrases the instruction while preserving the task intent, similar to the *Language* perturbations in LIBERO-Plus, and the *Object* axis changes object appearance and size. Notably, the most distinctive axes are *Position* and *Task*. *Position* swaps the relative locations of the pick and place objects, going beyond a simple initial-position shift: the policy must infer the operational relation between the two objects and recompute the corresponding action sequence. *Task* keeps the scene fixed but gives a novel instruction that asks the policy to manipulate a different target object, directly testing zero-shot novel instruction following. PRTS reaches 58.8% average SR, improving over $\pi_{0.5}$ by $+5.5$ points. The most significant improvement appears on *Task*: PRTS reaches 31.5%, while $\pi_{0.5}$ drops to 0.8% and the second-best VLA reaches only 22.3%, establishing a substantial absolute margin of 9.2 points. PRTS also leads on *Position* (24.3% vs. 20.8% for $\pi_{0.5}$). Especially, we provide a more detailed numerical results on these two axes in Table X in Appendix C. Together, these two axes highlight the core advantage of our pre-training design: rather than merely imitating demonstration trajectories, PRTS learns goal-conditioned representations that encode which state-action pairs are truly reachable under a given language instruction. This enables goal-reachability-aware decision making even when object layouts change or entirely new task instructions are introduced, precisely the capability targeted by contrastive RL pre-training.

4) SimplerEnv with Visual Matching: Table V evaluates PRTS on the WidowX robot in SimplerEnv, covering three pick-and-place tasks and one stacking task. PRTS is post-trained on the real-world BridgeData V2 dataset [57] with the same global batch size and number of gradient steps as GR00T-N1.5, the strongest prior result on this benchmark. PRTS reaches 77.1% average SR, improving over the strongest prior result GR00T-N1.5 by $+15.2$ points and over starVLA based on the same VLM backbone Qwen3-VL-4B-Instruct by $+16.2$ points. Notably, PRTS also reaches 100% on *Put Eggplant in Yellow Basket* while leading on *Put Spoon on Towel* and *Stack Green Block on Yellow Block*. These results show that PRTS can also achieve strong performance on a robot embodiment different from LIBERO, demonstrating its adaptability and generality across embodiments.

5) Real-World Evaluation: **RealMan dual-arm system.** Fig. 4a reports success rates (SR) on the 11-task RealMan Dual-Arm suite. To rigorously compare PRTS against other models, we deliberately deviate from the standard per-task fine-tuning paradigm. Instead, we post-train and evaluate a single shared checkpoint across all eleven tasks. This mixed-task post-training setting introduces inevitable cross-task interference, demanding a robust language-conditioned representation to maintain consistent goal-directed behavior across tasks. Furthermore, it establishes the foundation for the zero-shot instruction evaluation in Sec. VI-D.

Since all models share identical post-training data and recipes, the performance gap directly reflects the quality of the pre-trained representations. Overall, PRTS achieves a 95.9% average SR, outperforming $\pi_{0.5}$ (85.5%) and π_0 (67.3%). The advantage of PRTS is especially prononced on tasks requiring precise contact (e.g., *Stack Cups*, *Flip Tennis Tube*) and long-horizon manipulation. Notably, on the two-minute *Office Long Term* task, $\pi_{0.5}$ suffers from multi-task interference and collapses to 40% SR, whereas PRTS maintains a robust 95%. This contrast confirms that CRL pre-training injects a temporally consistent goal-reachability prior, preserving stable execution throughout long-horizon rollouts.

Flexiv single-arm system. To further assess the model's capability for fine-grained, contact-rich manipulation, we evaluate PRTS and baselines on the Flexiv platform under a per-task fine-tuning scheme. This benchmark emphasizes precise

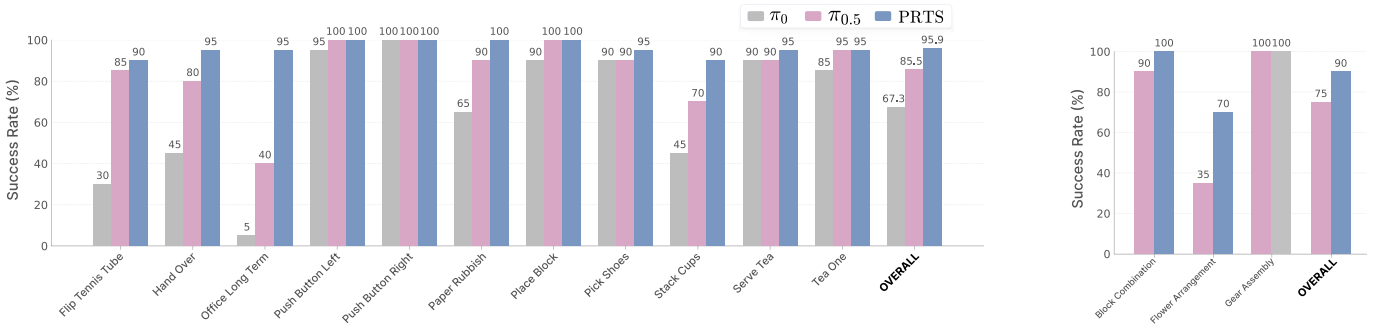
[1]Here we do not include the *Environment* perturbation because the official codebase has not yet released the corresponding assets.

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy [11]	78.5	87.5	73.5	64.8	76.1
OpenVLA [26]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [45]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [68]	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 [40]	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 [33]	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 [27]	97.6	98.4	97.9	94.5	97.1
X-VLA [70]	98.2	98.6	97.8	**97.6**	98.1
π_0-Fast [44] (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 [51] (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 [62] (bs=32, 30K steps)	<u>98.6</u>	<u>99.6</u>	<u>98.6</u>	94.8	97.9
ABot-M0 [62] (bs=64, 40K steps)	**98.8**	**99.8**	**99.0**	<u>96.6</u>	**98.6**
PRTS (Ours) **(bs=32, 30K steps)**	**98.8**	**99.8**	98.4	<u>96.6</u>	<u>98.4</u>

TABLE II: **Evaluation results on the LIBERO benchmark.** Post-training budget is indicated in gray next to the model names. Best results are in bold, and second-best results are underlined. PRTS matches the strongest prior results with global batch size 32 and 30K post-training steps; ABot-M0 under the same budget is included for a controlled comparison.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Average
OpenVLA [26]	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 [22]	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 [53]	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT [27]	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
π_0 [4] (bs=32, 30K steps)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast [44] (bs=32, 30K steps)	<u>65.1</u>	21.6	61.0	73.2	73.2	74.4	68.8	61.6
Qwen3-VL-PI [51] (bs=32, 30K steps)	43.6	46.6	81.8	88.4	89.4	70.1	74.7	68.9
ABot-M0 [62] (bs=32, 30K steps)	**65.7**	53.1	81.2	**96.8**	<u>95.0</u>	**87.5**	81.3	78.7
$\pi_{0.5}$ [5] (bs=256, 30K steps)	64.0	<u>58.0</u>	<u>88.5</u>	<u>96.6</u>	81.4	**87.5**	**85.9**	<u>80.7</u>
PRTS (Ours) **(bs=32, 30K steps)**	59.5	**72.4**	**89.6**	96.5	**96.7**	<u>81.3</u>	<u>83.3</u>	**81.4**

TABLE III: **Evaluation results on the LIBERO-Plus benchmark.** Zero-shot robustness evaluation of the LIBERO-finetuned checkpoint across seven perturbation axes [21]. Post-training budget is indicated in gray when available. Best results are in bold, and second-best results are underlined.



(a) RealMan task suite.

(b) Flexiv task suite.

Fig. 4: **Real-world evaluation.** In both suites, per-task success rates averaged over 20 real-world trials are reported. **(a) RealMan dual-arm task suite.** PRTS achieves 95.9% average SR and reaches at least 90% on all 11 tasks, outperforming $\pi_{0.5}$ and π_0 under the same evaluation protocol. **(b) Flexiv single-arm task suite.** PPRTS outperforms the baselines in all three tasks spanning spatial-aligned and fine-grained and contact-rich manipulation.

Method	Semantic	Object	Position	Task	Average
OpenVLA [26]	**97.2**	93.0	0.0	0.0	47.6
Qwen3-VL-PI [51] (bs=32, 30K steps)	95.5	70.4	4.3	3.9	43.5
ABot-M0 [62] (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 IV: **Evaluation results on the LIBERO-Pro benchmark.** Zero-shot generalization evaluation of the LIBERO-finetuned checkpoint across semantic, object, position, and task variations [72]. Post-training budget is indicated in gray when available. Best results are in bold, and second-best results are underlined.

execution through three challenging tasks: *Gear Assembly*—sequential high-precision insertion of three gears with varying sizes, *Block Combination*—fine-grained spatial alignment, and *Flower Arrangement*—a compounding long-horizon task requiring sequential placement of two distinct flowers. As shown in Fig. 4b, PRTS achieves a 90.0% average success rate, outperforming $\pi_{0.5}$ by +15.0%. While both models reach saturated performance on *Gear Assembly* (100.0%), PRTS demonstrates stronger spatial grounding on *Block Combination* (100.0% vs. 90.0%), and significantly alleviates the compounding execution errors that cause $\pi_{0.5}$ to struggle on the temporally extended *Flower Arrangement* sequence (70.0% vs. 35.0%). These results suggest that CRL pre-training equips the model with a stronger goal-reachability-aware representation, enabling more reliable action generation under both fine-grained spatial constraints and long-horizon sequential dependencies.

C. Effect of Contrastive RL

After the overall comparison, we now examine how much of PRTS's performance comes from the contrastive RL objective used during pre-training. To isolate this effect, we train a matched variant of PRTS using the same architecture, datasets, and training recipe as in Sec. VI-A, except that the CRL coefficient in Eq. (15) is set to $\lambda_{\text{crl}} = 0.0$ during pre-training. After pre-training, we attach the same flow-matching action expert to both PRTS and its non-CRL variant, and post-train them on standard LIBERO with global batch size 32 for 30K gradient steps. We evaluate the resulting models on standard LIBERO and further test them zero-shot on LIBERO-Plus and LIBERO-Pro without benchmark-specific adaptation. The results are shown in Table VI.

On standard LIBERO, CRL improves the average SR from 97.8% to 98.4% under the same post-training budget. The strongest improvements appear on *Spatial* and *Long* (both 1.0%), suggesting that the CRL-shaped representations benefit relational grounding and long-horizon execution on the in-distribution benchmark.

The advantage becomes much clearer under zero-shot evaluation on LIBERO-Plus. Specifically, CRL improves the average SR by 4.9%, demonstrating substantial gains along the *Robot* (16.3%) and *Noise* (9.0%), alongside notable improvements in the *Camera* (2.8%) and *Language* (2.3%). Across these permutation dimensions, the policy must execute the intended task consistently despite shifts in proprioceptive initialization, perceptual inputs, or language instructions. Under such distribution shifts, successful execution relies not on memorizing trajectories from post-training, but rather on preserving the language-conditioned goal and effectively re-grounding it within the perturbed observation stream. By explicitly aligning state representations with high-level task semantics during pre-training, CRL equips the policy with this capability. Consequently, PRTS w/ CRL exhibits enhanced robustness when the path from observation to action is disturbed while the underlying task semantics remain fixed. These results supports our hypothesis that contrastive pre-training inherently strengthens the goal-conditioned representations leveraged by the downstream action expert, moving beyond mere action imitation on the nominal training distribution.

The evaluation on the LIBERO-Pro further corroborates these findings by testing the model's capacity for zero-shot generalization under severe semantic and structural shifts.

Method	Put Spoon On Towel	Put Carrot On Plate	Stack Green Block On Yellow Block	Put Eggplant In Yellow Basket	Average
OpenVLA [26]	0.0	0.0	0.0	4.1	1.0
RT-1-X [12]	0.0	4.2	0.0	0.0	1.1
OpenVLA-OFT [27]	34.2	30.0	30.0	72.5	41.8
SpatialVLA [45]	16.7	25.0	29.2	**100.0**	42.7
CogACT [30]	71.7	50.8	15.0	67.5	51.3
F1-VLA [40]	50.0	**70.8**	50.0	66.7	59.4
Qwen3-VL-PI [51]	<u>78.1</u>	46.9	30.2	<u>88.5</u>	60.9
GR00T-N1.5 [3]	75.3	54.3	<u>57.0</u>	61.3	<u>61.9</u>
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 [44]	29.1	21.9	10.8	66.6	48.3
PRTS (Ours)	**83.3**	<u>66.6</u>	**58.3**	**100.0**	**77.1**

TABLE V: **Evaluation results on the SimplerEnv WidowX benchmark (Visual Matching).** All numbers are success rates (%). Best results are in bold, and second-best results are underlined.

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 VI: **Effect of contrastive RL pre-training across the LIBERO benchmark family.** Both variants are post-trained on the same standard LIBERO dataset and evaluated zero-shot on LIBERO-Plus and LIBERO-Pro. All numbers are success rates (%), and Δ reports PRTS w/ CRL minus PRTS w/o CRL.

Here, CRL yields a 5.0% increase in average SR, driven primarily by gains on the *Position* (10.5%) and *Task* (11.4%). Unlike lower-level visual variations, these dimensions demand a higher degree of cognitive flexibility. They require the policy to actively reinterpret redefined instruction goals, identify new target relations, and synthesize novel action sequences, rather than simply retrieving and replaying original demonstration trajectories. The suite-level breakdown in Table VII provides a more detailed view. CRL effectively mitigates the severe performance degradation observed in the non-CRL variant under complex re-grounding scenarios. For example, when evaluating on novel spatial configurations (*Object-Position*), the SR improves from 4.8% to 36.0%. Furthermore, CRL yields substantial gains in adapting to 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 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 [56]. Specifically, we sample trajectories from

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 VII: **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 Δ denotes PRTS w/ CRL minus PRTS w/o CRL.

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. 5.

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 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.

D. Controlled Real-World Generalization

The real-world results above evaluate whether PRTS can execute a broad set of tasks under the standard deployment setting. We next ask a stricter question: can the same policy preserve task success when the scene deviates from the post-training distribution in controlled ways? To answer Q3, we select four representative RealMan tasks, *Paper Rubbish*, *Place Block*, *Pick Shoes* and *Stack Cups*. These tasks cover bimanual receptacle interaction, precise target placement, object organization with closure and multi-object sequential stacking, respectively.

We evaluate each task under four axes of real-world generalization, visualized in Figs. 6 and 7. We arrange the axes from lower-level perceptual shifts to higher-level semantic shifts. *Illumination* changes the observation statistics by turning off the large background light behind the robot. *Spatial* moves the interaction objects to positions outside the post-training placement distribution. *Object* replaces the manipulated objects with same-category instances that differ in color or size while keeping the instruction semantics fixed. Finally, *Task* generalization recombines components from training tasks into new language instructions. Unlike object generalization, this axis changes the language-specified goal itself: the required skills have appeared during post-training, but they must be retrieved from different task contexts and recomposed in a new scene. These controlled perturbations serve as a physical counterpart to the distribution shifts studied in LIBERO-Plus and LIBERO-Pro, but expose the policy to real sensor, contact, and workspace changes.

Table IX shows that PRTS is highly robust to visual, layout, and object shifts in the real world. The advantage is consistent across task types: *Paper Rubbish* and *Place Block* remain at 100% success across these three axes, while *Pick Shoes* and *Stack Cups* stay at least 80% despite changes in object appearance and placement. This suggests that PRTS is not simply replaying trajectories tied to a narrow scene configuration, but can preserve the intended manipulation behavior under substantial visual and layout variation.

Task generalization is more challenging because the language instruction itself changes. This is where the gap to prior VLAs becomes largest: PRTS averages 73.8% success on the task-instruction axis, while $\pi_{0.5}$ reaches 35.0% and π_0 reaches only 13.8%. Importantly, these are not arbitrary unseen skills. Each task recombines primitives that appeared during post-training, but places them under a new instruction and scene context. In *Paper Rubbish*, the original trash-can grasp is replaced by grasping a flat wireless mouse from the long-horizon office task; this requires a qualitatively different, deeper and more planar grasp than the crumpled-paper grasp. Prior models tend to execute the original paper-rubbish grasping pattern and fail to stably grasp the mouse, while PRTS transfers the mouse-grasping skill into the new task and reaches 80% success. In *Place Block*, the target object is changed to a soy-sauce bottle that must be grasped at a narrow stable region, while both the

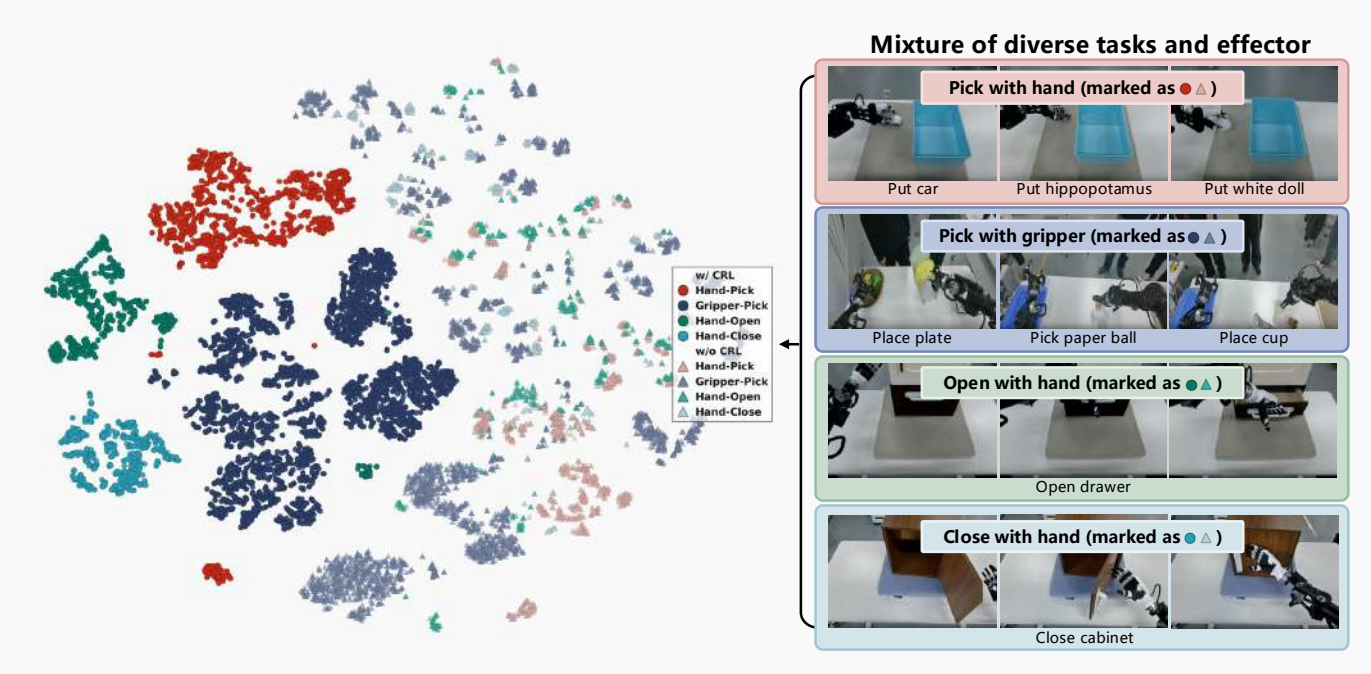


Fig. 5: **T-SNE visualizations of learned task representations.**

bottle and the original block are present. This tests whether the policy follows the language-specified object rather than the scene's habitual block action; PRTS reaches 55%, whereas π_0 mostly continues to grasp the block and $\pi_{0.5}$ often selects the bottle but drops it due to poor grasp placement.

The other two task-generalization cases further test whether the model can preserve task structure while replacing key objects or substeps. In *Pick Shoes*, the two shoes are replaced by a soy-sauce bottle and a paper cup, producing a visually different scene and requiring object-specific grasps before closing the box. π_0 often moves to where the shoes would have been and stalls, and $\pi_{0.5}$ can pick objects but frequently fails to place them into the box. PRTS instead completes the recomposed sequence with 85% success. In *Stack Cups*, the final paper-cup stacking step is replaced by dropping a crumpled paper ball into the third cup. This removes a key visual cue used for localizing the next stacking target and requires the policy to track the current execution stage precisely. PRTS reaches 75%, while prior models show larger precision degradation even in earlier cup-stacking stages. Overall, these results support our central claim that goal-reachability-aware pre-training yields a real-world policy that better preserves the connection between language goals and feasible state-action outcomes under distribution shifts, while also revealing a clear limitation: instruction-level recombination remains one of the hardest forms of real-world generalization.

E. Robustness and Recovery under Human Interventions

The generalization study above changes the initial scene before execution begins. We next consider a more dynamic question: can the policy recover when the world changes *during* execution? This setting is important because real deployments rarely proceed as a clean open-loop rollout. Objects may be moved by people, a grasped item may be taken away and reset, and a nearly completed task may be forced back to an earlier state. To answer Q4, we conduct intervention stress tests on four representative tasks: *Paper Rubbish*, *Pick Shoes*, and *Place Block* on the RealMan dual-arm platform, and *Gear Assembly* on the Flexiv platform.

Figure 8 visualizes the intervention protocol. Rather than only perturbing the initial condition, a human interrupts the rollout after the robot has already formed an intent or started executing a substep. The perturbation can invalidate the current action, reset an object to an earlier state, or change the progress of a multi-stage task. We therefore treat recovery as more than retrying a failed grasp: the policy must observe the new state, abandon stale assumptions about what has already been completed, and continue toward the original instruction. These trials are intended as qualitative stress tests rather than a fully instrumented benchmark, so we report the observed recovery behavior instead of a formal success-rate table.

Across roughly five intervention trials per task, PRTS consistently recovered from human perturbations and completed the task. *Place Block* provides a controlled example with three interruptions: moving the block before grasping, removing it during transport and resetting it to the grasping area, and moving it back after successful placement while the arm is returning home. PRTS responds by tracking the moving target, stopping stale actions when the block is removed, or re-entering the full task procedure after post-completion reset, rather than treating the task as already finished.

The other tasks test the same recovery principle in different

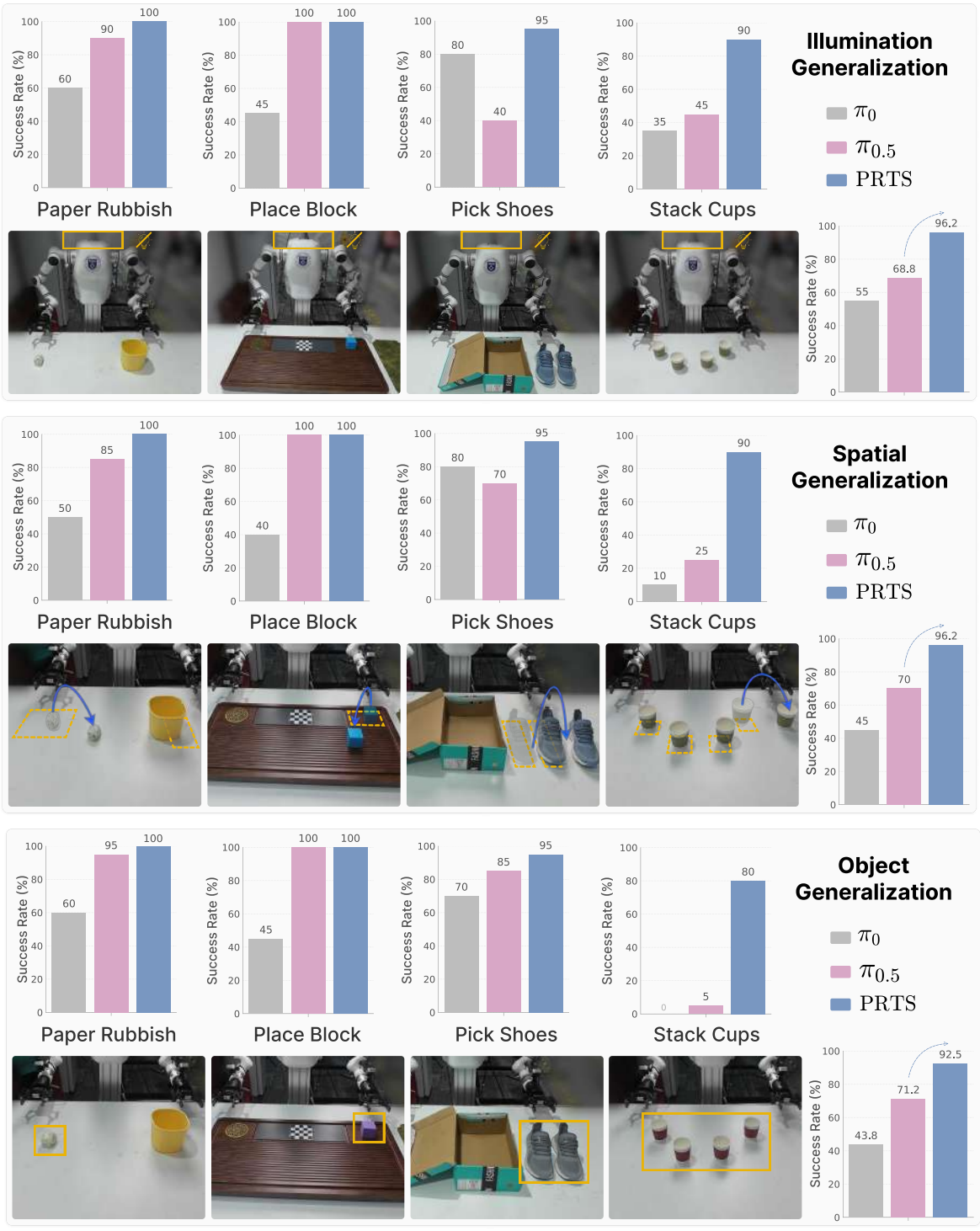


Fig. 6: **Controlled visual, layout, and object shifts on the RealMan platform.** We visualize the three lower-level generalization settings used in the RealMan study. Illumination generalization changes the lighting condition, spatial generalization moves the manipulated objects outside the post-training layout distribution, and object generalization changes the visual instance while preserving the semantic object category and the task instruction. Quantitative results are reported in Table IX.

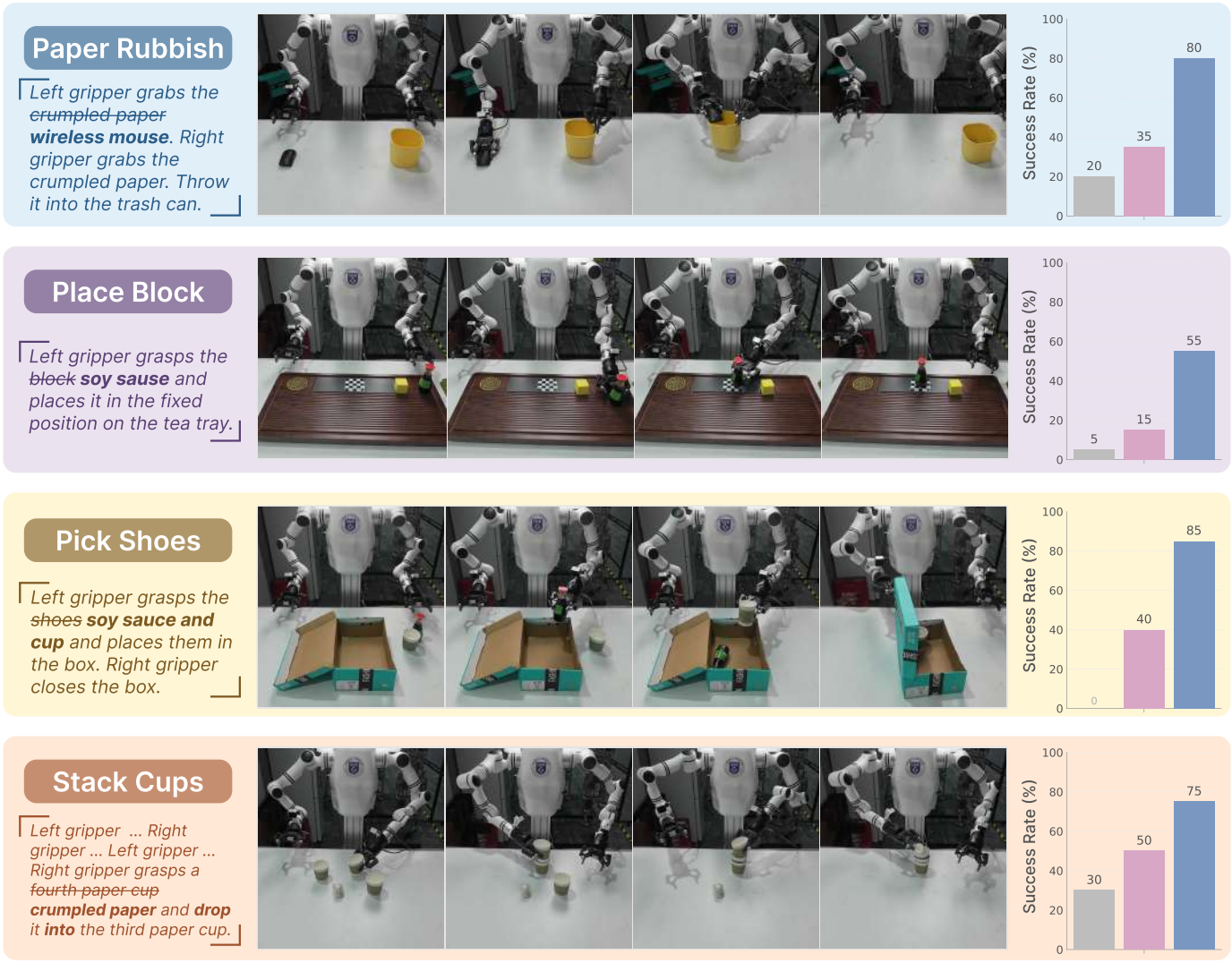


Fig. 7: **Controlled task-level generalization on the RealMan platform.** Task generalization changes the language goal by recombining components of seen tasks, making it semantically stricter than illumination, spatial, or object-level shifts. This setting tests whether the policy can bind a novel instruction to feasible manipulation steps instead of merely recognizing familiar objects or replaying a familiar trajectory. Quantitative results are reported in Table IX.

task structures. *Paper Rubbish* and *Pick Shoes* use similar object-removal and reset interventions, but require recovery in bimanual receptacle interaction and multi-object organization. The policy must re-localize the manipulated objects, infer which subgoals remain unfinished, and redo only the affected stages before completing the original instruction. *Gear Assembly* uses a heavy intervention: after the assembly sequence has been completed, human operators move one of the three gears back to its original position. PRTS still detects the changed state and executes the corresponding gear grasping and placement again, showing that recovery remains goal-conditioned even after apparent task completion and in a contact-rich setting.

The baseline failures reveal why this setting is stricter than standard real-world evaluation. π_0 largely ignores the changed state and continues replaying fixed task phases, so all three *Place Block* perturbations disrupt its execution. $\pi_{0.5}$ is more reactive under simple interruptions, such as tracking a moved object before grasping or retrying the current substep. However, under stronger interventions that revert a nearly completed or completed task to an earlier stage, it often treats tasks as completed and fails to re-enter the necessary earlier subgoals; this failure is especially clear in the post-completion *Place Block* reset and the heavy *Gear Assembly* intervention. In contrast, PRTS repeatedly re-selects actions based on the currently reachable path to the language goal. This supports the central role of goal-reachability-aware representations: robustness under intervention requires not only recognizing objects, but also judging which goal-conditioned state-action outcomes remain feasible after the execution history has been invalidated.

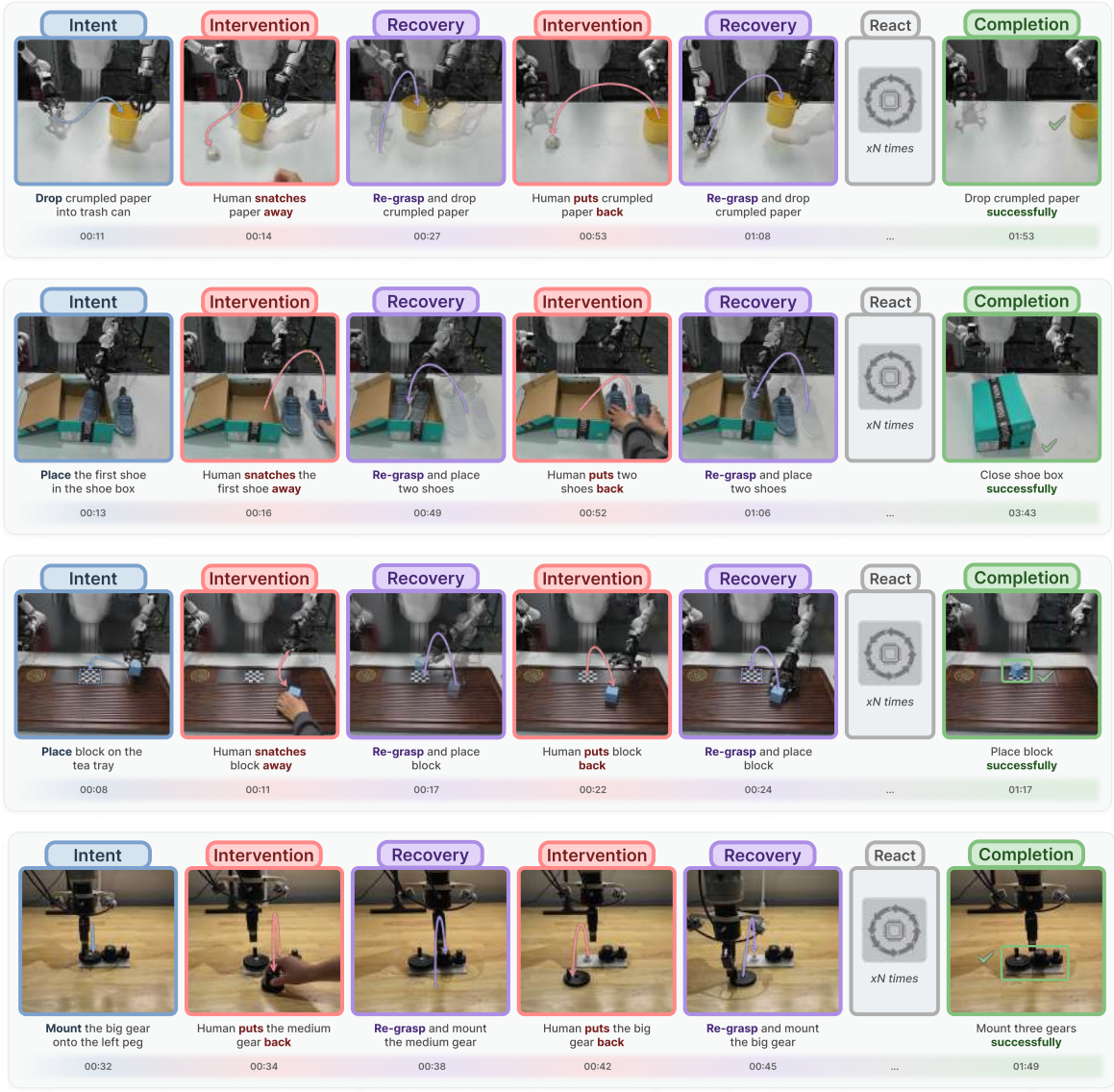


Fig. 8: **Robustness and recovery under human interventions.** We visualize representative intervention rollouts in which a human repeatedly perturbs task-relevant objects after the policy has committed to an action or subgoal. The traces cover bimanual receptacle interaction, multi-object organization, precise placement, and contact-rich assembly. Successful recovery requires the policy to recognize the changed state, re-estimate task progress, and choose the next action according to the still-unreached language goal.

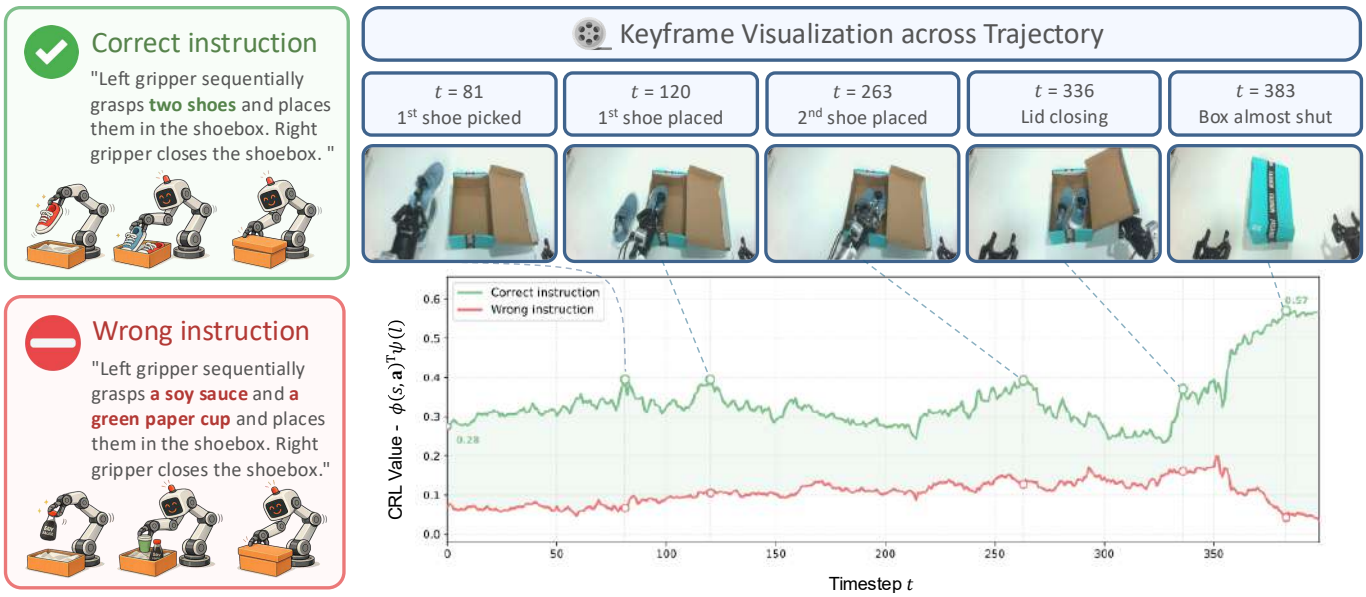


Fig. 9: **CRL value on a held-out *Pick Shoes* trajectory, scored by the pre-trained PRTS with no RealMan task suite fine-tuning.** *Left:* the two instructions used to score the same demonstration—a correct one (green, top) that describes what the trajectory actually does, and a wrong one (red, bottom) that swaps the named grasp targets to objects (soy sauce, green paper cup) that never appear in the scene. *Top right:* five keyframes from the trajectory with their timestep indices, marking the structural milestones of the task (first shoe lifted, first shoe placed, second shoe placed, lid closing, box almost shut). *Bottom right:* the contrastive value $\phi(s_t, \mathbf{a}_t)^\top \psi(l)$ as a function of timestep t, computed under the correct instruction (green) and the wrong instruction (red) on the same state-action sequence; dashed lines link each keyframe to its corresponding marker on the green curve. The green curve forms progress-sensitive local peaks at the five keyframes and rises sharply to 0.57 as the shoebox is closed, while the red curve stays in $0.04 \sim 0.20$ throughout and never crosses the green curve in any frames.

F. CRL Value Visualization

The previous experiments establish that CRL pre-training improves policy robustness under distribution shift. Here we directly inspect its value. Theorem 1 shows that the inner product $\phi(s_t, \mathbf{a}_t)^\top \psi(l)$ tracks the log-transformed goal-conditioned action value $\log Q_l^\pi(s_t, \mathbf{a}_t)$: the score should increase as the trajectory approaches goal completion under the correct instruction, and remain low under a mismatched instruction.

To this end, we evaluate the CRL encoder heads on the **pre-traiend PRTS** before any post-training on RealMan task suite as we mention in Sec. VI-B. Demonstrations from the eleven RealMan tasks are therefore out-of-distribution for both the backbone and the contrastive heads, which directly tests whether the CRL-shaped representation captures goal-reachability semantics on entirely new trajectories without any post-training adaptation. We randomly select a held-out *Pick Shoes* demonstration trajectory and score it under two instructions on the *same* state-action sequence $(s_t, \mathbf{a}_t)$: a correct instruction that describes the actual task (*grasp two shoes and place them into the shoebox*), and a deliberately mismatched instruction that replaces the grasp targets with absent objects (*soy sauce* and *green paper cup*), which is exactly the instruction edit used in the novel-task generalization study (Sec. VI-D). A faithful CRL value should assign consistently higher scores to the correct instruction throughout the episode.

Fig. 9 supports this behavior clearly. Along the *Pick Shoes* trajectory, the value under the correct instruction shows a clear overall upward trend as the rollout progresses toward task completion, with local peaks consistently aligning with key manipulation milestones—firmly grasping the first shoe (t=81), placing it into the shoebox (t=120), placing the second shoe (t=263), and finally closing the lid (t=336 and t=383). In particular, the score rises from 0.28 at the initial frame to 0.57 as the shoebox is nearly closed, with the largest jump concentrated in the final lid-closure phase. Although the curve is not strictly monotonic at every timestep, its major increases are tightly coupled with semantically meaningful task progress rather than tracking superficial visual changes or arm motion alone. This shows that the CRL head captures goal-reaching structure instead of merely reacting to visual changes: the model correctly assesses the goal-reaching feasibility of the current state-action pair on a per-frame basis.

In contrast, the value under the wrong instruction stays consistently low ($0.04 \sim 0.20$) and even drifts downward toward the end of the rollout, because the executed actions are shoe-grasping behaviors that do not move toward a soy-sauce or paper-cup goal at any frame. Notably, the red curve never crosses the green one across all timesteps, and the shaded green band visualizes this persistent margin. This confirms the two key properties expected from CRL: temporal-progress-sensitive value estimation and strong goal-conditioned dis-

criminability without any post-training adaptation.

G. 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.

Sec. III-B introduces the role-aware attention mask for the CRL token blocks and its CuTe-kernel implementation on top of FlashAttention [64], 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 [47].

Kernel microbenchmark. All rows in Fig. 10a 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. 10b plots aggregate tokens/s against GPU count for the full training step, using 4 packed sample 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.

VII. Conclusion

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. VI-G).

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. VI-B–VI-E). A direct examination of the value prediction further confirms that the same backbone produces a goal-reachability-aware and goal-discriminative score on out-of-distribution rollouts without any post-training adaptation (Sec. VI-F).

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 [58]—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. 7) and recovery under human interventions (highlighted in Fig. 8). 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.

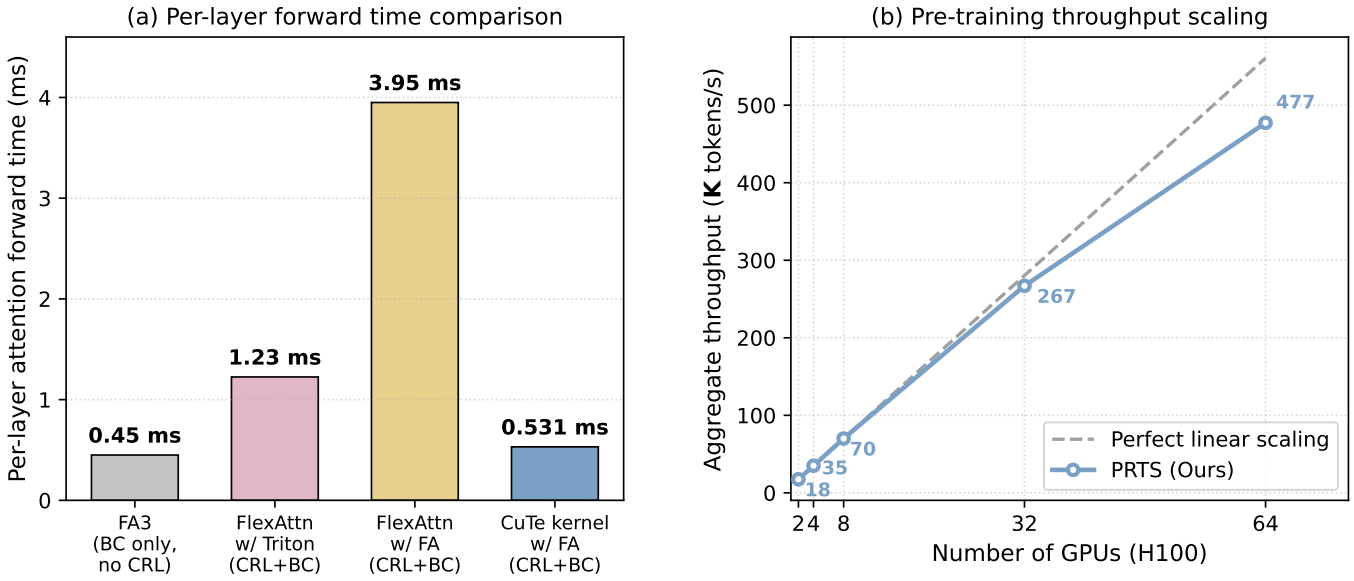


Fig. 10: **Pre-training efficiency of PRTS. (a)** Per-layer attention forward time on 2 H100 GPUs for BC-only FA3, two FlexAttention backends, and the fused CuTe kernel. **(b)** Aggregate token throughput vs. number of H100 GPUs; dashed line: linear scaling from the 2-GPU point.

[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] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[13] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi,

Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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/.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

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

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

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

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

[54] 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. pages 12706–12713, 10 2025. doi: 10.1145/3746027.3758209.

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

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

[57] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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. 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:

$$\mathcal{L}^{l\to\text{sa}}(\phi,\psi) = \mathbb{E}_{\mathcal{B}\sim\mathcal{D}^B}\left[\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left(-\sum_{j\in\mathcal{S}(l_i;\mathcal{B})} q_{ij}\log\frac{\exp(\psi(l_i)^\top\phi(s_j,a_j))}{\sum_{k\in\mathcal{B}}\exp(\psi(l_i)^\top\phi(s_k,a_k))}\right)\right], \tag{18}$$

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:

$$\mathcal{L}_l = -\sum_{r=1}^{m} q_r \log p_r, \tag{19}$$

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:

$$\begin{aligned}\psi(l)^\top\phi(s_{j_r},a_{j_r}) - \log Z &= \log q_r \\ &= (T_{j_r} - t_{j_r})\log\gamma - \log\left(\sum_{r'}\gamma^{T_{j_{r'}}-t_{j_{r'}}}\right),\end{aligned} \tag{20}$$

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:

$$\begin{aligned}\psi(l)^\top\phi(s_{j_r},a_{j_r}) = (T_{j_r} - t_{j_r})\log\gamma \\ + \underbrace{\left(\log Z - \log\sum_{r'}\gamma^{T_{j_{r'}}-t_{j_{r'}}}\right)}_{\triangleq C(l,\mathcal{B})}.\end{aligned} \tag{21}$$

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:

$$\begin{aligned}Q_l^{\pi^*}(s_t,a_t) &= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k\mathbb{I}[s_{t+k} = s_g] &(22)\\ &= (1-\gamma)\gamma^{T-t}, &(23)\end{aligned}$$

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:

$$(T-t)\log\gamma = \log Q_l^{\pi^*}(s_t,a_t) - \log(1-\gamma). \tag{24}$$

Substituting back:

$$\psi(l)^\top\phi(s,a) = \log Q_l^{\pi^*}(s,a) + \underbrace{(C(l,\mathcal{B}) - \log(1-\gamma))}_{\triangleq C(l)}. \tag{25}$$

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. ■

a) Comparison with Standard CRL.: Standard CRL [19] 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.

B. Detailed real-world task descriptions and illustrations

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. VI-A; here we focus on the task content, object arrangements, and the manipulation behavior required by each instruction.

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.

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

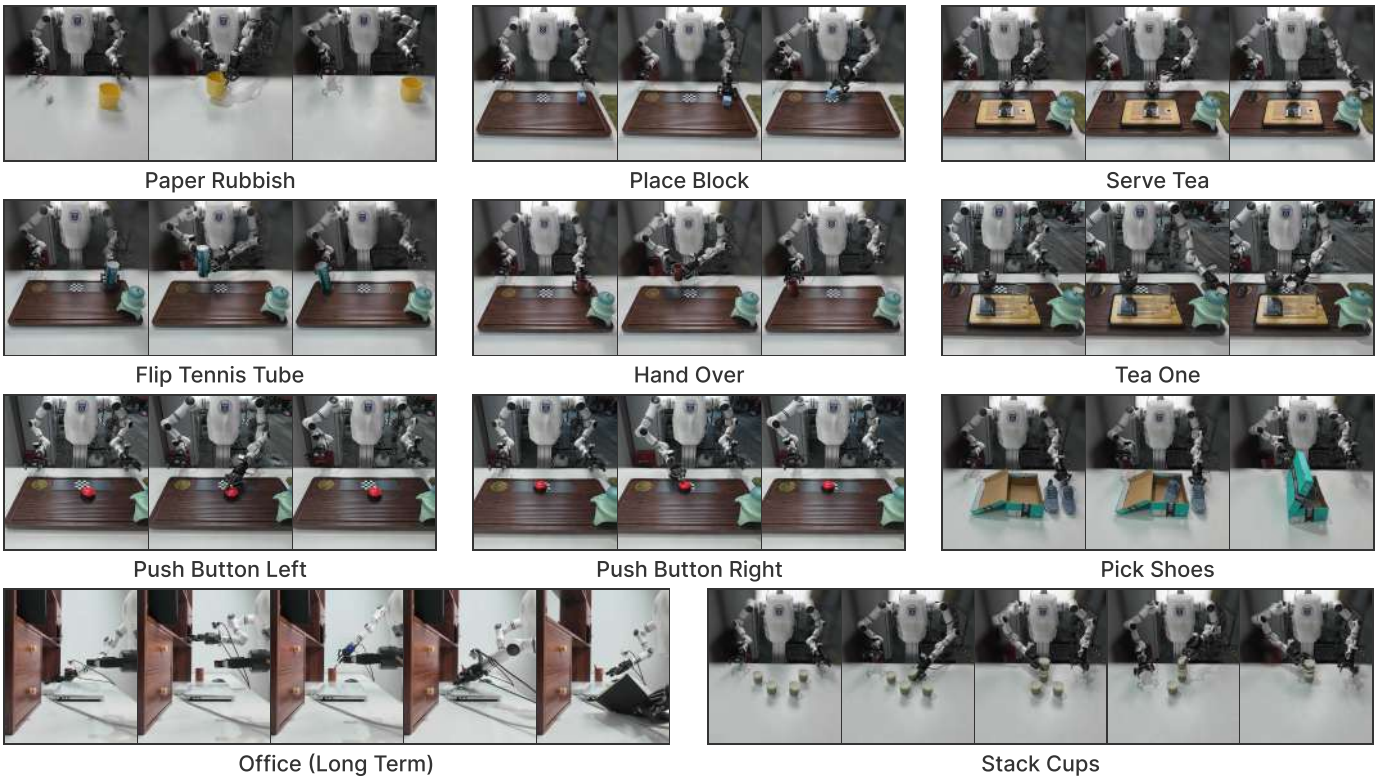


Fig. 11: **RealMan task suite.** We evaluate PRTS on eleven RealMan tasks, including ten short-horizon desktop manipulation tasks and one long-horizon office task. The montage visualizes representative rollouts for the task settings described in Appendix B.

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 single-arm tasks that require accurate geometric placement and contact-rich manipulation.

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.

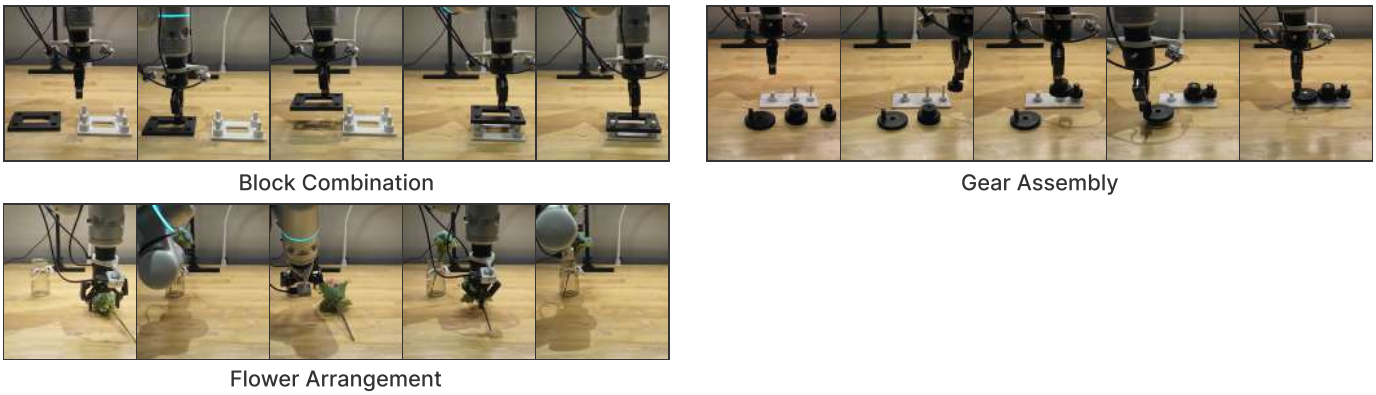


Fig. 12: **Flexiv task suite.** The Flexiv evaluation complements the RealMan suite with single-arm manipulation tasks that emphasize precise alignment, assembly, and insertion-like object placement.

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 VIII: **Real-world evaluation on the RealMan dual-arm platform.** Per-task success rates over 20 real-world trials. PRTS achieves 95.9% average SR and reaches at least 90% on all 11 tasks, outperforming $\pi_{0.5}$ and π_0 under the same evaluation protocol.

C. Detailed results on LIBERO-Pro Task and Position variations

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 IX: **Controlled real-world generalization on the RealMan dual-arm platform.** We report success rates under illumination (*Light*), spatial position (*Pos*), object instance (*Obj*), and task-instruction (*Task*) shifts for four representative RealMan tasks. The task-instruction axis is the most semantically demanding since the language goal changes rather than only the visual scene.

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.)
OpenVLA [26]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen3-VL-PI [51] (bs=32, 30K steps)	14.4	0.0	0.2	0.0	1.2	8.8	1.2	6.8
ABot-M0 [62] (bs=32, 30K steps)	13.4	<u>54.8</u>	9.0	<u>9.8</u>	5.6	10.6	0.2	<u>14.0</u>
π_0 [4] (bs=32, 30K steps)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\pi_{0.5}$ [5] (bs=256, 30K steps)	<u>20.0</u>	1.0	17.0	1.0	**38.0**	0.0	<u>8.0</u>	1.0
PRTS (Ours) (bs=32, 30K steps)	**26.8**	**62.2**	**36.0**	8.8	19.0	**33.8**	**15.4**	**21.2**

TABLE X: **Detailed performance comparison on the *Position* and *Task* variations of LIBERO-Pro benchmark.** Since VLAs are trained on a fixed instruction distribution, they perform poorly under task perturbations. While conventional VLAs suffer catastrophic performance drops under these distributional shifts, our PRTS model exhibits superior generalization capabilities over instruction following. The best results are highlighted in bold and second best results are underlined.