GTPO: GROUP TEACHING POLICY OPTIMIZATION VIA TEMPORAL OPTIMAL TRANSPORT IN VLA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

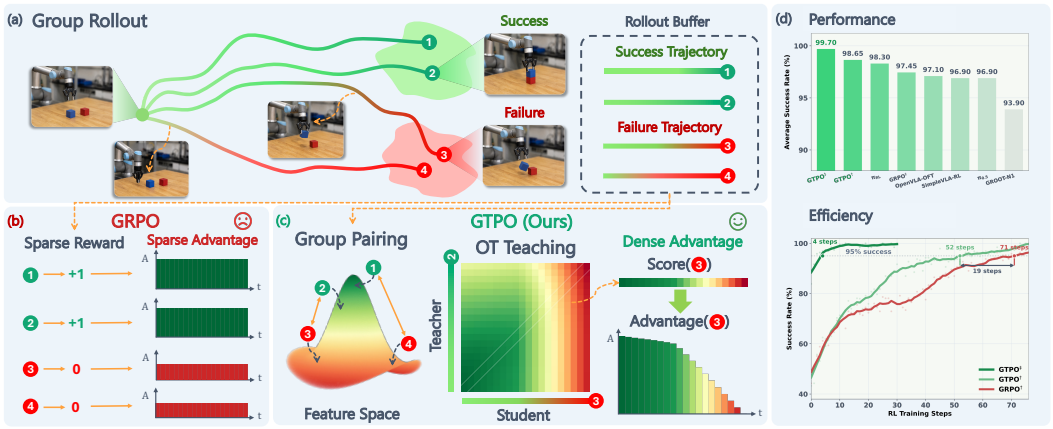

Figure 1: **GTPO overview. (a)** From the same initial environment state, the policy samples a group of successful and failed trajectories and stores them in the rollout buffer. **(b)** GRPO broadcasts each sparse trajectory-level advantage uniformly across all action chunks. **(c)** GTPO pairs each failure with its nearest successful rollout in visual feature space and uses temporal OT alignment to produce a dense advantage over time. **(d)** This dense credit assignment improves both average success rate and RL training efficiency under the same rollout budget and policy initialization.

ABSTRACT

Reinforcement learning can improve vision-language-action policies beyond their demonstrations, but costly robot interactions and sparse terminal feedback make this process inefficient. Reliable dense supervision could extract more useful information from each interaction, yet learned reward models often require substantial additional data construction and training. Can the policy's own successful and failed rollouts provide this supervision? We propose Group Teaching Policy Optimization (GTPO), which turns successful rollouts into teachers for failed attempts at the same task from similar initial states. Building on Group Relative Policy Optimization (GRPO) and drawing inspiration from inverse reinforcement learning, GTPO uses trajectory comparison to refine otherwise uniform trajectory-level advantages. Each failed rollout is paired with its nearest successful in-group teacher using the policy's visual features. Temporally masked optimal transport aligns their trajectories and produces normalized action-chunk weights that redistribute the original GRPO advantage while retaining terminal rewards. The teachers evolve with the policy, requiring no additional expert reference trajectories or separately trained reward model for teaching. We evaluate GTPO on LIBERO, RoboTwin, and five challenging real-world tasks requiring fine manipulation. Across four LIBERO suites, GTPO achieves 98.65% average success with one demonstration per task and 99.70% with the full demonstration set, requiring at least 22.2% fewer aggregate training steps to reach 95% success than matched GRPO. On the five real-world tasks, GTPO achieves 80–100% success after three on-policy rounds, exceeding GRPO by 10–35 percentage points. Project website: `https://gtpo-project.github.io/gtpo_page/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Vision-language-action (VLA) models provide a general interface from observations and language instructions to robot actions (Kim et al., 2024), but complex, fine-grained manipulation still requires task-specific post-training. Reinforcement learning (RL) offers a promising approach because interaction allows a policy to improve beyond its demonstrations (Hu et al., 2025; Tan et al., 2025; Li et al., 2026; Zang et al., 2025). Realizing this potential efficiently, however, is difficult in robotics, where interactions are costly and feedback is often limited to sparse terminal outcomes. Success or failure alone provides little guidance about which grasp, alignment, or placement decisions contributed to the result, making long-horizon policy optimization inefficient and unstable. Reliable dense process rewards or fine-grained credit assignment signals are therefore key to extracting more useful supervision from each costly interaction with the physical environment.

Obtaining reliable dense process-level supervision remains a significant challenge. Learned reward models often require a separate data construction and training pipeline beyond policy optimization (Lee et al., 2026; Chen et al., 2026; Fang et al., 2026). Data collection or synthesis, reward supervision, and model training introduce substantial overhead. These costs motivate us to consider the experience already available during policy learning: *can successful and failed rollouts generated by the policy itself provide dense process-level supervision, without additional external data for reward learning or the overhead of training a separate reward model?*

Our starting point is that successful rollouts can serve as teachers for failed attempts at the same task from similar initial states. Following GRPO's group-based formulation (Shao et al., 2024), we organize these attempts into rollout groups, providing behavioral references that evolve with the policy. However, standard GRPO broadcasts a single outcome-based advantage across each trajectory, leaving temporal differences between successful and failed behavior unused. We draw inspiration from inverse RL and trajectory matching (Ng & Russell, 2000; Abbeel & Ng, 2004; Zakka et al., 2021; Haldar et al., 2023; Luo et al., 2023; Fu et al., 2024) to exploit these differences, using successful in-group rollouts as references for finer-grained guidance along failed trajectories.

Turning this idea into useful supervision requires establishing temporal correspondence. Two attempts may execute at different speeds, pause at different stages, or terminate at different times. Comparing observations at identical timesteps can therefore confuse timing differences with differences in task progress. Matching only by visual similarity can also associate observations from temporally distant stages. A useful comparison must accommodate local variations in execution timing while restricting unrelated matches, allowing successful behavior to guide updates along a failed trajectory.

We propose **Group Teaching Policy Optimization** (GTPO) to implement this form of temporal teaching. Using the policy's visual features, GTPO selects the nearest successful in-group teacher for each failure according to an optimal transport (OT) distance. Temporally masked Sinkhorn OT aligns the trajectories, and alignment costs yield normalized action-chunk weights that reshape the failed trajectory's original GRPO advantage. The terminal reward is retained, while successful trajectories keep uniform weights. OT thus acts as a *credit allocator*, requiring no additional expert reference trajectories or separately trained reward model for teaching. SFT initialization and real-world corrective data remain components of the overall policy learning pipeline in our experiments.

We evaluate GTPO across four LIBERO suites, RoboTwin manipulation tasks, and five challenging real-world tasks requiring fine manipulation. On LIBERO, GTPO reaches 98.65% average success with one demonstration per task and 99.70% with the full demonstration set. It requires at least 22.2% fewer aggregate training steps to reach 95% success than matched GRPO. On the five real-world tasks, GTPO achieves 80–100% success after three on-policy rounds, exceeding GRPO by 10–35 percentage points. Ablations assess temporal alignment and teacher selection.

Our contributions are threefold: (1) an on-policy group-teaching framework that reuses successful rollouts as adaptive teachers for failed attempts, extracting supervision from the policy's own experience without additional expert reference trajectories or reward-model training; (2) a temporally masked OT mechanism that accommodates local differences in execution timing and converts visual alignment costs into normalized action-chunk weights, refining GRPO's trajectory-level advantages while retaining terminal rewards; and (3) evaluations across LIBERO, RoboTwin, and five challenging real-world manipulation tasks that demonstrate gains in success and training efficiency, together with controlled ablations of temporal alignment, teacher selection, and credit concentration.

2 Related Work

Reinforcement Learning for Robot Manipulation Robot RL combines demonstrations with online interaction to improve sample efficiency. RLPD mixes prior data with online replay (Ball et al., 2023); SERL adds learned rewards and automated resets (Luo et al., 2024a); and HIL-SERL uses human interventions for contact-rich manipulation (Luo et al., 2024b). RLDG distills specialist experience into generalist policies (Xu et al., 2025), while FLaRe scales RL fine-tuning to pretrained transformers (Hu et al., 2025). Together, these systems show that interaction can refine pretrained visuomotor representations while preserving the observation and action interfaces required for scalable VLA post-training.

RL has also become a post-training stage for VLA models. GRAPE learns from stage-wise preferences (Zhang et al., 2024); RIPT-VLA combines dynamic sampling with sparse binary feedback (Tan et al., 2025); SimpleVLA-RL studies scalable online optimization across manipulation benchmarks (Li et al., 2026); and RLinf-VLA provides unified PPO/GRPO infrastructure across VLAs and simulators (Zang et al., 2025). These works demonstrate the value of interaction, whereas GTPO addresses the complementary problem of temporal credit assignment in critic-free group-based RL. In particular, sparse outcome-based methods still assign the same rollout-level signal to every action, while critic-based alternatives require learning an additional value function from limited robot interaction. GTPO retains GRPO's critic-free optimization but introduces structure within each trajectory by comparing rollouts generated under matched initial conditions, reducing variation due to scene configuration.

Dense Reward Methods in Robotic Reinforcement Learning Dense reward learning provides intermediate supervision by estimating task progress from richer signals. REDS uses segmented videos and subtask conditioning (Kim et al., 2025a); ReWiND models progress and regression from language and rewound demonstrations (Zhang et al., 2025); and DEMO^3 jointly learns multi-stage rewards, policies, and world models (López Escoriza et al., 2025). For real-robot learning, VLAC predicts progress from paired observations (Zhai et al., 2025). Other methods exploit robot corpora, procedural descriptions, counterfactual failures, or learned dynamics (Lee et al., 2026; Wu et al., 2026; Feng et al., 2026; Chen et al., 2026; Lv et al., 2026; Fang et al., 2026). RoboReward, for example, uses large robot corpora and counterfactual negatives, whereas procedural reward models ground feedback in predicted task stages. Dynamics-aware approaches additionally estimate whether an observation represents physically plausible progress. Although these methods can produce detailed process supervision, they require a learned or prompted reward mechanism whose errors may alter the optimization objective. GTPO trains no reward model: it retains the terminal task reward and uses successful group members to reweight its advantage across time. This changes the policy-update weighting without introducing a separately learned progress reward.

Inverse Reinforcement Learning and Temporal Alignment IRL recovers a reward or progress signal from expert behavior when task rewards are unavailable or sparse. XIRL learns visual progress across embodiments (Zakka et al., 2021), while OT-based imitation aligns learner and expert trajectories or occupancy measures (Haldar et al., 2023; Luo et al., 2023). TemporalOT (Fu et al., 2024) introduces temporal locality into OT and converts similarity to a fixed expert demonstration into per-step rewards. GTPO transfers this principle to online group-based RL but differs in two respects: the nearest successful rollout in the current group acts as an adaptive teacher, and OT allocates an existing GRPO advantage instead of defining a new reward. The teacher thus improves together with the policy and needs no additional expert trajectory for OT teaching. Because unconstrained OT is insensitive to sequence order, a temporal support mask restricts matches between distant manipulation phases; it does not establish causal correspondence or enforce strict monotonicity.

3 Method

3.1 Problem Formulation

Given the current observation o_t and a language instruction ℓ, a policy π_θ predicts an action chunk

$$a_t = \left(a_t^0, \ldots, a_t^{n-1}\right) \sim \pi_\theta(\cdot \mid o_t, \ell)\,, \tag{1}$$

where n is the chunk size. The robot executes a_t, interacts with the environment, and receives the next observation o_{t+1} until reaching the goal or horizon T.

Both successful and failed trajectories are stored in the rollout buffer. A successful trajectory that reaches the goal at step $g \leqslant T$ and a failed trajectory that exhausts the horizon are, respectively,

$$\zeta^{\text{succ}} = \left(o_0, a_0, o_1, a_1, \ldots, o_{g-1}, a_{g-1}, o_g\right), \qquad \zeta^{\text{fail}} = \left(o_0, a_0, o_1, a_1, \ldots, o_{T-1}, a_{T-1}, o_T\right). \tag{2}$$

Each trajectory receives only a terminal reward indicating goal completion or horizon exhaustion,

$$r(\zeta) = \begin{cases} r_{\text{succ}} > 0\,, & \text{goal reached within } T\,, \\ r_{\text{fail}} < 0\,, & \text{otherwise}\,. \end{cases} \tag{3}$$

Training maximizes the expected terminal reward over tasks and policy-generated trajectories:

$$\theta^\star = \operatorname*{argmax}_\theta \; J(\theta)\,, \qquad J(\theta) = \mathbb{E}_{\ell,\, \zeta\sim\pi_\theta}\big[r(\zeta)\big]\,. \tag{4}$$

3.2 Preliminaries: Group Relative Policy Optimization

GRPO (Shao et al., 2024) avoids a value model by sampling G rollouts from the same initial state and normalizing their terminal rewards within the group:

$$A_i = \frac{r_i - \mu_g}{\sigma_g + \varepsilon_{\text{std}}}\,, \qquad \mu_g = \frac{1}{G}\sum_{j=1}^{G} r_j\,, \qquad \sigma_g = \sqrt{\frac{1}{G-1}\sum_{j=1}^{G}\big(r_j - \mu_g\big)^2}\,, \tag{5}$$

where $\varepsilon_{\text{std}} > 0$. With importance ratio $\rho_{i,t} = \pi_\theta(a_{i,t} \mid o_{i,t}, \ell)/\pi_{\theta_{\text{old}}}(a_{i,t} \mid o_{i,t}, \ell)$, its clipped objective is

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{\sum_{i,t} m_{i,t}} \sum_{i=1}^{G}\sum_{t=0}^{T_i-1} m_{i,t}\, \min\Big(\rho_{i,t}A_i,\; \text{clip}\big(\rho_{i,t}, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}}\big)A_i\Big)\,, \tag{6}$$

where T_i is the rollout length and $m_{i,t}$ masks padding. The same A_i is broadcast to every chunk, providing no within-trajectory credit variation (Figure 1(b)). GTPO instead derives a temporally shaped $\widehat{A}_{i,t}$ from same-state grouped rollouts while retaining this critic-free clipped-loss form.

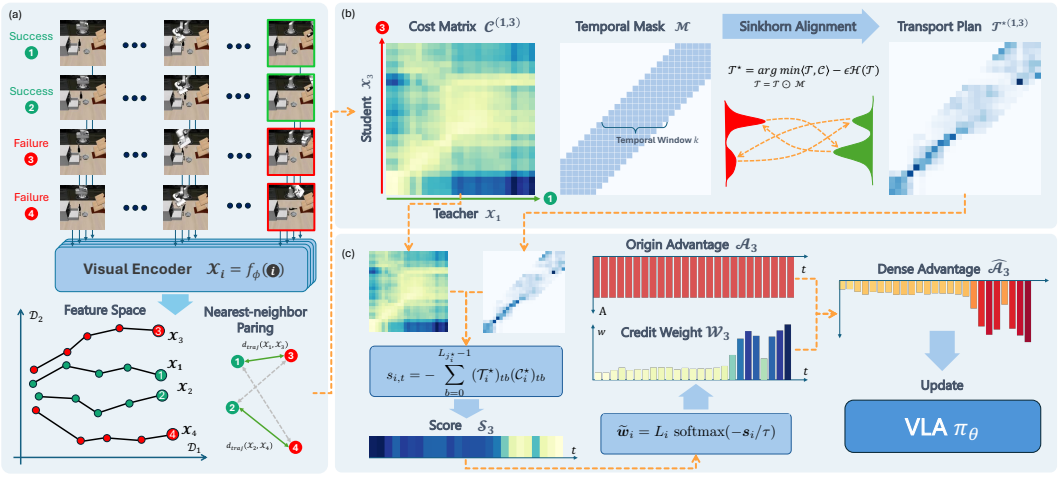

Figure 2: **GTPO pipeline. (a)** Rollouts are encoded as visual feature trajectories, and each failed rollout is paired with its nearest successful teacher. **(b)** A temporally masked Sinkhorn plan aligns the paired trajectories. **(c)** The alignment produces chunk-level credit weights that reshape the original GRPO advantage for policy optimization without altering the rollout collection procedure.

3.3 Nearest-Neighbor Group Teaching

We encode each rollout, select its nearest in-group success as teacher, and align the pair with temporal OT (Figure 2(a)).

Visual features and pairwise cost. For T_i valid action chunks, the policy encoder f_ϕ supplies the shifted feature sequence

$$x_{i,t} = f_\phi(o_{i,t}), \qquad \mathcal{X}_i = \big(x_{i,1}, \ldots, x_{i,T_i-1}\big) \in \mathbb{R}^{L_i \times D}, \quad L_i = T_i - 1. \tag{7}$$

Here $o_{i,t+1}$ follows $a_{i,t}$; the final action has no stored successor feature and retains weight 1. Let $\mathcal{S} = \{j : r_j > 0\}$ and $\mathcal{F} = \{i : r_i < 0\}$ denote successes and failures. For each failed–successful pair $(i, j) \in \mathcal{F} \times \mathcal{S}$, cosine distance gives

$$\mathcal{C}_{ab}^{(i,j)} = 1 - \frac{\langle x_{i,a+1}, x_{j,b+1} \rangle}{\|x_{i,a+1}\| \, \|x_{j,b+1}\| + \delta}, \tag{8}$$

where $0 \leqslant a < L_i$, $0 \leqslant b < L_j$, and $\delta > 0$ ensures numerical stability.

Temporally masked Sinkhorn alignment. Following TemporalOT (Fu et al., 2024), we restrict entropic Sinkhorn alignment (Peyré & Cuturi, 2019; Cuturi, 2013) to a temporal band:

$$\mathcal{M}_{ab}^{(i,j)} = \begin{cases} 1\,, & L_i = L_j \text{ and } |a-b| \leqslant k\,, \\ 1\,, & L_i \neq L_j \text{ and } \left| \frac{a}{\max(L_i-1,1)} - \frac{b}{\max(L_j-1,1)} \right| \leqslant \frac{k}{\max(L_i,L_j)}\,, \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{9}$$

where k controls temporal tolerance. With uniform marginals $\mu = \frac{1}{L_i}\mathbf{1}$ and $\nu = \frac{1}{L_j}\mathbf{1}$, the temporally masked transport plan is

$$\begin{aligned} \mathcal{T}^{\star(i,j)} = \underset{\substack{\mathcal{T} \geqslant 0 \\ \mathcal{T} = \mathcal{T} \odot \mathcal{M}^{(i,j)}}}{\operatorname{argmin}} \quad & \langle \mathcal{T}, \mathcal{C}^{(i,j)} \rangle - \varepsilon H(\mathcal{T}) \\ \text{s.t.} \quad & \mathcal{T}\mathbf{1} = \tfrac{1}{L_i}\mathbf{1}, \quad \mathcal{T}^\top \mathbf{1} = \tfrac{1}{L_j}\mathbf{1}. \end{aligned} \tag{10}$$

Here $\mathcal{T}_{ab}$ is transported mass, $H(\mathcal{T}) = -\sum_{a,b} \mathcal{T}_{ab} \log \mathcal{T}_{ab}$, and the mask zeros matches outside the band. Appendix A.2 derives the solution. Total transport cost defines the distance between shifted rollout sequences:

$$d_{\text{traj}}(\mathcal{X}_i, \mathcal{X}_j) = \langle \mathcal{T}^{\star(i,j)}, \mathcal{C}^{(i,j)} \rangle. \tag{11}$$

Nearest-neighbor teacher. Each failure selects the successful teacher with minimum transport cost:

$$j_i^\star = \underset{j \in \mathcal{S}}{\operatorname{argmin}} \, d_{\text{traj}}(\mathcal{X}_i, \mathcal{X}_j), \qquad i \in \mathcal{F}. \tag{12}$$

Pairs with $T_i < 2$ or $T_j < 2$ are not aligned. If no valid teacher is available, weights remain uniform, reducing the update to the original GRPO assignment.

3.4 Chunk-Level Credit Assignment

For the selected teacher, let $\mathcal{T}_i^\star = \mathcal{T}^{\star(i,j_i^\star)}$ and $\mathcal{C}_i^\star = \mathcal{C}^{(i,j_i^\star)}$. The per-chunk alignment scores are

$$s_{i,t} = -\sum_{b=0}^{L_{j_i^\star}-1} (\mathcal{T}_i^\star)_{tb} (\mathcal{C}_i^\star)_{tb}, \qquad 0 \leqslant t < L_i. \tag{13}$$

Negating these alignment scores yields penalty weights,

$$\widetilde{w}_{i,t} = L_i \frac{\exp(-s_{i,t}/\tau)}{\sum_{u=0}^{L_i-1} \exp(-s_{i,u}/\tau)}, \qquad \sum_{t=0}^{L_i-1} \widetilde{w}_{i,t} = L_i. \tag{14}$$

Higher cost means lower $s_{i,t}$ and larger $\widetilde{w}_{i,t}$, amplifying the magnitude of negative advantage on more divergent chunks. Temperature τ controls concentration. The shifted features already associate row t with action $a_{i,t}$: set $w_{i,t} = \widetilde{w}_{i,t}$ for $t < L_i$ and $w_{i,T_i-1} = 1$, giving $\sum_t w_{i,t} = T_i$. Rollouts with $T_i < 2$ retain uniform weights.

$$\widehat{A}_{i,t} = A_i \cdot w_{i,t} \cdot m_{i,t}\,. \tag{15}$$

The resulting dense advantage $\widehat{A}_{i,t}$ directly replaces A_i in the GRPO loss (6), preserving its clipped-loss form while changing the temporal weighting of policy updates.

Algorithm 1 gives the full procedure. Defaults are NNT assignment, $G = 8$, $k = 30$, $\varepsilon = 0.01$ (100 iterations), cosine distance, and exponential $\tau : 0.05 \to 1$; Section 4.4 ablates these choices.

Properties of the credit transformation. Since $\sum_t w_{i,t} = T_i$, the mean chunk advantage remains A_i before policy ratios and clipping. This normalization does not preserve the loss, gradient, or KL change of a GRPO update. The temporal mask restricts temporally distant matches, while nearest-neighbor selection adapts the reference to each failure. Successful rollouts and groups without a success retain uniform weights, so temporal shaping is applied only when the group provides direct evidence of better behavior.

4 Simulation Experiments

Our experiments test whether GTPO improves over GRPO and prior VLA RL methods, whether the gain is demonstration-efficient, and which design choices (temporal bandwidth, OT temperature and schedule, alignment mechanism, and teacher assignment) drive the improvement across suites with different task structures and horizons represented by the four LIBERO suites.

4.1 Setup

We report pass@1 success on all four LIBERO (Liu et al., 2023) suites using one third-person camera. We use two distinct LoRA-adapted OpenVLA-OFT (Kim et al., 2025b) SFT checkpoints, trained with one expert trajectory per task (†) or the full demonstrations (‡). Within each setting, GRPO and GTPO share the same initialization; these checkpoints are weaker than the published full model. Training uses RLinf (Yu et al., 2025) with $G = 8$, terminal binary rewards, and token-level optimization; trajectory advantages are broadcast to valid chunks before temporal reweighting. Published baselines (§) are included under their original settings (sources in Appendix B.6). This protocol isolates post-training gains while holding the policy backbone, rollout grouping, and optimization interface fixed throughout the matched comparison.

4.2 Main Results

Table 1: LIBERO success rate. †/‡: one/full demonstration warm start; §: published result under its original setting. Parentheses show gains over the matched SFT checkpoint.

Models	Paradigm	Object		Spatial		Goal		Long		Average	
		SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓
OpenVLA§	SFT	88.4	14	84.7	14	79.2	14	53.7	14	76.5	14
GR00T N1§	SFT	97.6	10	94.4	10	93.0	10	90.6	9	93.9	10
$\pi_0^{\S}$	SFT	98.8	5	96.8	8	95.8	8	85.2	10	94.2	9
$\pi_{0.5}^{\S}$	SFT	98.2	8	98.8	3	98.0	6	92.4	6	96.9	6
OpenVLA-OFT§	SFT	98.4	7	97.6	7	97.9	7	94.5	3	97.1	5
GRAPE§	RL	92.1	12	88.5	13	83.1	11	57.2	13	80.2	13
TGRPO§	RL	92.2	11	90.4	11	81.0	13	59.2	12	80.7	12
VLA-RL§	RL	91.8	13	90.2	12	82.2	12	59.8	11	81.0	11
SimpleVLA-RL§	RL	98.7	6	98.2	6	98.8	4	91.7	7	96.9	6
$\pi_{\mathrm{RL}}^{\S}$	RL	**100.0**	1	99.6	2	99.6	2	94.0	4	98.3	3
OpenVLA-OFT†	SFT	28.8	16	52.25	15	49.4	15	14.9	15	36.34	16
GRPO†	RL	97.7	9	94.8	9	94.0	9	91.0	8	94.38	8
GTPO† (Ours)	RL	**100.0** (+71.2)	1	98.6 (+46.35)	4	99.2 (+49.8)	3	96.8 (+81.9)	2	98.65 (+62.31)	2
OpenVLA-OFT‡	SFT	50.2	15	51.6	16	49.4	15	11.9	16	40.78	15
GRPO‡	RL	99.6	4	98.6	4	98.1	5	93.5	5	97.45	4
GTPO‡ (Ours)	RL	**100.0** (+49.8)	1	**100.0** (+48.4)	1	**99.8** (+50.4)	1	**99.0** (+87.1)	1	**99.70** (+58.93)	1

Table 2: Success rate (%) on four RoboTwin (Mu et al., 2025) manipulation tasks.

Method	Place Container Plate	Place Empty Cup	Beat Block Hammer	Pick Dual Bottles	Average
GRPO	95.3	94.5	96.1	92.3	94.55
GTPO (Ours)	**99.2**	**98.5**	**98.7**	**98.1**	**98.63**

(1) Strong performance with limited demonstrations. With only one demonstration per task, GTPO reaches 98.65% average success, already exceeding full-data GRPO (97.45%). Using the full demonstration set further raises performance to 99.70%, ranking first on all four LIBERO suites. Notably, the gap between the one- and full-demonstration settings is only 1.05 points for GTPO, compared with 3.07 points for GRPO, indicating substantially reduced dependence on large demonstration sets.

(2) Consistent gains from temporal group teaching. Compared with matched GRPO, GTPO improves every suite under both warm-start settings, with average gains of 4.27 and 2.25 points. The largest margins occur on `LIBERO-Long` (+5.8/ + 5.5), showing that temporally resolved credit is particularly valuable when sparse outcomes must be propagated through long action sequences. The persistent long-horizon advantage across both warm starts suggests that the gain is not merely compensation for a weak SFT model.

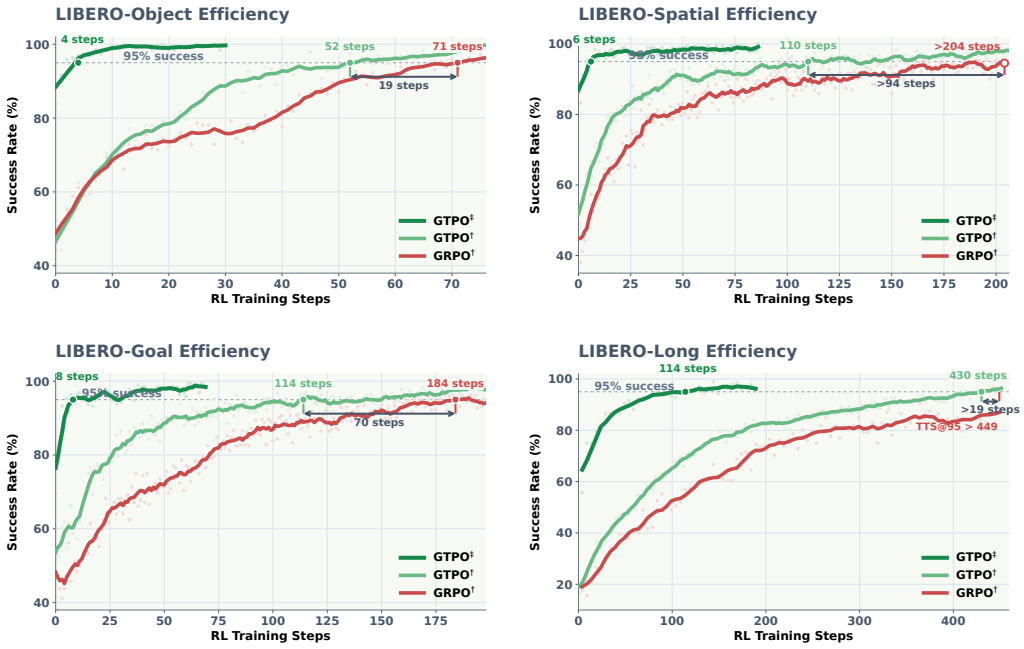

Figure 3: Training efficiency on LIBERO: Object (top left), Spatial (top right), Goal (bottom left), and Long (bottom right). Superscripts † and ‡ denote SFT initialization with one demonstration per task and the full demonstration set, respectively. Lines show seven-checkpoint moving averages of success rate; pale points show raw evaluations.

Training efficiency. We summarize Figure 3 using normalized AUC-SR, success at 25% of training (Early SR), and steps to 95% success (TTS@95), macro-averaged across the four suites. These metrics capture full-horizon sample efficiency, early learning progress, and the time needed to obtain a reliable policy, respectively.

Table 3: Aggregate training efficiency on LIBERO. For TTS@95, runs that do not reach 95% are censored at their training budget, making the reported improvement a conservative lower bound.

Method	AUC-SR ↑	Early SR@25% ↑	TTS@95 ↓
GRPO†	78.55	70.46	> 227.0
GTPO†	**85.92**	**81.63**	**176.5**
Improvement	+7.37	+11.17	$>$ 22.2% fewer steps

(3) Faster optimization and consistent cross-benchmark gains. GTPO gains 7.37 AUC-SR and 11.17 Early SR while requiring at least 22.2% fewer steps to reach 95% success. The curves therefore show an advantage throughout training rather than only at the final checkpoint. GTPO also improves every RoboTwin task, raising average success from 94.55% to 98.63%. The consistent ordering across LIBERO and fine-grained RoboTwin tasks indicates that the gain is not specific to a single benchmark or evaluation granularity.

4.3 OT Necessity Ablation

We compare GTPO with NNT + pointwise cosine matching (GTPO-pointwise) on `LIBERO-Object`. Pointwise matching replaces masked Sinkhorn while retaining nearest-success teaching and normalized chunk weights; Table 4 summarizes the three methods. GTPO exceeds pointwise matching by

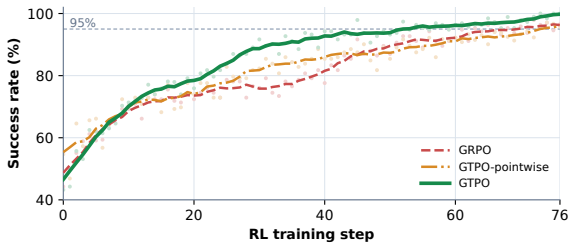

Figure 4: OT necessity ablation on `LIBERO-Object`. Curves show the recorded success rates for GRPO, GTPO-pointwise, and GTPO. Lines are seven-checkpoint moving averages; pale points are raw evaluations, not seed-level uncertainty.

Table 4: OT vs. pointwise matching on `LIBERO-Object`. Final SR is raw; other metrics use smoothed rates. Definitions: Appendix B.4.

Method	AUC-SR	Early SR	TTS@95	Final SR
GRPO	80.72	73.74	70	96.53
GTPO-pointwise	81.89	74.70	73	95.80
GTPO	**85.97**	**78.23**	**52**	**100.00**

4.08 AUC-SR points and reaches 95% success 21 steps earlier (Figure 4), supporting this comparison but not establishing universal OT necessity or causal localization.

4.4 Analysis of Group Teaching and Credit Assignment

On `LIBERO-Object`, we vary bandwidth, temperature, and teacher assignment one factor at a time, holding the remaining settings fixed. Figure 5 and Table 5 summarize all tested settings.

Temporal window. $k = 30$ gives the best AUC and TTS@95. Although $k = 50$ has the highest Early SR, neither $k = 1$ nor $k = 50$ reaches 95% within its budget, illustrating the trade-off between execution-speed tolerance and phase mixing.

Credit temperature. Temperature controls concentration of negative credit. Exponential annealing gives the best AUC; linear annealing reaches 95% three steps earlier but has lower Early SR (77.31 vs. 85.64).

Teacher assignment. ST uses one teacher per group, MTA averages successes, and NNT selects a teacher per failure (Appendix A.3). NNT leads all metrics; ST requires 81 steps and MTA never reaches 95%, supporting failure-specific teaching. Successful rollouts retain uniform weights.

Taken together, the three analyses indicate that GTPO benefits from a moderate temporal context, smooth credit concentration, and failure-specific teacher selection. These choices improve both the speed of reaching high success and the area under the learning curve, so we use the default setting ($k = 30$, $\tau = \text{Exp.}$, NNT) for simulation.

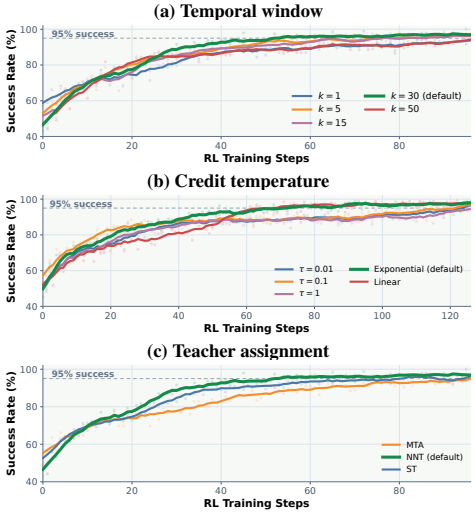

Table 5: Design-analysis metrics on `LIBERO-Object`. Exp. and Linear anneal τ from 0.05 to 1; $>$ denotes a threshold not reached within that analysis horizon.

Analysis	Setting	AUC	Early	TTS@95
Window k	1	84.02	77.83	> 96
	5	86.91	83.63	75
	15	85.95	82.07	77
	30	**87.96**	83.04	**52**
	50	84.25	**84.75**	> 96
Temperature τ	0.01	84.93	82.81	124
	0.1	86.90	**85.83**	122
	1	84.37	82.85	> 126
	Exp.	**89.38**	85.64	71
	Linear	87.11	77.31	**68**
Teacher	ST	85.99	78.91	81
	MTA	83.13	76.04	> 96
	NNT	**87.96**	**83.04**	**52**

Figure 5: Design analyses on `LIBERO-Object`; three efficiency curves are stacked in panels (a)–(c) and the complete metric table is shown at right.

Visualization of the OT-credit mechanism. Figure 6 illustrates a successful teacher and a failed student. Temporal reweighting applies to the student; successful rollouts retain uniform weights.

The failed rollout (bottom) contains an empty-space grasp marked in red. The weight profile illustrates how visual divergence from the teacher can concentrate negative advantage near an annotated error. This qualitative example explains the intended allocation mechanism; it does not establish that visual distance identifies the causal error or validate localization accuracy. Changes in appearance or delayed action effects can also affect the weights.

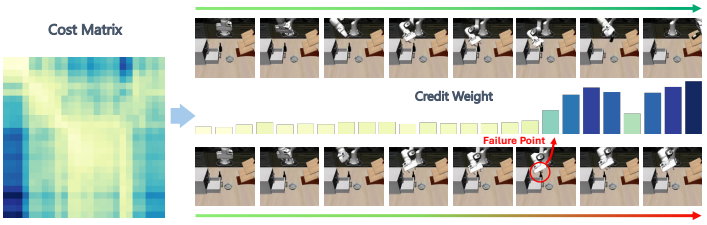

Figure 6: Illustration of OT credit allocation for a successful teacher (top) and failed student (bottom). The weight profile refers to the student; red marks the annotated failed grasp.

5 Real-World Experiments

We evaluate five physical tasks: block stacking, battery insertion, component soldering, screw driving, and bottle-cap removal, using PPO-style on-policy updates with intervention guidance as detailed in Appendix C.

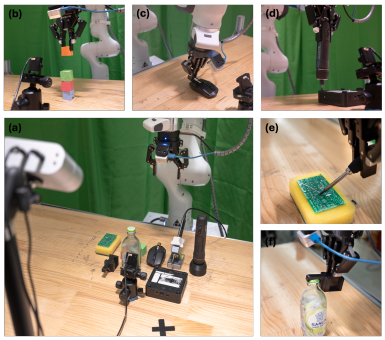

Figure 7: Real-world setup and tasks: **(a)** scene; **(b)** stack blocks; **(c)** insert battery; **(d)** solder component; **(e)** drive screw; and **(f)** pop the top.

Setup and training protocol. We deploy $\pi_{0.5}$ (Black et al., 2025) on a Franka Panda with three cameras. A checkpoint fine-tuned on 50 demonstrations initializes GRPO/GTPO for three on-policy rounds. Both use a PPO-style clipped probability-ratio loss: GRPO uses group-relative advantages, while GTPO applies temporal OT weights. Round one shares rollouts; later rounds collect fresh data with each method's latest policy. Each round collects 64 rollouts in 16 groups of four. Initial states are similar within groups; object positions vary across groups. Valid groups require an autonomous success. Near-error interventions mark interrupted prefixes as failures; corrections enter only auxiliary SFT. Neither method uses a critic or GAE.

Results. Each point uses 20 physical trials under shared poses and binary success criteria. Figure 8 reports two learning curves and final success for all tasks. GTPO reaches 80–100% after round three, exceeding GRPO by 10–35 percentage points, with the largest gain on screw driving under matched initialization and takeover conditions.

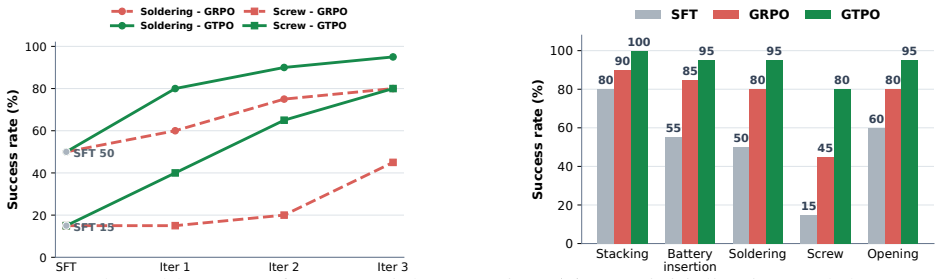

Figure 8: Real-world success from 20 trials per point. **(a)** SFT initialization and three on-policy training rounds for two representative tasks; solid/dashed lines denote GTPO/GRPO. **(b)** SFT initialization and final-iteration success on all five tasks.

6 Conclusion and Limitations

We introduced GTPO, which converts group-relative feedback into chunk-level credit by pairing failures with in-group successful teachers and aligning their visual trajectories with temporally masked OT. Across simulation and physical tasks, it improves success and efficiency without a separately trained process-reward model. **Limitations.** Real-world reinforcement learning remains inefficient under costly, safety-constrained interaction. Visual trajectory comparisons can misestimate GTPO weights under occlusion, viewpoint changes, or visually similar but dynamically different actions.

AI use statement. We used generative AI for language editing, presentation feedback, and figure/layout refinement. The authors produced and verified all method, data, analyses, and final content against implementation records.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, 2023.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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 *Conference on Robot Learning*, 2025.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. π_{rl}: Online RL fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025a.

Qianzhong Chen, Hau Zheng, Justin Yu, Suning Huang, Jiankai Sun, Ken Goldberg, Chuan Wen, Pieter Abbeel, Yide Shentu, Philipp Wu, and Mac Schwager. SARM^2: Multi-task stage aware reward modeling for self improving robotic manipulation. *arXiv preprint arXiv:2606.10305*, 2026.

Zengjue Chen, Runliang Niu, He Kong, Qi Wang, Qianli Xing, and Zipei Fan. Tgrpo: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization, 2025b.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

Yu Fang, Wanxi Dong, Jiaqi Liu, Yue Yang, Mingxiao Huo, Yao Mu, Huaxiu Yao, Li Erran Li, Daniel Szafir, and Mingyu Ding. DenseReward: Dense reward learning via failure synthesis for robotic manipulation. *arXiv preprint arXiv:2607.13033*, 2026.

Youhe Feng, Hansen Shi, Haoyang Li, Xinlei Guo, Yang Wang, Chengyang Zhang, Jinkai Zhang, Xiaohan Zhang, Jie Tang, and Jing Zhang. ProcVLM: Learning procedure-grounded progress rewards for robotic manipulation. *arXiv preprint arXiv:2605.08774*, 2026.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward, 2024.

Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In *Conference on Robot Learning*, 2023.

Jiaheng Hu, Rose Hendrix, Ali Farhadi, Aniruddha Kembhavi, Roberto Martín-Martín, Peter Stone, Kuo-Hao Zeng, and Kiana Ehsani. FLaRe: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. In *IEEE International Conference on Robotics and Automation*, 2025.

Changyeon Kim, Minho Heo, Doohyun Lee, Honglak Lee, Jinwoo Shin, Joseph J. Lim, and Kimin Lee. Subtask-aware visual reward learning from segmented demonstrations. In *International Conference on Learning Representations*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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. *arXiv preprint arXiv:2406.09246*, 2024.

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

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. RoboReward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. SimpleVLA-RL: Scaling vla training via reinforcement learning. In *International Conference on Learning Representations*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2023.

Adrià López Escoriza, Nicklas Hansen, Stone Tao, Tongzhou Mu, and Hao Su. Multi-stage manipulation with demonstration-augmented reward, policy, and world model learning. *arXiv preprint arXiv:2503.01837*, 2025.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. SERL: A software suite for sample-efficient robotic reinforcement learning. *arXiv preprint arXiv:2401.16013*, 2024a.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024b.

Yicheng Luo, Zhengyao Jiang, Samuel Cohen, Edward Grefenstette, and Marc Peter Deisenroth. Optimal transport for offline imitation learning, 2023.

Jindi Lv, Hao Li, Jie Li, Fankun Kong, Yang Wang, Pengfei Yi, Yifei Nie, Xiaofeng Wang, Zheng Zhu, Chaojun Ni, Qiuping Deng, Hengtao Li, Jiancheng Lv, and Guan Huang. ViVa: A video-generative value model for robot reinforcement learning. *arXiv preprint arXiv:2604.08168*, 2026.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27649–27660, 2025.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 663–670, 2000.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*, volume 11. Now Publishers, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

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

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

Yanru Wu, Weiduo Yuan, Ang Qi, Vitor Guizilini, Jiageng Mao, and Yue Wang. Large reward models: Generalizable online robot reward generation with vision-language models. *arXiv preprint arXiv:2603.16065*, 2026.

Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. RLDG: Robotic generalist policy distillation via reinforcement learning. In *Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.028.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, Zixiao Huang, Mingjie Wei, Yuqing Xie, Ke Yang, Bo Dai, Zhexuan Xu, Jiakun Du, Xiangyuan Wang, Xu Fu, Letong Shi, Zhihao Liu, Kang Chen, Weilin Liu, Gang Liu, Boxun Li, Jianlei Yang, Zhi Yang, Guohao Dai, and Yu Wang. Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation, 2025.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. XIRL: Cross-embodiment inverse reinforcement learning. *arXiv preprint arXiv:2106.03911*, 2021.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Peihong Wang, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, Zhihao Liu, Kang Chen, Wenhao Tang, Quanlu Zhang, Weinan Zhang, Chao Yu, and Yu Wang. RLinf-VLA: A unified and efficient framework for reinforcement learning of vision-language-action models. *arXiv preprint arXiv:2510.06710*, 2025.

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.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J. Lim, Jesse Thomason, Erdem Bıyık, and Jesse Zhang. ReWiND: Language-guided rewards teach robot policies without new demonstrations. In *Conference on Robot Learning*, 2025.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. GRAPE: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

A Method Details

A.1 Optimal Transport Background

Optimal Transport (OT) (Peyré & Cuturi, 2019) compares probability distributions by finding the minimum-cost coupling between them. Given discrete marginals $\mu \in \Delta^n$, $\nu \in \Delta^m$ and a non-negative cost matrix $\mathcal{C} \in \mathbb{R}_{\geqslant 0}^{n \times m}$, the Kantorovich formulation seeks a transport plan $\mathcal{T} \in \mathbb{R}_{\geqslant 0}^{n \times m}$ minimizing

$$\mathcal{T}^\star = \underset{\mathcal{T} \in \mathbb{R}_{\geqslant 0}^{n \times m}}{\operatorname{argmin}} \langle \mathcal{T}, \mathcal{C} \rangle \quad \text{s.t.} \quad \mathcal{T}\,\mathbf{1} = \mu, \;\; \mathcal{T}^\top \mathbf{1} = \nu \,, \tag{16}$$

where entry $\mathcal{T}_{ab}$ is the probability mass moved from source a to target b, and $\langle \cdot, \cdot \rangle$ is the Frobenius inner product. The minimum objective value is the Wasserstein distance between μ and ν under $\mathcal{C}$. Solving (16) exactly requires a linear program, which is expensive for long sequences. The Sinkhorn algorithm (Cuturi, 2013) adds an entropic regularizer $H(\mathcal{T}) = -\sum_{a,b} \mathcal{T}_{ab} \log \mathcal{T}_{ab}$ for a smooth, efficient relaxation:

$$\mathcal{T}_\varepsilon^\star = \underset{\mathcal{T} \geqslant 0}{\operatorname{argmin}} \langle \mathcal{T}, \mathcal{C} \rangle - \varepsilon\, H(\mathcal{T}) \,, \quad \text{s.t.} \quad \mathcal{T}\,\mathbf{1} = \mu, \;\; \mathcal{T}^\top \mathbf{1} = \nu \,, \tag{17}$$

whose solution decomposes as $\mathcal{T}_\varepsilon^\star = \operatorname{diag}(u)\, \mathcal{K}\, \operatorname{diag}(v)$ with kernel $\mathcal{K}_{ab} = \exp(-\mathcal{C}_{ab}/\varepsilon)$, where u, v are found by alternating row/column normalization in log-space. A key limitation of (17) is that it is *permutation-invariant*: shuffling the source or target sequence leaves the optimal cost unchanged, so temporal ordering must be imposed separately (Section 3), which is essential for sequential tasks such as robotic manipulation.

A.2 Temporally Masked Sinkhorn Derivation

We derive the solution of (10) for one trajectory pair and omit the superscript (i, j) for clarity. Let $\mathcal{I} = \{(a, b) : \mathcal{M}_{ab} = 1\}$ be the set of permitted matches. Using $-H(\mathcal{T}) = \sum_{a,b} \mathcal{T}_{ab} \log \mathcal{T}_{ab}$, and noting that $\sum_{a,b} \mathcal{T}_{ab} = 1$, we may write the objective equivalently as

$$\min_{\mathcal{T} \geqslant 0} \quad \sum_{(a,b)\in\mathcal{I}} \left[\mathcal{T}_{ab}\mathcal{C}_{ab} + \varepsilon \mathcal{T}_{ab}\big(\log \mathcal{T}_{ab} - 1\big)\right] \tag{18}$$

$$\text{s.t.} \quad \mathcal{T}\mathbf{1} = \mu, \qquad \mathcal{T}^\top \mathbf{1} = \nu, \qquad \mathcal{T}_{ab} = 0 \quad \forall (a, b) \notin \mathcal{I}.$$

The subtraction of 1 inside the entropy term changes the objective only by the constant $-\varepsilon$ and therefore does not change its minimizer.

Lagrangian and stationary point. Introduce multipliers $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^m$ for the row and column marginals. Restricting optimization to the permitted support $\mathcal{I}$, the Lagrangian is

$$\begin{aligned}\mathfrak{L}(\mathcal{T}, \alpha, \beta) = &\sum_{(a,b)\in\mathcal{I}} \left[\mathcal{T}_{ab}\mathcal{C}_{ab} + \varepsilon \mathcal{T}_{ab}\big(\log \mathcal{T}_{ab} - 1\big)\right] \\ &+ \sum_a \alpha_a \left(\mu_a - \sum_{b:(a,b)\in\mathcal{I}} \mathcal{T}_{ab}\right) + \sum_b \beta_b \left(\nu_b - \sum_{a:(a,b)\in\mathcal{I}} \mathcal{T}_{ab}\right).\end{aligned} \tag{19}$$

For every permitted entry, the stationarity condition is

$$\frac{\partial \mathfrak{L}}{\partial \mathcal{T}_{ab}} = \mathcal{C}_{ab} + \varepsilon \log \mathcal{T}_{ab} - \alpha_a - \beta_b = 0, \tag{20}$$

which gives

$$\mathcal{T}_{ab} = \exp\left(\frac{\alpha_a + \beta_b - \mathcal{C}_{ab}}{\varepsilon}\right) = u_a \mathcal{K}_{ab} v_b. \tag{21}$$

Here $u_a = \exp(\alpha_a/\varepsilon)$, $v_b = \exp(\beta_b/\varepsilon)$, and the masked Gibbs kernel is

$$\mathcal{K}_{ab} = \mathcal{M}_{ab} \exp(-\mathcal{C}_{ab}/\varepsilon). \tag{22}$$

Thus forbidden entries are exactly zero, while the complete solution has the matrix form

$$\mathcal{T}^\star = \operatorname{diag}(u)\mathcal{K}\operatorname{diag}(v). \tag{23}$$

Sinkhorn scaling. Substituting (23) into the marginal constraints yields

$$u \odot (\mathcal{K}v) = \mu, \qquad v \odot (\mathcal{K}^\top u) = \nu, \tag{24}$$

where $\odot$ denotes element-wise multiplication. Therefore, u and v can be obtained by alternating updates

$$u^{(q+1)} = \frac{\mu}{\mathcal{K}v^{(q)}}, \qquad v^{(q+1)} = \frac{\nu}{\mathcal{K}^\top u^{(q+1)}}, \tag{25}$$

where division is element-wise. Starting from $v^{(0)} = \mathbf{1}$, these updates are repeated until both marginal residuals $\|\mathcal{T}\mathbf{1} - \mu\|_1$ and $\left\|\mathcal{T}^\top \mathbf{1} - \nu\right\|_1$ fall below a tolerance.

Log-domain implementation. Direct evaluation of $\exp(-\mathcal{C}/\varepsilon)$ may underflow when ε is small. We instead define dual potentials $f = \varepsilon \log u$ and $g = \varepsilon \log v$ and perform the updates as

$$f_a = \varepsilon \log \mu_a - \varepsilon \operatorname{logsumexp}_{b:\mathcal{M}_{ab}=1}\left(\frac{g_b - \mathcal{C}_{ab}}{\varepsilon}\right), \tag{26}$$

$$g_b = \varepsilon \log \nu_b - \varepsilon \operatorname{logsumexp}_{a:\mathcal{M}_{ab}=1}\left(\frac{f_a - \mathcal{C}_{ab}}{\varepsilon}\right). \tag{27}$$

After convergence, the transport plan is recovered by

$$\mathcal{T}^\star_{ab} = \begin{cases} \exp\left((f_a + g_b - \mathcal{C}_{ab})/\varepsilon\right), & \mathcal{M}_{ab} = 1, \\ 0, & \mathcal{M}_{ab} = 0. \end{cases} \tag{28}$$

The temporal mask must contain at least one permitted entry in every row and column and admit a coupling with the specified marginals; we choose k accordingly. Each dense iteration costs $O(nm)$ time, or $O(|\mathcal{I}|)$ when the banded support is stored sparsely. Finally, the trajectory distance used for teacher selection is the unregularized transport cost $\langle \mathcal{T}^\star, \mathcal{C} \rangle$, as defined in (11); entropy is used only to obtain a stable alignment.

A.3 Teacher Assignment Strategies

We formalize the three teacher assignments compared in Section 4.4. Recall that $\mathcal{S}$ and $\mathcal{F}$ are the successful and failed rollouts in one GRPO group. We exclude rollouts with fewer than two valid chunks from alignment; if no valid teacher remains, weights are uniform. Below, $\mathcal{S}$ denotes the remaining teacher pool. For each valid pair (i, j), temporally masked OT on shifted features of lengths $L_i = T_i - 1$ and $L_j = T_j - 1$ produces the plan $\mathcal{T}^{\star(i,j)}$, cost matrix $\mathcal{C}^{(i,j)}$, and trajectory distance $d_{ij} = d_{\mathrm{traj}}(\mathcal{X}_i, \mathcal{X}_j)$. To make the dependence on teacher j explicit, define the pair-specific alignment score and normalized credit weight as

$$s_{i,t}^{(j)} = -\sum_{b=0}^{L_j-1} \mathcal{T}_{tb}^{\star(i,j)} \mathcal{C}_{tb}^{(i,j)}, \qquad 0 \leqslant t < L_i, \tag{29}$$

$$\widetilde{w}_{i,t}^{(j)} = L_i \frac{\exp\Big(-s_{i,t}^{(j)}/\tau\Big)}{\sum_{u=0}^{L_i-1} \exp\Big(-s_{i,u}^{(j)}/\tau\Big)}, \qquad \sum_{t=0}^{L_i-1} \widetilde{w}_{i,t}^{(j)} = L_i. \tag{30}$$

Set $w_{i,t}^{(j)} = \widetilde{w}_{i,t}^{(j)}$ for $t < L_i$ and $w_{i,T_i-1}^{(j)} = 1$. Thus $\sum_{t=0}^{T_i-1} w_{i,t}^{(j)} = T_i$. Higher transport cost gives a larger penalty weight for negative-advantage rollouts, consistently with Equation (14).

Single teacher (ST). ST chooses one successful rollout for the entire group. Let

$$j_{\mathrm{ST}}^{*} \in \operatorname*{argmax}_{j \in \mathcal{S}} r_j, \qquad w_{i,t}^{\mathrm{ST}} = w_{i,t}^{(j_{\mathrm{ST}}^{*})}, \quad i \in \mathcal{F}. \tag{31}$$

The same j_{ST}^{*} supervises every failure. When several successful rollouts have the same terminal reward, as in our binary-reward setting, one is selected once with a fixed seeded tie break and then shared by the group. ST is inexpensive, but a single teacher may be visually or behaviorally mismatched to some failures.

Multi-teacher averaging (MTA). MTA aligns each failed rollout with every successful rollout and averages the resulting teacher-specific credit weights:

$$w_{i,t}^{\mathrm{MTA}} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} w_{i,t}^{(j)}, \qquad i \in \mathcal{F}. \tag{32}$$

Thus MTA averages normalized temporal credits rather than raw feature trajectories or OT cost matrices. It uses all available successes, but irrelevant teachers can dilute the sharp temporal signal from a well-matched demonstration.

Nearest-neighbor teacher (NNT). NNT assigns a different teacher to each failure by minimizing the pairwise trajectory distance:

$$j_i^{*} = \operatorname*{argmin}_{j \in \mathcal{S}} d_{ij}, \qquad w_{i,t}^{\mathrm{NNT}} = w_{i,t}^{(j_i^{*})}, \quad i \in \mathcal{F}. \tag{33}$$

Unlike ST, NNT adapts the teacher to the current failure; unlike MTA, it avoids averaging incompatible behaviors. This is the default assignment used by GTPO.

For any assignment $q \in \{\mathrm{ST}, \mathrm{MTA}, \mathrm{NNT}\}$, the final chunk-level advantage is

$$\widehat{A}_{i,t}^{q} = A_i \, w_{i,t}^{q} \, m_{i,t}. \tag{34}$$

All three variants preserve $\sum_t w_{i,t}^{q} = T_i$, so they change only the temporal distribution of the original trajectory advantage. Successful rollouts, and all rollouts in a group with $\mathcal{S} = \varnothing$, retain $w_{i,t} = 1$.

A.4 GTPO Algorithm

Algorithm 1 GTPO: Group Teaching via Temporal Optimal Transport

Require: Scores $\{r_i\}_{i=1}^{G}$, shifted features $\mathcal{X}_i = (x_{i,1}, \ldots, x_{i,T_i-1})$, valid lengths T_i, masks $m_{i,t}$, bandwidth k, temperature τ, entropic weight ε

Ensure: Dense advantages $\{\widehat{A}_{i,t}\}$

$A_i \leftarrow (r_i - \mu_g)/(\sigma_g + \varepsilon_{\text{std}})$ for all $i \in \{1, \ldots, G\}$ ▷ Eq. (5)
$\mathcal{S} \leftarrow \{j : r_j > 0\}$, $\mathcal{F} \leftarrow \{i : r_i < 0\}$
$L_i \leftarrow T_i - 1$ for all i; $\mathcal{S}_{\text{valid}} \leftarrow \{j \in \mathcal{S} : T_j \geqslant 2\}$
for $i = 1, \ldots, G$ **do**
    $w_{i,t} \leftarrow 1$ for all valid t ▷ Also covers short rollouts
    if $i \in \mathcal{F}$ and $T_i \geqslant 2$ and $\mathcal{S}_{\text{valid}} \neq \varnothing$ **then**
        $j_i^\star \leftarrow \operatorname{argmin}_{j \in \mathcal{S}_{\text{valid}}} \text{SinkhornCost}(\mathcal{X}_i, \mathcal{X}_j; k, \varepsilon)$ ▷ Eq. (12)
        $(\mathcal{C}, \mathcal{M}, \mathcal{T}^\star) \leftarrow \text{MaskedOT}(\mathcal{X}_i, \mathcal{X}_{j_i^\star}; k, \varepsilon)$
        $s_{i,t} \leftarrow -\sum_{b=0}^{L_{j_i^\star}-1} \mathcal{T}^\star_{tb}\mathcal{C}_{tb}$ for $0 \leqslant t < L_i$ ▷ Eq. (13)
        $\widetilde{\boldsymbol{w}}_i \leftarrow L_i \operatorname{softmax}(-\boldsymbol{s}_i/\tau)$ ▷ Eq. (14)
        $w_{i,t} \leftarrow \widetilde{w}_{i,t}$ for $0 \leqslant t < L_i$; $w_{i,T_i-1} \leftarrow 1$
    end if
    $\widehat{A}_{i,t} \leftarrow A_i \cdot w_{i,t} \cdot m_{i,t}$ for all t ▷ Eq. (15)
end for
return $\{\widehat{A}_{i,t}\}$

B Simulation Experiment Details

B.1 Benchmark and Policy Initialization

The configuration below corresponds to the `LIBERO-Object` experiment with OpenVLA-OFT and GTPO. The suite contains ten object-manipulation tasks. We use two distinct LoRA SFT checkpoints: one trained with one expert demonstration per task (†), and the other with the full demonstration set (‡). Within each setting, GRPO and GTPO are initialized from the same checkpoint, ensuring a matched comparison of their RL updates. OpenVLA-OFT has approximately 8B parameters and combines a Llama-2-7B language backbone with fused DINOv2 and SigLIP visual encoders. Reinforcement learning updates all policy parameters (LoRA is disabled), without a value head or a critic.

Table 6: Simulation model and input configuration.

Parameter	Setting
Policy camera	One third-person camera
Image resolution	256×256 rendering; 224×224 center-cropped policy input
Wrist camera	Rendered by the environment; not provided to the policy
Action space	7D (translation, rotation, and gripper)
Action-chunk horizon	8
Action discretization	256 bins
Hidden / vocabulary size	4,096 / 32,000
Proprioception / FiLM	Both disabled
Maximum prompt length	128
Maximum generated length	1,024

B.2 Training and Inference Configuration

Each policy iteration collects rollouts from groups of matched initial states and then updates the actor. Episodes contain at most 320 environment steps, corresponding to at most 40 policy decisions with an action-chunk horizon of 8. Advantages are normalized within each group. Table 7 gives the rollout and optimization settings.

We compute cosine feature costs without temporal subsampling. The temporal OT settings for this configuration are listed in Table 8.

Evaluation uses the same stochastic decoding parameters as training. Table 9 summarizes evaluation and systems settings. Training uses a single node equipped with NVIDIA A800 GPUs.

Table 7: Simulation rollout and policy optimization.

Parameter	Setting
Training budget	At most 1,000 policy iterations
Rollout collection	8 rollout epochs per iteration; 128 parallel environments
Group size	$G = 8$
Optimizer	Adam; $(\beta_1, \beta_2) = (0.9, 0.999)$, $\varepsilon = 10^{-8}$
Learning rate	2×10^{-5}
Weight decay	0.01
Gradient clipping	Global norm of 1.0
Batch size	10,240 globally; micro-batch size 32 per GPU
Optimization epochs	One per collected batch
Log probabilities / entropy	Token-level
Loss aggregation	Token-mean
Clipping	Lower/upper ratios $0.20/0.28$; dual-clip coefficient 3.0
KL / entropy coefficients	Both zero
Reward processing	Action-level rewards, scaled by 5.0
Reward filtering	Enabled; lower/upper bounds $0.5/4.5$
Sampling	Stochastic; temperature 1.6, top-$p = 1$, no top-k truncation, repetition penalty 1.0

Table 8: Temporal optimal-transport configuration.

Parameter	Setting
Feature cost	Cosine distance
Temporal subsampling	None
Mask half-bandwidth	$k = 30$
Sinkhorn regularization	$\varepsilon = 0.01$
Sinkhorn iterations	100
Credit temperature	Exponential schedule from 0.05 to 1.0 over the first 80 policy iterations

Table 9: Simulation evaluation and systems configuration.

Parameter	Setting
Evaluation frequency	Every policy iteration
Evaluation environments	192 parallel environments with automatic resets
Evaluation rollouts	One rollout per fixed reset-state identifier
Episode horizon	320 environment steps, for both training and evaluation
Evaluation sampling	Temperature 1.6, top-$p = 1$, no top-k truncation, repetition penalty 1.0
Hardware	Single node with NVIDIA A800 GPUs
Training backend	Fully sharded data parallelism
Memory settings	Gradient checkpointing and actor/rollout offloading
Precision	bfloat16
Actor / environment seed	1234 / 0
Checkpoint interval	Every 500 policy iterations

B.3 Cosine Distance in the OT Ablation

Section 4.3 compares two ways of matching visual trajectories, both using cosine distance. GTPO-pointwise retains nearest-success teaching and normalized chunk weights but replaces masked Sinkhorn alignment with pointwise matching. Thus this comparison tests the alignment mechanism, rather than replacing cosine distance with a different feature metric.

Feature distance. For a failed rollout i and a successful teacher candidate j, we use the shifted features $\mathcal{X}_i = (x_{i,1}, \ldots, x_{i,T_i-1})$ and $\mathcal{X}_j = (x_{j,1}, \ldots, x_{j,T_j-1})$ from Equation (7). The feature following an action chunk represents its observed consequence. For valid indices $0 \leqslant a < L_i$ and $0 \leqslant b < L_j$, the cost is

$$c_{ab}^{(i,j)} = 1 - \frac{x_{i,a+1}^{\top} x_{j,b+1}}{\|x_{i,a+1}\|_2 \|x_{j,b+1}\|_2 + 10^{-8}}. \tag{35}$$

The computation uses float32 features and the stabilizer 10^{-8} in the denominator. For nonzero features, costs are approximately 0 for the same direction, 1 for orthogonal directions, and 2 for opposite directions. This measures visual dissimilarity, not task success or an action's physical correctness directly.

Pointwise matching versus OT. For equal-length trajectories indexed on the same time grid, pointwise comparison reads $c_{tt}^{(i,j)}$, comparing the two successor observations at each corresponding chunk. It does not search for an alternative match when one rollout executes a sub-action earlier or later. In contrast, GTPO uses the full pairwise matrix within the temporal mask and computes

$$s_{i,t}^{\mathrm{OT}} = -\sum_{b=0}^{L_j-1} \mathcal{T}_{tb}^{\star(i,j)} c_{tb}^{(i,j)}. \tag{36}$$

The transport plan can distribute a chunk's mass across nearby teacher chunks while satisfying both trajectory marginals. Consequently, a timing offset can increase a pointwise distance even when the two rollouts exhibit similar behavior, whereas OT can accommodate that offset within the permitted temporal band.

From scores to penalty weights. The shared credit transformation takes alignment scores $s_{i,t}$, with larger scores denoting closer matches, and applies

$$\widetilde{w}_{i,t} = L_i \frac{\exp(-s_{i,t}/\tau)}{\sum_{u=0}^{L_i-1} \exp(-s_{i,u}/\tau)}, \qquad \widehat{A}_{i,t} = A_i w_{i,t} m_{i,t}. \tag{37}$$

Here $w_{i,t} = \widetilde{w}_{i,t}$ for $t < L_i$, and the final chunk retains weight 1 because it has no stored successor feature. A larger mismatch yields a lower score and a larger weight; for a failed rollout with $A_i < 0$, this strengthens the negative advantage. Normalization preserves $\sum_t w_{i,t} = T_i$. Successful rollouts and failures without a valid teacher retain uniform weights. Score scale matters: the OT row cost in Equation (36) includes transport mass $1/L_i$, so replacing it by a raw cosine distance changes the effective temperature unless that scale is accounted for.

B.4 Evaluation Metrics

Efficiency metrics. We report normalized area under the success-rate curve over the common per-suite horizon (AUC-SR), success at 25% of that horizon (Early SR), and the first step reaching 95% success (TTS@95). We apply the seven-checkpoint moving average used in Figure 3, compute each metric per suite, and macro-average across the four suites. If a run never reaches 95%, TTS@95 is right-censored at its final training step; substituting that budget yields only a lower bound on GTPO's efficiency gain.

OT ablation metrics. For the comparison in Section 4.3, we apply a centered seven-checkpoint moving average to success rates, replicating boundary values at the endpoints. Normalized AUC-SR is the mean of the smoothed success rates; Early SR is the smoothed success rate at 25% of training, and TTS@95 is the first recorded step at which the smoothed success rate reaches 95%. Final SR is the final raw evaluation. These descriptive metrics are computed from one learning curve per method and do not quantify variability across independent training seeds.

Reward diagnostics. We measure reward–success separation $\mathbb{E}[R \mid \text{success}] - \mathbb{E}[R \mid \text{failure}]$, return-based success AUROC, Pearson/Spearman return–success correlation, and the rates of high-reward failures and low-reward successes. The latter use the 25th percentile of successful returns and the 75th percentile of failed returns, respectively.

B.5 Design-Analysis Protocol

On `LIBERO-Object`, we vary one factor at a time while holding the remaining settings at their defaults: temporal window $k \in \{1, 5, 15, 30, 50\}$; constant $\tau \in \{0.01, 0.1, 1\}$ or linear/exponential $0.05 \to 1$ schedules; and ST, MTA, or NNT teacher assignment. Efficiency metrics are computed from the seven-checkpoint moving averages over the common training horizon within each analysis.

B.6 External Baseline Sources

The published baselines marked § in Table 1 are OpenVLA (Kim et al., 2024), GR00T N1 (NVIDIA et al., 2025), π_0 (Black et al., 2024), $\pi_{0.5}$ (Black et al., 2025), OpenVLA-OFT (Kim et al., 2025b),

GRAPE (Zhang et al., 2024), TGRPO (Chen et al., 2025b), VLA-RL (Lu et al., 2025), SimpleVLA-RL (Li et al., 2026), and π_{RL} (Chen et al., 2025a). These entries retain their original experimental settings rather than the matched warm starts used for our GRPO/GTPO comparisons. Ranks are computed within each table column as one plus the number of entries with strictly higher success rates; ties share a rank.

C Real-World Experiment Details

C.1 Backbone and SFT Configuration

All real-world experiments use $\pi_{0.5}$ (Black et al., 2025) as the VLA backbone. For each task, we collect 50 demonstrations and fine-tune a task-specific SFT checkpoint. The same data and checkpoint initialize GRPO and GTPO. The reported SFT baseline is this checkpoint before takeover training. Initial SFT uses eight GPUs. Table 10 summarizes the shared interface and the separate SFT and on-policy RL settings.

Table 10: Real-world SFT and on-policy RL configuration.

Parameter	**Setting**
Shared optimizer and action interface	
Optimizer	AdamW; $(\beta_1, \beta_2) = (0.9, 0.95)$, $\varepsilon = 10^{-8}$
Gradient clipping	Global norm of 1.0
Visual input	Base, wrist, and nearby cameras at 224×224
Action space	Physical 7D TCP actions (xyz, rotvec, gripper), padded to 32 dimensions
Action convention	TCP pose actions are deltas; gripper command is absolute
Horizon / denoising steps	50 generated actions / 10 steps
Initial SFT	
Learning-rate schedule	1,000-step warmup to 2.5×10^{-5}; cosine decay to 2.5×10^{-6}
Weight decay	10^{-10}
Batch size	512 globally; 64 per GPU
Training / checkpoint steps	30,000 total; checkpoint every 2,500 steps
Precision	bfloat16
Flow-matching time	$u \sim \mathrm{Beta}(1.5, 1.0)$
On-policy RL	
Rollout budget	64 rollouts per task and round; 16 groups of $G = 4$
Policy epochs / batch	3 epochs; 8 chunks per optimizer step; forward batch 1 with accumulation
Learning-rate schedule	10-step warmup to 5×10^{-6}; cosine decay to 5×10^{-7} over 10,000 steps
Weight decay	0.01
PPO / dual clipping	Ratio interval $[0.8, 1.2]$; $c = 3$ for negative advantages
Loss coefficients	$\lambda_{\mathrm{RL}} = 2 \times 10^{-4}$; $\lambda_{\mathrm{SFT}} = 1.0$
Advantage normalization	Within-group sample standard deviation plus 10^{-6}; none after GTPO weighting
KL / entropy / critic / GAE	All disabled; approximate KL logged only
Training / grouping seed	42 / 0; RNG state restored on resume
Collection noise seed	Not explicitly fixed

C.2 Three-Round On-Policy Collection with Takeover

Starting from the common SFT checkpoint, GRPO and GTPO perform three rounds of robot inference, operator takeover, and policy update. Each round collects 64 rollouts per task, organized into 16 groups of four. During collection, the current policy controls the robot until it is about to make an error or an error has already occurred; only then does the operator take over and provide a corrective continuation. The policy-generated prefix preceding the takeover is terminated and labeled as a failed sample, irrespective of whether operator control later completes the task. This prevents a human recovery from being attributed to the policy. We split each round into two disjoint streams. The policy stream $\mathcal{D}^{(q)}_{\mathrm{pol},M}$ contains autonomous rollouts and the policy-generated prefixes before takeover; each such prefix receives a failure label. The takeover stream $\mathcal{D}^{(q)}_{\mathrm{to},M}$ contains only the operator-controlled corrective continuation. Both streams record the instruction, image observations,

executed action chunks, initial robot state, intervention indicator, and binary label. All methods use identical reset, takeover, and success criteria, with matched rollout budgets and update epochs in every round. For policy-generated chunks, the rollout buffer additionally stores the collecting policy version, its log probabilities, and the sampling variables needed to recompute those log probabilities during optimization.

Let $M \in \{\mathrm{GRPO}, GTPO\}$ index the training branch and $\theta_M^{(q)}$ its parameters after round q. Because all branches have identical parameters before round one, we collect one common pair of policy and takeover streams and use it to update every method. In rounds two and three, each method deploys its own latest checkpoint and trains only on the two streams generated during its own rollout:

$$\left(\mathcal{D}_{\mathrm{pol},M}^{(1)}, \mathcal{D}_{\mathrm{to},M}^{(1)}\right) = \left(\mathcal{D}_{\mathrm{pol}}^{(1)}, \mathcal{D}_{\mathrm{to}}^{(1)}\right), \tag{38}$$

$$\left(\mathcal{D}_{\mathrm{pol},M}^{(q)}, \mathcal{D}_{\mathrm{to},M}^{(q)}\right) \sim \mathrm{RolloutTakeover}\left(\pi_{\theta_M^{(q-1)}}\right), \qquad q \in \{2,3\}. \tag{39}$$

The shared first round controls for collection variability. Each round freezes the behavior policy at $\theta_{\mathrm{old},M}^{(q)} = \theta_M^{(q-1)}$ for collection and holds its recorded log probabilities fixed during the ensuing clipped update. Subsequent rounds collect fresh rollouts with each updated policy, without replaying earlier policy rollouts. Operator actions in $\mathcal{D}_{\mathrm{to},M}^{(q)}$ provide separate auxiliary SFT supervision.

C.3 Rollout Groups and Temporal Teaching

Round-wise rollout groups. Groups are constructed exclusively from the policy stream $\mathcal{D}_{\mathrm{pol},M}^{(q)}$. Because exact repeated resets are difficult on a physical robot, a group contains same-task rollouts with nearby initial visual features and proprioceptive states. Each group contains four rollouts ($G = 4$), while object initial positions vary across the 16 groups in a round. This variation provides diversity in object placement while preserving comparable initial conditions for within-group teaching. Groups without an autonomous success are skipped; within each valid group, binary outcomes are normalized to obtain the relative advantages A_i in Equation (5). Takeover segments never enter group normalization and can never be selected as teachers.

Temporal group teaching. Teacher selection and credit assignment follow the formulation in Equations (12)–(15), with separate real-world hyperparameters: mask half-bandwidth $k = 6$, Sinkhorn regularization $\varepsilon = 0.01$, at most 2,000 iterations, tolerance 10^{-5}, and fixed credit temperature $\tau = 0.01$. Each failed rollout selects its nearest successful teacher; larger post-action alignment costs receive larger positive weights, yielding $\widehat{A}_{i,t} = A_i w_{i,t} m_{i,t}$. Group advantages use the sample standard deviation ($\mathrm{ddof} = 1$) and $\varepsilon_{\mathrm{std}} = 10^{-6}$ in Equation (5). We do not re-standardize advantages after temporal weighting. Teachers and weights are computed from the current round and held fixed throughout its policy updates.

C.4 PPO-Style Clipped Policy Update

Single-step stochastic flow sampling. At each policy decision, $\pi_{0.5}$ generates a normalized action sequence of shape $H \times D = 50 \times 32$ from $z_0 \sim \mathcal{N}(0, I)$, using $N = 10$ denoising steps. Let z_n denote the denoising state after n steps, with flow time $u_n = 1 - n/N$. For each action chunk independently, sample $J \sim \mathrm{Uniform}\{0, \ldots, 8\}$. Only transition J uses Flow-SDE; the remaining transitions use Flow-ODE, including the last. The SDE noise strength is fixed at $a = 0.5$, with neither a learned noise network nor noise annealing. At the selected step, the transition has diagonal Gaussian form

$$z_{J+1} = \mu_\theta(z_J, o, \ell, J) + \sigma_J \odot \xi, \qquad \xi \sim \mathcal{N}(0, I), \tag{40}$$

where μ_θ is the Flow-SDE transition mean obtained from the policy velocity and σ_J is its transition standard deviation; a specifies the SDE noise strength, not an independently learned per-action variance. The robot executes at most the first 20 actions before observing again. The actual executed length $L_{i,t} \leqslant 20$ is determined by robot feedback; chunks with $L_{i,t} = 0$ are excluded from policy updates. Only the seven physical action coordinates contribute; the remaining 25 coordinates are padding. Translation and rotation-vector actions are deltas relative to the current state, while the gripper command is absolute. Actions use quantile normalization; the robot interface converts them to an eight-dimensional xyz–quaternion–gripper command. All three camera streams (base, wrist, and nearby) are active.

Policy ratios. Both branches use PPO-style clipping (Schulman et al., 2017) with group-relative advantages, without a critic or GAE. The non-joint score uses the selected stochastic transition, re-evaluated on the saved observations and denoising samples. For each retained action coordinate (h, d), its Gaussian log density is

$$\ell_{\theta,h,d} = -\frac{1}{2}\left[\frac{(z_{J+1,h,d} - \mu_{\theta,h,d})^2}{\sigma_{J,h,d}^2} + 2\log\sigma_{J,h,d} + \log(2\pi)\right]. \tag{41}$$

We sum over the executed prefix and the seven effective action dimensions:

$$s_{\theta,i,t} = \sum_{h=0}^{L_{i,t}-1}\sum_{d=1}^{7}\ell_{\theta,h,d}. \tag{42}$$

There is no division by $L_{i,t}$, 7, or $7L_{i,t}$. In implementation, the masked new-minus-old log densities are summed in float32 before exponentiation. The ratio is

$$\rho_{i,t}^{(q)}(\theta) = \exp\Big(s_{\theta,i,t} - \mathrm{stopgrad}\Big(s_{\theta_{\mathrm{old},M}^{(q)},i,t}\Big)\Big). \tag{43}$$

This is the unscaled product-density ratio of the retained coordinates of the selected Gaussian transition, conditional on the saved denoising input. It is not the marginal density ratio of the final physical action or a product over the full denoising chain. Only the subsequent loss is averaged over valid chunks, so each executed chunk carries equal weight regardless of $L_{i,t}$. Because the log ratio sums $7L_{i,t}$ terms, its scale can depend on execution length even though the clipping interval is shared.

Group-relative clipped objective. GRPO uses A_i, while GTPO uses $\widehat{A}_{i,t}$, in the same clipped surrogate. Write $\bar{\rho} = \mathrm{clip}(\rho, 0.8, 1.2)$ and $\ell_{\mathrm{PPO}}(\rho, A) = \max(-\rho A, -\bar{\rho}A)$. Dual clipping with $c = 3$ applies only to negative advantages:

$$\ell_{\mathrm{dual}}(\rho, A) = \begin{cases}\min(\ell_{\mathrm{PPO}}(\rho, A), -3A), & A < 0,\\ \ell_{\mathrm{PPO}}(\rho, A), & A \geqslant 0.\end{cases} \tag{44}$$

For the GTPO branch ($M = \mathrm{GTPO}$), the policy loss is

$$\mathcal{L}_{\mathrm{clip},M}^{(q)}(\theta) = \frac{1}{Z_{\mathrm{pol}}}\sum_{(i,t)\in\mathcal{D}_{\mathrm{pol},M}^{(q)}} m_{i,t}\,\ell_{\mathrm{dual}}\Big(\rho_{i,t}^{(q)}, \widehat{A}_{i,t}\Big), \tag{45}$$

where Z_{pol} counts retained policy chunks. GRPO replaces $\widehat{A}_{i,t}$ with A_i. Each optimizer step averages over up to eight chunks with positive executed length, using forward batches of one and gradient accumulation. Neither an entropy bonus nor a KL penalty is added; approximate KL is recorded only as a diagnostic.

Auxiliary takeover supervision. Only operator-controlled corrective segments receive the SFT flow-matching loss. For action chunk $a_{i,t}$, Gaussian noise z, and flow time $u \sim \mathrm{Beta}(1.5, 1.0)$, let $a_{i,t}^u = (1-u)z + ua_{i,t}$ and $v_{i,t}^\star = a_{i,t} - z$. Then

$$\mathcal{L}_{\mathrm{to\text{-}SFT}}^{(q)}(\theta) = \frac{1}{Z_{\mathrm{to}}}\sum_{(i,t)\in\mathcal{D}_{\mathrm{to},M}^{(q)}} m_{i,t}\left\|v_\theta(a_{i,t}^u, o_{i,t}, \ell, u) - v_{i,t}^\star\right\|_2^2, \tag{46}$$

where Z_{to} is the number of valid takeover chunks. The full objective for each branch is

$$\mathcal{L}_M^{(q)}(\theta) = \lambda_{\mathrm{RL}}\mathcal{L}_{\mathrm{clip},M}^{(q)}(\theta) + \lambda_{\mathrm{SFT}}\mathcal{L}_{\mathrm{to\text{-}SFT}}^{(q)}(\theta), \tag{47}$$

with $\lambda_{\mathrm{RL}} = 2 \times 10^{-4}$ and $\lambda_{\mathrm{SFT}} = 1.0$ for both branches. Operator-controlled actions enter neither the probability-ratio objective nor group-relative or temporal-OT advantage estimation.

C.5 Real-World RL Configuration

The on-policy settings in Table 10 apply separately from initial SFT. We train the Gemma-300M action expert, action input/output projections, and time MLP. SigLIP and the Gemma-2B backbone remain frozen; no LoRA or value head is used. Each collected round is optimized for three epochs. With N_{pol} retained policy chunks and a final partial batch, this gives $3\lceil N_{\mathrm{pol}}/8\rceil$ optimizer steps; the number can vary with the retained data despite a matched rollout budget. The learning-rate schedule follows the cumulative optimizer step and continues on resume.

C.6 Baselines and Evaluation

SFT denotes the common initialization before the three rounds. GRPO and GTPO share the clipped objective and auxiliary takeover SFT loss, differing only in the advantages in Equation (45). We evaluate SFT and each round's checkpoints without operator takeover, using the same pose protocol and binary success criteria with 20 trials per task and checkpoint.

C.7 Additional Real-World Learning Curves

Figure 9 extends the GRPO–GTPO learning-curve comparison in Figure 8 to stacking, battery insertion, and bottle-cap opening. Both methods start from the same task-specific SFT checkpoint and follow the protocol above. GTPO reaches 100% success on stacking after the second round and 95% on battery insertion and opening after the third round. Its final gains over GRPO are 10, 10, and 15 percentage points, respectively.

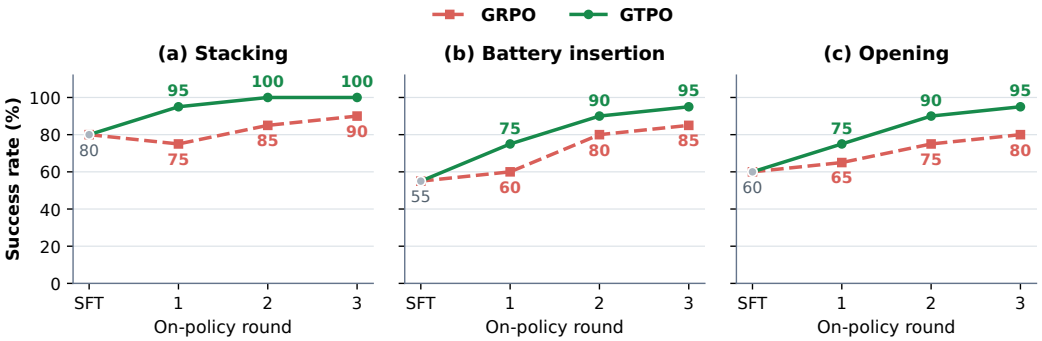

Figure 9: Additional real-world learning curves for (a) stacking, (b) battery insertion, and (c) opening. Each point reports success over 20 trials at SFT initialization or after an on-policy training round. Solid green lines denote GTPO; dashed red lines denote GRPO.