TRAINING WITH BORROWED EYES: DECENTRALIZED MULTI-ROBOT MANIPULATION VIA PRIVILEGED INFORMATION DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning decentralized policies provides a scalable paradigm for multi-robot manipulation by enabling flexible collaboration without centralized observations or communication at deployment. However, agents operating solely on local observations often struggle to infer the intentions and behaviors of their partners, resulting in degraded coordination. In this work, we present a decentralized manipulation framework that transfers coordination knowledge from a privileged centralized policy to locally conditioned agents. During training, a privileged teacher with access to global observations captures coordinated behaviors, while privileged information distillation transfers its action distribution to decentralized students conditioned only on local observations. At deployment, each agent independently predicts its own actions without access to the states, observations, or actions of partners. This formulation retains the scalability and robustness of fully decentralized execution while exploiting privileged global information during training to foster implicit coordination. Experiments on real-world multi-arm manipulation tasks show that our approach consistently outperforms centralized baselines and prior decentralized methods, providing a practical approach for transforming pretrained centralized policies into scalable decentralized robotic systems.

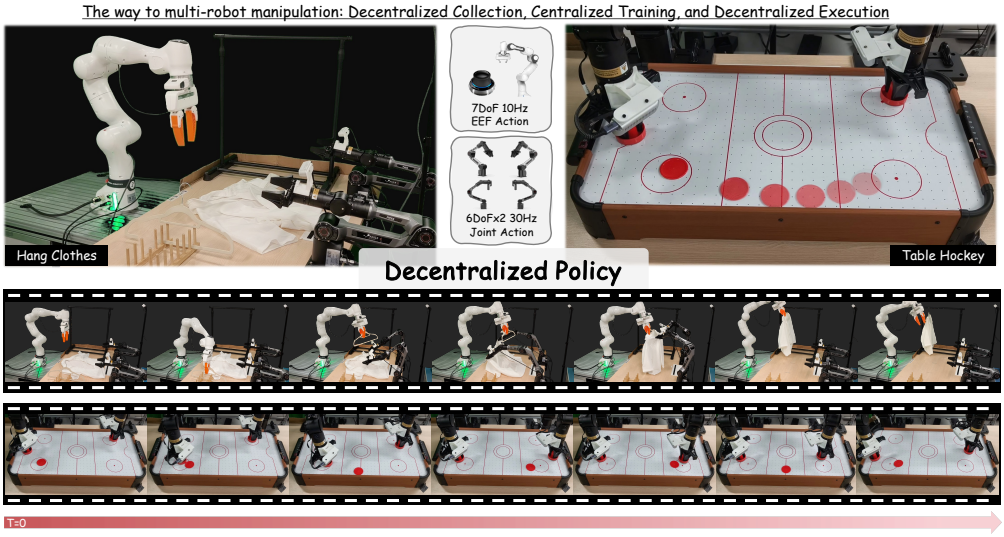


Figure 1: **Highlights overview.** Driven by decentralized policies, heterogeneous robots perform challenging real-world cooperative and competitive tasks including *Hang Clothes* and *Table Hockey*.

1 INTRODUCTION

With the rapid advancement of robot control policies (Chi et al., 2023; Black et al., 2025b;a), it has become increasingly feasible to instruct a single robot to perform a wide range of tasks across industrial, medical, and household domains. A natural question that follows is how to extend such policies

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.

to coordinate multiple robots for accomplishing more complex and challenging tasks. Existing approaches (Tung et al., 2021; Zhao et al., 2023; Black et al., 2025a) commonly adopt a centralized paradigm, where a single policy conditions on team-wide observations, jointly predicts actions for all robots in a single inference step, and subsequently dispatches these actions to individual robots for execution. However, this centralized design faces fundamental scalability limitations as the size of the robot team increases. In particular, the dimensionality of the joint observation context and action space grows rapidly, while the system becomes increasingly dependent on reliable real-time communication and vulnerable to single points of failure. These limitations substantially hinder the practical deployment of centralized policies in real-world multi-robot systems.

Decentralized policies offer a promising alternative (He et al., 2025; Dong et al., 2026; Park et al., 2026; Doshi et al., 2026). Under this paradigm, each robot performs inference and execution solely with its local policy, without requiring explicit inter-robot communication. By conditioning each robot's policy only on its local observations, decentralized execution naturally avoids many of the limitations inherent to centralized approaches. However, it also introduces a fundamental challenge: **How can an independently operating policy, equipped with only local observations, learn to coordinate effectively with its partners?** Figure 9 (a) in appendix illustrates the impact of naive decentralization on cooperative performance under identical settings. Several recent methods address this challenge. *LatentToM* (He et al., 2025) learns private ego and shared consensus representations for decentralized coordination, but relies on a shared centralized view. *MIMIC-D* (Dong et al., 2026) proposes a centralized-training, decentralized-execution diffusion framework that captures multimodal coordination behaviors, but requires access to the states of all robots during training. Park et al. (2026) instead distill privileged multi-agent dynamics into a collaborative latent that each agent predicts from local observations at deployment. Although effective, these methods learn task-specific policies rather than adapting large pretrained vision-language-action backbones, even though such models are typically pretrained under a centralized learning paradigm. *CHORUS* (Doshi et al., 2026) takes an initial step by demonstrating that pretrained coordination priors can be partially transferred to decentralized policies through simple weight sharing. However, weight sharing alone **does not explicitly bridge the gap between globally informed coordination knowledge and locally conditioned decentralized decision-making**.

To address this challenge, inspired by knowledge distillation (Hinton et al., 2015) and homogeneous distillation (Furlanello et al., 2018), we propose **Privileged Information Distillation (PID)**, a two-stage post-training framework that transforms a centralized pretrained model into decentralized policies as shown in Figure 2. Starting from the centralized paradigm of pretrained models, i.e., *Full Observation to Full Action (F2F)*, we first train a **Privileged Teacher** that models *Full Observation to Single Action (F2S)*. By retaining access to joint observations while predicting the action of a single robot, the teacher can fully exploit its partners' observations during action generation, thereby preserving the coordination knowledge inherited from the centralized model. Since its reliance on joint observations prevents the Privileged Teacher from decentralized deployment, we further derive a homogeneous student policy from the teacher, modeling *Local Observation to Single Action (L2S)*. During training, we distill the teacher's action flow field into the student, encouraging the student to identify coordination cues from local observations and develop representations that approximate those induced by joint observations. Compared with directly training a decentralized policy, our homogeneous privileged information distillation allows the student's action flow field to be shaped by privileged collaborative information, rather than merely imitating ground-truth actions under local conditioning.

In this paper, our contributions are threefold:

- We establish a decentralized policy framework and present a systematic study of multi-robot manipulation through transferring pretrained centralized policies in simulation. Our approach achieves state-of-the-art performance on the multi-arm manipulation benchmark RoboFactory.
- We introduce Privileged Information Distillation for decentralized policy learning, which transfers the coordination capability of a privileged teacher into a homogeneous decentralized student conditioned solely on local observations.
- We build a heterogeneous real-world multi-arm system and establish a complete pipeline of decentralized data collection, centralized training, and decentralized execution. The system successfully accomplishes four challenging multi-robot cooperative and competitive tasks, demonstrating the practical feasibility and strong potential of our decentralized framework.

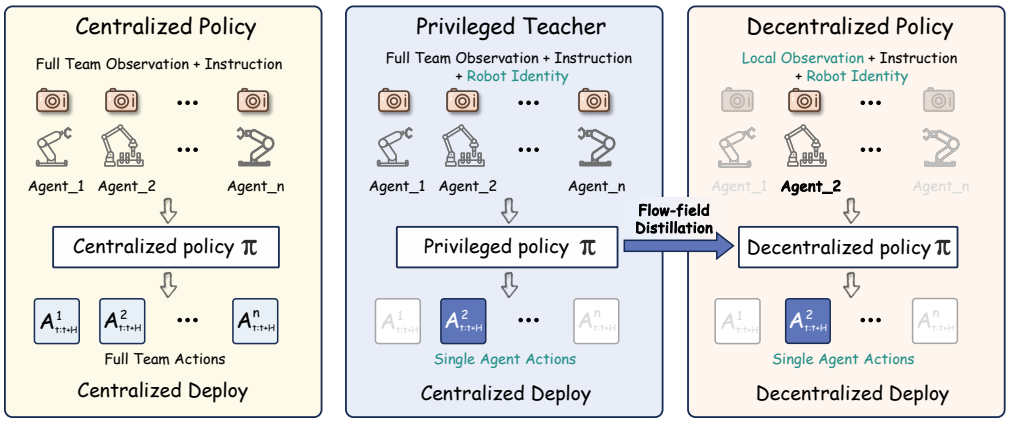


Figure 2: From centralized to decentralized manipulation via privileged distillation. The pretrained *F2F* policy is first adapted into an *F2S* privileged teacher, whose coordination knowledge is then distilled into an *L2S* decentralized policy.

2 Related Work

Vision-Language-Action Models for Manipulation. Early approaches such as ACT (Zhao et al., 2023) and Diffusion Policy (Chi et al., 2023) demonstrate the effectiveness of temporally coherent action chunks and generative action modeling for complex manipulation tasks. Building upon large-scale vision-language pretraining, Vision-Language-Action models (VLAs) further integrate semantic understanding with low-level robot control (Brohan et al., 2023; Kim et al., 2025; Liu et al., 2025; Black et al., 2025b;a). More recent models such as $\pi_{0.7}$ (Physical Intelligence et al., 2026) and InternVLA-A1.5 (Ma et al., 2026) further improve open-world generalization through richer context conditioning and latent foresight, respectively. Collectively, these advances in foundation models make it feasible to extend powerful pretrained manipulation policies to decentralized multi-robot scenarios, laying the foundation for our approach.

Decentralized Multi-Robot Manipulation Policy. Decentralized coordination has been extensively studied in multi-agent reinforcement learning (MARL), where agents are required to make decisions based on local observations while jointly optimizing a shared objective (Lowe et al., 2017; Rashid et al., 2018; Yu et al., 2022). These methods are primarily developed on low-dimensional multi-agent control domains and are difficult to directly scale to visually rich robotic manipulation from simulator environments. Recent works have extended decentralized learning to multi-robot manipulation through imitation learning. LatentToM (He et al., 2025) learns agent-specific and shared latent representations to enable implicit coordination between decentralized diffusion policies. MIMIC-D (Dong et al., 2026) adopts the CTDE paradigm to model multimodal coordinated behaviors with decentralized diffusion policies, while CLS-DP (Park et al., 2026) distills privileged multi-agent dynamics into collaborative latent representations that can be inferred from local observations at deployment. More recently, CHORUS (Doshi et al., 2026) shows that pretrained VLA visuomotor priors can support decentralized collaboration through a shared policy that independently controls heterogeneous robots from local observations. Leveraging these advances, we highlight the potential of combining large-scale pretrained policies with decentralized execution for scalable multi-robot manipulation.

Distillation in Robot Learning. Knowledge distillation has been widely adopted in robot learning to transfer knowledge from privileged or computationally expensive teacher policies to deployable student policies (Hinton et al., 2015). Prior works (Lee et al., 2020; Miki et al., 2022) employ teacher–student learning to transfer policies trained with privileged simulation information to controllers operating from onboard observations. Related strategies leverage privileged state information to guide vision-based agents in autonomous driving and robotic manipulation (Chen et al., 2020; Lee et al., 2022). More recently, Acar et al. (2024) transfer complementary visual information from multi-view teachers to single-view policies. These studies demonstrate the effectiveness of distillation in bridging the gap between information-rich training and constrained deployment. In contrast, our work focuses on coordination knowledge transfer in multi-robot manipulation.

3 METHOD

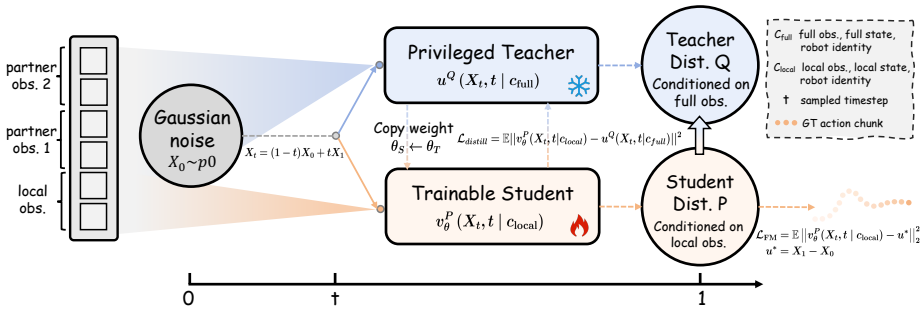


Figure 3: The detailed design of Privileged Information Distillation

3.1 PRELIMINARIES AND PROBLEM FORMULATION

Centralized Policy. Consider a robotic manipulation system consisting of $n \geq 1$ robots. At time step t, robot i receives an observation o_t^i, which typically contains its visual observations and proprioceptive states. We denote the joint observation and action as

$$\mathbf{o}_t = \{o_t^1, \ldots, o_t^n\}, \qquad \mathbf{a}_t = \{a_t^1, \ldots, a_t^n\}. \tag{1}$$

Modern Vision-Language-Action models (VLAs) additionally condition on a language instruction c and commonly predict an action chunk over a horizon of H steps. A centralized VLA policy can therefore be formulated as

$$\pi_\theta^{\text{cen}}\left(\mathbf{A}_t \mid \mathbf{o}_t, c\right), \qquad \mathbf{A}_t = \left\{A_t^1, \ldots, A_t^n\right\}, \tag{2}$$

where

$$A_t^i = (a_t^i, a_{t+1}^i, \ldots, a_{t+H-1}^i) \tag{3}$$

denotes the action chunk of robot i. Thus, the policy has access to observations from all robots and jointly models their actions. This formulation subsumes the standard single-robot VLA setting when $n = 1$.

Formulation of Decentralized Policy. We focus on multi-robot manipulation with $n \geq 2$, where each robot must act solely according to locally available information during execution. Specifically, robot i only observes its local observation o_t^i and predicts its own action chunk A_t^i:

$$\pi_{\theta_i}^{\text{dec}}\left(A_t^i \mid o_t^i, c^i\right), \qquad i \in \{1, \ldots, n\}. \tag{4}$$

Here, $c^i = [c; p_{\text{id}}^i]$ concatenates the instruction c with robot i's identity prompt p_{id}^i. Importantly, o_t^i does not contain the observations, proprioceptive states, or actions of other robots. Following CHORUS (Doshi et al., 2026), we assume that the robots share the same policy parameters, i.e., $\theta_i = \theta$. Consequently, the decentralized policy of the multi-robot system factorizes as

$$\pi_\theta^{\text{dec}}\left(\mathbf{A}_t \mid \mathbf{o}_t, \mathbf{c}\right) = \prod_{i=1}^{n} \pi_\theta^{\text{dec}}\left(A_t^i \mid o_t^i, c^i\right), \qquad \mathbf{c} = \{c^1, \ldots, c^n\}. \tag{5}$$

This formulation enforces fully decentralized execution: each robot independently produces its local action from its local observation and robot-specific prompt, without accessing global observations or explicitly communicating with other robots at inference time. Although decentralized agents do not access partner observations or global information, local observations are assumed to contain sufficient interaction cues, making the required actions theoretically inferable.

3.2 HOMOGENEOUS PRIVILEGED INFORMATION DISTILLATION

We transfer coordination knowledge from a privileged policy to a deployable local policy through a two-stage teacher–student framework. The teacher and student share the same model architecture and action-flow parameterization, but differ in their accessible observations as shown in Figure 3: the teacher conditions on the joint observation $\mathbf{o}_t$, whereas the student for robot i only observes o_t^i. Both policies predict the same robot-specific action chunk A_t^i conditioned on c^i.

Teacher–Student Data Pairing. For each action step t, we construct n paired instances by treating each robot $i \in \{1, \dots, n\}$ as the target. The teacher input is $(\mathbf{o}_t, c^i)$, the student input is (o_t^i, c^i), and both are paired with A_t^i. One shared teacher and one shared student are used for all i, yielding a homogeneous policy parameterization across robots. Each action chunk is transformed into the common model action space, where all output dimensions are supervised. Dataset partitioning is performed at episode level before robot-wise expansion to prevent trajectory leakage across splits.

Privileged Teacher Training. The teacher is initialized from a pretrained $\pi_{0.5}$ checkpoint and trained with the full privileged observation. Given a clean action chunk A_t^i for target robot i at action step t, Gaussian noise $\boldsymbol{\epsilon}_t^i$, and flow time $\alpha \sim \mathrm{Beta}(1, 1.5)$, we construct

$$x_{\alpha,t}^i = (1-\alpha)\boldsymbol{\epsilon}_t^i + \alpha A_t^i, \qquad u_t^{i,*} = A_t^i - \boldsymbol{\epsilon}_t^i. \tag{6}$$

The teacher velocity field $v_T = f_{\theta_T}(\mathbf{o}_t, c^i, x_{\alpha,t}^i, \alpha)$ is optimized by

$$\mathcal{L}_T = \mathrm{MSE}_{\mathcal{M}}(v_T, u_t^{i,*}), \tag{7}$$

where $\mathcal{M}$ is the action-validity mask.

Homogeneous Distillation. The student adopts the same architecture and action parameterization as the teacher, and is initialized from the trained teacher checkpoint. We query the privileged teacher and decentralized student at the identical noisy action state and flow time, enabling pointwise alignment of their conditional velocity fields. During distillation, the teacher is frozen and kept in evaluation mode. For each training sample, both models are evaluated on the same flow state $(x_{\alpha,t}^i, \alpha)$:

$$v_T = f_{\theta_T}(\mathbf{o}_t, c^i, x_{\alpha,t}^i, \alpha), \qquad v_S = f_{\theta_S}(o_t^i, c^i, x_{\alpha,t}^i, \alpha). \tag{8}$$

The student is jointly supervised by the ground-truth flow field and the teacher field:

$$\mathcal{L}_S = \lambda_{\mathrm{local}}\mathcal{L}_{\mathrm{local}} + \lambda_{\mathrm{field}}\mathcal{L}_{\mathrm{field}}, \tag{9}$$

where

$$\mathcal{L}_{\mathrm{local}} = \mathrm{MSE}_{\mathcal{M}}(v_S, u_t^{i,*}), \qquad \mathcal{L}_{\mathrm{field}} = \mathrm{MSE}_{\mathcal{M}}\left(v_S, \mathrm{sg}[v_T]\right). \tag{10}$$

The field-level term transfers information acquired from privileged observations, whereas the local flow-matching term anchors the student to the ground-truth action field. Their joint optimization therefore enables privileged information transfer while mitigating the influence of inaccurate teacher predictions during student training. We use $\lambda_{\mathrm{local}} = 1$ and $\lambda_{\mathrm{field}} = 0.5$ by default.

Teacher Quality-Aware Regularization. It is naturally observed that the teacher's predictive advantage can vary across samples. For example, the states of other robots may provide more useful information during handover or coordinated manipulation phases, whereas local observations may be sufficient during relatively independent motion phases. Motivated by teacher-reliability weighting (Saputra et al., 2019) and sample-wise distillation (Zhou et al., 2021), we adapt the field-distillation strength to the teacher's relative prediction quality. For a batch of B paired instances, indexed by b, we first compute the teacher and student flow-matching errors on the real action coordinates:

$$e_b^S = \mathrm{MSE}_{\mathcal{M}_{\mathrm{real}}}(v_{S,b}, u_b^*), \qquad e_b^T = \mathrm{MSE}_{\mathcal{M}_{\mathrm{real}}}(v_{T,b}, u_b^*). \tag{11}$$

Here $v_{S,b}$, $v_{T,b}$, and u_b^* denote the corresponding student, teacher, and target flow field for batch instance b. Using detached errors, we define

$$g_b = \sigma\left(\frac{e_b^S - e_b^T}{\kappa r}\right), \qquad r = \max\left(\frac{1}{B}\sum_{b=1}^{B}\frac{e_b^S + e_b^T}{2}, \varepsilon_{\min}\right). \tag{12}$$

Here $\varepsilon_{\min} > 0$ is a small numerical constant, and κ is the gate-temperature parameter. The gates are normalized to unit batch mean and contracted toward one so that their maximum is bounded by $w_{\max}$. The resulting detached weights w_b are directly incorporated into the field-distillation loss:

$$\mathcal{L}_{\mathrm{field}}^{\mathrm{QA}} = \frac{1}{B}\sum_{b=1}^{B} w_b \ \mathrm{MSE}_{\mathcal{M}}\left(v_{S,b}, \mathrm{sg}[v_{T,b}]\right). \tag{13}$$

Consequently, samples for which the teacher is more accurate receive stronger distillation supervision, while unreliable teacher outputs have a limited effect. The final student objective remains

$$\mathcal{L}_S = \lambda_{\mathrm{local}}\mathcal{L}_{\mathrm{local}} + \lambda_{\mathrm{field}}\mathcal{L}_{\mathrm{field}}^{\mathrm{QA}}, \tag{14}$$

with $\kappa = 1$ and $w_{\max} = 2$ in our default configuration. The local flow-matching term anchors the student to demonstrated actions, while the weighted field term transfers privileged guidance according to its relative predictive quality.

3.3 Decentralized Data Collection and Execution

Decentralized Collection. Let $\mathcal{R} = \{r_i\}_{i=1}^{n}$ denote n independently controlled robots. Robot i has embodiment metadata m_i, a local observation space, and a native action space. During episode e, it records only its local trajectory

$$\mathcal{D}^{e,i} = \left\{ \left(o_t^{e,i}, A_t^{e,i}, c^{e,i}, \tau^{e,i} \right) \right\}, \tag{15}$$

where $A_t^{e,i}$ is an H-step action chunk, $c^{e,i}$ contains the task instruction and robot identity, and $\tau^{e,i}$ is the local timestamp. Trajectories belonging to the same episode are associated by their episode identifier and timestamps; each robot retains its native sampling schedule. For the centralized baseline, Align_τ uses $\tau^{e,i}$ to align the robots' trajectories within each episode and resample each episode's local trajectories into synchronized joint samples. The decentralized and centralized training sets are therefore

$$\mathcal{D}_{\mathrm{dec}} = \bigcup_{e} \bigcup_{i=1}^{n} \mathcal{D}^{e,i}, \quad \mathcal{D}_{\mathrm{cen}} = \bigcup_{e} \mathrm{Align}_\tau \left(\mathcal{D}^{e,1}, \ldots, \mathcal{D}^{e,n} \right). \tag{16}$$

Decentralized Execution. At execution, each robot independently predicts and decodes its own chunk:

$$\widetilde{A}_t^i \sim \pi_\theta^{\mathrm{dec}} \left(\cdot \mid o_t^i, c^i \right) \tag{17}$$

It executes the first $K_i \leq H$ commands and replans from a fresh local observation. Robots operate asynchronously, which is particularly important for real-world deployment. In practice, individual robots are often connected through separate networks or computing instances and may experience different round-trip latencies. Asynchronous execution allows each robot to progress at the fastest rate permitted by its own communication and computation pipeline as Figure 4 shows, whereas a centralized policy must wait for the slowest robot at every joint inference step, substantially reducing execution efficiency. Moreover, decentralized execution enables robots to recover from transient desynchronization caused by out-of-distribution behaviors and eventually re-establish coordination.

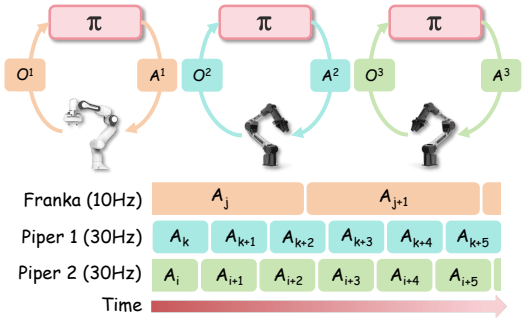


Figure 4: Illustration of decentralized execution.

4 Simulation Experiments

4.1 Experimental Setup

Benchmark. We evaluate our method on RoboTwin 2.0 (Chen et al., 2026) for bimanual tasks and RoboFactory (Qin et al., 2025) for multi-robot tasks. In RoboTwin 2.0, the original demonstrations are collected and organized in a centralized manner. We convert bimanual trajectories into decentralized data by assigning each arm the shared agent-view image and its own wrist-view image as local observations. Detailed information is provided in Appendix A.

Baselines. We use $\pi_{0.5}$ (Black et al., 2025a) as the centralized baseline, which takes full observations and jointly predicts actions for all robots. For decentralized baselines, we consider $\pi_{0.5}$-Dec. (Black et al., 2025a) and CHORUS (Doshi et al., 2026), both built on pre-trained $\pi_{0.5}$. All methods use identical observation/action spaces, chunk sizes, and other settings for each robot in both simulation and real-world experiments. On RoboFactory, we further compare with CLS-DP (Park et al., 2026) and several established centralized policies. Detailed configurations are provided in Appendix C.1.

4.2 Main Results

RoboTwin 2.0. Table 1 demonstrates results on bimanual benchmark RoboTwin 2.0 (Chen et al., 2026). The tasks are divided into non-cooperative and cooperative categories, with three examples and a group average reported for each. Given that $\pi_{0.5}$ is predominantly pretrained on centralized

Table 1: **RoboTwin 2.0 results.** Task success rate (%; higher is better). Each category shows three tasks and its full-category average; the final column averages all 50 evaluated tasks. $\pi_{0.5}$ uses full observations as a centralized approach; Others use local observations at deployment.

Method	Non-cooperative tasks (32)				Cooperative tasks (18)				Overall
	Move Can Pot	Place A2B Right	Stack Blocks (3)	**Group Avg.**	Handover Block	Handover Mic	Place Cans Plasticbox	**Group Avg.**	**Avg.**
Centralized									
$\pi_{0.5}$ (Black et al., 2025a)	84.0	**77.0**	64.0	**76.56**	41.0	60.0	**94.0**	**66.39**	**72.90**
Decentralized									
$\pi_{0.5}$-Dec. (Black et al., 2025a)	76.0	58.0	70.0	73.94	45.0	43.0	85.0	58.06	68.22
CHORUS (Doshi et al., 2026)	78.0	57.0	75.0	75.38	48.0	47.0	88.0	59.78	69.76
Ours	**88.0**	66.0	**78.0**	76.50	**52.0**	**62.0**	92.0	62.56	71.48

Note: **Bold** and underline indicate the best and second-best results in each column, respectively.

bimanual data, where joint observations are mapped directly to joint actions, its superior performance under the centralized setting is both expected and reasonable. Overall, $\pi_{0.5}$-Dec and CHORUS exhibit a larger performance drop on cooperative tasks than on non-cooperative tasks, which is consistent with our hypothesis that **collaboration places stronger demands on inferring partner intent from local observations**. Notably, our method suffers a substantially smaller degradation and achieves a **4.5%** average success rate improvement over $\pi_{0.5}$-Dec on cooperative tasks, demonstrating that **PID effectively enhances the policy's ability to infer coordination cues from local observations**. Further statistical support is presented in Figure 9 (b).

Table 2: **RoboFactory results.** Task success rate (%; higher is better). Methods are grouped by execution paradigm. *Average* is the mean over the displayed tasks, excluding missing results.

Method	Initialization	3 robots		4 robots		Average
		Camera Alignment	Three Robots Stack Cube	Take Photo	Long Pipeline Delivery	
Centralized						
DP (Chi et al., 2023)	From Scratch	3.0	0.0	0.0	0.0	0.8
DP3 (XYZ+RGB) (Ze et al., 2024)	From Scratch	18.0	0.0	11.0	—	9.7
GauDP (Wang et al., 2025)	From Scratch	26.0	0.0	3.0	—	9.7
$\pi_{0.5}$ (Black et al., 2025a)	Pretrained $\pi_{0.5}$	94.0	10.0	75.0	90.0	67.3
Decentralized						
Local-GauDP (Wang et al., 2025)	From Scratch	15.0	0.0	2.0	—	5.7
CLS-DP (Park et al., 2026)	From Scratch	55.0	20.0	8.0	—	27.7
$\pi_{0.5}$-Dec. (Black et al., 2025a)	Pretrained $\pi_{0.5}$	92.0	22.0	70.0	91.0	68.8
CHORUS (Doshi et al., 2026)	Pretrained $\pi_{0.5}$	96.0	29.0	73.0	94.0	73.0
Ours	Pretrained $\pi_{0.5}$	**97.0**	**36.0**	**78.0**	**96.0**	**76.8**

RoboFactory. Table 2 extends the comparison to multi-robot manipulation in RoboFactory (Qin et al., 2025). The selected tasks comprise all three-robot and four-robot tasks. While centralized policies perform competitively in bimanual tasks on RoboTwin, their performance degrades substantially as the number of robots increases to 3 or 4, falling behind decentralized alternatives. Under the same $\pi_{0.5}$ backbone, $\pi_{0.5}$-Dec already matches or surpasses the centralized baseline, while our method further improves upon it by **8.0** percentage points. These results provide strong evidence that decentralized policies **offer a more scalable control paradigm for multi-robot manipulation**.

4.3 Asynchronous Evaluation

We conduct this experiment to evaluate the policy's ability to recover from execution desynchronization. At the beginning of each task, we intentionally pause the left or right robot for 5, 10, or 15 action steps before resuming execution. This manually induced mismatch creates an out-of-distribution coordination state, allowing us to directly assess whether the policy can recover from desynchronization without relying on additional recovery demonstrations. In real-world deployments, such mismatches may arise at any time due to fluctuating network latency, temporary communication disruptions, or single-robot failures.

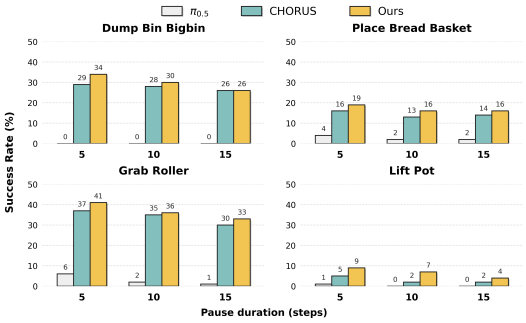


Figure 5: Asynchronous evaluation results on 4 cooperative tasks on RoboTwin 2.0.

As shown in Figure 5, the centralized policy exhibits little ability to recover once the robots become desynchronized, largely due to its tightly coupled joint-action training paradigm. In contrast, our decentralized policy substantially mitigates this issue and retains a degree of recovery capability even under the most severe 15-step delay setting. Such resilience to transient desynchronization is particularly important for robust real-world deployment.

4.4 Ablation Study

We further investigate two key components of our distillation scheme and their different combinations in this section. We also explore the effect of completely removing distillation supervision. The experimental results validate the effectiveness and rationale of our design choices.

Table 3: **Component-wise analysis.** Success rate (%) on RoboTwin 2.0. Each subsequent block varies one component while fixing the other; colored cells identify the component being varied.

Variant	λ_{field}	Regularization	Non-coop. SR (%)	Coop. SR (%)	Overall SR (%)
Teacher weight copy	0	None	75.32	60.34	69.93
Distillation strength	0.1	Teacher-aware	75.56	60.69	70.21
	0.25	Teacher-aware	76.29	61.72	71.04
	0.5*	Teacher-aware	**76.50**	**62.56**	**71.48**
	1.0	Teacher-aware	75.90	62.04	70.91
Regularization method	0.5	None	75.24	62.20	70.55
	0.5	Step-aware	75.88	62.08	70.91
	0.5	Teacher-aware*	**76.50**	**62.56**	**71.48**

Note: $\star$ marks the default configuration, repeated across blocks. $\lambda_{\text{field}} = 0$ retains local supervision but disables distillation.

Distillation strength. We analyze the contribution of privileged field supervision by varying $\lambda_{\text{field}} \in \{0, 0.1, 0.25, 0.5, 1.0\}$ with teacher-aware regularization fixed, as summarized in Table 3. The teacher weight copy variant with $\lambda_{\text{field}} = 0$ updates the student weights using only local flow-matching supervision. The nonzero settings test the balance between matching the teacher field and fitting the ground-truth actions, with Teacher-aware regularization as the default. As a result, $\lambda_{\text{field}} = 0.5$ is the suitable choice for the best trade-off between the distillation and ground-truth objectives, and it is used in our method.

Regularization. We compare None, Step-aware, and Teacher-aware regularization with a fixed $\lambda_{\text{field}} = 0.5$ in Table 3. Without regularization, all samples are assigned uniform distillation weights. Step-aware adaptively decreases the distillation weight during training, reducing reliance on teacher guidance as the student becomes more capable. In contrast, teacher-aware regularization dynamically adjusts weights based on the relative flow-matching errors of the teacher and student, emphasizing samples where the teacher provides more reliable supervision. The results show that Step-aware improves over uniform weighting, while Teacher-aware regularization achieves further gains. As shown in Figure 11, the teacher has a larger advantage over the student at early training stages and gradually decreases as the student improves, following a trend similar to Step-aware but providing finer-grained, sample-level adaptation within data batches.

Teacher-Student Analysis. As shown in Figure 6, we examine how teacher training steps affect both teacher quality and the performance of the distilled student. (1) When the teacher has not converged, ground-truth supervision keeps the student close to the directly trained decentralized baseline. (2) The best teacher (80k steps) does not yield the best student, which is instead distilled from the 60k checkpoint. Moderate overfitting may improve the teacher's own SR while reducing the transferability of its knowledge. (3) Distillation benefits cooperative tasks most; for non-cooperative tasks, local observations may already suffice, while additional observations may introduce distracting information and slightly degrade teacher performance.

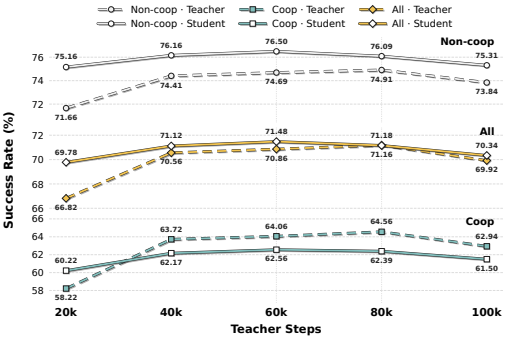


Figure 6: Teacher-Student analysis across teacher training steps.

5 Real-World Experiments

5.1 Experimental Setup

As shown in Figure 7, our system comprises three independently controlled robots: one 7-DoF Franka Research 3 robotic arm and two 6-DoF arms of an AgileX Piper platform. Each robot uses the local agent view, its own wrist view, and its native proprioceptive state. Franka operates at 10 Hz using a 10-dimensional end-effector (EEF) space with 6d rotation representation (Zhou et al., 2019), formulated as $\mathbf{s}^F, \mathbf{a}^F = [\mathbf{p}, \mathbf{r}_{6D}, g] \in \mathbb{R}^{10}$, whereas each Piper arm operates at 30 Hz in a 7-dimensional joint space, formulated as $\mathbf{s}^{P_i}, \mathbf{a}^{P_i} = [q_1, \ldots, q_6, g] \in \mathbb{R}^7$. Through this naturally heterogeneous setup, we aim to demonstrate that our decentralized framework has stronger adaptability and can accommodate diverse embodiments, control frequencies, and action spaces. We collect 40–60 demonstrations per task (184 in total). During evaluation, each task is evaluated over 20 trials for every method, and we report both SR and APR (average progress rate) as metrics.

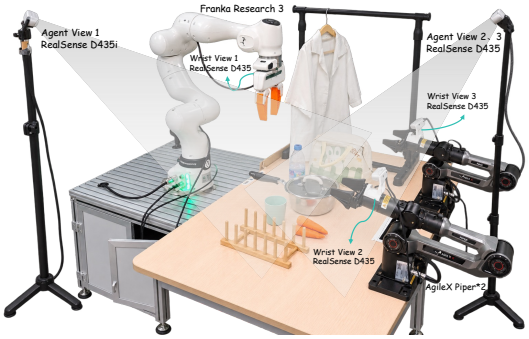


Figure 7: Real-world experimental setup.

5.2 Main Results

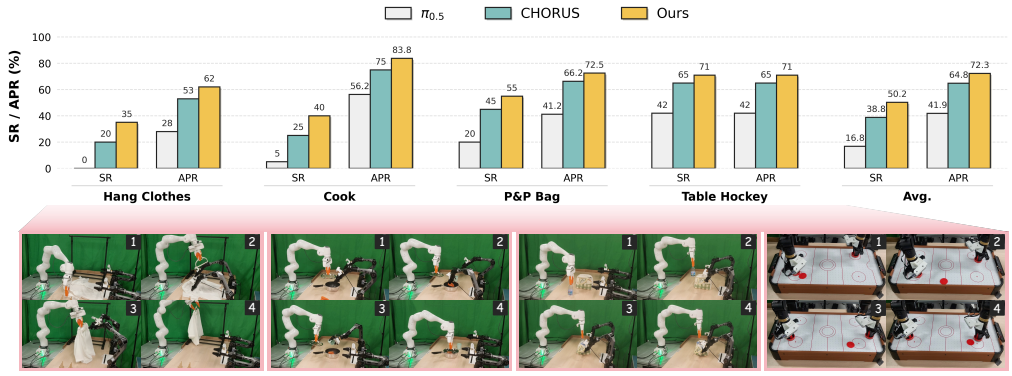


Figure 8: Results on real-world multi-robot manipulation tasks.

Figure 8 demonstrates that our approach consistently improves manipulation performance across all evaluated tasks. By aggregating knowledge from heterogeneous robots under decentralized paradigms, our method increases the average SR / APR from **16.8% / 41.9%** of the pretrained $\pi_{0.5}$ to **50.2% / 72.3%**, substantially narrowing the gap between individual robot capabilities and multi-robot deployment with different configurations. Compared with CHORUS, our approach further achieves higher performance on every task, yielding an average improvement of **11.4%** SR and **7.5%** APR. This validates the effectiveness of decentralized learning and our distillation approach for scalable multi-robot manipulation. The APR definition and computation, additional real-world experimental details are provided in Appendix B, with videos available in Supplementary Material.

6 Conclusion

In this work, we investigate the potential of decentralized policies for multi-robot manipulation. To leverage powerful foundation manipulation models pretrained under a centralized paradigm, we propose Privileged Information Distillation, a two-stage training framework that transfers the coordination capability of a privileged teacher into decentralized policies. We systematically evaluate decentralized control in simulation across varying team sizes, from two-arm to four-arm settings. We further build a heterogeneous three-robot real-world system and demonstrate several challenging collaborative manipulation tasks, highlighting the practical capability and potential of our framework. Future work will extend decentralized policy learning to a broader range of robot embodiments, particularly mobile manipulators, and explore more complex multi-robot tasks.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, implement methods, and clean and reformat datasets. We did not use generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, or interpret results. Formulating mathematical claims, generating synthetic datasets, developing theoretical models or conceptual frameworks, providing key ingredients for mathematical proofs, and assisting with proof writing are not applicable to this work. Additionally, we used generative AI tools to summarize or analyze existing literature, improve the readability of the manuscript, and identify relevant literature. All AI-assisted text was reviewed and revised by the authors for accuracy and consistency with the underlying research. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Our method and experimental setups are described in Sections 3, 4, and 5. To support the reliability of our findings, we repeated experiments as extensively as available computational and robotic hardware resources permitted. The supplementary materials provide clear videos of the real-world experiments. We will organize and publicly release the source code with the final version of the paper to facilitate reproducibility.

REFERENCES

Cihan Acar, Kuluhan Binici, Alp Tekirdağ, and Yan Wu. Visual-policy learning through multi-camera view to single-camera view knowledge distillation for robot manipulation tasks. *IEEE Robotics and Automation Letters*, 9(1):691–698, 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 *Proceedings of the 9th Conference on Robot Learning*, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, 2023.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proceedings of the Conference on Robot Learning*, 2020.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu

Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

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

Dayi Dong, Maulik Bhatt, Seoyeon Choi, and Negar Mehr. MIMIC-D: Multi-modal imitation for MultI-agent coordination with decentralized diffusion policies. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Ria Doshi, Tian Gao, Annie Chen, Chelsea Finn, and Jeannette Bohg. CHORUS: Decentralized multi-embodiment collaboration with one VLA policy. *arXiv preprint arXiv:2606.12352*, 2026.

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Chengyang He, Gadiel Mark Sznaier Camps, Xu Liu, Mac Schwager, and Guillaume Adrien Sartoretti. Latent theory of mind: A decentralized diffusion architecture for cooperative manipulation. In *Proceedings of the 9th Conference on Robot Learning*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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 *Proceedings of the 8th Conference on Robot Learning*, 2025.

Alex X. Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, Claudio Fantacci, Jose Enrique Chen, Akhil Raju, Rae Jeong, Michael Neunert, Antoine Laurens, Stefano Saliceti, Federico Casarini, Martin Riedmiller, Raia Hadsell, and Francesco Nori. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *Proceedings of the 5th Conference on Robot Learning*, 2022.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Zhaxizhuoma, Yuqiang Yang, Jiaqi Peng, Xueyuan Wei, Yangkun Zhu, Jiahao Jiang, Xing Gao, Hanqing Wang, Feng Yuan, Kailin Li, Xueyue Zhu, Tai Wang, Yan Ding, Jiangmiao Pang, Jia Zeng, Jingjing Zhang, Bowen Zhou, Yao Mu, Chunhua Shen, and Weinan Zhang. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

Chanyoung Park, Minsung Yoon, Andrew Jeong, and Sung-eui Yoon. Distilling collaborative dynamics into latent space for implicit coordination in decentralized multi-agent manipulation. *arXiv preprint arXiv:2606.22982*, 2026.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. RoboFactory: Exploring embodied agent collaboration with compositional constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Muhamad Risqi U. Saputra, Pedro P. B. de Gusmao, Yasin Almalioglu, Andrew Markham, and Niki Trigoni. Distilling knowledge from a deep pose regressor network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Albert Tung, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Learning multi-arm manipulation through collaborative teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Ziye Wang, Li Kang, Yiran Qin, Jiahua Ma, Zhanglin Peng, Lei Bai, and Ruimao Zhang. GauDP: Reinventing multi-agent collaboration through gaussian-image synergy in diffusion policies. In *Advances in Neural Information Processing Systems*, 2025.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, 2022.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023.

Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. In *International Conference on Learning Representations*, 2021.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

A Simulation Experiments on RoboTwin 2.0 and RoboFactory Benchmarks

A.1 Task Categorization by Bimanual Coupling

We categorize the 50 RoboTwin 2.0 tasks into two groups according to the degree of bimanual coordination required by the task. The categories are defined based on whether the two arms are independent or required to coordinate in space or time and even simultaneously manipulate the same object.

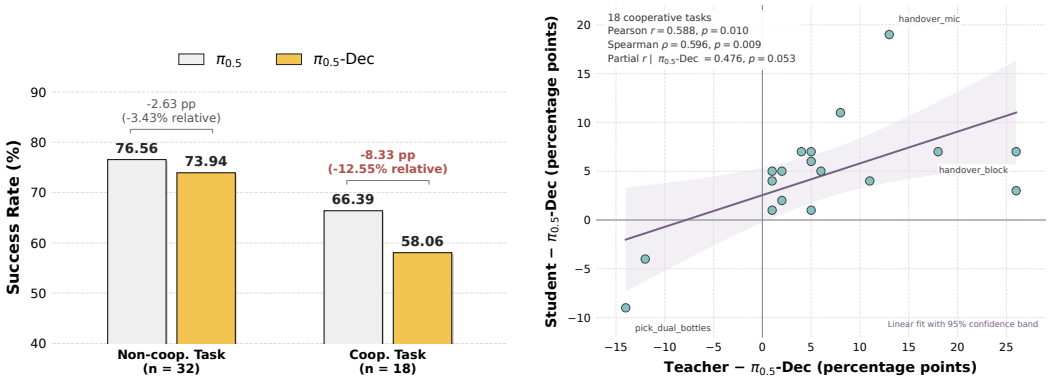
Non-cooperative tasks. Non-cooperative tasks are tasks in which only one arm is actively used, or the two arms can act without meaningful spatial or temporal coupling. This category contains 32 tasks:

adjust_bottle, *beat_block_hammer*, *blocks_ranking_rgb*, *blocks_ranking_size*, *click_alarmclock*, *click_bell*, *move_can_pot*, *move_pillbottle_pad*, *move_playingcard_away*, *move_stapler_pad*, *open_laptop*, *open_microwave*, *place_a2b_left*, *place_a2b_right*, *place_container_plate*, *place_empty_cup*, *place_fan*, *place_mouse_pad*, *place_object_scale*, *place_object_stand*, *place_phone_stand*, *place_shoe*, *press_stapler*, *rotate_qrcode*, *shake_bottle*, *shake_bottle_horizontally*, *stack_blocks_three*, *stack_blocks_two*, *stack_bowls_three*, *stack_bowls_two*, *stamp_seal*, and *turn_switch*.

Cooperative tasks. Cooperative tasks require coordination between the two arms in space or time, such as handover, sequential dependency, shared target regions, or one arm stabilizing or repositioning an object for the other arm, and even simultaneously grasping and manipulating the same object. This category contains 18 tasks:

dump_bin_bigbin, *grab_roller*, *handover_block*, *handover_mic*, *hanging_mug*, *lift_pot*, *pick_diverse_bottles*, *pick_dual_bottles*, *place_bread_basket*, *place_bread_skillet*, *place_burger_fries*, *place_can_basket*, *place_cans_plasticbox*, *place_dual_shoes*, *place_object_basket*, *put_bottles_dustbin*, *put_object_cabinet*, and *scan_object*.

A.2 Quantitative Statistics and Analysis



(a) **(b)**

Figure 9: (a) Success rates of $\pi_{0.5}$ and $\pi_{0.5}$-Dec. on non-cooperative and cooperative tasks. (b) Per-task teacher and student improvements over $\pi_{0.5}$-Dec. on the 18 cooperative tasks.

As shown in Figure 9 (a), when adapting $\pi_{0.5}$ from centralized training data to decentralized training, the performance degradation is only 2.63 percentage points on non-cooperative tasks, while it increases to **8.33** percentage points on cooperative tasks (a relative drop of **12.55%**). This observation suggests that cooperative tasks rely more heavily on partner-related information for coordination. Therefore, when explicit communication with partners is unavailable, enabling agents to infer coordination-related information from local observations becomes crucial.

To investigate whether privileged-information distillation effectively transfers coordination knowledge from the centralized teacher to decentralized students, we analyze the performance improvements of the teacher and student policies across 18 cooperative tasks. As shown in Figure 9 (b), the teacher's improvement over $\pi_{0.5}$-Dec is positively associated with the student's improvement over the same baseline. Specifically, the Teacher–$\pi_{0.5}$-Dec gap and Student–$\pi_{0.5}$-Dec improvement achieve Pearson correlation $r = 0.588$ and Spearman correlation $\rho = 0.596$. This trend suggests that tasks where the centralized teacher captures stronger coordination behaviors also benefit more from distillation, indicating that privileged information enables the transfer of implicit partner modeling and cooperative behaviors to decentralized policies.

A.3 Robot Identity Descriptions

Our decentralized policy receives robot identity through text prefixes concatenated with task instructions, using **identical templates for training and inference**.

RoboTwin 2.0. The two arms are independent agents, identified as `left` and `right`. Each prompt specifies the controlled arm and its partner so the shared policy generates the corresponding arm's actions:

- **Left arm:** `You are the left robot arm. Act as the left arm only and coordinate with the right arm. Task: {instruction}`
- **Right arm:** `You are the right robot arm. Act as the right arm only and coordinate with the left arm. Task: {instruction}`

RoboFactory. For N robots, identities are `agent_i`, where $i = 0, \ldots, N-1$. In our data conversion, `agent_i` maps to `Franka_panda_i` and its local camera observations, state, and actions. The identity prefix distinguishes each robot within the common task:

- **Agent i:** `[You are the agent_{i}] {task_instruction}`

Here, i indexes robots from zero, and `task_instruction` is shared by all robots. Identity is supplied only through text, without appending numerical identity codes. Each robot's state remains eight-dimensional (seven joint positions and one gripper dimension).

A.4 Task-wise Simulation Results

As shown in Table 4, we report the detailed success rate of each task under different methods.

Table 4: Task-wise RoboTwin 2.0 success rates (%) in the multi-task simulation setting. All methods use 2,500 demonstrations across 50 tasks and are evaluated in the clean setting. Ours denotes the decentralized student. Tasks are grouped by coupling type; averages are task-weighted. The best result in each row is bolded, including ties.

Simulation Task	$\pi_{0.5}$	$\pi_{0.5}$**-Dec.**	**CHORUS**	**Ours**
Non-cooperative Tasks				
Adjust Bottle	**100.0**	99.0	**100.0**	**100.0**
Beat Block Hammer	87.0	88.0	**91.0**	81.0
Blocks Ranking RGB	80.0	83.0	**88.0**	86.0
Blocks Ranking Size	50.0	53.0	**55.0**	49.0
Click Alarmclock	89.0	84.0	86.0	**90.0**
Click Bell	**96.0**	85.0	85.0	93.0
Move Can Pot	84.0	76.0	78.0	**88.0**
Move Pillbottle Pad	**90.0**	88.0	87.0	83.0
Move Playingcard Away	**88.0**	83.0	81.0	67.0
Move Stapler Pad	72.0	70.0	**73.0**	65.0
Open Laptop	**96.0**	91.0	92.0	95.0
Open Microwave	36.0	**38.0**	35.0	**38.0**
Place A2b Left	76.0	71.0	73.0	**77.0**
Place A2b Right	**77.0**	58.0	57.0	66.0
Place Container Plate	91.0	91.0	**95.0**	**95.0**
Place Empty Cup	**97.0**	95.0	**97.0**	**97.0**
Place Fan	65.0	69.0	76.0	**85.0**
Place Mouse Pad	33.0	28.0	27.0	**40.0**
Place Object Scale	**73.0**	64.0	62.0	63.0
Place Object Stand	**92.0**	90.0	**92.0**	91.0
Place Phone Stand	**60.0**	55.0	56.0	56.0
Place Shoe	**48.0**	42.0	42.0	47.0
Press Stapler	**85.0**	81.0	82.0	80.0
Rotate QRcode	70.0	73.0	**79.0**	**79.0**
Shake Bottle	**100.0**	**100.0**	**100.0**	**100.0**
Shake Bottle Horizontally	**100.0**	**100.0**	**100.0**	**100.0**
Stack Blocks Three	64.0	70.0	75.0	**78.0**
Stack Blocks Two	**98.0**	95.0	96.0	**98.0**
Stack Bowls Three	**75.0**	71.0	72.0	74.0
Stack Bowls Two	95.0	95.0	**98.0**	95.0
Stamp Seal	54.0	53.0	**56.0**	**56.0**
Turn Switch	29.0	27.0	26.0	**36.0**
Category Average	**76.56**	73.94	75.38	76.50
Cooperative Tasks				
Dump Bin Bigbin	**90.0**	82.0	82.0	89.0
Grab Roller	**100.0**	99.0	99.0	**100.0**
Handover Block	41.0	45.0	48.0	**52.0**
Handover Mic	60.0	43.0	47.0	**62.0**
Hanging Mug	14.0	12.0	11.0	**17.0**
Lift Pot	**95.0**	90.0	**95.0**	91.0
Pick Diverse Bottles	**68.0**	64.0	66.0	66.0
Pick Dual Bottles	74.0	83.0	**85.0**	74.0
Place Bread Basket	78.0	78.0	**81.0**	74.0
Place Bread Skillet	**63.0**	5.0	4.0	8.0
Place Burger Fries	**89.0**	82.0	84.0	86.0
Place Can Basket	79.0	81.0	81.0	**86.0**
Place Cans Plasticbox	**94.0**	85.0	88.0	92.0
Place Dual Shoes	26.0	22.0	21.0	**28.0**
Place Object Basket	**83.0**	71.0	72.0	75.0
Put Bottles Dustbin	**65.0**	50.0	52.0	55.0
Put Object Cabinet	21.0	22.0	28.0	**33.0**
Scan Object	**55.0**	31.0	32.0	38.0
Category Average	**66.39**	58.06	59.78	62.56
Overall Average	**72.90**	68.22	69.76	71.48

B Real World Experiment Task Settings

B.1 Data Collection

During data collection, we use a 3D space mouse teleoperation interface to control the Franka robot arm and employ an additional matched-arm teleoperation setup for the AgileX Piper robot. The number of demonstrations collected for each task is reported in Table 5.

Table 5: Number of demonstrations collected for each real-world task.

Task	`cook`	`hang_clothes`	`pick&place_bag`	`table_hockey`	**Total**
Demonstrations	40	54	50	40	**184**

B.2 Language Instructions

Table 6 lists the language prompts used for the four real-world tasks. The base prompt describes the overall task procedure and is used to train the centralized baseline policy. The blue, green, and purple text denotes the robot identity descriptions for the three different robots, which are used to train all decentralized policies.

Table 6: Task-level and robot-specific language instructions used in real-world experiments.

Prompt	**Language Instruction**
Cook `(cook)`	
Base	Prepare the carrots together: lift and hold the pan lid, place the carrot pieces into the pan, pour the water from the cup into the pan, then replace the lid.
Franka	You control the Franka arm. Prepare the carrots together: lift and hold the pan lid, place the carrot pieces into the pan, pour the water from the cup into the pan, then replace the lid.
Left Piper	You control the left Piper arm. Prepare the carrots together: lift and hold the pan lid, place the carrot pieces into the pan, pour the water from the cup into the pan, then replace the lid.
Right Piper	You control the right Piper arm. Prepare the carrots together: lift and hold the pan lid, place the carrot pieces into the pan, pour the water from the cup into the pan, then replace the lid.
Hang Clothes `(hang_clothes)`	
Base	Hang the white shirt together: hold the clothes hanger steady, lift the shirt by both shoulder areas, guide it onto the hanger, and leave it hanging securely. Finally, hang the clothes on the drying rack.
Franka	You control the Franka arm. Hang the white shirt together: hold the clothes hanger steady, lift the shirt by both shoulder areas, guide it onto the hanger, and leave it hanging securely. Finally, hang the clothes on the drying rack.
Left Piper	You control the left Piper arm. Hang the white shirt together: hold the clothes hanger steady, lift the shirt by both shoulder areas, guide it onto the hanger, and leave it hanging securely. Finally, hang the clothes on the drying rack.
Right Piper	You control the right Piper arm. Hang the white shirt together: hold the clothes hanger steady, lift the shirt by both shoulder areas, guide it onto the hanger, and leave it hanging securely. Finally, hang the clothes on the drying rack.
Pick-and-Place Bag `(pick_and_place_bag)`	
Base	Pack the water bottle together: hold the paper bag open and steady, lift the water bottle, lower it fully into the bag, release it, and leave the bag upright.
Franka	You control the Franka arm. Pack the water bottle together: hold the paper bag open and steady, lift the water bottle, lower it fully into the bag, release it, and leave the bag upright.
Left Piper	You control the left Piper arm. Pack the water bottle together: hold the paper bag open and steady, lift the water bottle, lower it fully into the bag, release it, and leave the bag upright.
Right Piper	You control the right Piper arm. Pack the water bottle together: hold the paper bag open and steady, lift the water bottle, lower it fully into the bag, release it, and leave the bag upright.

Continued on the next page.

Table 6 (continued): Language instructions.

Prompt	Language Instruction
Table Hockey	`(table_hockey)`
Base	Play table hockey together: use the two red strikers to keep the small red puck in play, intercept it on each half of the table, and send it cleanly back across the center line.
Left Piper	You control the left Piper arm. Play table hockey together: use the two red strikers to keep the small red puck in play, intercept it on each half of the table, and send it cleanly back across the center line.
Right Piper	You control the right Piper arm. Play table hockey together: use the two red strikers to keep the small red puck in play, intercept it on each half of the table, and send it cleanly back across the center line.

B.3 APR Evaluation Standard

We adopt Average Progress Rate (APR) as an additional evaluation metric for real world tasks. Specifically, each task is decomposed into several task-specific milestones, and scores are assigned according to the completion quality of each stage. This design enables a finer-grained evaluation of intermediate manipulation performance instead of relying solely on binary task success.

For each task τ, the final APR score is computed as the average normalized progress score over its K evaluation trials:

$$\mathrm{APR}_\tau = \frac{1}{K}\sum_{k=1}^{K}\frac{s_{\tau,k}}{S_\tau^{\max}}, \tag{18}$$

where $s_{\tau,k}$ is the progress score obtained in the k-th trial of task τ, and $S_\tau^{\max}$ is the maximum possible score for that task. Note that for the *Table Hockey* task, due to its unique task objective and scoring protocol, we set SR and APR to the same value.

The detailed scoring criteria for each task are described below.

Hang Clothes. Hang the white shirt together: hold the clothes hanger steady, lift the shirt by both shoulder areas, guide it onto the hanger, and leave it hanging securely. Finally, hang the clothes on the drying rack.

- Franka grasping and lifting the hanger: +1
- Both the left and right Piper arms grasping the shirt collar: +1
- Successfully placing the shirt onto the hanger: +2
- Hanging the hanger with the shirt on the clothes rail: +1

Maximum score: 5

Cook. Prepare the carrots together: lift and hold the pan lid, place the carrot pieces into the pan, pour the water from the cup into the pan, then replace the lid.

- Franka grasping and lifting the pan lid: +1
- The left Piper arm placing the carrots into the pan: +1
- The right Piper arm pouring the water from the cup into the pan: +1
- Franka replacing the lid on the pan: +1

Maximum score: 4

Pick-and-Place Bag. Pack the water bottle together: hold the paper bag open and steady, lift the water bottle, lower it fully into the bag, release it, and leave the bag upright.

- Franka grasping and lifting the water bottle: +1
- Both the left and right Piper arms grasping the bag and pulling it open: +2
- Franka placing the water bottle into the bag: +1

Maximum score: 4

Table Hockey. Play table hockey together: use the two red strikers to keep the small red puck in play, intercept it on each half of the table, and send it cleanly back across the center line.

- Hitting the puck from the striking arm's side across the center line: +1 per successful hit, counted over one minute

Maximum score: 10 (Scores exceeding 10 are clipped to 10.)

B.4 Visualization of Real World Tasks

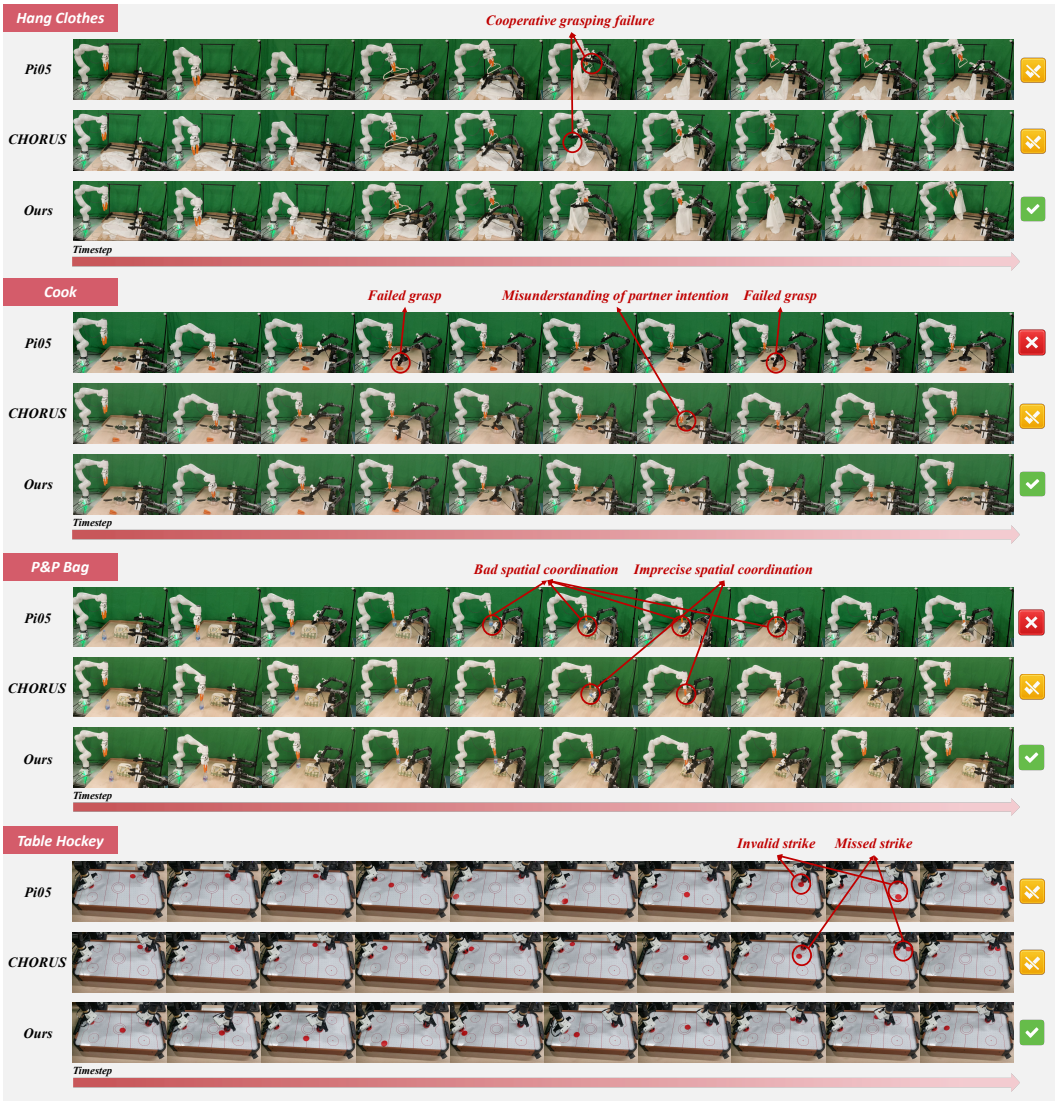


Figure 10: Detailed visualizations of the four real-world tasks across different methods

Figure 10 presents qualitative comparisons of execution results on four challenging real-world tasks across three methods. In the *Hang Clothes* task, both $\pi_{0.5}$ and CHORUS fail to achieve reliable cooperative grasping of the deformable cloth, while our method maintains stable coordination. In the *Cook* task, the centralized $\pi_{0.5}$ policy jointly predicts actions for three heterogeneous arms, leading to degraded manipulation performance and even failure in grasping the carrot. Meanwhile, CHORUS suffers from incorrect partner intention understanding during the pouring stage, where it prematurely closes the pot lid by mistakenly assuming that the partner has completed the pouring operation. In contrast, our approach rarely encounters such coordination failures. In the *P&P Bag* task, $\pi_{0.5}$ and CHORUS exhibit varying degrees of inaccurate spatial coordination, resulting in failed bottle placement or collisions. These qualitative results demonstrate the advantage of our decentralized formulation and distillation framework in handling heterogeneous multi-robot coordination.

C Model Details and Training Configurations

C.1 Training Configurations

We perform full-parameter fine-tuning on the open-source $\pi_{0.5}$ model using the Flow Matching objective. For RoboTwin 2.0 (Chen et al., 2026), RoboFactory (Qin et al., 2025), and real-world experiments, we individually adopt a multi-task training paradigm to fine-tune the model on all tasks jointly. Detailed training configurations are provided in Table 7.

Table 7: **Training configurations.** Default settings for the privileged teacher and decentralized student.

Configuration	Value
Data & Input	
Backbone	$\pi_{0.5}$
RoboTwin 2.0 training set	50 tasks, 2,500 demonstrations
RoboFactory training set	4 tasks, 400 demonstrations
Real-world training set	4 tasks, 184 demonstrations
Action chunk length H	50
Optimization	
Batch size	128
Optimizer	AdamW
Peak learning rate	2.5×10^{-5}
Decay learning rate	2.5×10^{-6}
Gradient Clip	1.0
Teacher objective	Flow-matching MSE
Student objective	$\lambda_{\text{local}}\mathcal{L}_{\text{local}} + \lambda_{\text{field}}\mathcal{L}^{\text{QA}}_{\text{field}}$
Local loss weight λ_{local}	1
Field loss weight λ_{field}	0.5
Teacher initialization	Pretrained $\pi_{0.5}$ checkpoint
Student initialization	Trained teacher checkpoint
Teacher checkpoint for default student	60k steps
Student checkpoint for evaluation	50k steps
Quality-Aware Distillation	
Gate temperature κ	1
Maximum gate weight $w_{\max}$	2
Teacher checkpoints analyzed	20k, 40k, 60k, 80k, 100k steps
Teacher during distillation	Frozen and evaluation only
Infrastructure	
Training GPUs	8 × H100
Inference GPUs	RTX 5090

C.2 Details of Teacher Quality-Aware Regularization

Figure 11 tracks the fraction of samples for which the teacher has a lower flow-matching error than the student ($e_b^T < e_b^S$) with $\lambda_{\text{field}} \in \{0.1, 0.25, 0.5, 1.0\}$, using a frozen teacher trained for 60k steps. By jointly analyzing Table 3 and Figure 11, we find that an overly small λ_{field} reduces the student's ground-truth error and teacher-better sample fraction, but weakens the transfer of coordination knowledge. Conversely, an overly large λ_{field} increases the propagation of teacher errors to the student, also degrading task performance.

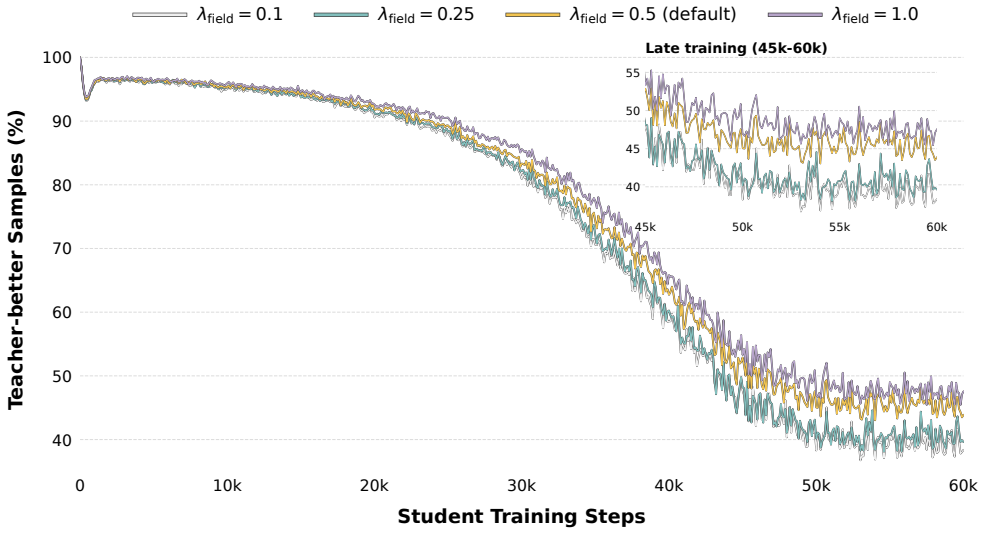


Figure 11: **Teacher-better sample fraction during student training.** The inset enlarges the 45k–60k interval.