

Before We Act: Learning Action-Conditioned Team-Response Beliefs for Decentralized Multi-Robot Manipulation

Zihao Li^{1,2}, Yanan ZHOU¹, Ranpeng Qiu², Guoqiang Ren², Weiming Zhi^{1,*}

¹The University of Sydney

²Zeno AI

The best action depends on what the teammate will do next

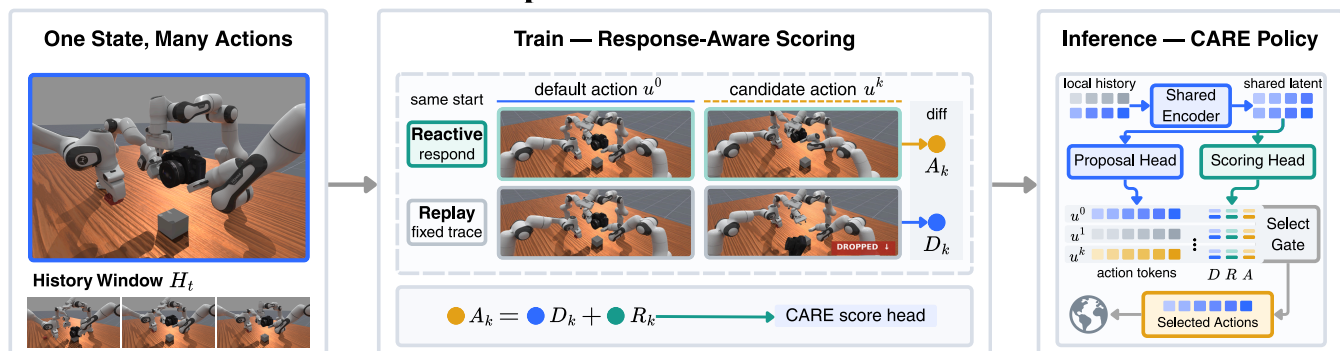


Fig. 1: **Before acting, predict what teammate response contributes.** From the same local history (left), CARE compares candidate and nominal actions under reactive and replayed teammate behavior during training (center). The difference between total advantage A_k and direct advantage D_k supplies the response label R_k . At execution (right), the response readout uses predictive memory to guide selection, with calibrated fallback to the nominal action.

Abstract—In multi-robot manipulation, an action’s value depends on how teammates respond. We present CARE (Cooperative Action–Response Belief Estimation), a framework for learning *team-response beliefs* from local history. A predictive belief encoder represents the interaction and conditions a nominal action. Matched reactive and replay continuations separate each candidate’s direct advantage from the utility contributed by teammate adaptation. A candidate-conditioned readout learns this response distribution alongside total advantage, and calibrated selection chooses an alternative or retains the nominal action. Our analysis characterizes when response changes the preferred action and how prediction and source errors limit the recoverable gain. With the representation, proposals, and data fixed, explicit response supervision improves candidate ranking and reduces selection regret relative to learning the same information through branch utilities. The benefit is concentrated in decisions where teammate adaptation changes the preferred action. Across RoboFactory, the MARS Control Track, and physical collaboration tasks, CARE achieves the highest average success among the evaluated policies, combining a capable nominal policy with selective coordination adjustments. Together, these results show that CARE turns predictions of teammate adaptation into effective decentralized action selection. Project page: <https://jeong-zju.github.io/care/>.

I. INTRODUCTION

Consider a robot carrying an object with a partner and deciding when to turn. Turning now shortens its own path but can destabilize the object if the partner is still regrasping, whereas waiting briefly makes little direct progress but lets the partner finish the grasp. The uncertainty that matters is how the team will respond to the robot’s next action. The

same decision arises when releasing an object in a handoff, yielding a shared workspace, or synchronizing a lift.

We define a *team-response belief* as the distribution of a candidate’s gain or loss in team utility due to teammate adaptation, relative to replaying nominal teammate commands. As in partially observable control [1], this belief summarizes uncertainty from local history. Latent teammate modeling [2], scene representations [3], shared policies [4], language negotiation [5], and memory [6] provide complementary tools for coordination. Our focus is the action-conditioned utility contrast and its prediction from local history.

Demonstrations train capable imitation policies [7], [8], [9], [10], [11], [12], [13], but record only the action taken and the response that followed. They do not specify how teammates would have responded to alternatives. Multi-agent critics [14], [15], [16] can learn their combined value without separating teammate adaptation. CARE compares reactive and replayed continuations to measure whether this adaptation helps or hurts team utility. Our hypothesis is that explicitly supervising this contribution improves finite-data learning when response changes the preferred action.

CARE, short for *Cooperative Action–Response Belief Estimation*, connects predictive representation learning with *paired interaction augmentation*. Its *predictive belief encoder* represents the evolving interaction from local history, supporting both nominal action generation and response inference. Paired continuations compare each candidate with the nominal action under reactive teammates and teammates replaying nominal commands. Subtracting the replay advantage from the reactive advantage isolates the response interaction. A

*Corresponding author: Weiming Zhi.
weiming.zhi@sydney.edu.au.

candidate-conditioned response model connects this supervision to the predictive representation. We retain the original demonstrations and add explicit interaction supervision, complementing methods that reshape demonstrations [17], [18], [19].

We train CARE in two stages. We first learn the predictive belief encoder and its action interface on demonstrations, then freeze them as the reference policy π_{ref} . A fixed paired dataset trains the response readout, whose total-advantage lower quantile is calibrated for selection [20], [21], [22]. This separation lets us compare response targets with total-value and information-matched branch-utility targets while holding representation, proposals, and data fixed. Closed-loop success separately evaluates the complete policy π_{CARE} . Its selector-off control retains the predictive encoder and measures the nominal path.

This paper makes three contributions.

- We formulate action-conditioned team-response beliefs through matched reactive and replay contrasts and learn them from local predictive memory.
- We characterize when response changes the preferred action and how source and prediction errors limit the gain available to a learned selector.
- We demonstrate the finite-data benefit of response supervision with fixed-proposal, information-matched comparisons, alongside complete-policy evaluation in simulation and on hardware.

II. LEARNING PREDICTIVE TEAM BELIEFS

A. Defining Team-Response Belief

Robot i acts from local history $H = H_t^i = (o_{\leq t}^i, u_{< t}^i)$, including its RGB observations, proprioception, executed actions, and task instruction. CARE constructs a finite candidate set $\mathcal{U}(H) = \{u^0, \dots, u^{K-1}\}$ around its nominal action chunk u^0 . Alternatives such as waiting, retreating, or slowing down use only the acting robot’s chunk and joint state (Section IV). Let $U(Y)$ be a fixed bounded team utility over a given horizon, specified experimentally in Section IV. Candidates, utility, horizon, and reference teammates define the response target.

B. Predictive Belief Encoder

An instantaneous local observation may be compatible with different stages of a teammate’s motion. CARE therefore learns a predictive latent belief from history, so that action generation and response estimation can depend on temporal evidence. The *predictive belief encoder* parameterizes

$$p_\phi(z | H) = \prod_{j,f} \text{Cat}(z_{jf}; p_{\phi,jf}(H)), \quad (1)$$

where j indexes interaction slots and f categorical factors. This learned distribution represents predictive alternatives without assigning semantic labels to hidden teammate intent. Let $Z_b(H)$ denote a continuous embedding of its categorical probabilities and $Z_e(H)$ retained evidence of past prediction discrepancies. Their combined representation $Z(H) = [Z_b(H); Z_e(H)]$ supports response inference. To

make the belief relevant to control, CARE conditions the nominal action on Z_b as

$$u^0(H) = G_\theta(H) + g(H) \odot \Delta_\theta(H, Z_b). \quad (2)$$

Here G_θ is a temporal imitation policy, Δ_θ a belief-conditioned correction, and $g \in [0, 1]$ an evidence-dependent gate. The correction allows the same local observation to yield different actions when its preceding interaction differs.

Learning makes the belief informative about demonstrated actions and subsequent interaction, using team and future evidence available only during training (Section III-C). Predicting demonstrated interactions alone does not identify responses to untried actions, motivating paired supervision.

C. Paired Interaction Augmentation

To distinguish an action’s direct effect from the response it elicits, CARE compares two teammate behaviors under a fixed reference policy $\pi_{\text{ref}}(H) = u^0(H)$. Let $\bar{u}^{-i}$ be the teammate command sequence induced by the nominal ego action. For each candidate u^k , define Y_k^R as the outcome when teammates respond according to π_{ref} , and Y_k^P as the outcome when they replay $\bar{u}^{-i}$. The continuations share the initial interaction context and coupled exogenous randomness. Replay holds teammate commands fixed while preserving physical evolution. The paired advantages are

$$D_k = U(Y_k^P) - U(Y_0^P), \quad A_k = U(Y_k^R) - U(Y_0^R), \quad (3)$$

$$R_k = A_k - D_k.$$

Here D_k measures direct advantage with teammate commands fixed, and R_k measures the additional gain or loss associated with their adaptation. Equivalently, R_k compares the reactive-replay utility gap of the candidate with that of the nominal action. Physical interaction remains in both branches, with direct advantage defined by holding teammate commands fixed. The components may be dependent and are defined relative to the reference teammates and nominal trace. Simulator or model-generated labels must preserve this matched intervention. Altering observations alone does not identify R_k .

The *team-response belief* for candidate k is $b_H^k = \mathcal{L}(R_k | H)$, with $u^k \in \mathcal{U}(H)$ fixed by the proposal rule. It describes uncertainty about response utility, whereas $p_\phi(z | H)$ represents predictive interaction state. A candidate-conditioned model estimates the former from the latter’s representation. CARE predicts direct, response, and total quantiles separately because dependent marginal quantiles do not generally add.

D. Why Representation and Pairing Are Complementary

Two teammate policies may agree on demonstrated actions yet respond oppositely to a new candidate. Paired branches define a response target beyond those observations, while predictive memory supplies temporal evidence for estimating it locally. The readout estimates quantiles conditional on $(Z(H), u^k, u^0)$, so the representation must retain the temporal evidence needed to predict response utility. Predicting all four branch utilities provides the same label information as their contrasts. Directly learning R_k directs shared features toward the utility of adaptation, making this contribution explicit

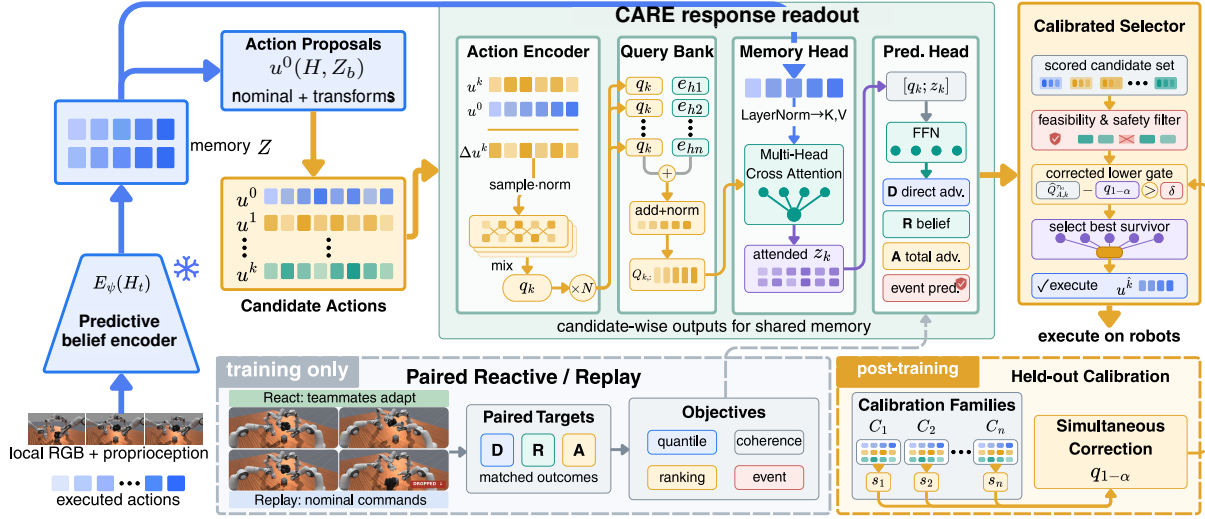


Fig. 2: CARE learns and acts on team-response beliefs. A predictive belief encoder represents the interaction from local history and conditions the nominal action (left). Paired reactive and replay contrasts distinguish teammate adaptation from direct effects through $R = A - D$ (center). A candidate-conditioned readout estimates these advantages from the learned representation, which remains fixed during response learning. Calibrated selection chooses a supported alternative or retains u^0 (right). Execution requires only local history.

during training. Q4 tests this learning bias with both the representation and proposals fixed.

E. Decision Value of Response Information

To characterize when teammate response changes the preferred action, let $d_k = \mathbb{E}[D_k | H]$, $r_k = \mathbb{E}[R_k | H]$, and $a_k = d_k + r_k$, with $d_0 = r_0 = a_0 = 0$. Under a fixed tie-breaking rule let $k_D \in \arg \max_k d_k$ and $k_* \in \arg \max_k a_k$. We define the decision value of response information as

$$\Gamma(H) = a_{k_*} - a_{k_D} = \max_k \{ (r_k - r_{k_D}) - (d_{k_D} - d_k) \} \geq 0. \quad (4)$$

Response changes the optimal value exactly when some candidate's relative response gain exceeds its direct disadvantage. Waiting is useful, for example, when the partner's additional progress outweighs its direct cost. A large $|r_k|$ alone is not enough, nor is a reversal between two inferior candidates. Equation (4) defines the strata of Q2.

Recoverable decision value. At a fixed history, let $\beta(H)$ bound the candidate-wise Wasserstein distance between source and target laws of (D_k, R_k) under the ℓ_1 transport cost. Suppose estimated component means differ from their source means by at most ϵ_D and ϵ_R , uniformly over candidates. Since $(D, R) \mapsto D + R$ is 1-Lipschitz, $\hat{a}_k = \hat{d}_k + \hat{r}_k$ obeys $|\hat{a}_k - a_k| \leq e$, where $e = \beta + \epsilon_D + \epsilon_R$. The mean-based choice $\hat{k} \in \arg \max_k \hat{a}_k$ then satisfies

$$a_{\hat{k}}(H) - a_{k_D}(H) \geq \Gamma(H) - 2e(H), \quad (5)$$

Indeed, $a_{k_*} \leq \hat{a}_{k_*} + e \leq \hat{a}_{\hat{k}} + e \leq a_{\hat{k}} + 2e$. Subtracting a_{k_D} gives (5). A unique best candidate is recovered when its lead exceeds $2e$, and $\Gamma > 2e$ ensures improvement over direct-only choice. Thus response magnitude matters through its effect on the decision margin. The same bound applies to an equally accurate total-value predictor. The benefit of decomposition lies in learning accurate decision scores from finite paired data, as examined in Q4.

Reference teammates and deployment. Freezing π_{ref} fixes its parameters, not its actions. Reactive teammates

still act on the observations induced by each candidate, so paired labels capture adaptation through the nominal policy. At deployment, teammates can also select alternatives. These interventions and the histories they induce create the additional shift. When a selector abstains, it executes exactly π_{ref} at its current history. The shared nominal policy anchors deployment, although earlier interventions can change the history. Concurrent overrides can also reinforce or counteract one another beyond the response represented in the labels. Thus CARE evaluates candidates against reference teammates without solving for mutually consistent selector decisions. These differences enter β alongside outcome-model mismatch. Equation (5) describes their effect on mean-based decisions.

III. FROM BELIEF TO ACTION

A. Action-Conditioned Response Readout

The response model must distinguish different actions in the same interaction. CARE therefore conditions its advantage predictions jointly on the predictive representation and candidate as

$$\hat{Q}_{X,k}^\tau(H) = f_{\omega,X}^\tau(Z(H), u^k, u^0), \quad X \in \{D, R, A\}. \quad (6)$$

Here $\tau \in \mathcal{T}$ indexes quantile levels. Response quantiles summarize $\hat{b}_H^k$, while total-advantage quantiles support conservative choice. Shared readout features let response supervision inform total-value estimation without imposing quantile additivity. An additional probability estimate describes adverse events. Training holds Z and the proposal rule fixed (Section III-C).

B. Calibrated Selection under Uncertainty

After freezing the response readout, we calibrate its lower quantile on M_{cal} families, where family j contains all candidates and matched repeats $\ell \in \mathcal{R}_j$ of one snapshot with paired advantages $A_{j,k,\ell}$ and $A_{j,0,\ell} = 0$. For a lower level $\tau_{10} \in \mathcal{T}$, the family score

$$s_j = \max_{k > 0, \ell \in \mathcal{R}_j} [\hat{Q}_{A,k}^{\tau_{10}}(H_j) - A_{j,k,\ell}]_+ \quad (7)$$

is the largest overshoot in a family. Set $q_{1-\alpha}$ to the $\lceil (M_{\text{cal}} + 1)(1 - \alpha) \rceil$ -th smallest score, or $+\infty$ if this index exceeds M_{cal} . The calibrated bound $\underline{A}_k(H) = \hat{Q}_{A,k}^{\tau_{10}}(H) - q_{1-\alpha}$ shifts all candidates equally, preserving their ordering while raising the threshold for intervention. Among alternatives with predicted adverse-event probability at most η , the policy picks $\hat{k} \in \arg \max_k \underline{A}_k(H)$ and executes

$$\pi_{\text{CARE}}(H) = \begin{cases} u^{\hat{k}}, & \underline{A}_{\hat{k}}(H) > \delta, \\ u^0, & \text{otherwise,} \end{cases} \quad (8)$$

keeping u^0 when no alternative is feasible. The margin δ raises the evidence required to leave the nominal action and the threshold η shrinks the set of actions considered, and both are fixed before any closed-loop evaluation.

Role of calibration. For a frozen readout and exchangeable calibration and test families, the order-statistic construction covers all candidates and repeats of a new family with probability at least $1 - \alpha$. The maximum accounts for selection among one robot’s candidates, allowing dependence within a family. It does not cover joint interventions by multiple selectors. Coverage is marginal over families from the reference distribution, and teammate interventions or hardware shift can break exchangeability. Q5 evaluates calibration through task success and intervention frequency during collective execution.

C. Training Objectives

Predictive representation. Imitation supervises nominal actions, while prediction of future features and teammate behavior shapes the belief. Categorical KL alignment transfers predictive structure from a teacher with team observations and future evidence to local inference. Separate gradients update each distribution, while teacher reconstruction preserves future information. Prediction targets and regularization are specified below.

Response learning. With the encoder and nominal policy fixed, paired advantages train only the response readout through

$$\mathcal{L} = \mathcal{L}_q + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{safe}} \mathcal{L}_{\text{safe}} + \lambda_{\text{coh}} \mathcal{L}_{\text{coh}}. \quad (9)$$

$\mathcal{L}_q$ averages pinball loss over D, R, A , non-nominal candidates, and quantile levels, estimating the distributions needed for conservative choice. Pairwise logistic ranking on median A emphasizes action ordering, and cross-entropy supervises adverse-event filtering. The coherence term $|\hat{Q}_{A,k}^{\tau} - \hat{Q}_{D,k}^{\tau} - \hat{Q}_{R,k}^{\tau}|$ softly couples the heads during training. This is a training regularizer rather than an assumption of quantile additivity. Selection uses the separately predicted total quantiles, and Q4 evaluates the contribution of this coupling. Calibration follows training. Loss coefficients are specified with the experimental settings.

IV. EXPERIMENTS

We evaluate how CARE combines predictive memory, response learning, and calibrated selection to coordinate decentralized manipulation. Q1 and Q6 assess the complete policy in simulation and on hardware. Q2 identifies the decisions that benefit from teammate response, Q3 and Q4

TABLE I: Closed-loop SR (%) on RoboFactory and the MARS Control Track.

RoboFactory							
Policy	LB ₂	CA ₃	LP ₄	TP ₄	PS ₂	PF ₂	Avg.
<i>No external robot-action pretraining</i>							
ACT [8]	90	50	80	0	5	100	54.17
Diffusion Policy [9]	65	20	0	5	0	35	20.83
LatentToM [†] [2]	100	35	95	35	95	75	72.50
GauDP [†] [3]	70	10	0	0	0	70	25.00
ManiFlow [10]	75	50	0	40	20	70	42.50
<i>External robot-action pretraining</i>							
RDT-IB [11]	35	25	0	40	20	60	30.00
OpenVLA-OFT [12]	10	55	0	15	0	15	15.83
$\pi_{0.5}$ [13]	85	65	55	65	70	15	59.17
<i>CARE: visual pretraining only</i>							
CARE (ours)	100	60	90	100	100	65	85.83

MARS Control Track						
Policy	PC ₂	SC ₂	TS ₃	FS ₄	Avg.	Mean
<i>No external robot-action pretraining</i>						
ACT [8]	30	45	15	0	22.50	38.33
Diffusion Policy [9]	0	0	0	0	0.00	10.42
LatentToM [†] [2]	30	35	0	0	16.25	44.38
GauDP [†] [3]	0	0	0	0	0.00	12.50
ManiFlow [10]	25	35	0	0	15.00	28.75
<i>External robot-action pretraining</i>						
RDT-IB [11]	15	35	10	0	15.00	22.50
OpenVLA-OFT [12]	0	0	0	0	0.00	7.92
$\pi_{0.5}$ [13]	15	55	15	0	21.25	40.21
<i>CARE: visual pretraining only</i>						
CARE (ours)	15	95	10	20	35.00	60.42

Subscripts denote robot counts. [†]Run under the shared decentralized interface. LB Lift Barrier, CA Camera Alignment, LP Long Pipeline Delivery, TP Take Photo, PS Pass Shoe, PF Place Food, PC Place Cube in Cup, SC Strike Cube Hard, TS Three Robots Place Shoes, FS Four Robots Stack Cube.

examine candidate selection and supervision, and Q5 studies when the policy should intervene. Together, these comparisons connect task performance to the decisions underlying it.

A. Benchmarks and Protocol

We evaluate six collaborative RoboFactory tasks [23] and four hidden-seed MARS Control Track tasks [24], including teams of three and four robots (Fig. 3). Within each benchmark, policies use the same demonstrations, local observation boundary, action interface, control rate, episode limits, and shared arm weights. Each CARE robot runs the same frozen π_{CARE} from its own local history, while label generation uses the shared reference policy π_{ref} defined below and baselines retain their own policies. Only the external-pretraining group may use robot-action data outside the benchmark. At execution, each arm observes only its RGB history, proprioception, executed-action history, and task instruction, never teammate observations or simulator state.

Evaluation objective. Team utility measures task progress and completion together with coordination costs under a fixed scalarization. Progress is clipped to $[-1, 1]$, and completion indicates success within the horizon. Conflict, duplicate work, and deadlock are measured as fractions of control steps. Activity imbalance is the range of robot activity fractions, while delay is time to success divided by the horizon, or one if unsuccessful. Progress and completion contribute positively, and the remaining terms are costs. Collisions and drops supply a separate adverse-event label. All response variants share this objective and its scale.

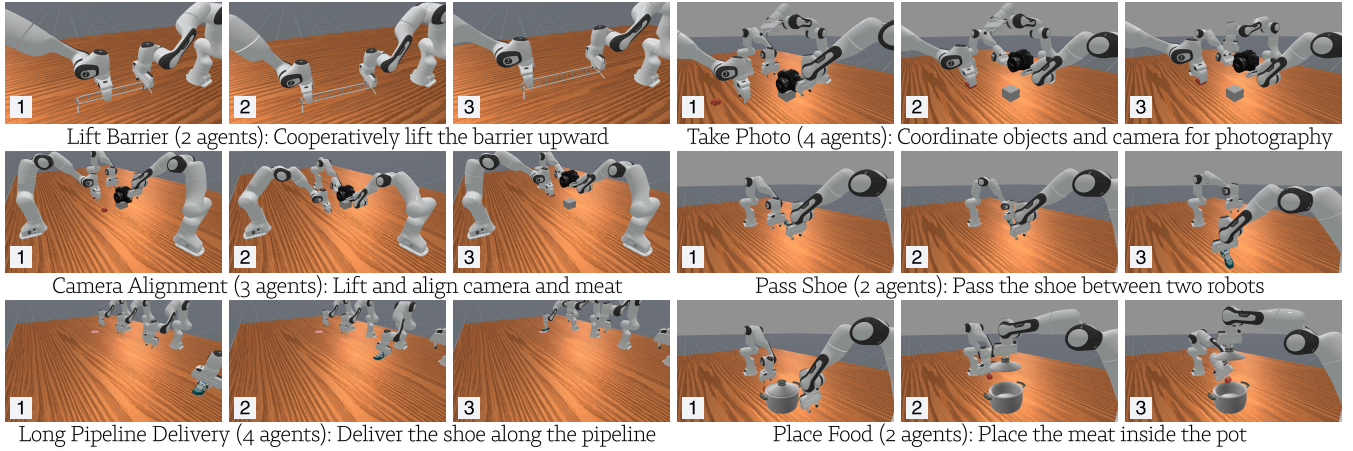


Fig. 3: **Collaborative tasks in RoboFactory.** Representative sequences of the six tasks with teams of two to four robots.

Nominal policy and predictive representation. Frozen DINO features [25], proprioception, executed actions, and task instruction form a 16-step local history. A temporal Transformer and action-query decoder predict joint-target chunks [8]. The predictive belief encoder uses causal attention and discrepancy-based event memory, with ego, teammate, and interaction slots. Its belief-conditioned residual follows (2). Predictive training uses future visual features at 0.2, 0.4, 0.8, and 1.6 seconds, teammate state and action targets, and teacher reconstruction. Regularization includes exchange consistency, a feature-variance floor, and action ranking under matched versus shuffled belief representations. These demonstration-based objectives are distinct from supervision from reactive and replay branches. DINO supplies visual pretraining only.

Training order and reference policy. Demonstration training uses team and future information for supervision, while execution uses only local history. We freeze the nominal path as $\pi_{\text{ref}}(H) = u^0(H)$, retaining its predictive belief encoder. Paired continuations match physical state, observation history, policy memory, pending commands, and random-number state. Reactive teammates recompute the nominal action, whereas replay uses recorded commands. All response variants share one frozen dataset and reference policy. Only the unmatched control changes pairing. Candidate differences and horizon embeddings query the frozen belief and event representation by cross-attention. We train the shared readout and calibrate selection, with no iterative relabeling. World-model augmentation supplies observations after altered actions, but response targets still require both matched branch outcomes. The complete π_{CARE} runs independently on every robot.

Settings. CARE scores $K = 6$ candidates over a 16-step horizon. Besides u^0 , five transforms are held over the 8-step commitment window, namely waiting for 4 steps, retreating by half the nominal displacement, advancing or delaying the gripper by 4 steps, and slowing motion to $0.6\times$. Quantile levels are $\mathcal{T} = \{.05, .25, .5, .75, .95\}$ with $\tau_o = .05$. Calibration uses $M_{\text{cal}} = 24$ families at $1 - \alpha = .9$, with $\delta = 0$, $\eta = .25$, and loss weights $(.20, .10, .10)$ in (9). The seven utility components listed above use relative weights

$(30, 30, 8, 6, 6, 3, 5)$, normalized by their sum. Each family contains two repeats of K reactive and K replay continuations of at most 64 steps, from which all horizons are read, giving at most 1,536 simulated control transitions per family and about 276,000 over 180 families. This budget covers simulator labeling rather than world-model augmentation, and deployment uses neither simulator nor world-model rollouts.

Metrics. SR is terminal success over 20 rollouts per task, equally weighted, and OR is the percentage of control steps replacing u^0 . Offline, headroom is the reactive oracle’s realized advantage over u^0 , Recov. the fraction recovered, Regret the gap to the best candidate, and Rank the accuracy of candidate ordering against paired reactive outcomes. R MAE is the mean absolute residual of a variant’s response predictor on held-out paired labels (Q4). Headroom, Regret, R MAE, and Γ use the common utility scale defined above.

Splits and statistics. The snapshot evaluation uses 24 held-out test families. Table III reports the deployed seed. Figure 6 instead summarizes three seeds paired by family and seed. We also use trajectory-grouped five-fold cross-validation over all 180 families, with calibration excluded from the test trajectory. Confidence intervals use 10,000 bootstrap resamples clustered by trajectory and family, keeping snapshots, candidates, repeats, and both branches from one trajectory together. Table II reports grouped-CV stratum averages. SR in the mechanism tables is a 120-rollout consistency check. Arrows mark the preferred direction and bold the best entry.

B. Q1. How Capable Is CARE?

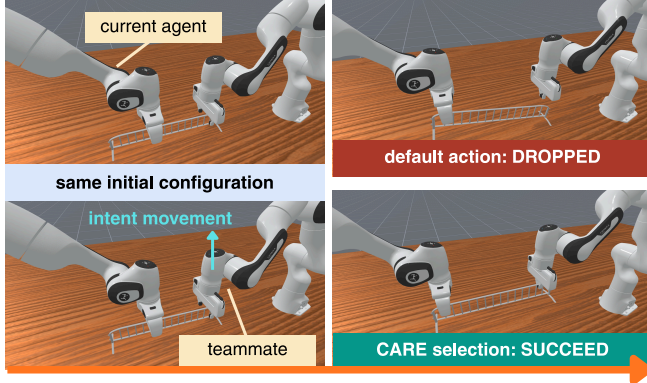
Table I compares CARE with action-only policies and the collaboration-specific LatentToM and GauDP. The pretraining partition separates external robot-action pretraining from visual-only pretraining. This comparison evaluates complete training recipes, including CARE’s paired interaction supervision. Q4 then controls the available supervision to examine the contribution of its response targets.

On RoboFactory, CARE achieves the highest average success. Its strongest results occur on Take Photo and Pass Shoe, where holding or waiting for a partner is central to completion. CARE succeeds in every rollout on both tasks. LatentToM is the strongest baseline and also performs well on Pass Shoe, reinforcing the importance of anticipating partner

TABLE II: Response strata on the 180 held-out decisions under trajectory-grouped cross-validation.

Stratum	n	Γ (utility units)			Recov. $\uparrow$				Regret $\downarrow$			
		Mean	Median	P90	Direct	Total	Branches	CARE	Direct	Total	Branches	CARE
Response-dependent	108	0.0124	0.0108	0.0240	0.060	0.390	0.450	0.560	0.0076	0.0059	0.0054	0.0046
positive reversal	68	0.0148	0.0131	0.0265	0.040	0.440	0.500	0.630	0.0084	0.0054	0.0049	0.0039
negative reversal	40	0.0083	0.0074	0.0178	0.090	0.305	0.365	0.440	0.0062	0.0068	0.0063	0.0058
Response-invariant	72	0	0	0	0.000	0.000	0.000	0.000	0.0000	0.0012	0.0010	0.0009
All decisions	180	0.0074	0.0065	0.0180	0.036	0.234	0.270	0.336	0.0046	0.0040	0.0036	0.0031

Direct, Total, and Branches denote the evaluators of Table III under CARE’s proposals and selection rule.



Selection and Execution using Different Choices

Fig. 4: **Teammate response matters for action selection.** From the same Lift Barrier configuration (left), the default action drops the barrier (top right), while CARE’s response-aware selection follows the teammate’s upward motion and completes the lift (bottom right).

behavior. Precision remains a complementary requirement, as other policies lead on Camera Alignment, Long Pipeline Delivery, and Place Food. Selection still depends on the execution accuracy of available actions. CARE’s nominal path already achieves 80.8% success, and response-based selection raises it to 85.8% with sparse overrides (Table IV). Predictive memory supports the nominal policy, while the response readout identifies when a local adjustment can improve the team’s continuation.

On MARS, CARE also achieves the highest average success, led by Strike Cube Hard and the only successful trials on Four Robots Stack Cube. CARE trails the strongest baselines on the other two tasks, and the low success rates across methods show the difficulty of these settings. Together, the two benchmarks demonstrate that CARE’s overall advantage extends across different collaborative tasks and team sizes.

C. Q2. When Does Team Response Matter for Choice?

We characterize response relevance using $\Gamma(H)$ from (4). Candidate means are estimated by averaging two paired simulator repeats, and strata are frozen before selector comparison. *Response-dependent* decisions have positive estimated Γ beyond the evaluation tolerance. In *response-invariant* decisions, a direct-only choice remains optimal within that tolerance. Positive reversals involve a candidate with $d_k \leq 0 < a_k$, while negative reversals involve one with $d_k > 0 \geq a_k$. The reversals distinguish helpful adaptation from disruptive responses.

Table II shows that teammate response changes the preferred action in 108 of the 180 held-out decisions. CARE has the highest recovery and lowest regret in this stratum, with its clearest advantage on positive reversals. Waiting

TABLE III: Mechanism studies with fixed proposals.

Proposals / selector	SR $\uparrow$	Recov. $\uparrow$	Regret $\downarrow$	OR
Perturbed / CARE	84.2	0.450	0.0055	6.5
CARE / Random	63.3	0.140	0.0086	96.7
CARE / Direct-only	80.0	0.040	0.0096	42.0
CARE / Total-only	81.7	0.280	0.0072	28.0
CARE / CARE	85.8	0.520	0.0048	6.0
Supervision variant	SR $\uparrow$	R MAE $\downarrow$	Rank $\uparrow$	Regret $\downarrow$
Total only (A)	81.7	—*	0.548	0.0072
Total + direct ($A + D$)	83.3	0.0310 [†]	0.571	0.0059
Total + branch utilities	85.0	0.0610 [†]	0.603	0.0061
Direct only (D)	80.8	—*	0.452	0.0083
Unmatched branches	78.3	0.1290	0.438	0.0096
CARE w/o coherence	84.2	0.0520	0.587	0.0057
CARE (full)	85.8	0.0140	0.671	0.0048

Snapshot split, deployed seed. [†]Contrast of marginal medians. *No response head (Q4).

TABLE IV: Selection rules with the same trained response readout.

Selector	OR (%)	SR (%) $\uparrow$
Nominal path (selector off)	0.0	80.8
Posterior mean	98.0	42.5
Uncalibrated lower quantile	38.0	78.3
Candidate-wise calibration	18.0	89.2
Simultaneous calibration	6.0	85.8
CARE (+ safety filter)	6.0	85.8

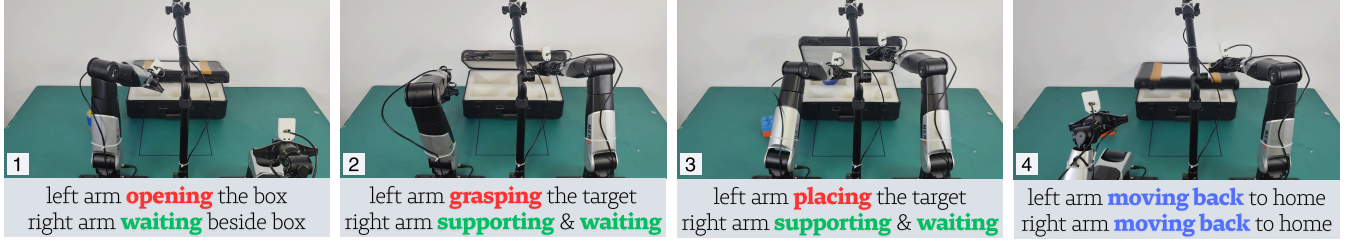
or maintaining support can look unproductive locally while enabling a partner to finish its motion. Negative reversals test the complementary ability to avoid seemingly useful actions that disrupt coordination. Total-only prediction is less reliable here, whereas CARE again achieves the lowest regret. In response-invariant decisions, Direct is already optimal and CARE incurs a small residual regret. Zero recovery in this stratum reflects the absence of positive reactive headroom in these samples. On Lift Barrier (Fig. 4), moving before the partner lifts drops the barrier, while CARE follows its upward motion and preserves support.

D. Q3. Does the Response Readout Improve Selection?

The upper block of Table III compares evaluators over shared candidates with the proposal model fixed. A separate proposal control uses first-action perturbations of u^0 , including a one-step hold, time warps, and a frozen gripper. These brief changes provide less opportunity for adaptation than CARE’s sustained behavioral alternatives.

Random selection wastes the candidates’ concentrated headroom by often replacing a capable nominal action with an inferior alternative. Direct-only selection also intervenes often but recovers little gain, showing why immediate effects are insufficient for coordination. Total-only prediction improves recovery, yet CARE recovers roughly twice as much with far fewer overrides. CARE intervenes at about 6% of steps while recovering half the available headroom. The weaker

Opening Box: Two robots collaboratively open the box, with one stabilizing while the other grasps target object



Folding Bed: Two robots collaboratively fold the quilt, coordinating to hold, pull, and smooth its edges

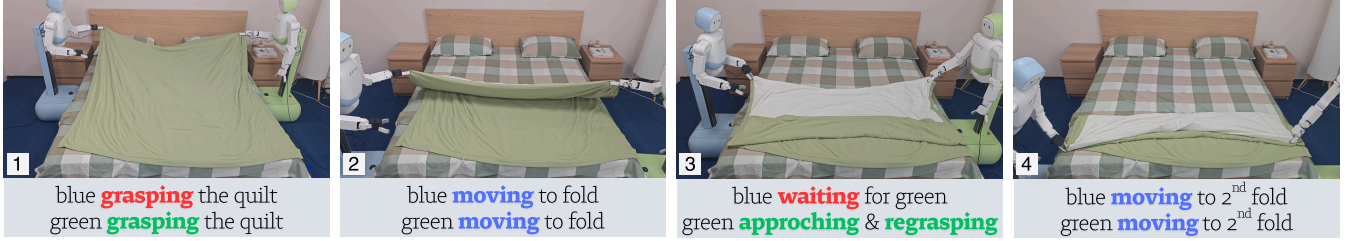


Fig. 5: **CARE on two physical collaboration tasks.** In *Opening Box* (top), one arm supports the lid and waits while its partner grasps and places the object. In *Folding Bed* (bottom), one arm holds the quilt edge and waits for the partner to regrasp. Keyframes highlight sustained support and waiting.

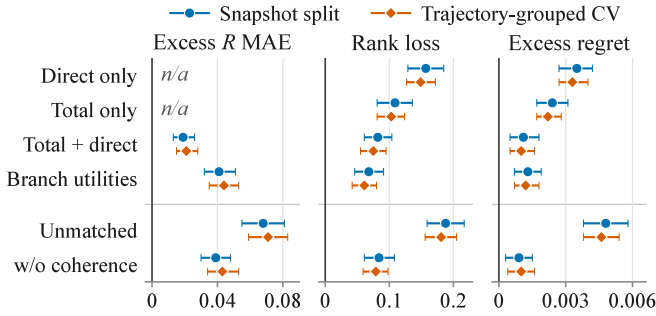


Fig. 6: **Response-readout ablations across three seeds.** Mean paired differences from CARE with 95% bootstrap intervals on the snapshot split (circles) and trajectory-grouped CV (diamonds). Table III reports one deployed seed, not these three-seed means. Upper rows vary the supervision targets and lower rows the coupling.

first-action perturbations support pairing sustained behavioral alternatives with a readout that evaluates their team benefit.

E. Q4. Does Explicit Response Supervision Help?

The lower block of Table III retrain the readout with proposals, representation, data, budget, and selection rule fixed. Total-only and $A + D$ use fewer targets from the same paired outcomes. The stronger branch-utility control predicts all four utilities and total advantage, providing every outcome needed to construct R . Unmatched branches remove seed coupling, while the coherence ablation removes the penalty linking the heads. The table reports the deployed seed, and Fig. 6 averages paired differences across three seeds.

Response targets improve action evaluation. CARE improves ranking accuracy by 0.068 and reduces regret by 0.0013 relative to branch-utility supervision. Both gains persist in the three-seed snapshot comparison, whose paired confidence intervals exclude zero (Fig. 6). Both methods learn a total-advantage head, so response targets improve the selection score. Grouped cross-validation also favors CARE in ranking, and Table II locates its regret advantage in response-dependent decisions. Although branch utilities contain the

TABLE V: Real-world dual-robot evaluation.

Policy	FB	OB	SR	↑ OR
ACT [8]	12/20	10/20	55.0	n/a
$\pi_{0.5}$ [13]	9/20	14/20	57.5	n/a
CARE, selector off	18/20	15/20	82.5	0.0
CARE	20/20	17/20	92.5	5.8

FB Folding Bed, OB Opening Box. Successes out of 20 trials.

same information, explicitly supervising adaptation helps the readout distinguish team value from direct effects. These controlled decision gains complement the closed-loop results.

Pairing and coupling support response learning. Removing seed coupling gives the weakest ranking and largest regret in the supervision study. Removing coherence also increases response error and worsens selection. These results support matching branch conditions and jointly learning the related targets. Here R MAE measures $\mathbb{E}|g(H) - R_k|$, minimized by the conditional median. CARE and its unmatched and coherence variants use $\hat{Q}_{R,k}^5$, while $A + D$ and branch supervision use contrasts of predicted marginal medians. Medians do not commute with subtraction, so this diagnostic compares the resulting response estimators. Rank and Regret directly assess their decisions. Total-only and Direct-only have no response predictor.

F. Q5. Can Selection Balance Intervention and Abstention?

Table IV holds CARE’s readout fixed and varies the decision rule. Posterior-mean selection overrides almost continuously and sharply reduces success. An uncalibrated lower quantile still falls below nominal, motivating a calibrated threshold for intervention.

Calibration supplies this threshold. Candidate-wise calibration achieves the highest success, while simultaneous calibration retains a gain over nominal with one third as many overrides. Since the common shift preserves candidate order, calibration governs abstention and complements the ranking improvements in Q4. The safety filter leaves aggregate rates unchanged. All robots run the tested selector, so these gains include the effects of teammates departing from π_{ref} .

Sparse intervention retains the shared nominal policy at most control steps, providing a practical basis for reference-based evaluation. However, aggregate OR does not measure concurrent overrides or their downstream effects. The results establish effective collective execution, while isolating the policy mismatch requires varying which robots intervene and how strongly.

G. Q6. How Does CARE Perform on Physical Tasks?

We compare CARE with ACT and $\pi_{0.5}$ on *Folding Bed* and *Opening Box* (Fig. 5), with 20 trials per task and policy. Policies share local observations, an 8-D joint-target action space, control frequency, and task budget. Policies are trained on task demonstrations and frozen. CARE arms share a checkpoint but use their own histories. The selector-off control retains the encoder and nominal path. Response supervision uses simulated-twin continuations, with a world model generating observations after altered actions. Calibration uses held-out simulator families.

Table V shows that CARE completes 37 of 40 trials, achieving the highest success across the two physical tasks. The nominal path already outperforms ACT and $\pi_{0.5}$, and selection adds four successful trials. Overrides are sparse and occur during holding and waiting phases. Supporting the quilt edge lets the partner regrasp, while holding the box lid preserves access to the object. Maintaining support thus enables the partner’s progress.

The physical results complement Q3–Q5 by demonstrating coordinated execution with all selectors active. Execution requires one additional readout forward pass per control query and no simulator or world-model rollouts.

V. CONCLUSION AND FUTURE WORK

We presented CARE, a framework that connects local predictive memory, explicit teammate-response supervision, and calibrated action selection. Matched reactive and replay continuations identify how teammate adaptation changes an action’s value, and the analysis characterizes when that contribution changes the preferred decision. With representation, proposals, and data fixed, response supervision improves ranking and regret over learning the same information through branch utilities. Results on RoboFactory, MARS, and two physical tasks demonstrate effective decentralized manipulation through a capable nominal policy and selective coordination adjustments. Future work can adapt proposal distributions [26], extend response learning to changing teammate policies, and combine selection with runtime monitoring [27], [28].

REFERENCES

- [1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998.
- [2] C. He, G. M. S. Camps, X. Liu, M. Schwager, and G. A. Sartoretti, “Latent theory of mind: A decentralized diffusion architecture for cooperative manipulation,” in *Proceedings of the 9th Conference on Robot Learning*, vol. 305, 2025, pp. 392–405.
- [3] Z. Wang, L. Kang, Y. Qin, J. Ma, Z. Peng, L. Bai, and R. Zhang, “GauDP: Reinventing multi-agent collaboration through gaussian-image synergy in diffusion policies,” in *Advances in Neural Information Processing Systems*, 2025.
- [4] R. Doshi, T. Gao, A. Chen, C. Finn, and J. Bohg, “CHORUS: Decentralized multi-embodiment collaboration with one VLA policy,” *arXiv preprint arXiv:2606.12352*, 2026.
- [5] Z. Mandi, S. Jain, and S. Song, “RoCo: Dialectic multi-robot collaboration with large language models,” *arXiv preprint arXiv:2307.04738*, 2023.
- [6] Z. Li, R. Qiu, Y. Chen, G. Ren, and W. Zhi, “TRACE: Trajectory-routed causal memory for delayed-evidence visuomotor imitation,” *arXiv preprint arXiv:2606.14551*, 2026.
- [7] W. Zhi, T. Lai, L. Ott, and F. Ramos, “Diffeomorphic transforms for generalised imitation learning,” in *Learning for Dynamics and Control Conference*, vol. 168, 2022, pp. 508–519.
- [8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” in *Robotics: Science and Systems*, 2023.
- [9] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Robotics: Science and Systems*, 2023.
- [10] G. Yan, J. Zhu, Y. Deng, S. Yang, R.-Z. Qiu, X. Cheng, M. Memmel, R. Krishna, A. Goyal, X. Wang, and D. Fox, “ManiFlow: A general robot manipulation policy via consistency flow training,” in *Proceedings of the 9th Conference on Robot Learning*, vol. 305, 2025, pp. 2268–2293.
- [11] S. Liu *et al.*, “RDT-1B: A diffusion foundation model for bimanual manipulation,” *arXiv preprint arXiv:2410.07864*, 2024.
- [12] M. J. Kim, C. Finn, and P. Liang, “Fine-tuning vision-language-action models: Optimizing speed and success,” in *Robotics: Science and Systems*, 2025.
- [13] K. Black *et al.*, “ $\pi_{0.5}$: A vision-language-action model with open-world generalization,” *arXiv preprint arXiv:2504.16054*, 2025.
- [14] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [15] S. Iqbal and F. Sha, “Actor-attention-critic for multi-agent reinforcement learning,” in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 2961–2970.
- [16] S. Zhang, O. So, M. Black, Z. Serlin, and C. Fan, “Solving multi-agent safe optimal control with distributed epigraph form MARL,” in *Robotics: Science and Systems*, 2025.
- [17] Y. Chen, R. Qiu, Z. Li, Y. Zhou, G. Ren, and W. Zhi, “Robots that collaborate: Sequential asymmetric imitation for learning coupled robot policies,” *arXiv preprint arXiv:2606.16490*, 2026.
- [18] J. Zhao, M. Ba, and W. Zhi, “Tri-manual visuomotor imitation learning of robot policies,” *arXiv preprint arXiv:2607.25731*, 2026.
- [19] J. Zhao, J. Tang, M. Ba, and W. Zhi, “NestDex: Nested policy learning with copilot assisted teleoperation for dexterous manipulation,” *arXiv preprint arXiv:2608.13362*, 2026.
- [20] Y. Romano, E. Patterson, and E. J. Candès, “Conformalized quantile regression,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [21] A. N. Angelopoulos and S. Bates, “A gentle introduction to conformal prediction and distribution-free uncertainty quantification,” *arXiv preprint arXiv:2107.07511*, 2021.
- [22] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, “Implicit quantile networks for distributional reinforcement learning,” in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 1096–1105.
- [23] Y. Qin, L. Kang, X. Song, Z. Yin, X. Liu, X. Liu, R. Zhang, and L. Bai, “RoboFactory: Exploring embodied agent collaboration with compositional constraints,” in *IEEE/CVF International Conference on Computer Vision*, 2025.
- [24] L. Kang *et al.*, “Advances and innovations in the multi-agent robotic system (MARS) challenge,” *arXiv preprint arXiv:2601.18733*, 2026.
- [25] M. Caron, H. Touvron, I. Misra, *et al.*, “Emerging properties in self-supervised vision transformers,” *arXiv preprint arXiv:2104.14294*, 2021.
- [26] J. Long, D. Liu, W. Cai, I. Manchester, and W. Zhi, “Constraining streaming flow models for adapting learned robot trajectory distributions,” *arXiv preprint arXiv:2602.15567*, 2026.
- [27] H. Cheng, T. Zheng, T. Zhang, M. Johnson-Roberson, and W. Zhi,

- “DOSE3: Diffusion-based out-of-distribution detection on $SE(3)$ trajectories,” *arXiv preprint arXiv:2502.16725*, 2025.
- [28] Y. Zhou, R. Qiu, Y. Chen, J. Cui, and W. Zhi, “PATCH: Action-chunk-conditioned latent patch innovation monitoring for robot manipulation,” *arXiv preprint arXiv:2606.16690*, 2026.