Q-CoMove: Compact Circuit-Inspired Priors for Coordinated Motion in Multi-Component Embodied Systems

Ke Shi[1,2] **Fanqi Kong**[1] **Yuchen Wang**[1]
Zhipeng Liu[1] **Tingting Li**[1] **Ziming Zhao**[1*]
[1]Zhejiang University
[2]University of Science and Technology of China

Abstract

Mobile manipulation requires coordinated control of the base, arm, camera, and gripper as geometry and target visibility change. Existing approaches use learned whole-body policies or explicit geometric constraints to coordinate motion, leaving the design of compact interaction features for predictive control open to investigation. We propose Q-CoMove, a history-aware framework that combines circuit-inspired coordination features with physics-residual prediction. Component encoders and a shared recurrent network summarize observation histories. Goal-conditioned local rotations produce bounded component features and explicit pairwise products, which a classical decoder combines into a compact coordination representation. This representation complements direct component information to condition a joint policy and a hybrid dynamics predictor, followed by one-step candidate-action selection and geometric filtering. The design exposes cross-component interactions while retaining detailed inputs for control and permits fully classical evaluation without entangling operations. Across four MuJoCo tasks, Q-CoMove achieves an equal-task average success rate of 80.58%, improving on the strongest baseline by 3.05 percentage points under the same training and control pipeline.

1 Introduction

Mobile manipulation couples base reachability, arm clearance, camera visibility, and gripper alignment, especially with moving targets or obstacles. Embodied systems support language grounding [17], robotic policies [3], multimodal reasoning [11], and vision-language-action transfer [38]. Their observations must support coordinated responses to changing geometry and limited visibility.

Sequence models [35], graph models [32], and predictive control [27] support temporal reasoning, interactions, and action selection. We investigate a compact, bounded coordination representation that links component histories to predictive control while retaining detailed component information.

Q-CoMove combines history encoders for base, arm, camera, gripper, and environment observations; goal-conditioned rotation/product features; and physics-residual prediction. Detailed component features and the coordination latent jointly condition prediction and policy. A shared one-step controller scores joint-action candidates before geometric filtering (Figure 1).

Five simulated single-qubit states and two local rotation blocks use 35 gates to produce 15 local expectations and ten pairwise products. A classical decoder fuses them. This separable map admits direct classical evaluation: it supplies a circuit-inspired prior within an entirely classical pipeline.

*Corresponding author. Contact: zhaoziming@zju.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

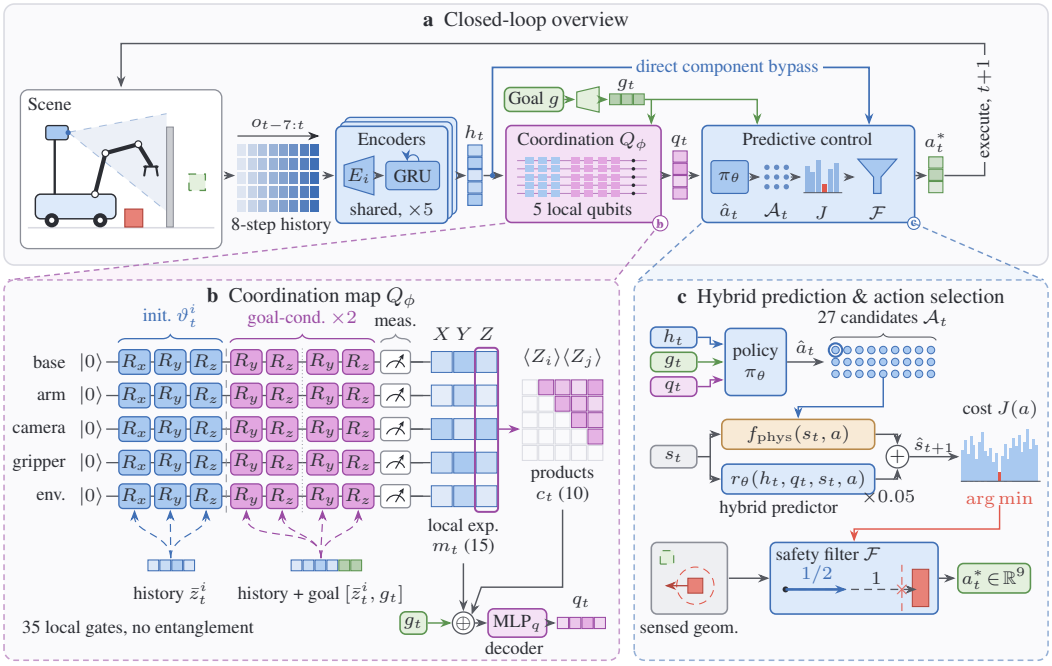


Figure 1: Q-CoMove architecture.

Our contributions are:

- We develop a history-aware framework that integrates rotation and product coordination features, physics-residual prediction, and joint action generation.
- We compare seven coordination representations across four simulated tasks, achieving the highest mean on each task and 80.58% equal-task success, and evaluate structural choices and data budgets.
- We conduct ablations and controlled perturbation experiments to examine the contributions of temporal context, coordination features, and dynamics prediction.

2 Related Work

Embodied Policies and Temporal Modeling. Visuomotor learning supports grasping [22] and manipulation [20]. SayCan grounds language in skills [17]; RT-1 learns policies from demonstrations [3]. PaLM-E integrates multimodal inputs [11], RT-2 transfers web knowledge to control [38], and Octo adapts across observation and action spaces [29]. AC-DiT conditions whole-body actions on base motion [6]; visibility-aware control coordinates base, arm, and head through line-of-sight constraints [16]. Decision Transformer [5] and Trajectory Transformer [18] model sequences; Diffusion Policy combines denoising and receding-horizon execution [7].

Relational Representations. Interaction networks model physical relations [1]. Graph physics engines model interactions [32], and graph networks formalize relational inductive biases [2]. Attention combines tokens [35]; Tensor Fusion Network exposes cross-modal interactions through outer products [37], providing a classical precedent for multiplicative fusion. Q-CoMove uses bounded coordinates, selected pairwise products, and a component bypass.

Prediction and Model-Based Control. Model predictive control uses predicted transitions and constraints [27]. Perceptive MPC integrates visual obstacle avoidance [30]; whole-body MPC handles task-priority transitions [36]. PILCO supports policy search [10] and probabilistic ensembles improve data efficiency [9]. PlaNet learns latent dynamics for planning from images [14], while Dreamer learns behavior through latent imagination [13]. Our physical transition plus learned residual follows

discrepancy modeling [33]; residual reinforcement learning instead learns an action correction to a controller [19]. Simulator-informed latent-state modeling [12] differs from our state-space residual correction.

Circuit-Inspired Features. Quantum features [15] and encoding ensembles [24] motivate our map. Classical circuit-like learning [21] highlights the distinction between circuit structure and computational advantage [4]. Our separable rotations admit classical Bloch-vector evaluation. Ablations test this prior; barren-plateau analyses study trainability [28]. Reliability research covers noise-robust QuST [23] and quantum-program bug datasets and benchmarking [25]. Evaluation is noiseless.

3 Q-CoMove

3.1 Problem Formulation and History Encoding

We consider a platform with component set $\mathcal{C} = \{\mathrm{base}, \mathrm{arm}, \mathrm{cam}, \mathrm{grip}, \mathrm{env}\}$. The environment entry is an observation block, not an actuator. At time t, the controller receives $o_t = [o_t^i]_{i\in\mathcal{C}}$ and task goal g. The robot's configuration and velocity state is $s_t \in \mathbb{R}^{22}$, the target of dynamics prediction. A joint command $a_t \in \mathbb{R}^9$ controls the base, arm, camera, and gripper.

Observations consist of proprioception, camera orientation, target visibility, visible target-relative position and horizontal orientation, and idealized obstacle positions, velocities, and sizes. The evaluation uses state-based sensing. Target features are zeroed under occlusion or outside the camera field of view. Hidden target states are used to generate observations and evaluation metrics, but are not supplied directly to the learned policy. A length-eight history $o_{t-7:t}$ provides temporal context.

Let $\bar{z}_t^i$ be the history encoding for component i and let $h_t = [\bar{z}_t^1, \ldots, \bar{z}_t^5]$. A goal-conditioned map produces a coordination representation $q_t = Q_\phi(h_t, g)$. The state predictor combines a coarse physical transition with a learned residual [33]:

$$\hat{s}_{t+1} = f_{\mathrm{phys}}(s_t, a_t) + 0.05\, r_\theta(h_t, q_t, s_t, a_t). \tag{1}$$

The coarse model approximates kinematics and actuator lag; the residual learns discrepancies from simulator transitions.

The policy proposes $\hat{a}_t = \pi_\theta(h_t, g_t, q_t)$, where g_t is the encoded goal. A shared one-step controller scores candidate commands using the predictor, then clips actuator limits and backtracks against sensed obstacle geometry. The executed command is $a_t^* = \mathcal{F}(\hat{a}_t; h_t, q_t, s_t, o_t, g)$, where $\mathcal{F}$ combines selection and filtering.

Framework Overview and History Encoding. Q-CoMove uses five simulated single-qubit states with $B = 2$ local blocks without entangling operations, evaluated by differentiable, noiseless PyTorch state vectors on classical hardware.

Figure 1(a) summarizes closed-loop control and the direct/coordination pathways; (b,c) detail coordination and predictive control. The component bypass feeds both policy and residual; repeated tokens denote shared inputs.

Observations use field-specific representations and scaling rather than a common mean–standard-deviation standardization. Each component is encoded as $z_t^i = E_i(o_t^i) \in \mathbb{R}^{64}$ by a $24 \to 64 \to 64$ GELU MLP. A 64-unit GRU [8] processes each component sequence independently to obtain $\bar{z}_t^i$; its weights are shared across components and $L = 8$. A separate $6 \to 32 \to 32$ goal MLP produces $g_t \in \mathbb{R}^{32}$. All learned alternatives share these encoders.

3.2 Rotation and Product Coordination Features

We associate one simulated qubit with each of the five components. The initial angles are $\vartheta_t^i = \pi \tanh(W_i \bar{z}_t^i + b_i) \in \mathbb{R}^3$, preparing

$$|\psi_t^{(0)}\rangle = \bigotimes_{i=1}^{5} \left[R_z(\vartheta_{t,3}^i) R_y(\vartheta_{t,2}^i) R_x(\vartheta_{t,1}^i)|0\rangle\right]. \tag{2}$$

Each subsequent block applies local updates,

$$|\psi_t^{(b+1)}\rangle = U_{\text{loc}}^{(b)}|\psi_t^{(b)}\rangle, \quad U_{\text{loc}}^{(b)} = \bigotimes_i [R_z(\lambda_{t,b}^i)R_y(\kappa_{t,b}^i)], \tag{3}$$

for $b = 0, \ldots, B-1$, where $(\kappa_{t,b}^i, \lambda_{t,b}^i)$ come from a linear head on $[\bar{z}_t^i, g_t]$ followed by $\pi \tanh$. The updates preserve the local product structure. For each fixed input, the state remains $|\psi_t\rangle = \bigotimes_i |\psi_t^i\rangle$, with

$$G_{\text{local}}(N, B) = 3N + 2NB = 35 \quad (N = 5, B = 2). \tag{4}$$

The readout contains local expectations and pairwise products,

$$m_t^i = [\langle X_i\rangle, \langle Y_i\rangle, \langle Z_i\rangle], \qquad c_t^{ij} = \langle Z_i Z_j\rangle = \langle Z_i\rangle\langle Z_j\rangle. \tag{5}$$

This gives $3N + N(N-1)/2 = 25$ features. Pairwise products expose cross-component interactions to downstream classical networks. Separability also permits evaluation through five Bloch vectors.

The goal-conditioned decoder is

$$q_t = \text{MLP}_q([\{m_t^i\}, \{c_t^{ij}\}, g_t]) \in \mathbb{R}^{64}, \tag{6}$$

with dimensions $57 \to 64 \to 64$ and GELU. The shared latent conditions prediction and action generation.

3.3 Predictive Coordination and Learning

With $h_t \in \mathbb{R}^{320}$ and $s_t \in \mathbb{R}^{22}$, prediction follows Equation 1. The residual r_θ is a $415 \to 64 \to 22$ MLP. The joint policy is $\hat{a}_t = \tanh(\text{MLP}_\pi([h_t, g_t, q_t]))$, using $416 \to 64 \to 9$. The $\tanh$ output bounds normalized commands to $[-1, 1]$, which are then converted to actuator velocities.

A shared one-step controller uses the predictor to score 27 candidate commands: three base/arm scales $(1, 0.75, 0.5)$, crossed with three wrist alternatives and three gripper alternatives (policy, observed-error feedback, zero). The cost combines target position, alignment terms when applicable, policy deviation, and obstacle proximity. The planning horizon is one control step.

Actions pass through velocity/joint clipping and exact-geometry backtracking at scales $1, 0.5, 0.25, 0$, using an 18 mm margin and 0.1 s obstacle extrapolation. The filter checks candidate steps against actuator limits and nearby sensed geometry.

Training Objective and Optimization. Training combines supervisor imitation, short-horizon prediction, and auxiliary terms:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{imit}} + 0.2\mathcal{L}_{\text{pred}} + 0.1\mathcal{L}_{\text{geom}} + 0.002\mathcal{L}_{\text{task}} \\ + 0.05\mathcal{L}_{\text{vel}} + 0.02\mathcal{L}_{\text{smooth}} + 0.05\mathcal{L}_{\text{safety}}.\end{aligned} \tag{7}$$

The imitation term is action MSE against an observation-based supervisor at visited states. Prediction uses recorded executed actions and three-step state errors, with episode-boundary masks and weights proportional to 0.9^k at prediction step k. Geometric consistency compares predicted and observed next-state end-effector position and horizontal orientation. The remaining terms penalize visible-target error, excessive predicted velocity, disagreement with supervisor action changes, and sensed obstacle proximity, respectively.

For full-data experiments, all models use the objective in Equation 7 and are trained with AdamW [26] for 40 epochs (the reduced-data study uses matched optimizer updates; Section 4.3), with batch size 256, an initial learning rate of 3×10^{-4}, and weight decay 10^{-4}. We use cosine annealing of the learning rate and gradient clipping with a maximum norm of 1.0. Computation is float32 with TF32 disabled. Three-step prediction increases free-rollout probability linearly from zero to 0.5, keeping the encoded context fixed during each short rollout. The checkpoint minimizes validation action MSE plus 0.2 prediction MSE. Early stopping, AMP, and learning-rate warmup are not used.

4 Experiments

We train separate models for mobile reaching, tracking during navigation, pre-contact alignment, and dynamic-obstacle reaching in MuJoCo [34]. Representation comparisons vary only the coordination module; data, histories, encoders, prediction/policy architectures, loss, training budget, candidate controller, and filter are shared. All computation is classical and observations are states, not images.

Simulation and Learning Setup. We use MuJoCo, Python, and PyTorch [31] at 20 Hz: each 0.05 s interval has ten 0.005 s physics steps. Episodes last at most 240 control steps (12 s). Offline-supervised models act online in closed loop. Main comparisons and success ablations share 200 test scenarios per task and five seeds. Experiments used two AutoDL instances. Each instance was equipped with one NVIDIA GeForce RTX 5090 GPU with 32 GB of VRAM, 25 vCPUs from an Intel Xeon Platinum 8470Q processor, and 90 GB of system RAM.

Dataset Construction and Model Selection. Each task has 160 training and 32 validation trajectories. Training, validation, and test scenarios are generated independently and share no trajectories. An observation-based rule teacher collects trajectories; perturbed and random actions cover deviations and recovery. All methods/seeds share the fixed task dataset. Validation error selects checkpoints; 200 independent common test scenarios evaluate them.

Success Criteria and Episode Termination. Table 1 summarizes all tasks. Reaching, alignment, and dynamic tasks use strict thresholds and consecutive holds; violations reset the counter, and success terminates the episode. Alignment conditions hold simultaneously. Tracking requires current 3D end-effector–moving-target distance < 7 cm for 10 consecutive steps (0.5 s), with the hold completed by 6 s; violations reset its counter. Final success also requires the tail-window criteria, so arrival alone does not terminate tracking. The maximum duration is 12 s; any collision makes an episode unsuccessful.

Table 1: Task success criteria.

Task	Required conditions	Time requirement
Reaching	3D end-effector–target distance < 7 cm	0.5 s (10 steps)
Alignment	Position error < 7 cm, horizontal orientation error $< 12^\circ$, and total gripper-opening error < 8 mm	0.5 s (10 steps)
Dynamic	3D end-effector–target distance < 7 cm	0.25 s (5 steps)
Tracking	Distance < 7 cm for 0.5 s, completed by 6 s; tail position RMSE ≤ 9 cm; tail visibility $\geq 65\%$	Final 6 s

Collision Metric. Collision means MuJoCo robot–obstacle penetration deeper than 1 mm, including collision-enabled base, arm, and gripper geometries but excluding self-collision, ground, and target contact. All collision-free claims use this criterion.

4.1 Main Comparison

Comparators and Evaluation Protocol. We compare concatenation, pairwise relations, graph message passing, component self-attention, a structure-matched low-rank Bilinear adaptation, and Local-Product. Local-Product replaces the rotation map with five goal-conditioned MLP heads and the same pairwise-product readout; its construction and the other ablation variants are defined in Section 4.2.1.

The 200 common test scenarios per task are independent of training, validation, and structure selection. We average four tasks equally within each seed, then report the five-seed mean and sample SD (Table 2; bold indicates highest means). Repeated configurations reuse checkpoints/evaluation records; distinct ablations are separately trained. The complete model's reference success is 80.58%. Sample SD summarizes performance variation across the five training seeds.

Main Comparison. Q-CoMove achieves the highest mean success in all four tasks (Table 2), reaching 80.58% equal-task success compared with 77.53% for the strongest baseline, GNN: a gain of 3.05 percentage points (pp). The largest task-level margins over GNN occur in dynamic-obstacle reaching and pre-contact alignment, both at 5.70 pp. These results compare coordination representations under a shared controller across all four tasks.

Table 2: Success rates across four tasks.

Method	Reaching	Tracking	Alignment	Dynamic	Average
Concat-MLP	88.90 ± 3.58	69.90 ± 6.72	71.70 ± 17.62	72.00 ± 6.03	75.63 ± 3.35
Pairwise	81.60 ± 9.35	66.30 ± 11.77	75.00 ± 15.17	70.00 ± 4.51	73.23 ± 6.03
GNN	88.70 ± 7.13	74.50 ± 4.06	78.70 ± 11.13	68.20 ± 6.79	77.53 ± 3.83
Component attention	85.20 ± 6.12	66.70 ± 6.83	68.60 ± 12.82	64.50 ± 2.26	71.25 ± 4.66
Bilinear	85.10 ± 8.49	67.70 ± 7.60	79.70 ± 13.27	72.80 ± 3.82	76.33 ± 4.17
Local-Product	85.70 ± 8.37	71.30 ± 1.96	73.00 ± 11.47	66.20 ± 4.35	74.05 ± 3.75
Q-CoMove	$\mathbf{89.00 \pm 1.84}$	$\mathbf{75.00 \pm 3.18}$	$\mathbf{84.40 \pm 5.58}$	$\mathbf{73.90 \pm 8.84}$	$\mathbf{80.58 \pm 3.36}$

4.2 Ablation Study

Ablations examine observation history, coordination features, pairwise products, the component bypass, local parameterization, and physical and learned dynamics. Each variant is retrained per task using the same data and loss for 40 epochs with five seeds. Q-CoMove denotes the full model.

4.2.1 Variant Definitions

Representation Replacements. Q-CoMove uses goal-conditioned rotations to produce a three-coordinate unit Bloch vector for each component, followed by ten products between the third coordinates of component pairs. **Local-Product** replaces those rotations with five independent goal-conditioned MLP heads. Each head produces three bounded coordinates without a unit-norm constraint; the 15 coordinates and ten third-coordinate products retain the same 25-dimensional readout. **Sphere-Product** uses the same heads but normalizes each three-coordinate output to unit Euclidean norm before forming the products. It therefore preserves the unit-sphere constraint and product readout while replacing the rotation parameterization. The local feature maps have 3,015 parameters for either MLP control and 2,915 for Q-CoMove, excluding the shared decoder. The $57 \to 64 \to 64$ decoder adds 7,872 parameters in each case.

Feature and Information-Path Ablations. **No Products** masks the ten pairwise readout entries during training and evaluation while preserving the 25-dimensional interface. **No History** uses only the current observation in both policy and prediction contexts, retaining a one-step GRU. **No Coordination** removes state-dependent coordination features and retains direct component features and a goal-only decoder. The **Component Bypass** is the direct 320-dimensional path from the concatenated component histories to the policy and residual predictor. Disabling it zeros those inputs after the coordination representation is constructed, so the encoders still feed the coordination branch. Goal information, physical state, action, controller observations, and geometric filtering remain available. These controls change training and evaluation together.

Prediction Ablations. **Physics Only** removes the learned residual and uses the coarse physical transition alone. **Learned Delta** retains the residual architecture but replaces the physical transition with the identity state update. Q-CoMove combines the coarse transition and learned correction as in Equation 1. The physics-only model has 93,100 total parameters; learned delta and Q-CoMove each have 121,154.

4.2.2 Temporal Context and Coordination Structure

Figure 2(a) shows the main information pathways, (b) reports three-step prediction RMSE, and (c) examines local parameterization and product features. The labels Full, Hybrid, and Rotation with products denote Q-CoMove. Success error bars indicate one sample SD across five seed-level equal-task averages; panel (b) shows mean prediction RMSE. Panels (a,c) compare Q-CoMove with the history, coordination, and product ablations on 200 common scenarios per task. The success rates (mean $\pm$ sample SD) are $80.58 \pm 3.36\%$ for Q-CoMove, $77.90 \pm 4.57\%$ without products, $66.53 \pm 4.99\%$ without history, and $77.40 \pm 4.08\%$ without coordination. Removing these elements reduces mean success by 2.68, 14.05, and 3.18 percentage points, respectively. History provides the largest improvement among these interventions, supporting temporal encoding when target observations are intermittent. Product and coordination ablations show additional mean gains alongside detailed inputs.

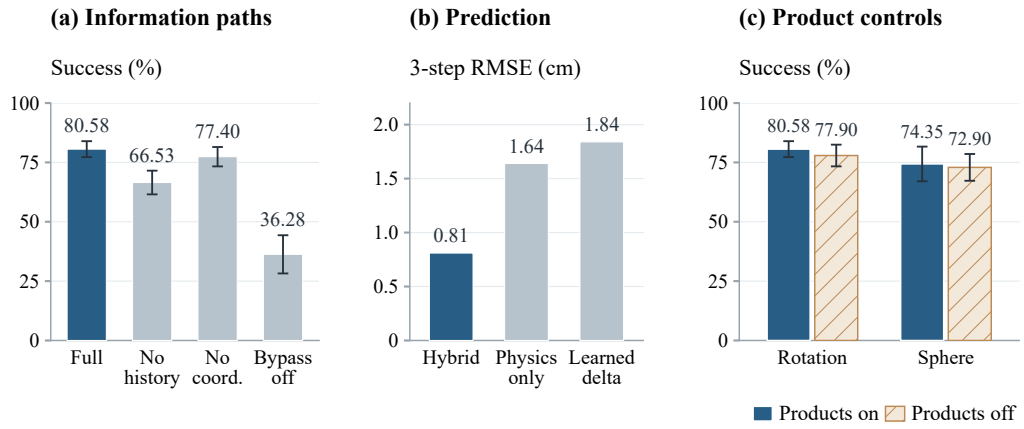


Figure 2: Q-CoMove ablations of information pathways, prediction, and coordination features.

The product/bypass study evaluates four configurations: products and bypass both enabled (A, Q-CoMove); products disabled and bypass enabled (B); products enabled and bypass disabled (C); and both disabled (D). Their success rates (mean $\pm$ sample SD) are $80.58 \pm 3.36\%$, $77.90 \pm 4.57\%$, $36.28 \pm 8.04\%$, and $32.98 \pm 5.78\%$, respectively. All settings share data, training budgets, and 200 scenarios per task. Figure 2(a) includes the complete and bypass-disabled configurations; panel (c) isolates product effects with the bypass retained.

Products improve mean success by 2.68 percentage points with the bypass and 3.30 points without it. Keeping the bypass improves success by 44.30 points with products and 44.92 points without products. The direct path supplies detailed component information for precise actions and prediction, while the compact coordination representation adds structured interaction features. The joint bypass ablation evaluates these detailed inputs to the policy and residual predictor as part of the complete controller; combining both paths achieves the highest mean success.

Representation replacements retain histories, goal access, bypass, policy, predictor, and execution. On the main test scenarios, Q-CoMove exceeds Local-Product and Sphere-Product by 6.53 and 6.23 percentage points (Table 3), supporting rotation parameterization beyond bounded coordinates and products alone.

Rotation and Product Controls. Figure 2(c) and Table 4 compare Rotation (Q-CoMove's local map) with Sphere (normalized MLP outputs), with products masked when disabled. Products add 2.68 and 1.45 percentage points, respectively; Q-CoMove has the highest mean. Values are equal-task mean $\pm$ sample SD over five seeds.

4.2.3 Physical Prediction and Learned Correction

We assess predictive accuracy using a set of 50 held-out scenarios per task with shared supervisor/perturbed rollouts. All predictors receive identical recorded action sequences and fixed initial history context. Up to 4,096 uniformly spaced starting indices are used per task, excluding rollouts crossing episode boundaries. Figure 2(b) and Table 3 report mean three-step end-effector RMSE over task/seed combinations, rather than a pooled RMSE. Success entries report mean $\pm$ sample SD over five random seeds. These measurements assess the Q-CoMove configuration used for control.

Table 3: Representation and prediction ablations.

Variant	Success (%)	RMSE (cm)
Q-CoMove	$\mathbf{80.58 \pm 3.36}$	$\mathbf{0.81}$
Local-Product	74.05 ± 3.75	0.82
Sphere-Product	74.35 ± 7.30	0.82
Physics only	79.78 ± 4.14	1.64
Learned delta	78.65 ± 4.23	1.84

Table 4: Rotation and product controls.

Configuration	Success (%)
Q-CoMove	$\mathbf{80.58 \pm 3.36}$
Rotation without products	77.90 ± 4.57
Sphere-Product	74.35 ± 7.30
Sphere without products	72.90 ± 5.64

Removing the residual or physical transition reduces success by 0.80 or 1.93 percentage points. Hybrid RMSE is 0.81 cm versus 1.64 cm (physics) and 1.84 cm (learned delta): reductions of 50.6% and 56.0%. Hybrid dynamics improve prediction and mean success under the shared controller.

4.3 Training-Data Efficiency

Each method trains separately per task. Shared nested subsets contain 40/80/120 of 160 training trajectories (25%/50%/75%), selected in fixed random order; validation is unchanged. All fractions share the full-data 40-epoch update budget, cycling reshuffled smaller subsets. Batch size, update-indexed learning-rate schedule, loss, and validation evaluation count are fixed. The main comparison's 200 test scenarios/task, controller, candidate-action scoring, geometric filtering, and success criteria remain unchanged. The 100% column reuses references from Tables 2 and 3. We first equally average four-task success rates per seed, then compute the mean and sample SD across five seeds.

Table 5: Training-data efficiency: success (mean $\pm$ sample SD, %) with a fixed update budget.

Method	25%	50%	75%	100%
Q-CoMove	$\mathbf{65.45 \pm 5.59}$	$\mathbf{71.83 \pm 4.67}$	$\mathbf{78.13 \pm 3.71}$	$\mathbf{80.58 \pm 3.36}$
Sphere-Product	59.38 ± 3.95	64.63 ± 3.91	71.23 ± 3.50	74.35 ± 7.30
Local-Product	57.83 ± 2.98	65.70 ± 3.89	71.98 ± 3.29	74.05 ± 3.75

4.4 Depth and Connection Structure

Table 6 compares nine depth/connectivity designs on 200 common structure-selection scenarios per task, separate from main testing, over five seeds. Two-block local has the highest equal-task mean, 78.80%, versus 77.58% and 75.53% at one/three blocks. Sparse blocks use 6, 6, and 5 component-pair connections; dense blocks use all ten pairs; local blocks have none. Parentheses count gates, not measured runtime. The selected design uses 35 gates.

Table 6: Depth and connectivity selection.

Blocks	Sparse entangling	Dense entangling	Local, no entangling gates
1	75.55 ± 4.72 (37)	76.73 ± 5.17 (45)	77.58 ± 3.64 (25)
2	75.40 ± 2.59 (59)	73.95 ± 4.22 (75)	$\mathbf{78.80 \pm 4.28}$ (35)
3	76.30 ± 2.25 (79)	75.68 ± 5.35 (105)	75.53 ± 6.62 (45)

4.5 Robustness Analysis

Frozen checkpoints from all seven methods are evaluated over all five training seeds without retraining or changing success criteria. Figure 3 reports mean success under occlusion (200 common scenarios/condition), frequency, and noise (100 each). Methods/conditions share scenarios within each sweep; sweeps use different sets, so zero-perturbation means are not interchangeable. Each curve summarizes success over all five training seeds.

Target-Visibility Occlusion. On 200 shared tracking scenarios, each episode has two scripted target-observation masks, each lasting 0, 0.3, 0.6, or 1.0 s. Scene geometry, target motion, and mask starts are fixed. Four durations and five seeds yield 4,000 episodes per method. Masks suppress state observations, not RGB images; zero duration does not remove geometric visibility checks.

Success generally declines as observations are hidden for longer. Q-CoMove has the highest observed mean at every scripted duration, retaining 69.6% success at 1.0 s versus 68.1% for GNN and 66.6% for Local-Product. Together with the history ablation, this result highlights temporal context as a useful part of the complete controller during intermittent visibility.

Target-Motion Frequency. On 100 new shared tracking scenarios, target-motion frequency is multiplied by 1, 1.25, or 1.5, holding the other sampled scene attributes fixed. No model is retrained. Q-CoMove achieves 68.2%, 50.6%, and 28.2%, highest among the seven methods at each frequency. All methods deteriorate, motivating longer-horizon prediction and action selection.

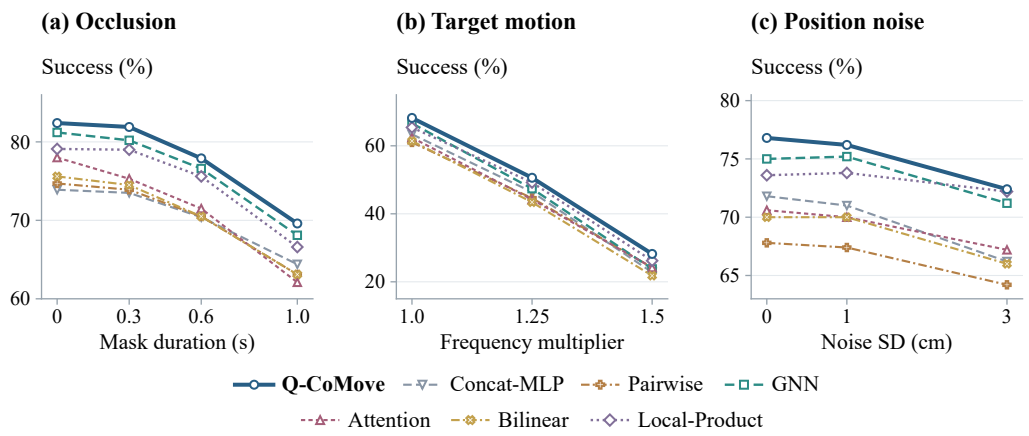


Figure 3: Robustness across seven methods.

Target-Position Observation Noise. On a separate set of 100 shared tracking scenarios, independent per-axis Gaussian noise with standard deviation 0, 1, or 3 cm is added only to visible target-position observations. Standardized draws are indexed by scene and control step and shared across methods and levels. Invisible target features remain masked; simulation truth and success evaluation are unchanged. At 1 cm, mean success changes little; at 3 cm, Q-CoMove retains 72.4%, versus GNN's 71.2% and Local-Product's 72.2%, the highest observed mean.

4.6 Paired Qualitative Tracking Cases

Figure 4 compares three scenarios replayed with fixed Q-CoMove and GNN checkpoints and the same controller. A category-based rule was specified before inspecting candidate outcomes: in main-test order, select the first shared success, the first differing outcome regardless of the successful method, and the first remaining Q-CoMove failure. Panels (a)–(c) show these categories. Q and G denote Q-CoMove and GNN. The rows show projected paths, 3D position error, and normalized command activity for the base, arm, camera, and gripper. Shading and dotted lines mark observation masks; hidden target truth is unavailable to the policy.

Using the tracking criteria in Table 1, in panel (a), both methods succeed, with tail RMSEs of 8.00 cm for Q-CoMove and 8.52 cm for GNN. In panel (b), Q-CoMove succeeds and GNN fails, with corresponding errors of 7.76 and 9.22 cm. Panel (c) is a challenging case for both methods, with errors of 10.30 and 11.76 cm. All six replays are collision-free. Q-CoMove has lower tail RMSE in every selected case, while the error traces and command maps show how coordinated actions evolve through interrupted visibility.

5 Discussion and Limitations

Coordination and Prediction. History and direct component information are important in ablations, rotation/product features lead the tested controls, and hybrid dynamics reduce three-step error. Together they support a compact coordination prior that supplements detailed inputs in predictive control.

Interpretation and Scope. The separable map is classically evaluable; five-seed comparisons (sample SD where available) do not establish quantum advantage. Tests cover a fixed five-component simulator, narrow initial distributions, and idealized state/obstacle sensing. Real-robot transfer, learned perception, and broader shifts remain untested. Pairwise readout scales as $O(N^2)$; runtime advantages are unmeasured.

Future Directions and Broader Impacts. Faster-motion failures motivate longer horizons; learned perception and varied environments would test assistive/service deployment. Independent safety

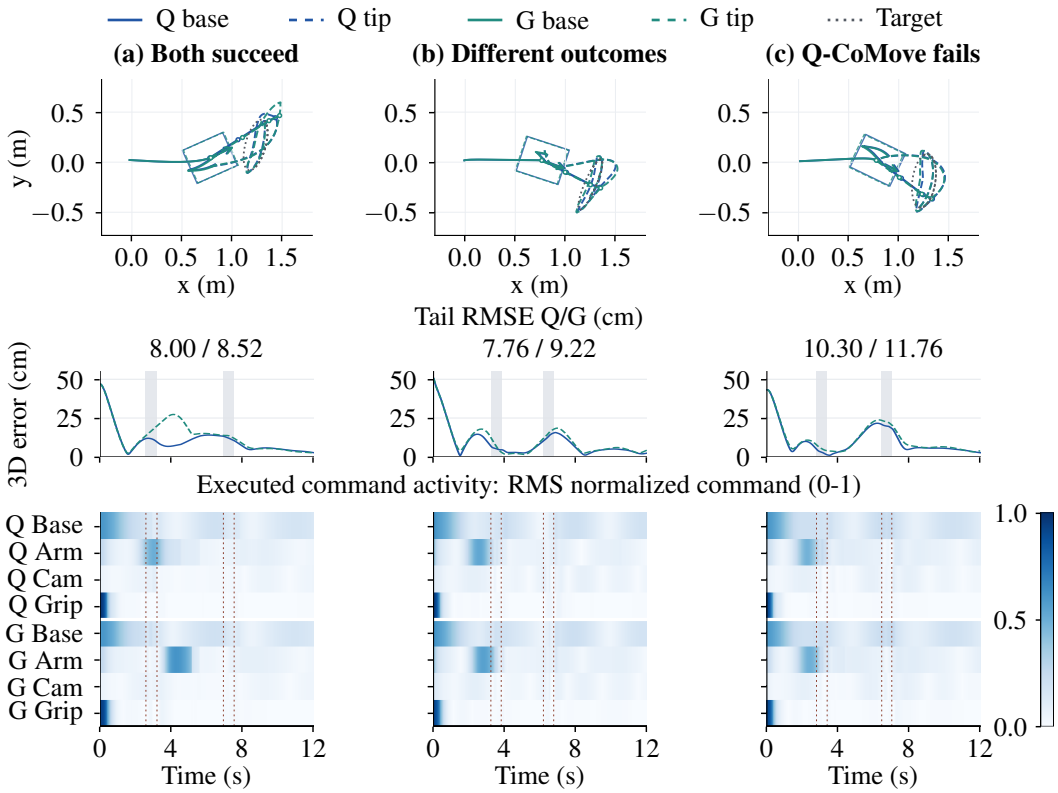


Figure 4: Paired tracking cases for Q-CoMove and GNN.

validation and sensor-data privacy protections are needed: prediction/filtering errors can cause collisions, and local filtering offers no global collision-free guarantee.

6 Conclusion

Q-CoMove combines history-aware rotation/product features, physics-residual prediction, and joint action generation for multi-component embodied control. It leads all four simulated tasks in the seven-method comparison, with 80.58% equal-task success. Ablation experiments support the complementary contributions of temporal context, direct component information, and coordination features, while prediction controls demonstrate the accuracy of the hybrid dynamics model.

Acknowledgments and Disclosure of Funding

This work was supported by the Ningbo Yongjiang Talent Programme (2025A-387-G), the Zhejiang Provincial Natural Science Foundation of China (Grant No. LQN26F020008), the CCF-Kuaishou Large Model Explorer Fund (No. CCF-KuaiShou 2025001), the CAAI-Ant Group Research Fund (2025CAAI-ANT-15), the Science and Technology Innovation Project of Ningbo Global Innovation Center, Zhejiang University (Grant No. S02J2604), and the China Postdoctoral Science Foundation under Grant No. GZB20260393.

References

[1] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4502–4510, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/3147da8ab4a0437c15ef51a5cc7f2dc4-Abstract.html.

[2] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://doi.org/10.15607/RSS.2023.XIX.025.

[4] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.*, 2(9):567–576, 2022. doi: 10.1038/S43588-022-00311-3. URL https://doi.org/10.1038/s43588-022-00311-3.

[5] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

[6] Sixiang Chen, Jiaming Liu, Siyuan Qian, Han Jiang, Zhuoyang Liu, Chenyang Gu, Xiaoqi Li, Chengkai Hou, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, and Shanghang Zhang. AC-DiT: Adaptive coordination diffusion transformer for mobile manipulation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5c6c63b7ec141825ff1327563e699f76-Abstract-Conference.html.

[7] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

[8] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

[9] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

[10] Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 465–472. Omnipress, 2011. URL https://icml.cc/2011/papers/323_icmlpaper.pdf.

[11] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 8469–8488. PMLR, 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

[12] Katharina Ensinger, Sebastian Ziesche, and Sebastian Trimpe. Learning hybrid dynamics models with simulator-informed latent states. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(11), pages 11892–11900, 2024. doi: 10.1609/aaai.v38i11.29075. URL https://doi.org/10.1609/aaai.v38i11.29075.

[13] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *CoRR*, abs/1912.01603, 2019. URL http://arxiv.org/abs/1912.01603.

[14] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

[15] Vojtech Havlícek, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nat.*, 567(7747):209–212, 2019. doi: 10.1038/S41586-019-0980-2. URL https://doi.org/10.1038/s41586-019-0980-2.

[16] Kerry He, Rhys Newbury, Tin Tran, Jesse Haviland, Ben Burgess-Limerick, Dana Kulić, Peter Corke, and Akansel Cosgun. Visibility maximization controller for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(3):8479–8486, 2022. doi: 10.1109/LRA.2022.3188430. URL https://doi.org/10.1109/LRA.2022.3188430.

[17] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, Proceedings of Machine Learning Research, pages 287–318. PMLR, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

[18] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1273–1286, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

[19] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 6023–6029. IEEE, 2019. doi: 10.1109/ICRA.2019.8794127. URL https://doi.org/10.1109/ICRA.2019.8794127.

[20] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *CoRR*, abs/1806.10293, 2018. URL http://arxiv.org/abs/1806.10293.

[21] Naoko Koide-Majima and Kei Majima. Quantum circuit-like learning: A fast and scalable classical machine-learning algorithm with similar performance to quantum circuit learning. *CoRR*, abs/2003.10667, 2020. URL https://arxiv.org/abs/2003.10667.

[22] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. In Dana Kulic, Yoshihiko Nakamura, Oussama Khatib, and Gentiane Venture, editors, *International Symposium on Experimental Robotics, ISER 2016, Tokyo, Japan, October 3-6, 2016*, Springer Proceedings in Advanced Robotics, pages 173–184. Springer, 2017. doi: 10.1007/978-3-319-50115-4_16. URL https://doi.org/10.1007/978-3-319-50115-4_16.

[23] Tingting Li, Liqiang Lu, Ziming Zhao, Ziqi Tan, Siwei Tan, and Jianwei Yin. QuST: Optimizing quantum neural network against spatial and temporal noise biases. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(4):1434–1447, 2025. doi: 10.1109/TCAD.2024.3471949. URL https://doi.org/10.1109/TCAD.2024.3471949.

[24] Tingting Li, Ziming Zhao, Liqiang Lu, and Jianwei Yin. QuFrame: A novel encoding ensemble framework for quantum neural networks. In *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pages 583–590. IEEE, 2025. doi: 10.1109/QCNC64685.2025.00097. URL https://doi.org/10.1109/QCNC64685.2025.00097.

[25] Tingting Li, Jiongchi Yu, Zhikang Fan, Junqing Wang, Yao Zhai, Shenglong Zhang, Yu Peng, Yuhan Chen, Junyu Chen, Fanqi Kong, Zhaoxuan Li, Xiaofei Yue, Ziming Zhao, and Jianwei Yin. Towards quantum program bug datasets and benchmarking. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 9338–9348. ACM, 2026. doi: 10.1145/3770855.3817560. URL https://doi.org/10.1145/3770855.3817560.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[27] David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Autom.*, 36(6):789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9. URL https://doi.org/10.1016/S0005-1098(99)00214-9.

[28] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *CoRR*, abs/1803.11173, 2018. URL http://arxiv.org/abs/1803.11173.

[29] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. URL https://arxiv.org/abs/2405.12213.

[30] Johannes Pankert and Marco Hutter. Perceptive model predictive control for continuous mobile manipulation. *IEEE Robotics and Automation Letters*, 5(4):6177–6184, 2020. doi: 10.1109/LRA.2020.3010721. URL https://doi.org/10.1109/LRA.2020.3010721.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[32] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin A. Riedmiller, Raia Hadsell, and Peter W. Battaglia. Graph networks as learnable physics engines for inference and control. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 4470–4479. PMLR, 2018. URL http://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

[33] Matteo Saveriano, Yuchao Yin, Pietro Falco, and Dongheui Lee. Data-efficient control policy search using residual dynamics learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4709–4715. IEEE, 2017. doi: 10.1109/IROS.2017.8206343. URL https://doi.org/10.1109/IROS.2017.8206343.

[34] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[36] Yushi Wang, Ruoqu Chen, and Mingguo Zhao. Whole-body model predictive control for mobile manipulation with task priority transition. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://wbmpc.github.io/.

[37] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114. Association for Computational Linguistics, 2017. doi: 10.18653/v1/D17-1115. URL https://aclanthology.org/D17-1115/.

[38] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, Proceedings of Machine Learning Research, pages 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the paper's core claims about structured coordination priors, physics-residual dynamics, and coordinated control, and the supporting empirical evidence is reported in Sections 1, 4, and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses simulation and sensing assumptions, five-seed empirical evaluation, classical computational interpretation, $O(N^2)$ readout scaling, and the locality of geometric filtering.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theorem-proof results; it introduces a modeling framework, optimization objective, and empirical evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3.1–4 describe the architecture, training objective, baselines, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open Access to Data and Code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper package does not include a public release of code or data, or reproduction scripts. The experimental setup is summarized in Section 4.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3 specifies the loss weights, optimizer, batch size, training duration, rollout schedule, and checkpoint selection; Section 4 describes the task setup and seed averaging.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 4 reports means and sample standard deviations over five random seeds for the main comparison, reduced-data study, and structure-selection results; prediction controls report mean task/seed RMSE. Success-ablation panels report five-seed means with sample-SD error bars; the RMSE panel has no error bars because uncertainty is unreported. Robustness curves are descriptive means without error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments used two AutoDL instances. Each instance was equipped with one NVIDIA GeForce RTX 5090 GPU with 32 GB of VRAM, 25 vCPUs from an Intel Xeon Platinum 8470Q processor, and 90 GB of system RAM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Section 5 discusses deployment-related safety, predictability, and sensor-data handling considerations, and the research does not involve practices that conflict with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 5 discusses deployment potential together with risks related to safety, predictability, and sensor-data handling in human-shared environments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped dataset, or broadly deployable foundation model artifact that requires special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for Existing Assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper credits existing experimental resources through citations, but does not explicitly document all applicable licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper package does not release a new dataset or a documented software artifact.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject studies that would require IRB review or equivalent approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM Usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core method, training procedure, or evaluation pipeline described in the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.