UNINAVMANIP: GRAPH-CONSTRAINED BELIEF-SPACE CONTROL FOR UNIFIED MOBILE NAVIGATION AND MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon mobile manipulation must coordinate task progress, visual memory, and heterogeneous base–arm control under partial observability. Existing visuomotor policies often model motion, task structure, or visual history separately, leaving mode transitions and verified progress weakly coupled. We present UNINAVMANIP, a vision-language-action controller that combines a language-grounded TaskGraph, typed head–wrist memory, mode-routed hybrid actions, and receding-horizon candidate selection. The graph restricts proposals using verified predecessors and confirmed preconditions; the controller executes short prefixes and updates the graph only after fresh-observation verification. In simulation-only physics-based Isaac Sim evaluations on 240 held-out episodes from three scenes, UNINAVMANIP reaches 61.7% complete-task success, compared with 57.1% for the strongest adapted public baseline and 49.6% for a matched internal hierarchical baseline. It also records the lowest collision rate at 0.15 per episode.

1 INTRODUCTION

Long-horizon mobile manipulation alternates between scene-scale navigation and contact-rich interaction. Base motion changes viewpoint, local alignment changes arm reachability, and manipulation changes the facts required by later actions. Task state, visual evidence, and control mode therefore evolve together under partial observability.

Vision-language-action models provide visuomotor priors across tasks and embodiments (Brohan et al., 2023a; Driess et al., 2023; Brohan et al., 2023b; Team et al., 2024; Kim et al., 2024; Black et al., 2025a;b). At each interaction boundary, a mobile-manipulation controller must identify a feasible subgoal and mode, retrieve the relevant head- and wrist-camera evidence, and choose an action that advances the task without excessive execution risk. In the tool-transfer example of Figure 1, departing the pickup station before verifying the grasp breaks the prerequisite for delivery. Correct local motion alone is therefore insufficient to establish task progress.

These requirements differ across modes: NAVIGATE needs scene-scale head evidence and sparse waypoints, ALIGN synchronizes base and arm commands, and MANIPULATE depends on local wrist evidence and contact-sensitive control. A monolithic action sequence can lose prerequisite structure, while a symbolic plan without visual memory and physical checks cannot recover from an unverified grasp, placement, or reachability condition.

UNINAVMANIP connects task feasibility, visual evidence, and action selection at each execution boundary. A TaskGraph projects subgoal–mode proposals onto the set allowed by verified predecessors and preconditions. Typed dual-view memory supplies scene-scale head evidence or local wrist evidence according to the active mode. Mode-specific heads generate two-level navigation controls, synchronized alignment, or manipulation chunks. Before execution, an ensemble model rolls out each feasible candidate for H macro transitions; the controller executes a short prefix of the highest-utility candidate and replans after verification. Under the shared embodiment, observation, and action protocol, this combination raises complete-task success by 4.6 points over the strongest adapted public baseline and by 12.1 points over the matched internal hierarchical baseline.

Our contributions are:

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

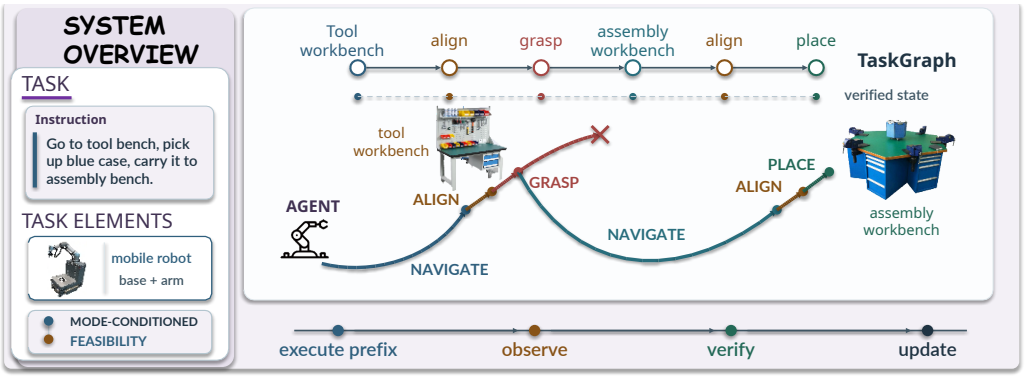


Figure 1: A blue tool case is transferred from the tool workbench to the assembly workbench. Blue/cyan denote navigation; amber, coral, and teal denote alignment, grasp, and place. The dashed branch leaves the tool workbench before grasp verification.

- **An execution-conditioned task interface.** A language-grounded TaskGraph projects subgoal–mode proposals onto verified predecessor, precondition, and mode constraints, updates unfinished nodes from verified effects, and formalizes graph-support preservation.
- **Typed multi-view belief and mode-routed control.** A bounded head–wrist memory indexed by camera role, time, and spatial identity provides mode-conditioned inputs to hierarchical navigation, synchronized base–arm alignment, and flow-matched manipulation.
- **Risk-aware receding-horizon selection.** An ensemble world–action model scores graph-feasible candidates by progress, collision risk, disagreement, and prior, executes a short prefix, and replans after verification; a conditional selection-error bound separates proposal coverage error from four score-error terms.

2 Related Work

Vision-language-action policies for mobile manipulation. Vision-and-language navigation studies instruction following in unseen environments (Anderson et al., 2018; Krantz et al., 2020), while embodied benchmarks add object interaction and long-horizon household tasks (Shridhar et al., 2020; Puig et al., 2018; Szot et al., 2021; Li et al., 2024). Generalist policies learn from action chunks, cross-embodiment data, and flow matching (Brohan et al., 2023a; Ebert et al., 2022; Open X-Embodiment Collaboration, 2024; Team et al., 2024; Kim et al., 2024; Black et al., 2025a;b; Lipman et al., 2023); related systems study diffusion control, low-cost bimanual manipulation, whole-body execution, and open-vocabulary interaction (Chi et al., 2023; Zhao et al., 2023; Fu et al., 2024; Yen-Chen et al., 2023).

Task planning and neuro-symbolic constraints. POMDPs formalize decision making under partial observation (Kaelbling et al., 1998), while task planning and temporal abstraction organize behavior across symbolic decisions and reusable skills (Sutton et al., 1999); integrated task-and-motion planning connects symbolic constraints to geometric feasibility (Kaelbling & Lozano-Pérez, 2013; Garrett et al., 2021). Recent embodied systems use high-level skills, predicates, and language plans to structure long-horizon behavior (Ahn et al., 2022; Huang et al., 2022). Within this line of work, UniNavManip exposes the graph frontier, preconditions, effects, and admissible modes as an online feasible-set interface and updates them from observations.

World models, risk, and memory. Latent world models predict action-conditioned futures (Hafner et al., 2019; 2025); probabilistic model-based control uses matched rollouts to compare candidate futures (Chua et al., 2018). Safe learning and deep ensembles provide tools for risk and uncertainty estimation (Brunke et al., 2022; Lakshminarayanan et al., 2017). Recurrent states, keyframes, maps, and multimodal memories retain complementary history (Fang et al., 2019; Pashevich et al.,

2021). UNINAVMANIP applies these ideas to graph-feasible candidate selection and mode-specific head–wrist evidence.

3 PROBLEM FORMULATION

At decision boundary t, the controller receives instruction I, head/wrist images and proprioception $o_t = (o_t^h, o_t^w, p_t)$, typed history H_t, mutable TaskGraph $G_t = (\mathcal{V}_t, \mathcal{E}_t, C_t, Q_t, D_t, \kappa_t)$, and kinematic feasibility estimate q_t. Its fields C_t, Q_t, D_t, and κ_t store confirmed facts, tentative facts, completed nodes, and failure counts, respectively. The learned belief representation is

$$b_t = F_\theta(I, o_t^h, o_t^w, p_t, \bar{H}_t, G_t, q_t) \in \mathbb{R}^d. \tag{1}$$

Here *belief* denotes a predictive state, not a calibrated Bayesian posterior: for supervised future variables $Y_{t:t+H}$ (modes, subgoals, controls, effects, progress, and risk), training seeks $p(Y_{t:t+H} \mid I, o_{\le t}, H_t, G_t, q_t) \approx p_\theta(Y_{t:t+H} \mid b_t)$, while future-state and consistency losses align imagined and observation-conditioned beliefs. Accordingly, b_t is trained as the predictive state for these control targets. The controller chooses $h_t = (g_t, m_t, c_t)$: a subgoal, a mode in {NAVIGATE, MANIPULATE, ALIGN, OBSERVE, RECOVER, DONE}, and a mode-specific control block. A horizon-H candidate is $\eta_k = \{(g_{t+j,k}, m_{t+j,k}, c_{t+j,k})\}_{j=0}^{H-1}$, whose mode/subgoal sequence follows the admissible transition relation $R(G_t)$.

After a prefix, the verifier returns $e_{t+1} = (e_{t+1}^+, e_{t+1}^-, e_{t+1}^?, e_{t+1}^{\text{done}})$, where e_{t+1}^- contains verified facts to delete, and applies

$$\begin{aligned} C_{t+1} &= (C_t \setminus e_{t+1}^-) \cup e_{t+1}^+, \\ D_{t+1} &= D_t \cup e_{t+1}^{\text{done}}, \\ Q_{t+1} &= (Q_t \cup e_{t+1}^?) \setminus C_{t+1}, \\ G_{t+1} &= \mathcal{U}(G_t, o_{t+1}, e_{t+1}). \end{aligned} \tag{2}$$

Revisions preserve graph acyclicity, completed nodes, and their ordering. For ordinary task actions (special-mode rules appear in Section 4.6), let $\text{Avail}(G_t)$ denote unfinished nodes whose predecessors are complete; then

$$\mathcal{F}_t^{gm} = \{(g, m) : g \in \text{Avail}(G_t),\ \text{pre}(g) \subseteq C_t,\ m \in \text{Modes}(g)\}. \tag{3}$$

Mode-specific decoding and physical checks produce the control-feasible set $\mathcal{F}_t$. Candidate evaluation uses its length-H extension

$$\mathcal{H}_t^H = \left\{ \eta = (h_{t+j})_{j=0}^{H-1} : \begin{array}{l} h_{t+j} = (g_{t+j}, m_{t+j}, c_{t+j}),\ \hat{G}_t^\eta = G_t,\ \hat{q}_t^\eta = q_t, \\ ((g_{t+j}, m_{t+j}))_{j=0}^{H-1} \in R_H(G_t; \hat{e}^\eta),\ c_{t+j} \in \mathcal{C}_j(m_{t+j}, \hat{q}_{t+j}^\eta)\ \ \forall j \end{array} \right\}, \tag{4}$$

where $R_H(G_t; \hat{e}^\eta)$ recursively applies predicted effects to a scoring-only branch graph and enforces node availability, branch-predicted preconditions, and allowed modes at every imagined boundary. The estimate $\hat{q}_{t+j}^\eta$ supplies branch-local kinematic and collision checks. Hence every ordinary-task candidate $h \in \mathcal{F}_t$ has $(g, m) \in \mathcal{F}_t^{gm}$; special modes follow their separate transition rules. These branch states are used only for scoring; only observations update G_{t+1}.

4 UNINAVMANIP

UNINAVMANIP forms a shared belief from current observations, typed history, TaskGraph state, and kinematic context. The graph and active mode restrict subgoal–mode predictions, visual evidence, and action experts. An ensemble ranks graph-feasible candidates by progress, collision risk, disagreement, and prior; after short-prefix execution, fresh-observation verification updates the belief, memory, and graph. Figures 2–4 summarize the architecture, memory routing, and closed-loop selection.

4.1 TASKGRAPH AS A FEASIBLE-SET OPERATOR

At episode start, a low-rate adapted Qwen3-VL planner maps the instruction and head-view summary to a subgoal DAG (Figure 4). Under a fixed skill catalog, expert TaskGraph traces and

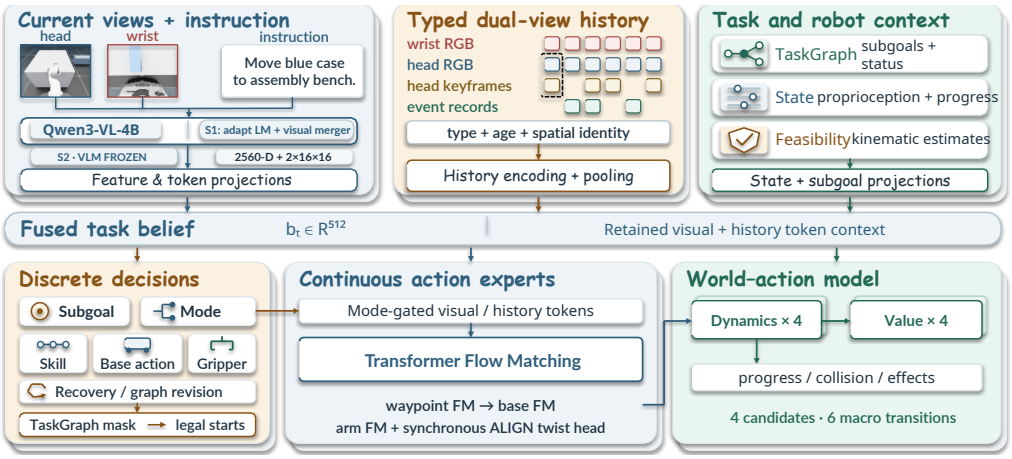


Figure 2: Model architecture of UNINAVMANIP during control learning. Stage 1 freezes the ViT and adapts the Qwen language backbone and visual merger; Stage 2 freezes the adapted Qwen3-VL and trains the policy modules shown below. The encoder supplies a 2560-D feature and two 16×16 grids (512 current visual tokens); trainable projections, typed-history encoding, and task-context fusion form a 512-D belief. Retained visual/history tokens condition the action experts, while dynamics/value ensembles evaluate graph-feasible candidates before prefix execution (Figure 4).

simulator-state labels supervise its nodes, edges, preconditions, add/delete effects, and modes. Outputs must be acyclic and schema-, skill-, and mode-valid; inferred facts remain in Q_t until confirmed. Post-prefix labels train effect/completion heads for e_{t+1}, while online updates use only camera and proprioception, revise unfinished nodes, and preserve completed history.

4.2 TASK-AWARE DUAL-VIEW MEMORY

The history H_t is the union of four bounded groups: wrist-local frames H_t^W, recent head frames H_t^R, head keyframes H_t^K, and decision/event records H_t^E. For a token r, its initial representation and typed-history encoding are

$$\begin{aligned} e_r &= E_H(r) + E_{\text{type}}(\rho_r) + E_{\text{age}}(\Delta t_r) + E_{\text{spatial}}(\sigma_r), \\ (\tilde{e}_r)_{r \in H_t} &= \text{TH}_\psi\left((e_r)_{r \in H_t}\right), \\ \bar{H}_t &= \text{Pool}\left((\tilde{e}_r)_{r \in H_t}\right), \end{aligned} \tag{5}$$

where ρ_r, Δt_r, and σ_r are its camera role, temporal age, and spatial identity, TH_ψ is a typed-history encoder, and Pool is its attention pooling operation. This pooled summary is the $\bar{H}_t$ used in (1). For a candidate mode m, learned camera gates $a_m^h, a_m^w \in [0, 1]$ and history attention produce

$$\alpha_{m,r} = \text{softmax}_{r \in H_t}(v_m^\top \tilde{e}_r), \qquad r_t(m) = \left[b_t;\, a_m^h s_t^h;\, a_m^w s_t^w;\, \sum_{r \in H_t} \alpha_{m,r} \tilde{e}_r \right], \tag{6}$$

Here s_t^h, s_t^w encode the current views and v_m is a mode query. NAVIGATE emphasizes head history, ALIGN combines both cameras, and MANIPULATE emphasizes wrist-local evidence.

4.3 MODE-ROUTED HYBRID ACTION HEADS

The shared belief predicts the mode, subgoal, recovery reason, and graph revision. Given a feasible pair (g, m), $r_t(m)$ conditions the skill, base-action, and gripper heads together with the continuous control branches. Figure 3 expands the action branches. NAVIGATE uses two conditional Flow Matching levels:

$$\begin{aligned} w_t[0 : K_s - 1] &= \text{FM}_\theta^{\text{wp}}(r_t(\text{NAVIGATE}))\,, \\ u_t^b[0 : K_d - 1] &= \text{FM}_\theta^{\text{base}}(r_t(\text{NAVIGATE}), w_t[0 : K_s - 1])\,, \end{aligned} \tag{7}$$

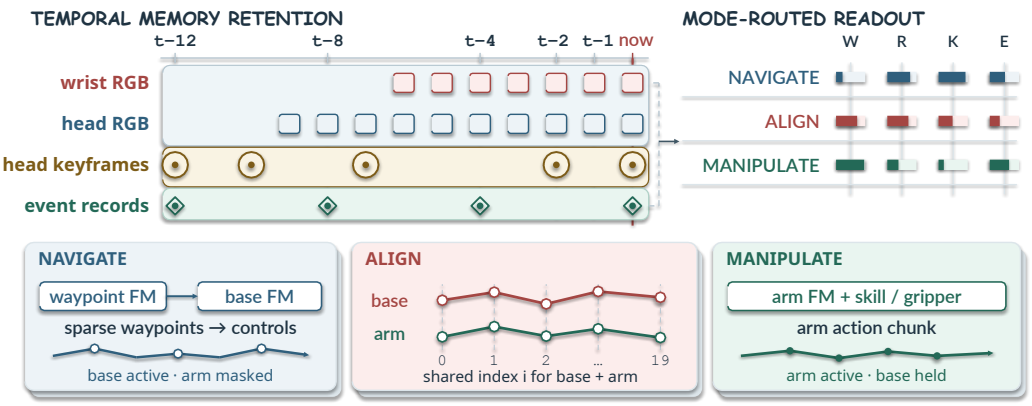


Figure 3: Typed memory and mode-routed control. Recent wrist/head rows contain RGB image tokens; head keyframes are a sparse subset of the head stream, while event records store nonvisual decision and verification metadata. Time positions follow the execution order, and token counts and attention bars show the allocation of context. NAVIGATE maps 8 sparse waypoints to 20 dense controls; ALIGN pairs base and arm commands at shared indices; MANIPULATE holds the base. Curves show command variation.

where sparse waypoints use $(x, y, \sin\psi, \cos\psi)$ and dense base actions use (v, ω). We index a decoded block locally, so $w_t[i]$ and $u_t^b[i]$ denote waypoint and base controls at block index i. ALIGN combines a fine twist $\xi_t[i] \in \mathbb{R}^2$ with the arm flow $u_t^a[i] \in \mathbb{R}^7$:

$$U_t[i] = [\xi_t[i]; u_t^a[i]], \qquad i = 0, \dots, L-1. \tag{8}$$

NAVIGATE masks the arm; ALIGN shares one clock across base and arm; MANIPULATE holds the base and activates the arm, skill, and gripper outputs. Other modes use zero-motion or reason-conditioned masks.

4.4 World-Action Candidate Rollout

At each decision boundary (Figure 4), the policy samples $N = 4$ graph-feasible candidates $\eta_k \in \mathcal{H}_t^H$. Here t, j, and i index decision boundaries, imagined macro transitions, and controls within a block. Each of the $H = 6$ ensemble transitions uses the first $H_{\text{exec}} = 5$ controls of a 20-control block (0.25 s at 20 Hz):

$$\begin{aligned} \hat{b}_{t,k}^{(e)} &= b_t, \\ \hat{b}_{t+j+1,k}^{(e)} &= T_{\phi,e}\Big(\hat{b}_{t+j,k}^{(e)}, \hat{\boldsymbol{h}}_{t+j,k}[0 : H_{\text{exec}} - 1]\Big), \\ & \qquad j = 0, \dots, H-1, \;\; e = 1, \dots, E, \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{h}}_{t+j,k} = (\hat{h}_{t+j,k}[i])_{i=0}^{19}$ and $\hat{h} = (\hat{g}, \hat{m}, \hat{c})$. A rollout covers 30 ticks (1.5 s); predicted effects define a scoring-only branch graph, while actual updates require observation. The selected first-block prefix is executed and its suffix discarded. For progress $\hat{V}_k$, collision $\hat{C}_k$, disagreement $\hat{S}_k$, and prior $\hat{\ell}_k$, we select

$$\hat{U}_k = \hat{V}_k - \lambda_c \hat{C}_k - \lambda_u \hat{S}_k + \lambda_p \hat{\ell}_k, \qquad k^* = \arg\max_k \hat{U}_k. \tag{10}$$

Independently replayed simulator branches from the same state provide progress, collision, and effect targets, separating candidate ranking from observation-based verification (Supplementary Appendix B).

Feasible-set and selection properties. For a proposal with positive mass on the feasible set, TaskGraph projection is

$$\pi_{\mathcal{F}}(h \mid b_t, G_t, q_t) = \frac{\pi_\theta(h \mid b_t)\mathbf{1}[h \in \mathcal{F}_t]}{Z_t}, \qquad Z_t = \int_{\mathcal{F}_t} \pi_\theta(h \mid b_t)\, dh > 0. \tag{11}$$

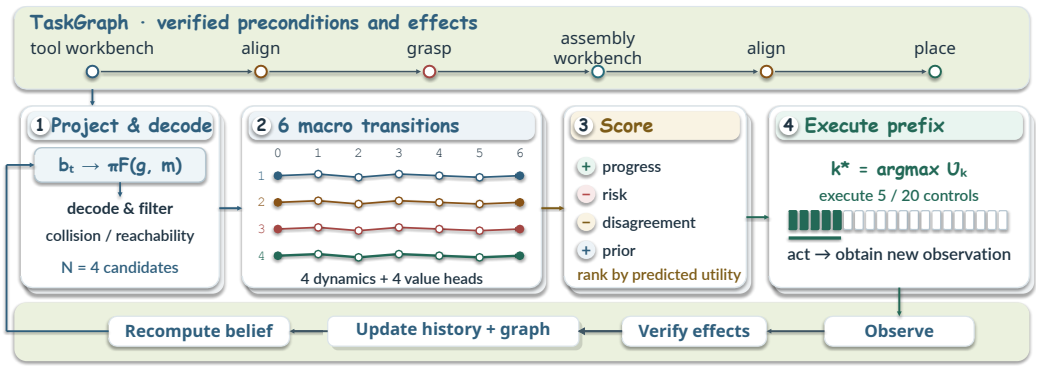


Figure 4: Graph-constrained selection for the task in Figure 1. Four candidates are decoded and filtered, then each ensemble member rolls out six macro transitions (30 nominal 20-Hz control ticks). Utility combines progress, collision risk, model disagreement, and prior. Each decoded block contains 20 controls; the controller executes its first 5 before observation, effect verification, history/graph updates, and belief recomputation. Curves and bars show the candidate score components.

Under the stated graph and revision assumptions, projected ordinary-task prefixes remain in $\mathcal{F}_t^{gm}$; physical filters apply before execution. Let $\mathcal{K}_t$ be the retained set, $U_k^\star$ its matched utility, and $U_{\text{all}}^\star = \sup_{\eta \in \mathcal{H}_t^H} U^\star(\eta)$. If termwise score errors are bounded by $\epsilon_V, \epsilon_C, \epsilon_S, \epsilon_\ell$, selecting $k^* = \arg\max_{k \in \mathcal{K}_t} \hat{U}_k$ gives

$$U_{k^*}^\star \geq U_{\text{all}}^\star - \Gamma_{\text{cov}} - 2\delta, \qquad \Gamma_{\text{cov}} = U_{\text{all}}^\star - \max_{k \in \mathcal{K}_t} U_k^\star, \qquad \delta = \epsilon_V + \lambda_c \epsilon_C + \lambda_u \epsilon_S + \lambda_p \epsilon_\ell. \quad (12)$$

Here Γ_{cov} is coverage error and 2δ is retained-set ranking error. The termwise assumptions imply $|\hat{U}_k - U_k^\star| \leq \delta$ for every $k \in \mathcal{K}_t$; applying this to the estimated and true retained maximizers yields 2δ, while Γ_{cov} extends the comparison to the full feasible horizon. Conditional on the retained set, no candidate independence is required; Supplementary Appendix B gives assumptions, proofs, and an iid hit corollary.

Verified-predecessor invariant. If completion is emitted only after g's incoming predecessors $\text{Pred}_{G_t}(g)$ are complete, (2) preserves

$$g \in D_t \implies \text{Pred}_{G_t}(g) \subseteq D_t.$$

Starting from downward-closed D_0, induction adds a node only after its predecessors; revisions preserve completed nodes and their order. With $\text{Avail}(G_t)$ and C_t in (3), ordinary successors therefore require completed predecessors and confirmed preconditions. Verification and physical checking remain separate; Appendix B states the full assumptions.

4.5 Learning Objective

For each decision prefix we supervise the current mode and subgoal, branch-specific actions, effects, and future-state predictions. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{mode}} + \mathcal{L}_{\text{subgoal}} + \mathcal{L}_{\text{branch}} + \mathcal{L}_{\text{flow}} \quad (13)$$

$$+ \lambda_f \mathcal{L}_{\text{future}} + \lambda_{\text{con}} \mathcal{L}_{\text{consistency}} + \lambda_v \mathcal{L}_{\text{value}} + \lambda_e \mathcal{L}_{\text{effect}}. \quad (14)$$

The branch term collects the mode-masked discrete heads and synchronized ALIGN supervision. The sparse-waypoint, waypoint-conditioned dense-base, and arm branches use conditional Flow Matching. For branch $r \in \{\text{wp}, \text{base}, \text{arm}\}$, with expert target y_r, Gaussian noise ϵ_r, and $\tau \sim \text{Unif}(0,1)$, define $z_{r,\tau} = (1-\tau)\epsilon_r + \tau y_r$ and

$$\mathcal{L}_{\text{flow}} = \sum_{r \in \{\text{wp,base,arm}\}} \mathbb{E}\left[\| v_{\theta,r}(z_{r,\tau}, \tau, \text{cond}_r) - (y_r - \epsilon_r) \|_2^2 \right]. \quad (15)$$

The base branch additionally conditions on its waypoint scaffold, and the arm branch on mode and skill; mode masks activate the relevant terms. With $\bar{b}_{t+1} = E^{-1} \sum_e \hat{b}^{(e)}_{t+1}$, the consistency term $\mathcal{L}_{\text{consistency}} = \|\text{sg}(b^{\text{obs}}_{t+1}) - \bar{b}_{t+1}\|_2^2$ aligns the one-step imagined belief with its observation-conditioned counterpart. We use cross-entropy for mode, subgoal, and effect labels, Huber regression for future/value targets, and binary cross-entropy for collision probabilities; ensemble members are fitted on bootstrap resamples.

4.6 Closed-Loop Inference

The controller projects, filters, and ranks candidates with (11) and (10). If none survives, it invokes RECOVER; OBSERVE verifies with zero motion, and DONE requires all nodes verified. After H_{exec} controls, a fresh observation defines b^{obs}_{t+1} and updates memory and graph. Thus $t+1$ indexes the executed prefix, and imagined states never enter either state store.

Mode transitions. Modes switch only at macro-block boundaries: NAVIGATE enters ALIGN after interaction-region verification, ALIGN enters MANIPULATE after reachability and alignment hold, and MANIPULATE returns through OBSERVE before enabling a successor. RECOVER is available after empty filtering or a confirmed failure predicate and also returns through OBSERVE. DONE lies outside $\text{Avail}(G_t)$ and is enabled exactly when all required nodes are in D_t, excluding unverified transitions from $\mathcal{H}^H_t$.

The model is trained in two stages, with expert action/effect supervision and additional recovery and verification targets. Episode construction, exact dimensions and clocks, target formation, domain randomization, and other reproduction details are provided in the supplementary material.

5 Experiments

We evaluate UniNavManip on held-out physics-based episodes, then isolate the contributions of graph projection, graph revision, dual-view memory, synchronized alignment, and world-action selection. All methods use the same observation interface, action clock, training data, and evaluation episodes.

5.1 Experimental Setup

Tasks and protocol. We evaluate in Isaac Sim (NVIDIA, 2025) with Isaac Lab (Mittal et al., 2025). The dataset contains 2,400 episodes across 30 factory, office, and home scenes and eight task templates; evaluation uses 240 episodes from three held-out scenes. Each instruction requires long-range approach, local alignment, manipulation, effect verification, and continuation or termination. An episode succeeds when every required TaskGraph node is verified from agent observations before timeout; simulator state supplies supervision and evaluation labels. The scene split, simulator versions, model configuration, and training protocol are in Supplementary Appendix A.

Baselines. We compare two matched internal baselines: a flat Qwen3-VL chunk policy (Bai et al., 2025) and hierarchical waypoint-plus-arm control. We also evaluate adapted public VLA policies, $\pi_{0.5}$ fine-tuned (Black et al., 2025b) and SERF-VLA adapted (Kim et al., 2026). Demonstrations, observation interface, action clock, low-level controller, and evaluation protocol are held fixed; each policy uses its own policy and planning interface.

Metrics. Task SR is complete-task success; Subgoal and Manip. measure required-node and verified-manipulation completion; Continue measures success after a verified manipulation when further navigation is required. Illegal transitions diagnose graph/planner admissibility, while Continue exposes whether observation-verified effects support the correct successor. Alignment error is the episode-averaged end-effector-to-target distance after ALIGN, in centimeters. Cost/oracle is the episode-averaged ratio of executed cost to matched-expert cost. Results use episode-level aggregation; definitions appear in Supplementary Appendix A.

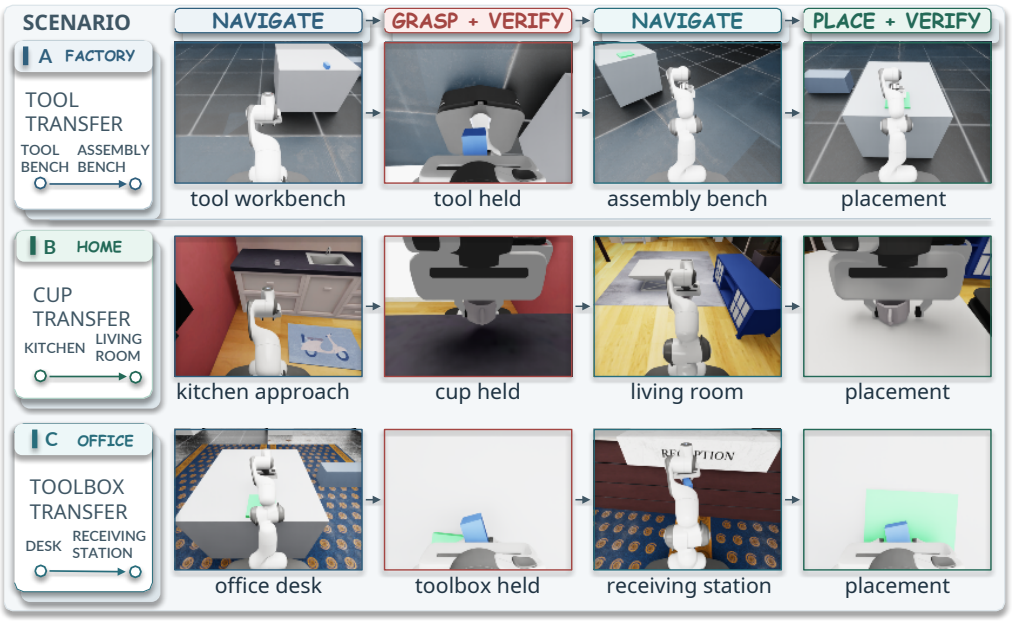


Figure 5: Isaac Sim RGB traces for factory tool transfer (top), home cup transfer (middle), and office toolbox transfer (bottom). Head/wrist frames show approach, object holding, target approach, and verified placement in each scenario. The shared header summarizes task phases.

5.2 Main Results

Table 1 reports complete-episode Task SR; Subgoal, Manip., and Continue localize failures before contact and after verified effects. Collision and cost remain separate outcome measures.

Table 1: Main results on held-out physics-based mobile-manipulation episodes.

Method	Task SR (%)↑	Subgoal (%)↑	Manip. (%)↑	Continue (%)↑	Coll./ep. ↓	Cost/oracle ↓
Flat Qwen3-VL chunk policy	38.8	64.2	54.8	46.5	0.43	1.31
Hierarchical waypoint + arm	49.6	73.6	66.1	58.7	0.29	1.20
$\pi_{0.5}$ fine-tuned	52.1	75.8	68.7	61.9	0.27	1.17
SERF-VLA adapted	57.1	79.6	72.4	66.7	0.22	1.13
UniNavManip (ours)	61.7	82.1	75.9	70.8	0.15	1.08

UniNavManip improves Task SR by 4.6 points over SERF-VLA and 12.1 over the hierarchical baseline, with the lowest collision rate and cost. Table 2 examines component contributions.

5.3 Component Ablations

Table 2 removes one component at a time with fixed training data and a 2048 visual-token capacity. Graph revision is disabled while boundary-level projection is retained; each success value is paired with its associated diagnostic.

Graph projection most affects Task SR and Illegal transitions; synchronized ALIGN reduces alignment error, while typed memory and graph revision preserve Continue. Removing rollout lowers Task SR from 61.7% to 54.2%; Supplementary Appendix A specifies the matched replacements.

Table 2: Component ablations under matched training data and a fixed 2,048-token visual budget.

Variant	Task SR (%)↑	Illegal ↓	Align err. (cm)↓	Continue (%)↑
Full UNINAVMANIP	61.7	0.6	3.7	70.8
– graph projection	52.1	8.7	4.1	61.0
– graph revision	59.2	1.5	4.0	65.8
– typed memory	57.9	1.2	5.5	62.9
– synchronized ALIGN	54.6	1.8	8.3	61.7
– world-action rollout	54.2	2.4	4.5	68.5

5.4 ROBUSTNESS AND RECOVERY

We test camera corruption, dynamics shift, target occlusion, and their combination. Recovery SR measures RECOVER events followed by a feasible action before timeout; the supplement defines each perturbation.

Table 3: Robustness under controlled test-time perturbations; Recovery is 0 without RECOVER.

Setting	Task SR (%)↑	Recovery (%)↑	Continue (%)↑	Coll./ep. ↓	Cost/oracle ↓
Nominal	61.7	81.2	70.8	0.15	1.08
Camera corruption	57.1	76.5	65.5	0.18	1.13
Dynamics shift	54.6	72.7	62.8	0.24	1.19
Target occlusion	52.1	68.8	59.9	0.20	1.17
Combined shift	45.8	61.4	54.2	0.31	1.29
Combined, no RECOVER	37.5	0.0	45.0	0.39	1.37

Combined shift yields 45.8% Task SR and 0.31 collisions; target occlusion is the strongest single perturbation. Replacing RECOVER with OBSERVE yields 37.5% and 0.39, respectively.

5.5 SYSTEM EFFICIENCY

Table 4 reports rollout cost; matched measurement settings and evaluation definitions are given in Supplementary Appendix A.

Table 4: System efficiency under matched observation/action settings.

Method	Visual cap. (tokens)↓	GPU memory (GB)↓	Latency (ms)↓
Dense-history hybrid policy	1536	22.6	187
Hierarchical waypoint + arm	512	15.8	101
UNINAVMANIP without rollout	2048	17.2	118
UNINAVMANIP (ours)	2048	19.6	171

On one NVIDIA RTX 4090, batch-1 bfloat16 measurements average 1,000 decisions after 100 warm-up steps. Rollout adds 53 ms and 2.4 GB for the 7.5-point gain in Table 2; 171 ms remains below the 250 ms interval.

Conclusion. UNINAVMANIP couples graph selection, typed memory, hybrid Flow Matching control, and world–action scoring. Across 240 held-out episodes, it achieves the highest success and lowest collision rate among evaluated policies, improves over component ablations, and remains effective under controlled perturbations.

AI use statement. Generative AI assisted language editing and organization; the authors remain responsible for all content.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems XXI*, 2025a. doi: 10.15607/RSS.2025.XXI.010.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX*, 2023a. doi: 10.15607/RSS.2023.XIX.025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023b.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. A survey of safe learning for robot control. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

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.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2023.

Frederik Ebert, Yilun Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In *RSS*, 2022.

Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *CVPR*, pp. 538–547, 2019. doi: 10.1109/CVPR.2019.00063.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. doi: 10.1146/annurev-control-091420-084139.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9–10):1194–1227, 2013. doi: 10.1177/0278364913484072.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024.

Sunghwan Kim, Byeonghyun Pak, Kehan Long, Yulun Tian, and Nikolay Atanasov. Serf: Spatiotemporal environment and robot feature map for long-horizon mobile manipulation. *arXiv preprint arXiv:2606.12956*, 2026.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Lingelbach, Jiankai Sun, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. In *CoRL*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICML*, 2023.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

NVIDIA. NVIDIA Isaac Sim. Version 5.1.0, 2025. URL `https://docs.isaacsim.omniverse.nvidia.com/5.1.0/`.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, pp. 15922–15932, 2021. doi: 10.1109/ICCV48922.2021.01564.

Xavier Puig, Kevin Ra, Marko Boben, Jialin Li, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities for the common sense of artificial intelligence. In *CVPR*, 2018.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, 2020.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Kreiman, Tianhe Xu, et al. Octo: An open-source generalist robot policy. In *RSS*, 2024.

Lin Yen-Chen, Eric Undersander, Andrey Yang, Sanjay Arnaud, Justin Chang, Di Gao, Rafal Jain, Zsolt Kira, Vladlen Koltun, Jitendra Malik, et al. Homerobot: Open-vocabulary mobile manipulation. In *CoRL*, 2023.

Tony Z. Zhao, Vikas Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016.