CST-WM: A CAUSALLY STRUCTURED WORLD MODEL FOR EMBODIED VISUAL TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied visual tracking requires a robot to choose actions that keep a moving target observable at a suitable distance, and to recover it after occlusion, out-of-view drift, or distractor crossings. We cast the task as planning over future target evidence with an action-conditioned world model. In logged tracking data, however, the behavior policy's actions are correlated with where the target is, so a generic predictor can learn a shortcut: it writes the current action directly into its prediction of target evidence, instead of letting the action affect that evidence only by moving the robot and changing what it observes. We call this failure causal hallucination; the resulting rollouts look plausible but rank candidate actions for the wrong reason. We propose CST-WM, a causally structured world model whose state is split into target-evidence, robot, and observation branches. Its transition removes the same-step edge from action to target evidence but keeps the path through robot motion and the resulting views, so candidate actions are still distinguished by their predicted ego-motion. With rollout-based model-predictive control, a single model handles both steady following and re-acquisition after target loss. On EVT-Bench and Habitat 3.0, covering standard tracking, target-loss recovery, and cross-dataset transfer, CST-WM improves following, distance-range control, safety, and re-acquisition over reactive trackers and world-model baselines, and removing the action mask causes the largest drop in re-acquisition among our ablations. Offline, CST-WM has lower multi-step rollout error, and its ranking of candidate actions agrees better with the simulator's. On a Unitree Go2 quadruped, CST-WM succeeds in 20 of 30 real-world trials under occlusion, distractor crossing, and fast motion, against 14 for TrackVLA (Wang et al., 2025).

1 INTRODUCTION

Embodied visual tracking (Luo et al., 2018; Zhong et al., 2019; 2024; Wang et al., 2025) appears simple: a robot only needs to stay with a moving target. In practice, reliable following requires the robot to preserve target observability, regulate following distance, and remain safe while both agents move through cluttered, partially observable environments (Maalouf et al., 2024; Zhong et al., 2023; Ye et al., 2024). The hardest moments are not when the target is clearly visible, but when it is temporarily occluded, leaves the field of view, or becomes confusable with distractors (Zhong et al., 2021; Liu et al., 2025a; Li et al., 2020). The value of an action therefore depends not only on how it changes the robot state, but also on how it changes future target visibility and apparent scale. Embodied visual tracking is thus a predictive decision problem rather than a reactive mapping from the current frame to the next control (Ha & Schmidhuber, 2018; Hafner et al., 2019; Bar et al., 2025). Learning the predictor from logged tracking data has a pitfall: the recorded robot turns toward and slows down for the target, so its actions are correlated with target evidence. We call this failure mode causal hallucination: an action-conditioned predictor can exploit that correlation by attributing to the current action a direct effect on target evidence, instead of letting the action influence it only through robot motion and the resulting change of view.

Neither dominant formulation covers these moments. Reactive trackers (Devo et al., 2021; Zhong et al., 2024; Wang et al., 2025; Scofano et al., 2025; Zhong et al., 2019; Luo et al., 2018; Zhang et al., 2025; Kim et al., 2024; Black et al., 2024) are myopic: once the target disappears or becomes ambiguous, they offer little basis for choosing actions that support multi-step recovery. A generic action-conditioned world model (Bar et al., 2025; Hafner et al., 2020; 2025; Bruce et al., 2024) is not sufficient either: a predictor that writes the current action directly into its target-related state produces plausible futures with the wrong semantics for re-acquisition. The error is invisible on logged action

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.

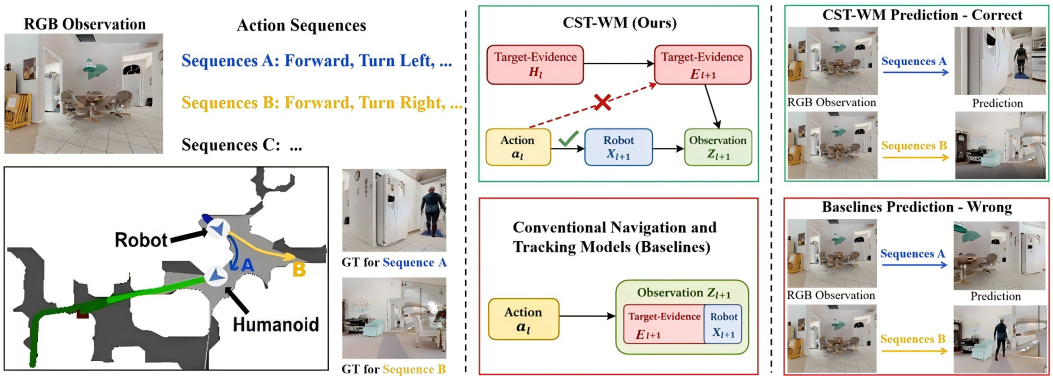


Figure 1: Causal structure of action-conditioned prediction in embodied visual tracking. Given an egocentric observation and candidate action sequences (A: forward, turn-left; B: forward, turn-right), the model must imagine which sequence preserves target observability and following distance. Conventional world models route the current action directly into the target-evidence update, producing plausible rollouts that fail to discriminate A from B. CST-WM removes this direct edge: the target-evidence branch $E_{\ell+1}$ is updated without action injection, and the action only reaches future observations through the robot branch $R_{\ell+1}$. The resulting predictions distinguish the two sequences and agree with the simulator.

sequences, where both explanations predict similar futures, and surfaces where planning relies on the model, under candidate actions that the recording robot never took.

As illustrated in Figure 1, tracking does not require recovering the target's full state at every step: for planning, a compact target evidence that says whether the target is observable and whether its apparent scale is compatible with a valid following distance suffices. We propose CST-WM, a causally structured world model with a target-evidence branch, a robot branch, and an observation branch. Its transition updates the target-evidence branch without the current action, which enters only the robot branch and reaches future observations through that route, and the target evidence of an imagined future is read from the observation branch by an evidence head distilled from the detector. Different candidate actions thus change later target evidence through the views they produce; while the target is hidden, candidates are ranked by whether their imagined views bring it back, so one rollout-based planner supports both stable following and re-acquisition.

We summarize our contributions as follows: (i) we formulate embodied visual tracking as planning over future target evidence and propose CST-WM, a causally structured world model that separates target evidence, robot motion, and observation dynamics while blocking direct action injection into the target-evidence branch; (ii) we introduce a compact target-evidence representation of observability and apparent scale for following, recovery, and safety-aware planning, read on imagined futures by an evidence head distilled from an open-vocabulary detector, with no privileged geometric supervision at test time; and (iii) on EVT-Bench single-target tracking (Wang et al., 2025) and Habitat 3.0 (Puig et al., 2024), against reactive trackers and an adapted NWM planner that differs only in the transition, CST-WM improves following, distance-range control, safety, and re-acquisition, and its imagined futures have higher rollout fidelity and rank counterfactual candidates closer to the simulator, with structural diagnostics, planning cost, and real-robot trials.

2 Related Work

2.1 Embodied visual tracking

Embodied visual tracking requires a robot to follow a moving target from egocentric observations while preserving visibility, regulating following distance, and remaining safe (Ye et al., 2024; 2025; Francis et al., 2025). Detection-plus-planning systems achieve basic following but degrade when the target becomes occluded or visually ambiguous in cluttered scenes. A major recent direction is reactive policy learning: reinforcement-learning-based active trackers (Devo et al., 2021; Bajcsy et al., 2024; Zhong et al., 2021; Zeng et al., 2024), offline-RL extensions such as EVT (Zhong et al., 2024; Shah et al., 2022; Liu et al., 2024a), and occlusion-aware trackers with explicit re-acquisition modes (Sun et al., 2026), alongside social-aware and vision-language-action variants that map rich visual context to control (Scofano et al., 2025; Wu et al., 2025; Bayraktar, 2025; Liu et al., 2025b; Zhang et al., 2025; Wang et al., 2025; Liu et al., 2025a). These methods strengthen robustness

under occlusion and distractors, but they still mainly map recent observations to actions rather than comparing how alternative actions affect future target observability and scale over multiple steps.

2.2 World models

World models instead learn latent dynamics from observation histories and support planning by rolling out futures before action execution. From early latent-dynamics methods (Ha & Schmidhuber, 2018; Hafner et al., 2019) to the Dreamer family and its embodied extensions (Hafner et al., 2020; 2021; 2025; Wu et al., 2023), and to broader generative or driving environments (Bruce et al., 2024; Wang et al., 2024a;b; Li et al., 2025; Hu et al., 2023), this paradigm enables comparing multiple candidate futures before acting. The most directly relevant prior work in embodied navigation is Navigation World Models (Bar et al., 2025), which shows that diffusion-based world modeling can support egocentric planning. WoMAP (Yin et al., 2025) distills open-vocabulary detector confidences into a reward head of a latent world model for active object localization, so imagined states are scored without decoding them. CST-WM reads imagined target evidence the same way and additionally constrains how the action may reach it. A generic action-conditioned transition can write the current action directly into its target-evidence branch, a tracking-specific form of causal hallucination related to shortcut learning (Geirhos et al., 2020) and to causal confusion in imitation learning (de Haan et al., 2019), where a policy relies on features correlated with the expert's actions; here the confounded quantity is the model's prediction of target evidence, which planning queries under actions the data never contain. CST-WM therefore excludes the current action from the target-evidence branch while it still drives the robot and observation branches.

3 Method

3.1 Problem Definition

At each step ℓ the robot receives an egocentric RGB observation $O_\ell \in \mathbb{R}^{h\times w\times 3}$ and executes an action $a_\ell = (v_\ell, \omega_\ell) \in \mathcal{A} \subseteq \mathbb{R}^3$, with planar linear velocity $v_\ell \in \mathbb{R}^2$ and angular velocity $\omega_\ell \in \mathbb{R}$. The robot never observes the target's state directly, and an episode succeeds if it keeps facing the moving target at a following distance of 1–3 m. Because the target moves on its own and the scene can hide it, an action is judged mainly by what the robot sees afterwards: whether the target stays in view, and how large it appears. We therefore predict these consequences with a world model and choose actions by planning over them (Figure 2).

3.2 Causally Structured World Model

3.2.1 Structured State

The planner needs to know whether the target is visible and how far away it is, and both can be read from the image: GroundingDINO (Liu et al., 2024b) returns a confidence $H_\ell^{\text{vis}} \in [0,1]$ and a box whose relative area gives the apparent scale $H_\ell^{\text{area}} \in [0,1]$:

$$H_\ell = \Gamma(O_\ell) = [\, H_\ell^{\text{vis}},\, H_\ell^{\text{area}} \,] \in [0,1]^2. \tag{1}$$

The score varies monotonically with distance over most of the following range and saturates only at very close range (Figure 7 in Appendix D.3), so H_ℓ serves as an observability and distance signal without privileged geometry. The rest of the scene enters through the latent $Z_\ell = \phi(O_\ell)$ of a pre-trained VAE (Blattmann et al., 2023) with decoder $\mathcal{D}$, and the robot pose $x_\ell \in \mathbb{R}^3$ (planar position and yaw) is integrated from past actions, $x_\ell = \mathcal{F}(a_{0:\ell-1})$. The state is $S_\ell = [H_\ell, Z_\ell, x_\ell]$; the transition receives H_ℓ as the scalar $0.4\, H_\ell^{\text{vis}} + 0.6\, (H_\ell^{\text{area}})^{1/2}$ and conditions on the current step only, without observation history, while the planner uses the two components read by the evidence head.

3.2.2 Where the Action May Enter

Physically, the action can change target evidence only by moving the camera: $a_\ell \to x_{\ell+1} \to Z_{\ell+1} \to H_{\ell+1}$. Logged data offer a second route. The behavior policy that records a tracking dataset turns toward a target drifting out of view and slows down near it, so the current action and the next target evidence share a common cause, the target's position. A transition whose target representation reads a_ℓ can fit this correlation and learn what the behavior policy usually sees after an action rather than what the action does to the view; this is causal hallucination (Section 1). On logged actions both routes predict similar futures, so training hardly separates them, but planning scores counterfactual candidates that the behavior policy never chose.

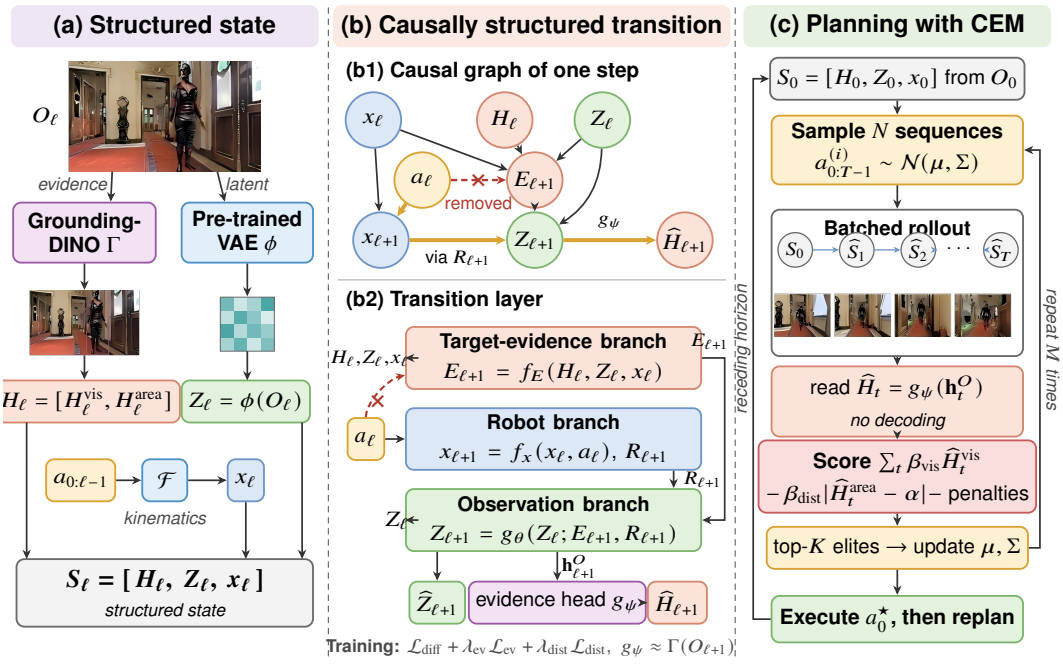


Figure 2: Overview of CST-WM. (a) *State.* GroundingDINO reads the target evidence H_ℓ (confidence, box area) from O_ℓ, a frozen VAE encodes O_ℓ into Z_ℓ, and the pose x_ℓ is integrated from past actions. (b) *Transition.* (b1) The edge $a_\ell \to E_{\ell+1}$ is removed, and the action reaches the next target evidence only along the highlighted path through $R_{\ell+1}$ and $Z_{\ell+1}$. (b2) Each layer updates the target-evidence, robot, and observation branches in this order, and the evidence head g_ψ, distilled from the detector, reads $\widehat{H}_{\ell+1}$ from the observation branch. (c) *Planning.* CEM rolls out N sampled sequences, scores the evidence read from the imagined views, refits to the top K, and executes the best first action.

We therefore give the action only the physical route, with two requirements:

$$\widehat{H}_{\ell+1} = g_\psi(\mathbf{h}^O_{\ell+1}), \qquad \frac{\partial E_{\ell+1}}{\partial a_\ell} = 0. \tag{2}$$

The first reads the target evidence of a predicted state from $\mathbf{h}^O_{\ell+1}$, the observation-branch tokens that generate the predicted view $\widehat{Z}_{\ell+1}$, with an evidence head distilled from the detector, $g_\psi(\mathbf{h}^O_{\ell+1}) \approx [H^{\text{vis}}_{\ell+1}, (H^{\text{area}}_{\ell+1})^{1/2}]$. Evidence thus follows the view, and, as with WoMAP's reward head (Yin et al., 2025), imagined states are scored without decoding. The second requires $E_{\ell+1}$, the transition's internal representation of the target, to be computed without the current action. Together they factorize the one-step transition as

$$p(x_{\ell+1}, Z_{\ell+1}, H_{\ell+1} \mid S_\ell, a_\ell) = p(x_{\ell+1} \mid x_\ell, a_\ell)\; p(Z_{\ell+1} \mid Z_\ell, E_{\ell+1}, R_{\ell+1})\; p(H_{\ell+1} \mid Z_{\ell+1}), \tag{3}$$

with $E_{\ell+1} = f_E(H_\ell, Z_\ell, x_\ell)$, $R_{\ell+1}$ the robot representation defined below, and $p(H_{\ell+1} \mid Z_{\ell+1})$ realized by g_ψ on the tokens that generate $Z_{\ell+1}$. The constraint removes only the direct edge: the action still shapes all later target evidence through the view, so candidates that move the camera differently still receive different evidence. The mask fixes the route; Section 4.4 tests how the model ranks actions the behavior policy never took, under paired interventions from the same simulator state.

3.2.3 Transition Architecture

The transition realizes Eq. (3) as a stack of causal layers. Each layer updates three groups of tokens in a fixed order, and attention masks decide what each group may read. The target-evidence tokens are updated first, $E_{\ell+1} = f_E(H_\ell, Z_\ell, x_\ell)$, and attend only to target-evidence, current-pose, and current-observation tokens, none of which carries a_ℓ. The robot tokens are updated next, $x_{\ell+1} = f_x(x_\ell, a_\ell)$ and $R_{\ell+1} = [x_{\ell+1}, W_a(a_\ell)]$, so $R_{\ell+1}$ is the only representation that carries the current action. The observation tokens come last, $Z_{\ell+1} = f_O(Z_\ell; R_{\ell+1}, E_{\ell+1})$, where f_E and f_O are attention and feed-forward blocks. The transition $\mathcal{T}_\theta : (S_\ell, a_\ell) \mapsto S_{\ell+1}$ thus has the paths $a_\ell \to R_{\ell+1} \to Z_{\ell+1} \to H_{\ell+1}$ and $H_\ell \to E_{\ell+1} \to Z_{\ell+1}$ and none from a_ℓ to $E_{\ell+1}$, so the zero Jacobian of Eq. (2) holds by construction (Section 4.5).

3.2.4 TRAINING

We train on transitions $(O_\ell, a_\ell, O_{\ell+1})$ from EVT-Bench (Wang et al., 2025) and Habitat 3.0 (Puig et al., 2024) (Appendix C). Only the next observation latent is generated, since the next pose follows from kinematics and the next evidence from the head. We noise $Z_{\ell+1}$ under a DDPM schedule, $\tilde{Z}_u = \sqrt{\bar{\gamma}_u}\, Z_{\ell+1} + \sqrt{1-\bar{\gamma}_u}\, \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, and train the denoiser ϵ_θ with

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\ell,u,\varepsilon}\Big[\lambda(u)\, \big\|\varepsilon - \epsilon_\theta(\tilde{Z}_u; S_\ell, a_\ell)\big\|_2^2\Big]. \tag{4}$$

The evidence head is trained on the same forward pass. It average-pools the final-layer observation tokens $\mathbf{h}^O_{\ell+1}$ at noise level u and regresses the detector's reading of the real next frame, $\mathcal{L}_{\text{ev}} = \mathbb{E}_{\ell,u,\varepsilon}\, \text{SL}_1\big(g_\psi(\mathbf{h}^O_{\ell+1}) - [H^{\text{vis}}_{\ell+1}, (H^{\text{area}}_{\ell+1})^{1/2}]\big)$ with $H_{\ell+1} = \Gamma(O_{\ell+1})$ and SL_1 the smooth-ℓ_1 loss; the square root spreads out the small boxes of distant targets. Because the observation branch attends to $E_{\ell+1}$ and $R_{\ell+1}$, this gradient reaches all three branches and shapes the dynamics toward what the planner reads. A small head g_{dist} on $E_{\ell+1}$, used only in training, predicts whether the next distance, taken from the simulator, lies in the following range (the auxiliary distance-aware loss $\mathcal{L}_{\text{dist}}$, Appendix A). The total loss is $\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{ev}}\mathcal{L}_{\text{ev}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}}$ with $\lambda_{\text{ev}} = 1$ and $\lambda_{\text{dist}} = 0.2$ (Appendix A). A rollout applies $\widehat{S}_{\ell+1} = \mathcal{T}_\theta(\widehat{S}_\ell, a_\ell)$ recursively, and the head reads each step's evidence at $u = 0$.

3.3 PLANNING WITH THE WORLD MODEL

Given the current state $S_0 = [H_0, Z_0, x_0]$, we search for an action sequence whose imagined future keeps the target in view at the right scale, or brings it back into view, without unsafe motion. We use the Cross-Entropy Method (CEM) (de Boer et al., 2005) with horizon $T = 10$, $N = 128$ candidates, $K = 16$ elites, and $M = 4$ iterations, with $S = 20$ DDIM sampling steps per imagined step. Each iteration samples $a^{(i)}_{0:T-1} \sim \mathcal{N}(\mu^{(m)}, \Sigma^{(m)})$, rolls out all candidates as one batch, scores them, and refits $(\mu^{(m+1)}, \Sigma^{(m+1)})$ to the elites (Appendix A). A candidate is scored by

$$\begin{aligned}\mathcal{V}(S_0, a_{0:T-1}) = \sum_{t=1}^{T}\Big[\beta_{\text{vis}}\, \widehat{H}^{\text{vis}}_t - \beta_{\text{dist}}\, \big|\widehat{H}^{\text{area}}_t - \alpha\big|\Big] \\ - \lambda_{\text{valid}} \sum_{t=0}^{T-1} \mathbb{I}\big(a_t \notin \mathcal{A}_{\text{valid}}\big) - \lambda_{\text{safe}} \sum_{t=1}^{T} \mathbb{I}\big(\widehat{S}_t \notin \mathcal{O}_{\text{safe}}\big),\end{aligned} \tag{5}$$

where $\widehat{H}_t = [\widehat{H}^{\text{vis}}_t, \widehat{H}^{\text{area}}_t] = g_\psi(\mathbf{h}^O_t)$ is read by the evidence head from each imagined view, $\widehat{H}^{\text{area}}_t$ being its estimate of $(H^{\text{area}}_t)^{1/2}$, and GroundingDINO runs only on the current observation O_0. The first term rewards seeing the target, the second rewards an apparent scale near the reference α, and the penalties count actions outside the velocity and smoothness limits $\mathcal{A}_{\text{valid}}$ and predicted states outside $\mathcal{O}_{\text{safe}}$, a kinematic clearance proxy on the predicted robot displacement with margin $d_{\text{safe}} = 0.20\,\text{m}$ that uses no map or depth (Appendix A). We set $\beta_{\text{vis}} = \beta_{\text{dist}} = \lambda_{\text{valid}} = \lambda_{\text{safe}} = 1$ and $\alpha = 0.5$. The robot executes the first action of the best sequence and replans at the next step (pseudocode in Appendix B).

Ranking candidates while the target is hidden. When the target is out of sight, $H^{\text{vis}}_0 \approx 0$, yet the candidates still predict different views, because each moves the camera differently. The observation branch combines that motion with the target-evidence branch, which carries the last evidence and the context in which the target was seen, so a candidate whose imagined views show the target again at step t collects the visibility and scale terms from t on. With the shortcut, candidates would instead be ranked by their resemblance to the behavior policy. If no candidate brings the target back within T steps, only the penalties remain, and the robot takes a safe first action and replans.

4 EXPERIMENTS

The experiments follow the chain of our claims: CST-WM tracks and recovers targets better than reactive trackers and an unstructured world model planned the same way (Sections 4.2–4.3); it does so because its imagined futures rank counterfactual candidates more like the real world (Section 4.4), and they do so because the action is kept out of the target-evidence branch (Sections 4.5 and 4.7).

4.1 EXPERIMENTAL SETTING

Benchmarks and metrics. We use EVT-Bench (Wang et al., 2025) and Habitat 3.0 (Puig et al., 2024) with egocentric observations and scene-disjoint splits. On EVT-Bench we evaluate single-target

Table 1: Main results. Left: EVT-Bench, single-target tracking (STT). Middle: Habitat 3.0, standard tracking from a visible target. Right: transfer from EVT-Bench to Habitat 3.0. †Released pipeline without the diffusion planner (Appendix C). Best in **bold**, second underlined.

EVT-Bench (STT)				Habitat 3.0, standard tracking					Cross-dataset transfer to Habitat 3.0				
Method	SR↑	TR↑	CR↓	Method	F↑	DRS↑	CR↓	ES↑	Method	F↑	DRS↑	CR↓	ES↑
Uni-NaVid	25.7	39.5	41.9	Habitat 3.0 baseline	0.29	0.47	0.48	0.40	Uni-NaVid	0.40	0.56	0.39	0.45
OA-VAT†	32.4	63.9	3.70	Uni-NaVid	0.34	0.52	0.43	0.47	TrackVLA	0.38	0.54	0.41	0.43
TrackVLA	85.1	78.6	1.65	Adapted NWM	0.41	0.61	0.33	0.49	Adapted NWM	0.43	0.58	0.35	0.47
TrackVLA++	86.0	81.0	2.10	SDA-S2	0.39	0.63	0.57	0.43	TrackVLA++	0.42	0.61	0.38	0.43
Ours	**88.7**	**83.4**	**1.41**	Ours	**0.53**	**0.70**	**0.27**	**0.61**	Ours	**0.48**	**0.65**	**0.30**	**0.54**



Figure 3: Recovery with CST-WM (Habitat 3.0 target-loss protocol). Each row is one episode: visible, at the image edge, lost, re-acquired; the blue curve is the planned path.

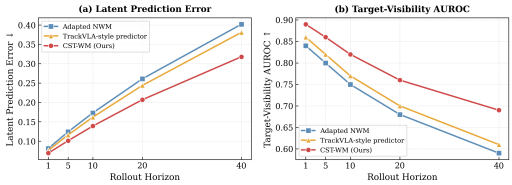


Figure 4: Offline rollout fidelity against horizon: (a) latent prediction error, (b) target-visibility AUROC.

tracking (STT) with success rate (SR), tracking rate (TR), and collision rate (CR). On Habitat 3.0 we evaluate tracking from a visible target with following rate (F), distance-range success (DRS), CR, and episode success (ES), and recovery under simulated short and long occlusions, out-of-view drift, and distractor crossings with re-acquisition success (Re-acq.), time to re-acquire in steps (TTR), following rate after re-acquisition (Post-F), and recovery episode success (Rec-ES). Transfer trains on EVT-Bench and tests on Habitat 3.0. We report means over three seeds (Appendix C).

Baselines. The first group consists of complete tracking systems with their own inputs, training data, and action selection: Uni-NaVid (Zhang et al., 2025), TrackVLA (Wang et al., 2025), TrackVLA++ (Liu et al., 2025a), the Habitat 3.0 baseline (Puig et al., 2024), SDA-S2 (Scofano et al., 2025), and OA-VAT (Sun et al., 2026), an occlusion-aware active tracker that we run on EVT-Bench STT with its released pipeline and PID control. The second group isolates our contribution. The adapted NWM (Bar et al., 2025) keeps NWM's non-factorized CDiT-XL/2 transition, in which the action conditions every token, and shares data, optimizer, evidence read-out, objective (Eq. (5)), CEM budget, and action bounds with CST-WM, at 6.6× our parameters and 3.6× our compute per pass, so any difference comes from the transition. Table 10 in Appendix C lists what every method receives.

4.2 Quantitative Comparison

Table 1 reports the three settings. On EVT-Bench STT, CST-WM reaches 88.7 SR and 83.4 TR, 2.7 and 2.4 points above TrackVLA++, with the lowest collision rate (1.41). On Habitat 3.0 it leads on all four metrics (F 0.53, DRS 0.70, CR 0.27, ES 0.61, against best baselines of 0.41, 0.63, 0.33, and 0.49). Against the adapted NWM, which differs only in the transition, the factorized model improves F and ES by 0.12 each and lowers CR by 0.06, although the adapted NWM is the larger model. The advantage carries over to transfer (+0.05 F, +0.07 DRS, +0.07 ES, −0.05 CR).

4.3 Target Loss and Re-acquisition

For each cause of target loss (Table 2), CST-WM has the highest Re-acq., Post-F, and Rec-ES and the shortest TTR under all four conditions. Its margin over the adapted NWM is 0.09 in Re-acq. under short occlusion and 0.15–0.16 under long occlusion, out-of-view drift, and distractor crossing, where the target stays hidden longest; it also re-acquires in 6.4–10.6 steps against 8.1–14.0. With objective and budget shared, the gap lies in which candidates each model imagines bringing the target back (Figure 3).

4.4 Offline World-Model Evaluation

Causal hallucination shows up only under counterfactual actions (Section 3.2), so we compare models offline under paired interventions: from each of 500 saved simulator states, the same 32 candidate action sequences are executed in the simulator and imagined by every model, including a TrackVLA-style predictor (a non-factorized transition conditioned on the action at every token); each model reads imagined evidence with its own head.

Table 2: Recovery after temporary target loss (Habitat 3.0 target-loss protocol). Re-acq.: re-acquisition success; TTR: time to re-acquire (steps); Post-F: following rate after re-acquisition; Rec-ES: recovery episode success. Best in **bold**, second underlined.

	Short occlusion		Long occlusion		Out-of-view drift		Distractor crossing	
Method	Re-acq.↑	TTR↓	Re-acq.↑	TTR↓	Re-acq.↑	TTR↓	Re-acq.↑	TTR↓
TrackVLA	0.71	8.9	0.48	14.6	0.52	12.8	0.43	15.2
Adapted NWM	0.75	8.1	0.54	13.2	0.57	11.7	0.49	14.0
Ours	**0.84**	**6.4**	**0.69**	**9.8**	**0.73**	**8.7**	**0.65**	**10.6**
Method	Post-F↑	Rec-ES↑	Post-F↑	Rec-ES↑	Post-F↑	Rec-ES↑	Post-F↑	Rec-ES↑
TrackVLA	0.58	0.47	0.41	0.29	0.45	0.33	0.37	0.24
Adapted NWM	0.62	0.50	0.45	0.34	0.49	0.37	0.41	0.29
Ours	**0.71**	**0.61**	**0.57**	**0.47**	**0.60**	**0.50**	**0.54**	**0.43**

Table 3: Offline diagnostics (Habitat 3.0). Top: ranking vs. simulator, the Plan. ρ of Tables 6–8 (TrackVLA-style: ρ 0.52, τ 0.37). Middle: action in $E_{\ell+1}$. Bottom: humanoid still, robot moving.

Diagnostic	Adapted NWM	Ours
Ranking: Spearman ρ↑	0.47	**0.68**
Ranking: Kendall τ↑	0.33	**0.51**
$\lvert\partial E_{\ell+1}/\partial a_\ell\rvert$↓	0.112	**0.001**
Var. under action swap↓	0.084	**0.006**
W_1 under action swap↓	0.127	**0.011**
Stability of $E_{\ell+1}$↑	0.61	**0.98**
View sensitivity↑	0.78	**0.84**

Table 4: Reading imagined target evidence on held-out 8-step rollouts, against GroundingDINO on the real frame: visibility AUROC after 1 and 8 steps, confidence MAE after 1 step, and Δ_{turn}, the change of predicted confidence after 8 steps when every turn is reversed (not ranked).

Read-out	1 step↑	8 steps↑	MAE↓	Δ_{turn}
Copy current evidence	**0.936**	0.589	**0.054**	–
Detect on decoded view	0.730	0.512	0.276	–
Separate latent head	0.875	0.623	0.113	–
Head, first sampling step	0.881	0.693	0.128	0.030
Head, u=0 (ours)	0.880	0.675	0.130	0.034
Head, w/o masking	0.860	**0.701**	0.131	0.025

4.4.1 Multi-Step Rollout Fidelity

We compare predicted and simulated futures at horizons of 1, 5, 10, 20, and 40 steps. CST-WM has lower latent prediction error and higher target-visibility AUROC, scored against GroundingDINO on the simulated frames, at every horizon (Figure 4). The gap grows with the horizon, so the structured transition accumulates less error over the multi-step rollouts that planning relies on.

4.4.2 Planning-Value Consistency

Accurate futures help only if they order candidates correctly. For each state we score its 32 candidates with Eq. (5) and compare the ranking with the one given by their outcomes in the simulator from that state. CST-WM agrees best with the simulator (Spearman ρ 0.68 against 0.47 for the adapted NWM and 0.52 for the TrackVLA-style predictor; Table 3).

4.4.3 Reading Imagined Target Evidence

The ranking rests on what the evidence head reads from imagined states. On held-out 8-step rollouts under logged actions, we compare each read-out with GroundingDINO on the real future frame (Table 4; protocol in Appendix D.2). After one step, copying the current evidence is hard to beat, since the view barely changes; after 8 steps the evidence head reaches a visibility AUROC of 0.675, against 0.589 for copying, 0.512 for detecting on the decoded view, and 0.623 for a separately trained latent head. Reversing the turn of every action changes the head's predicted confidence after 8 steps by 0.034 with the mask and by 0.025 without it; with the mask it can only arise through the view. Without the mask the head predicts the logged future better after 8 steps (0.701 against 0.675), yet it ranks the counterfactual candidates worse (ρ 0.55 against 0.68, Table 6) and recovers worse in closed loop (0.61 against 0.76). Accuracy on the actions the behavior policy took does not carry over to the actions it did not, which is the pattern a shortcut predicts.

4.5 Structural and Invariance Analysis

The offline gains should come from the constraint of Section 3.2, not from a better backbone. Two diagnostics check it: the Jacobian $\lvert\partial E_{\ell+1}/\partial a_\ell\rvert$ and the change of $E_{\ell+1}$ when a_ℓ is swapped for other valid actions (Table 3). Both are near zero for CST-WM, as the mask requires, whereas the adapted NWM and leaky variants absorb action information when allowed to (mean Jacobian 0.11 and up



Figure 5: Real-world tracking with CST-WM in a library, office corridors, and a parking lot at night. Each row is one episode, left to right in time; the blue curve is the planned path.

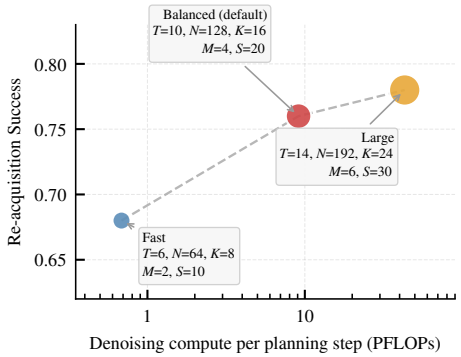


Figure 6: Re-acquisition against denoising compute per planning step; bubble size $\propto N$.

Table 5: Real-world tracking, 10 trials per scenario and method. SR/TR: success/tracking rate (%); 95% intervals in Appendix C.

	TrackVLA		Ours	
Scenario	SR↑	TR↑	SR↑	TR↑
Occlusion	40	55.8	**80**	**64.1**
Distractor	**60**	47.9	**60**	**56.2**
Fast motion	40	43.5	**60**	**51.2**

to 0.33). The mask does not guarantee a third property: with the humanoid held still and the robot moving, $E_{\ell+1}$ can change through Z_ℓ and x_ℓ. CST-WM keeps it stable (0.98 against 0.61) while its predicted views follow the viewpoint (0.84 against 0.78). The action thus moves the view but not the target.

4.6 Runtime and Real-World Tracking

Planning cost. A planning step runs MTS batched denoising passes over the N candidates, so its wall-clock time follows the budget. On one RTX A6000 a pass over 128 candidates takes 0.36 s, 2.8× faster than NWM's backbone. From observation to action, the default budget of 800 passes takes 267 s, the fast budget of Figure 6 22.5 s, and the robot budget below 8.2 s (Table 16). The evidence head reads 128 imagined views in 0.2 ms, against 11 ms and 38 ms per view to decode and detect. The simulator waits for the planner, so these times do not enter the simulated results, and a 4.7× larger budget adds only 0.02 re-acquisition. Under NWM's single-trajectory protocol, CST-WM matches NWM's reported times on the same GPU (Appendix D.4).

Real-world tracking. We follow the real-robot protocol of TrackVLA (Wang et al., 2025): occlusion, distractor crossing, and fast motion, with 10 trials per scenario and method (Appendix C). The robot is a Unitree Go2 with an Intel RealSense D435i RGB camera; as in TrackVLA and OA-VAT, planning runs on an offboard RTX 6000 Ada GPU with a reduced budget (T=4, N=24, K=8, M=2, 10 sampling steps), one planning step every 1.4 s; for deployment, the evidence head is further trained on 100k real frames of people from public videos together with 100k simulator frames (Appendix C).

Table 6: One-component ablations, Habitat 3.0.

Variant	F↑	DRS↑	Re-acq.↑	Plan. ρ ↑
W/o masking	0.47	0.63	0.61	0.55
W/o factorization	0.49	0.66	0.64	0.58
W/o dist.-aware loss	0.49	0.65	<u>0.72</u>	<u>0.64</u>
Deterministic	<u>0.50</u>	<u>0.67</u>	0.69	0.60
Full model	**0.53**	**0.70**	**0.76**	**0.68**

Table 7: Architectural variants, Habitat 3.0.

Variant	F↑	Re-acq.↑	Plan. ρ ↑	CR↓
Partial masking	0.46	0.68	<u>0.62</u>	<u>0.30</u>
Shuffled order	0.46	0.65	0.58	0.32
Deterministic	<u>0.50</u>	<u>0.69</u>	0.60	<u>0.30</u>
Full model	**0.53**	**0.76**	**0.68**	**0.27**

The model keeps no target identity, so a trial in which the robot follows the distractor counts as a failure. CST-WM succeeds in 8, 6, and 6 of 10 trials, against 4, 6, and 4 for TrackVLA (Table 5; episodes in Figure 5 and the video).

4.7 Ablation Study

Each ablation changes one component (Table 6); w/o masking removes only the target-evidence mask. Removing the mask costs the most re-acquisition (0.76 to 0.61) and planning-value consistency (0.68 to 0.55). A single-branch transition lowers F and consistency, dropping the distance-aware loss lowers all four by 0.04–0.05, and a deterministic predictor weakens recovery. Partial masking and a shuffled order also lower re-acquisition (Table 7); seeds vary by $\leq$0.02 (Table 15).

Target evidence: source and read-out. Table 8 varies where the planner's target evidence comes from. Our signal comes close to the simulator's true distance (0.76 against 0.78 re-acquisition). Detecting on simulator-rendered future views gives the highest planning-value consistency (0.74) and the lowest CR (0.24), but, reading raw detector scores, it reaches F and re-acquisition (0.48, 0.64) close to the raw-score row rather than our trained head. Among the deployable read-outs, the jointly trained head, whose loss also shapes the dynamics as in WoMAP (Yin et al., 2025), leads on every metric (re-acquisition 0.76, against 0.43 on a frozen transition, 0.57 for a separate latent head, and 0.46 for decode-and-detect) and reads an imagined step in 0.018 s instead of 5.8 s. The raw detector score lowers DRS to 0.60 and re-acquisition to 0.66; the distance-aware loss raises the signal's rank correlation with true distance from 0.71 to 0.83 across humanoids (Table 14).

Table 8: Target evidence for planning on Habitat 3.0: privileged sources, imagined-step read-outs, and signal variants; transition, objective, and CEM budget fixed. Time: read-out per imagined step.

	Target evidence	F↑	DRS↑	Re-acq.↑	CR↓	Plan. ρ ↑	Time (s)↓
Privileged	Simulator true distance	0.54	0.72	0.78	0.26	–	–
	Detector on rendered future views	0.48	0.67	0.64	0.24	0.74	5.2
Read-out	Detector on decoded views	0.38	0.46	0.46	0.40	0.58	5.8
	Separate latent head on $\widehat{Z}$	0.45	0.60	0.57	0.33	0.54	0.030
	Head, frozen transition	0.34	0.37	0.43	0.39	0.62	0.021
Signal	Raw detector score	0.46	0.60	0.66	0.32	–	–
	W/o distance-aware loss	0.49	0.65	0.72	0.29	0.64	–
Ours	Evidence head, joint training	0.53	0.70	0.76	0.27	0.68	0.018

5 Conclusion

CST-WM plans embodied visual tracking over imagined target evidence: its transition keeps the current action out of the target-evidence branch, so the action enters only the robot branch and reaches later evidence through the views it produces, and a detector-distilled head reads the evidence of each imagined future from the predicted view. Under paired interventions from the same simulator state, its imagined futures rank candidates closer to their simulated outcomes than the transition without the mask (ρ 0.68 against 0.55), and the gain carries to closed loop (re-acquisition 0.76 against 0.61). One model improves following, safety, and re-acquisition on EVT-Bench and Habitat 3.0, transfers from the first to the second, and holds on a real robot.

Limitations. CST-WM plans over detector-derived evidence, not the target's state: it inherits detector errors, keeps no target identity, and does not track a hidden target past the horizon (Appendix E). Its transition conditions on the current view only, so the target's motion is inferred from one frame, and each planning step runs its denoising passes in sequence, so the robot budget trades horizon and candidates for time (Appendix D.4).

AI USE STATEMENT

We used large language model assistants for editing (grammar, spelling, and word choice). All AI-assisted text was reviewed by the authors, and we take responsibility for the final content of this work.

ETHICS STATEMENT

The real-world trials involve lab members who consented to being recorded. The evidence head used on the robot is additionally trained on frames from publicly available online videos that contain people; the frames are used only to train the detector-distilled read-out, no identity information is extracted or used, and the frames are not redistributed.

REPRODUCIBILITY STATEMENT

The structured state, the causal transition, the evidence head, and the planning objective are specified in Section 3.2 and Section 3.3, with the full loss and CEM update equations in Appendix A and pseudocode in Appendix B. Data splits, recovery protocols, baselines, and evaluation metrics are described in Appendix C, and additional results are reported in Appendix D.

REFERENCES

Andrea Bajcsy, Antonio Loquercio, Ashish Kumar, and Jitendra Malik. Learning vision-based pursuit-evasion robot policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9197–9204. IEEE, 2024.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801, 2025.

Ertugrul Bayraktar. ReTrackVLM: Transformer-enhanced multi-object tracking with cross-modal embeddings and zero-shot re-identification integration. *Applied Sciences*, 15(4):1907, 2025. ISSN 2076-3417.

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. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623, 2024.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Alessandro Devo, Alberto Dionigi, and Gabriele Costante. Enhancing continuous control of mobile robots for end-to-end visual active tracking. *Robotics and Autonomous Systems*, 142:103799, 2021.

Anthony Francis, Claudia Pérez-D'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, Hao-Tien Lewis Chiang, Michael Everett, Sehoon Ha, Justin Hart, Jonathan P. How, Haresh Karnan, Tsang-Wei Edward Lee, Luis J. Manso, Reuth Mirsky, Sören Pirk, Phani Teja Singamaneni, Peter Stone, Ada V. Taylor, Peter Trautman, Nathan Tsoi, Marynel Vázquez, Xuesu Xiao, Peng Xu, Naoki Yokoyama, Alexander Toshev, and Roberto Martín-Martín. Principles and guidelines for evaluating social robot navigation algorithms. *ACM Transactions on Human-Robot Interaction*, 14(2):1–65, 2025.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2024.

Jing Li, Jing Xu, Fangwei Zhong, Xiangyu Kong, Yu Qiao, and Yizhou Wang. Pose-assisted multi-camera collaboration for active object tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):759–766, 2020.

Xiaofan Li, Chenming Wu, Zhao Yang, Zhihao Xu, Yumeng Zhang, Dingkang Liang, Ji Wan, and Jun Wang. DriVerse: Navigation world model for driving simulation via multimodal trajectory prompting and motion alignment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 9753–9762, 2025.

Jiahang Liu, Yunpeng Qi, Jiazhao Zhang, Minghan Li, Shaoan Wang, Kui Wu, Hanjing Ye, Hong Zhang, Zhibo Chen, Fangwei Zhong, Zhizheng Zhang, and He Wang. TrackVLA++: Unleashing reasoning and memory capabilities in VLA models for embodied visual tracking. *arXiv preprint arXiv:2510.07134*, 2025a.

Jinyi Liu, Yi Ma, Jianye Hao, Yujing Hu, Yan Zheng, Tangjie Lv, and Changjie Fan. A trajectory perspective on the role of data sampling techniques in offline reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*, pp. 1229–1237, Auckland, New Zealand, 2024a. IFAAMAS.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, pp. 38–55, 2024b.

Xuexin Liu, Zhuojun Zou, and Jie Hao. Adaptive text feature updating for visual-language tracking. In *Pattern Recognition*, pp. 366–381, Cham, 2025b. Springer Nature Switzerland. ISBN 978-3-031-78110-0.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3286–3295, 2018.

Alaa Maalouf, Ninad Jadhav, Krishna Murthy Jatavallabhula, Makram Chahine, Daniel M. Vogt, Robert J. Wood, Antonio Torralba, and Daniela Rus. Follow anything: Open-set detection, tracking, and following in real-time. *IEEE Robotics and Automation Letters*, 9(4):3283–3290, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations (ICLR)*, 2024.

Luca Scofano, Alessio Sampieri, Tommaso Campari, Valentino Sacco, Indro Spinelli, Lamberto Ballan, and Fabio Galasso. Following the human thread in social navigation. In *International Conference on Learning Representations (ICLR)*, 2025.

Dhruv Shah, Arjun Bhorkar, Hrishit Leen, Ilya Kostrikov, Nicholas Rhinehart, and Sergey Levine. Offline reinforcement learning for visual navigation. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 44–54, 2022.

Haowei Sun, Kai Zhou, Hao Gao, Shiteng Zhang, Jinwu Hu, Xutao Wen, Qixiang Ye, and Mingkui Tan. Instance-level visual active tracking with occlusion-aware planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. TrackVLA: Embodied visual tracking in the wild. In *Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4139–4164, 2025.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 55–72, 2024a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024b.

Kui Wu, Shuhang Xu, Hao Chen, Churan Wang, Zhoujun Li, Yizhou Wang, and Fangwei Zhong. VLM can be a good assistant: Enhancing embodied visual tracking with self-improving vision-language models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13154–13161, 2025.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World models for physical robot learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240, 2023.

Hanjing Ye, Jieting Zhao, Yu Zhan, Weinan Chen, Li He, and Hong Zhang. Person re-identification for robot person following with online continual learning. *IEEE Robotics and Automation Letters*, 9(11):9151–9158, 2024.

Hanjing Ye, Kuanqi Cai, Yu Zhan, Bingyi Xia, Arash Ajoudani, and Hong Zhang. RPF-Search: Field-based search for robot person following in unknown dynamic environments. *IEEE/ASME Transactions on Mechatronics*, 30(6):4129–4141, 2025.

Tenny Yin, Zhiting Mei, Tao Sun, Lihan Zha, Emily Zhou, Jeremy Bao, Miyu Yamane, Ola Shorinwa, and Anirudha Majumdar. WoMAP: World models for embodied open-vocabulary object localization. In *Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3605–3630, 2025.

Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. PoliFormer: Scaling on-policy RL with transformers results in masterful navigators. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 408–432, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Robotics: Science and Systems (RSS)*, 2025.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. AD-VAT: An asymmetric dueling mechanism for learning visual active tracking. In *International Conference on Learning Representations (ICLR)*, 2019.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Towards distraction-robust active visual tracking. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 12782–12792. PMLR, 2021.

Fangwei Zhong, Xiao Bi, Yudi Zhang, Wei Zhang, and Yizhou Wang. RSPT: Reconstruct surroundings and predict trajectory for generalizable active object tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):3705–3714, 2023.

Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline RL. In *European Conference on Computer Vision (ECCV)*, pp. 139–155, 2024.

A METHOD DETAILS

Auxiliary distance-aware loss. The full form of the auxiliary distance-aware loss attached to the target-evidence branch is

$$\mathcal{L}_{\text{dist}} = -\mathbb{E}_{\ell}\big[\, y^{\text{dist}}_{\ell+1} \log g_{\text{dist}}(E_{\ell+1}) + (1 - y^{\text{dist}}_{\ell+1}) \log(1 - g_{\text{dist}}(E_{\ell+1}))\,\big], \tag{6}$$

where $y^{\text{dist}}_{\ell+1} = \mathbb{I}(d^- \le d_{\ell+1} \le d^+)$ is the binary in-range distance label derived from the simulator-only relative distance $d_{\ell+1}$, with $[d^-, d^+]$ the following range, and g_{dist} is a lightweight prediction head on the target-evidence branch $E_{\ell+1}$. The distance is used only in training; the planner does not use the head.

Evidence head. The labels of $\mathcal{L}_{\text{ev}}$ are GroundingDINO (Swin-T) readings of the 224×224 training frames with the prompt "humanoid" and box and text thresholds of 0.20; the top-scoring box gives H^{vis} (its confidence) and H^{area} (its area over the image area). The head g_ψ is LayerNorm, a 512×512 linear layer, GELU and a 512×2 linear layer with a sigmoid, applied to the mean of the 28×28 final-layer observation tokens (0.26M parameters). We add the head and $\mathcal{L}_{\text{ev}}$ to a transition trained with $\mathcal{L}_{\text{diff}}$ for 12k steps and continue training all parameters with the total loss of Section 3.2 (smooth-ℓ_1 with threshold 0.05 for $\mathcal{L}_{\text{ev}}$). At planning time the head is read once per imagined step, from the tokens of the final denoising pass ($u = 0$), so no extra pass is run.

CEM update. At iteration m, with $\mathcal{I}^{(m)}$ denoting the indices of the top-K candidates, the sampling distribution is updated by

$$\mu^{(m+1)} = \tfrac{1}{K} \sum_{i \in \mathcal{I}^{(m)}} a^{(i)}_{0:T-1}, \quad \Sigma^{(m+1)} = \tfrac{1}{K} \sum_{i \in \mathcal{I}^{(m)}} \big(a^{(i)}_{0:T-1} - \mu^{(m+1)}\big)\big(a^{(i)}_{0:T-1} - \mu^{(m+1)}\big)^\top. \tag{7}$$

Admissible action and safe rollout sets. The admissible action set enforces platform velocity and smoothness limits,

$$\mathcal{A}_{\text{valid}} = \big\{\, a_t = [v_t, \omega_t] \;\big|\; \|v_t\|_\infty \le v_{\max},\ |\omega_t| \le \omega_{\max},\ \|a_t - a_{t-1}\|_\infty \le \delta_{\max} \,\big\}, \tag{8}$$

with a_{-1} the last executed action, and the safe rollout set is defined on the predicted robot branch alone, without map, depth, or simulator geometry. With $\widehat{p}_t$ the planar position in $\widehat{x}_t$ and $\Delta p_t = \|\widehat{p}_t - \widehat{p}_{t-1}\|_2$ the displacement predicted for one control period Δt, the clearance proxy is $d_{\text{clr}}(\widehat{S}_t) = v_{\max}\Delta t + d_{\text{safe}} - \Delta p_t$ and $\mathcal{O}_{\text{safe}} = \{\, \widehat{S}_t \mid d_{\text{clr}}(\widehat{S}_t) \ge 0 \,\}$ with $d_{\text{safe}} = 0.20\,\text{m}$: a predicted state is unsafe when its predicted displacement in one period exceeds the platform maximum $v_{\max}\Delta t$ by more than the margin d_{safe}. The proxy is not a distance to obstacles. Motion limits are matched to the simulator bounds used by all baselines.

B ALGORITHM SUMMARIES

Algorithm 1 summarizes training and Algorithm 2 planning.

C DETAILED EXPERIMENTAL SETTING

Benchmarks and tasks. We evaluate on EVT-Bench (Wang et al., 2025) and Habitat 3.0 (Puig et al., 2024). In both, the robot receives only egocentric observations and platform actions at test time; no humanoid state, future trajectory, or distance is available. We evaluate EVT-Bench's single-target tracking task (STT). Habitat 3.0 has richer scene geometry and more frequent occlusion and reappearance. On Habitat 3.0 we evaluate standard tracking from a visible target and a target-loss protocol in which the target becomes unavailable through short occlusion, long occlusion, out-of-view drift, or distractor crossing. For cross-dataset transfer, models trained on EVT-Bench are evaluated on Habitat 3.0 under the standard-tracking protocol.

Algorithm 1: Training procedure for CST-WM.

Input: transition dataset $(O_\ell, a_\ell, O_{\ell+1})$, model parameters θ, diffusion scheduler, optimizer.
Output: trained world model $\theta^\star$.

1. Encode O_ℓ and $O_{\ell+1}$ with the visual encoder to obtain the observation latents Z_ℓ and $Z_{\ell+1}$.
2. Read the target evidence H_ℓ and $H_{\ell+1}$ from O_ℓ and $O_{\ell+1}$ with the detector (confidence and normalized target area).
3. Form the structured state $S_\ell = [H_\ell, Z_\ell, x_\ell]$, with H_ℓ in its scalar form, and the robot inputs x_ℓ, a_ℓ.
4. Sample a diffusion step and corrupt the next observation latent $Z_{\ell+1}$ with Gaussian noise.
5. Predict the diffusion noise with the masked transition network (target-evidence branch without a_ℓ, robot branch with a_ℓ, observation branch reading both).
6. Compute the denoising objective, the evidence-head objective $\mathcal{L}_{\text{ev}}$ against $H_{\ell+1}$ on the pooled observation tokens, and the auxiliary distance-aware objective on the target-evidence branch (disabled only in the w/o distance-aware loss ablation); the head and $\mathcal{L}_{\text{ev}}$ are added after a 12k-step warm start with the denoising objective alone.
7. Update model parameters with AdamW.
8. Update the exponential moving average of the parameters.
9. Repeat until convergence and keep the validation-best checkpoint.

Algorithm 2: MPC planning procedure for CST-WM with the Cross-Entropy Method.

Input: current observation O_0 with its detector reading H_0, current state $S_0 = [H_0, Z_0, x_0]$, planning hyperparameters T, N, K, M, S.
Output: next control action $a_0^\star$.

1. Initialize the Gaussian sampling distribution over action sequences.
2. For each CEM iteration, sample N candidate action sequences.
3. Roll out the world model over horizon T for all candidates as one batch; read the target evidence of every predicted step with the evidence head g_ψ.
4. Score each candidate with the planning objective (Eq. (5)) on the evidence read from its imagined views.
5. Select the top-K elite sequences and update the sampling distribution.
6. After the final iteration, choose the best sequence and execute its first action.
7. Replan in receding-horizon form at the next step.

Data construction and splits. For each benchmark we build one-step transitions $(O_\ell, a_\ell, O_{\ell+1})$ from consecutive frames sampled at the simulator control frequency, keeping simulator metadata only for offline analysis and for the label of $\mathcal{L}_{\text{dist}}$. We remove corrupted frames, invalid robot states, reset fragments, terminal fragments without motion, and transitions that violate the platform action limits. Train, validation, and test splits are scene-disjoint (Table 9).

Training and model selection. All compared world models are trained on the same transitions and split with the settings of Table 9; ablations keep the backbone, schedule, and planner fixed and change one component. Checkpoints are selected on the validation split by the mean of F and DRS, with ties (to three decimals) broken by lower CR, and every learned model is trained with three seeds.

Metrics. On EVT-Bench, success rate (SR), tracking rate (TR), and collision rate (CR) measure task completion, tracking continuity, and safety. On Habitat 3.0, following rate (F) measures how consistently the robot follows, distance-range success (DRS) the fraction of time within the following interval, CR safety, and episode success (ES) whether the episode ends with valid following. Under target loss, re-acquisition success (Re-acq.) and time to re-acquire (TTR) measure recovery, and following rate after re-acquisition (Post-F) and recovery episode success (Rec-ES) measure whether tracking resumes. Re-acquisition is scored against the original target, so following the distractor does not count. The offline diagnostics are paired interventions: 500 simulator states are saved from held-out episodes, and from each state the same 32 candidate action sequences are executed in the simulator and imagined by every model, so the simulated outcome of each candidate is its interventional outcome from that state. *Rollout fidelity* compares predicted and simulated futures at horizons $\{1, 5, 10, 20, 40\}$ by latent error and by the AUROC of the evidence head's visibility reading against GroundingDINO on the simulated frame. *Planning-value consistency* scores the 32 candidates of each state with Eq. (5) and compares the ranking with the one given by their simulated outcomes by Spearman's ρ and Kendall's τ.

Table 9: Implementation details: dataset scale, architecture, optimization, planner, and compute.

Item	Details
Data	EVT-Bench / Habitat 3.0; scene-disjoint splits. Train scenes: 64 / 72; val: 8 / 9; test: 8 / 9. Trajectories: 48,000 / 61,000. One-step clips: 1.92M / 2.44M. Filtering removes corrupted frames, invalid robot states, reset fragments, terminal-only collision fragments, and action-invalid transitions.
Observation and action	Observations sampled at the simulator control frequency; consecutive frames without temporal skipping; actions at the same rate.
Model	VAE (Blattmann et al., 2023) (sd-vae-ft-ema; 34.2M-parameter encoder, 49.5M-parameter decoder), 224×224 views, latent $4\times28\times28$. Target-evidence branch input: the scalar $0.4\,H^{\mathrm{vis}} + 0.6\,(H^{\mathrm{area}})^{1/2}$ (the evidence head outputs the two components); conditioning on the current step only: one observation latent, no history; robot state: planar position and yaw; action (v, ω). Transition: 12 causal layers, each with target-evidence, robot (a pose sub-layer and an action sub-layer, giving $R_{\ell+1} = [x_{\ell+1}, W_a(a_\ell)]$), and observation sub-layers (attention + FFN), hidden 512, 8 heads, FFN 2048; 153.6M parameters, 89 GFLOPs per denoising pass. Evidence: GroundingDINO Swin-T (172.8M parameters), prompt "humanoid", views resized to 800×800, on the current observation and for training labels; evidence head g_ψ (0.26M parameters) on imagined steps.
Diffusion training	Linear DDPM with fixed variance, 1000 train steps, ε-prediction on the next observation latent, 20 DDIM inference steps; AdamW, lr 8×10^{-5}, no weight decay, batch 16, 150 epochs, grad-norm clip 10, EMA 0.9999, bf16 mixed precision; 3 seeds; evidence-head loss weight $\lambda_{\mathrm{ev}} = 1$ (joint training from the 12k-step transition); auxiliary distance-aware loss weight $\lambda_{\mathrm{dist}} = 0.2$.
Planning	T=10, N=128, K=16, M=4, 20 DDIM steps; β_{vis}=β_{dist}=λ_{valid}=λ_{safe}=1.0; α=0.5; d_{safe}=0.20 m; one stochastic rollout per candidate; target evidence read by g_ψ at every imagined step.
Compute	Training: a single GPU per model, bf16 mixed precision, deterministic evaluation seeds. Runtime: measured on one RTX A6000, and on an RTX 6000 Ada for the NWM-protocol trajectory times and the robot-budget planning step; per-component times, memory, and the NWM-protocol trajectory times are in Appendix D.4 and Table 17. Random seeds per main model: 3. Evaluation episodes per model: 500 starting states $\times$ 32 candidate sequences for offline diagnostics, plus standard EVT-Bench / Habitat 3.0 splits for downstream metrics.

Baselines. The reactive policies (Uni-NaVid (Zhang et al., 2025), TrackVLA (Wang et al., 2025), TrackVLA++ single-view (Liu et al., 2025a), and OA-VAT (Sun et al., 2026) on EVT-Bench; the Habitat 3.0 baseline (Puig et al., 2024) and SDA-S2 (Scofano et al., 2025) on Habitat 3.0) act from their own inputs and training data and serve as system-level references. The adapted NWM (Bar et al., 2025) isolates the transition. We retrain NWM's CDiT-XL/2 once on our trajectories with our optimizer settings and replace its goal-image objective, which tracking cannot provide, with Eq. (5). It reads current evidence with GroundingDINO and imagined evidence with an evidence head of our architecture, trained with $\mathcal{L}_{\mathrm{ev}}$ on its own final-layer observation tokens, and it shares the observation interface, planning horizon, candidate budget, CEM iterations, sampling steps, and action bounds. What remains different is the transition: the action conditions every token of the adapted NWM and of the TrackVLA-style predictor, whereas our target-evidence branch is masked. Table 10 summarizes what each method receives.

OA-VAT on EVT-Bench. OA-VAT (Sun et al., 2026) is evaluated on UnrealCV and a real drone. We run its released code on the 1,404 episodes of EVT-Bench STT: first-frame target search by prototype matching (YOLOE candidates and DINOv3 features), ORTrack tracking with its confidence-aware Kalman filter, prediction and re-acquisition modes, online prototype updates, and its PID control law. The diffusion planner checkpoint is not released, so we run the PID variant that OA-VAT reports in its own ablation. The reference image of the target is the first frame, with the simulator's box at $t = 0$ standing in for the box a user would draw; the simulator box is not used afterwards. The control law uses EVT-Bench's image-space gains and set-point, since OA-VAT's drone gains are in

Table 10: What each compared method receives at test time. "Same as ours" for the adapted NWM and the ablations means identical training data, optimizer settings, evidence read-out, planning objective (Eq. (5)), and CEM budget (T=10, N=128, K=16, M=4, 20 DDIM steps); ablations differ from the full model only in the named component. †Run by us on EVT-Bench STT with the released pipeline, without the diffusion planner.

Method	Type	Target evidence	Action selection	Model and compute
Uni-NaVid, TrackVLA, TrackVLA++	reactive VLA	learned, from RGB and instruction	one policy pass per step	own architecture and training data
OA-VAT†	reactive tracker	own detector, tracker, and re-identification	PID control law	released weights, no planner
Habitat 3.0 baseline, SDA-S2	reactive	own perception	one policy pass per step	own architecture and training data
Adapted NWM	world model, CEM	GroundingDINO on current view, evidence head on imagined views	Eq. (5), same budget	CDiT-XL/2, 1.01B, 321 GFLOPs per pass; same as ours otherwise
Ablations (Table 6)	world model, CEM	GroundingDINO on current view, evidence head on imagined views	Eq. (5), same budget	full model with one component changed
CST-WM (ours)	world model, CEM	GroundingDINO on current view, evidence head on imagined views	Eq. (5)	causal transition, 0.15B, 89 GFLOPs per pass

UnrealCV units. Thresholds are those of the released evaluation script. Episodes are terminated and scored with EVT-Bench's own evaluation code.

Real-world protocol. Following TrackVLA (Wang et al., 2025), each method runs 10 trials per scenario on the same course. In *occlusion*, the target walks behind bookshelves and around corridor corners; in *distractor*, a second person in similar clothing crosses between the robot and the target; in *fast motion*, the target walks quickly along a winding path. A trial succeeds if the robot follows the target to the end of the course without collision and without losing it for more than 6.8 s. The tracking rate is the fraction of time the target is in view at 1–3 m, measured from the recorded video. Following the distractor instead of the target is a failure. With 10 trials per scenario, the 95% Wilson intervals of the success rates are 49–94% for 8/10, 31–83% for 6/10, and 17–69% for 4/10; over all 30 trials they are 49–81% for CST-WM (20/30) and 30–64% for TrackVLA (14/30). To run on the robot, planning uses a smaller budget than in simulation, T=4, N=24, K=8, M=2, and 10 sampling steps, on an offboard RTX 6000 Ada: one planning step takes 1.4 s (Table 16), and between planning steps the robot executes the last chosen action.

Evidence head for deployment. All simulation results, including Table 8, use the evidence head trained on EVT-Bench and Habitat 3.0 as described in Section 3.2. For the real-world trials only, the evidence head is further trained on 100k real frames containing people, taken from publicly available online videos, mixed with 100k frames collected in EVT-Bench and Habitat 3.0, with the transition frozen. Real frames are labeled by GroundingDINO as for $\mathcal{L}_{\text{ev}}$, and their camera motion, which the videos do not record, is estimated by visual odometry and used as the action input of the forward pass.

D Additional Experimental Results

D.1 Structural Diagnostics

These diagnostics complement Table 3. For our model the action-dependence of $E_{\ell+1}$ is zero by construction, so its rows verify the implementation of the mask; the informative rows are those of

models whose target representation may read the action, which show how much action information such a representation absorbs when it is allowed to. Table 11 breaks the Jacobian $|\partial E_{\ell+1}/\partial a_\ell|$ down by target visibility: the adapted NWM and the leaky variants depend on the action both in and out of view, and their dependence correlates negatively with tracking performance across episodes (-0.36 to -0.53). Table 12 varies how deep the mask reaches: without a mask (A, single branch) the mean Jacobian is 0.19, a shallow mask (B) raises it to 0.33, and partial and deeper masks (C, D) lower it to 0.09 and 0.04. Table 13 replaces a_ℓ with other actions from $\mathcal{A}_{\text{valid}}$ at a fixed state and measures the spread of the resulting target representation. The robot-motion-only rows of Table 3 test what the mask does not enforce: with the humanoid held still, stability is one minus the normalized variation of $E_{\ell+1}$, which can change through Z_ℓ and x_ℓ, and view sensitivity is the normalized variation of $\widehat{Z}$ under viewpoint change.

Table 11: Jacobian $|\partial E_{\ell+1}/\partial a_\ell|$ of the target-evidence representation over all rollout states and by target visibility, and its correlation with per-episode tracking performance. Best in **bold**, second underlined (Jacobian columns).

Method	All states↓	In-view↓	Out-of-view↓	Corr. with performance
Adapted NWM	0.112	0.096	0.131	-0.41
Leaky variant B	0.331	0.304	0.357	-0.53
Leaky variant C	<u>0.086</u>	<u>0.071</u>	<u>0.102</u>	-0.36
Ours	**0.001**	**0.001**	**0.002**	-0.04

Table 12: Mean Jacobian $|\partial E_{\ell+1}/\partial a_\ell|$ for masks of increasing depth: A, none (single branch); B, shallow; C, partial; D, deeper.

Mask	A	B	C	D	Ours
Jacobian	0.19	0.33	0.09	<u>0.04</u>	**0.00**

Table 13: Target representation under action swap at a fixed state: variance across actions and mean pairwise W_1 distance.

Method	Variance↓	Pairwise W_1↓
Adapted NWM	<u>0.084</u>	<u>0.127</u>
Leaky variant	0.146	0.214
Ours	**0.006**	**0.011**

D.2 Reading Imagined Target Evidence

These protocols belong to the read-out experiments of Sections 4.4 and 4.7 (Tables 4 and 8).

Offline protocol. Table 4 uses 192 starts on held-out trajectories, half chosen among starts whose target visibility changes within 8 steps and half at random, so that the steps a planner must anticipate are well represented. From each start, a model rolls out 8 steps with 20 DDIM steps per step, EMA weights, and the logged actions and poses; the sampled latent becomes the next context, and the head's own reading, $0.4\,\widehat{H}^{\text{vis}} + 0.6\,\widehat{H}^{\text{area}}$ (the head outputs the root of the area), becomes the next evidence input. All read-outs use the same sampled latents, so they differ only in how they read them. Labels are GroundingDINO readings of the real frames, obtained as for $\mathcal{L}_{\text{ev}}$; a frame counts as visible when its confidence exceeds 0.35. The separate latent head is a three-stage convolutional network (64/128/256 channels, global average pooling, two-layer MLP) trained on encoder latents of the training frames with the same labels and with additive latent noise, blur, shifts, and horizontal flips as augmentation. The full model and the w/o-masking variant are trained from the same transition with the same schedule; the latter lets the target-evidence branch attend to the action token and is otherwise identical. For Δ_{turn}, the angular velocity of every action is negated and the rollout is repeated with the same noise.

Steps at which visibility changes. Restricting the evaluation to the steps at which the target's visibility differs from the start, the steps a planner has to anticipate, the evidence head reaches a visibility AUROC of 0.467 with the mask and 0.446 without it, against 0.530 for detecting on the decoded view and 0.417 for the separate latent head; copying the current evidence is wrong on these steps by definition.

Closed-loop protocol. Table 8 runs the Habitat 3.0 standard-tracking and target-loss episodes with the full transition and changes only the target evidence. The true-distance reference gives the planner the simulator's robot–target distance in place of the apparent-scale term. For the rendered-view reference, each CEM candidate is executed from a saved copy of the simulator state, the humanoid replays its recorded motion for the next T steps, and GroundingDINO reads the rendered 224×224 views with the thresholds of $\mathcal{L}_{\text{ev}}$. For decode-and-detect, the final denoised latent of every imagined step is decoded by $\mathcal{D}$ and read by the same detector. The frozen-transition head has the architecture of g_ψ and is trained with $\mathcal{L}_{\text{ev}}$ with all transition parameters fixed; the joint head follows the same schedule with all parameters updated, so the two differ only in whether $\mathcal{L}_{\text{ev}}$ shapes the dynamics. Read-out time is for one imagined step, a batch of $N{=}128$ views, measured inside the closed-loop planner, excluding denoising, as the median over imagined steps after warm-up; Table 18 times the head's forward pass alone.

D.3 Target-Evidence Signal and Seeds

Figure 7 plots the detector's score against the relative distance on EVT-Bench: it tracks the distance over most of the following range and saturates only very close to the target, which is why the planner can use it as a distance signal. Table 14 repeats the signal comparison across humanoids of different body scale, height, appearance, clothing, and gait: the distance-aware loss raises the Spearman correlation between the signal and the (sign-inverted) true distance from 0.71 to 0.83 and improves all four downstream metrics. Table 15 shows the full model on Habitat 3.0 standard tracking for three seeds, which vary by at most 0.02, against a gap of 0.12 in F to the strongest baseline in Table 1.



Figure 7: GroundingDINO target-evidence score against relative robot–human distance on EVT-Bench. These readings are the current evidence and the training labels of the evidence head.

Table 14: Signal across humanoids of different appearance and gait on Habitat 3.0. ρ: Spearman correlation between the signal and the sign-inverted true distance.

Variant	$\rho\uparrow$	F↑	DRS↑	Re-acq.↑	CR↓
W/o distance-aware loss	0.71	0.47	0.62	0.68	0.31
With distance-aware loss	**0.83**	**0.52**	**0.69**	**0.75**	**0.28**

Table 15: Full model per seed, Habitat 3.0 standard tracking.

Seed	F↑	DRS↑	CR↓
1	0.52	0.69	0.28
2	**0.54**	**0.71**	**0.26**
3	0.53	0.70	0.27

D.4 Runtime Details

Unless marked Ada, times are measured on one NVIDIA RTX A6000 (48 GB) in a workstation, as the median of repeated runs after warm-up, with the world model in bf16 autocast and eager mode (PyTorch 2.4) and GroundingDINO in fp32 as in its reference inference code (PyTorch 2.5); no quantization or distillation is used.

One planning step, from observation to action. A CEM planning step with N candidates, horizon T, M iterations, and S sampling steps runs MTS denoising passes, each over all N candidates as one batch, in sequence: the S passes of one imagined step, the T steps of one rollout, and the M iterations of CEM each depend on the previous one. The evidence head reads the MT imagined steps from the tokens of the final pass of each step, GroundingDINO reads the current observation once, and the current frame is encoded once. Table 16 gives the wall-clock time of the whole step for the

three budgets, measured by running the full step, with the count of every stage; the detector and the encoder take 0.1 s of it and the head reads under 0.02 s, so the denoising passes account for the rest. Denoising compute per step is $MTSN \times 89$ GFLOPs, the horizontal axis of Figure 6. In simulation the environment is stepped synchronously, so the planner's time does not affect the simulated results; on the robot, planning runs offboard with the robot budget, and the real-world protocol of Appendix C states the resulting planning rate.

Table 16: Wall-clock time of one planning step from the received observation to the issued action, for the three budgets, on one RTX A6000 (median of full steps after warm-up); the robot budget is also given on the offboard RTX 6000 Ada of the real-world trials. Denoising passes run in sequence, each over the N candidates as one batch; per-pass times are in Table 18.

Stage Budget (T, N, M, S)	Default (10,128,4,20)	Fast (6,64,2,10)	Robot (4,24,2,10)	Robot, Ada (4,24,2,10)
GroundingDINO, current view	0.09 s	0.09 s	0.09 s	–
VAE encoder, current frame	7 ms	7 ms	7 ms	–
Denoising passes (sequential, batch N)	800	120	80	80
Evidence-head reads (batch N)	40	12	8	8
Time per action (whole step)	267 s	22.5 s	8.2 s	1.4 s
Planning steps per second	0.0038	0.044	0.12	0.71

Components and the NWM protocol. Table 18 lists the components of a planning step at the candidate batch sizes of the three budgets; its per-pass times are isolated calls, and inside a full planning step a pass takes 3–8% less (Table 16 divided by its pass count). Table 17 follows the NWM protocol (Bar et al., 2025): one trajectory is simulated at batch size 1, and each state costs a VAE encoding of the 4-frame context, the diffusion sampler, and a VAE decoding, exactly as in the NWM inference code (our transition reads only the last frame of that context); the NWM CDiT-XL/2 backbone is timed with the same script on the same GPU, and our model is also timed on an RTX 6000 Ada, the GPU on which NWM reports its times. This protocol times the throughput of one trajectory, not a planning step. As in NWM, the 4-bit column divides the 6-step time by 4 without running quantization. NWM obtains its 6-step column by distillation; our 6-step column times a 6-step DDIM sampler. With `torch.compile`, a batch-1 denoising pass takes 24.7 ms and the 6-step trajectory of Table 17 1.53 s; at batch sizes of 64 and above the compiled model is not faster, so all planning-step times use eager mode.

Table 17: Time (s) to simulate one trajectory at batch size 1 (NWM protocol): 16 states with 250 steps (Base), 8 states (+Skip), 6 sampling steps, and 4-bit quantization. Ada: RTX 6000 Ada. *Estimated as in NWM: 6-step time divided by 4. †NWM distills to 6 steps; we time a 6-step sampler.

Model, GPU	Base	+Skip	+6 steps†	+4-bit*
NWM, Ada (reported)	30.3	14.7	0.4	0.1
NWM, A6000	166.2	84.0	2.30	0.57
Ours, A6000	135.6	67.7	1.88	0.47
Ours, Ada	23.4	12.3	0.34	0.08

Table 18: Per-component runtime and peak allocated GPU memory on one RTX A6000. Denoising passes are timed at the candidate batch sizes of the robot budget and of the fast, default, and large budgets of Figure 6; the NWM backbone is listed for reference. The evidence head reads imagined views during planning, GroundingDINO reads the current observation, and the decoder serves visualization and the decode-and-detect read-out of Table 8. *Batch size 1 includes CPU resizing in the reference code.

Component	Size	Batch	Time	Peak memory
VAE encoder, 1 / 4 frames	34.2M	1	7 / 23 ms	–
Ours, one denoising pass	153.6M, 89 GFLOPs	1 / 24 / 64	31 / 106 / 198 ms	1.2 / 1.6 / 2.4 GB
		128 / 192	362 / 572 ms	3.6 / 4.8 GB
NWM CDiT-XL/2, one denoising pass	1.01B, 321 GFLOPs	1 / 128	39 / 1017 ms	–
Evidence head g_ψ	0.26M	24 / 64 / 128 / 192	0.21 / 0.20 / 0.20 / 0.27 ms	–
VAE decoder, per view	49.5M	1 / 64	15.6 / 10.6 ms	–
GroundingDINO Swin-T, per view*	172.8M, 800×800	1 / 8	90 / 40 ms	1.0 / 3.5 GB
		32	38 ms	12.0 GB

E LIMITATIONS

Proxy, not target state. CST-WM is designed for planning-oriented embodied visual tracking rather than full target-state recovery. Its target evidence summarizes observability and apparent scale, so re-acquisition means that imagined views contain the target again within the planning horizon; the model does not represent the hidden target's trajectory, and when no candidate brings the target back within T steps the planner acts on the validity and safety terms until new evidence arrives (Section 3.3).

Detector dependence. Target evidence is read by an evidence head trained on GroundingDINO labels, so planning inherits the detector's errors: miscalibrated confidence under partial visibility, weak responses for very small or distant targets, saturation of the area term at very close range (Figure 7), and confusion between similar-looking humans. The model has no persistent notion of target identity: its evidence says that a humanoid the detector accepts is in view, so several candidate humans or a distractor that looks like the target can redirect the planner. The distractor-crossing condition of Table 2 and the cross-humanoid evaluation (Table 14) cover part of these conditions.

Evaluation scope. EVT-Bench results cover its single-target tracking task (STT); its distracted (DT) and ambiguous (AT) tracking tasks require the robot to keep the identity of one target among similar people, which is outside the scope of CST-WM.