DYNAMICS-INFORMED ACTION LEARNING FOR GOAL-CONDITIONED LATENT PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models enable visual control by planning action sequences in compact representation spaces, but a gap remains between how latent dynamics are learned and how they are used for planning. First, teacher-forced training does not expose the dynamics model to the recursively predicted latent states encountered during planning. Second, sampling-based planners typically initialize action search from goal-agnostic proposals, without exploiting the action-conditioned dynamics learned by the world model. We introduce Dynamics-Informed Action Learning (DIAL) to bridge these two gaps. During training, rollout alignment exposes the forward model to its own autoregressive latent predictions, while multi-step inverse supervision trains a lightweight inverse dynamics head to infer actions from latent-state transitions. We further apply the inverse head to model-generated transitions with its parameters held fixed, encouraging predicted rollouts to retain action-relevant information. At test time, DIAL autoregressively couples the inverse and forward dynamics to construct a dynamics-informed action sequence toward the goal, which initializes the mean of the Cross-Entropy Method (CEM) proposal distribution. The subsequent CEM search then refines this initialization using the standard sampling and updating procedure. xperiments across four visual control benchmarks demonstrate that DIAL outperforms existing latent world models and planning baselines in both standard and long-horizon scenarios.

1 INTRODUCTION

World models enable agents to predict the consequences of their actions before execution Ha & Schmidhuber (2018); Hafner et al. (2019b;a), allowing them to learn from imagined rollouts rather than environment interactions. This paradigm has demonstrated strong performance across diverse control settings, with recent world-model agents achieving broad task coverage and performance surpassing human levels Hafner et al. (2023; 2025). In visual planning, latent world models bypass pixel-level reconstruction and operate directly in a compact latent space Maes et al. (2026). During training, forward dynamics models learn how actions transform the current latent state into a future state. At planning time, goal-conditioned control generates candidate action sequences and selects those whose predicted outcomes best match the goal Sun et al. (2026). However, this forward model defines a one-way, action-conditioned mapping from actions to state transitions: it predicts the resulting state given an action, but does not directly infer which action would induce a desired latent-state transition. This asymmetry creates a gap between learned dynamics and planning.

This asymmetry appears in two complementary ways. First, there is an inconsistency in the state inputs received by the model during the training and planning phases. During training, each prediction relies on latent states derived from encoded observation frames. Conversely, during planning, the model must autoregressively feed its own predictions into subsequent steps, causing errors to accumulate during the rollout process Li et al. (2026b). Consequently, accurate single-step prediction does not necessarily translate into reliable multi-step planning. Second, existing planning methods fail to fully leverage the action-conditioned dynamics learned by the world model when initializing the action search. Planners typically initialize candidate action sequences based on generic uniform or Gaussian distributions De Boer et al. (2005); Sun et al. (2026), without using this learned information to identify promising regions of the action space toward the goal. As a result, under limited search budgets or large goal offsets, such uninformed initialization forces early iterations to explore irrelevant regions of the action space, reducing search efficiency.

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.

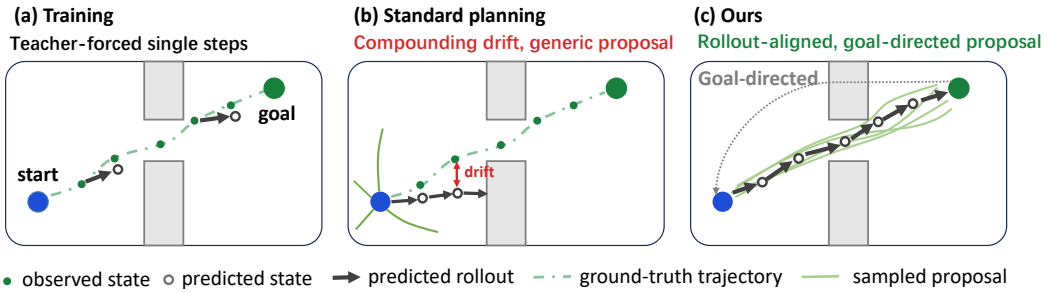

Figure 1: **DIAL aligns latent dynamics learning with goal-directed planning.** (a) During training, teacher forcing performs one-step prediction from observed latent states. (b) At planning time, autoregressive rollouts can accumulate prediction drift, while generic action proposals lack goal-directed initialization. (c) DIAL addresses both gaps by training on model-generated rollouts and using learned inverse dynamics to construct a dynamics-informed CEM proposal.

Existing approaches address these challenges mainly from the modeling and trajectory-evaluation perspectives. On the modeling side, methods such as DINO-WM Zhou et al. (2024), PLDM Sobal et al. (2022; 2025), and LeWM Maes et al. (2026) improve latent representations and dynamics modeling. On the trajectory-evaluation side, methods such as RC-aux Li et al. (2026b) and TRM Li et al. (2026a) improve candidate evaluation. These approaches, however, do not directly connect learned action-conditioned dynamics to the initialization of goal-conditioned action search. As illustrated in Fig. 1, autoregressive prediction and action-search initialization therefore remain two separate aspects of the planning pipeline.

To this end, we propose **D**ynamics-**I**nformed **A**ction **L**earning (**DIAL**) for latent world models. DIAL connects latent dynamics learning with autoregressive planning and uses learned inverse dynamics to inform the initialization of action search. During training, rollout alignment exposes the forward model to its own predicted latent states, while multi-step inverse supervision trains a lightweight inverse dynamics head to recover actions from latent-state transitions. We further apply the inverse head to model-generated transitions with the forward model held fixed, encouraging rollouts to preserve action-relevant information. During planning, the inverse head infers an action toward the goal from the current predicted latent state, and the forward model predicts the resulting latent state for the next step. The resulting action sequence initializes only the mean of the Cross-Entropy Method (CEM) proposal distribution, while the subsequent sampling and proposal-refitting procedure remains unchanged. DIAL therefore transfers action-conditioned information learned during world-model training into goal-directed search without introducing an additional planning stage or increasing the search budget.

We instantiate DIAL on LeWorldModel (LeWM), a compact reconstruction-free JEPA world model for goal-conditioned planning from pixels Maes et al. (2026). DIAL preserves the original LeWM backbone and CEM planning procedure, allowing us to isolate the effects of rollout-aligned training and dynamics-informed action initialization. We evaluate DIAL on the original LeWM task family under matched training and planning protocols, including different planning horizons, search budgets, and ablation settings. DIAL consistently improves goal-reaching performance, with particularly pronounced gains under limited search budgets and long-horizon goals. These results demonstrate the benefit of combining rollout-aligned latent dynamics with dynamics-informed action initialization for efficient latent planning. Our contributions can be summarized as:

1. We propose DIAL, a dynamics-informed framework that connects latent dynamics learning with goal-conditioned planning by using learned inverse dynamics to construct initial CEM proposal.
2. We introduce rollout alignment and multi-step inverse supervision to improve autoregressive dynamics and preserve action-relevant information in model-generated latent transitions.
3. We provide controlled empirical evaluation across multiple goal offsets, search budgets, and training ablations, demonstrating consistent improvements in goal-reaching performance, particularly under limited search budgets and long-horizon planning.

2 Related Work

World models for representation. World models aim to learn predictive models of environment dynamics from data, enabling agents to reason about future states through imagined rollouts. Early approaches rely on generative models that explicitly model future observations, such as IRIS Micheli et al. (2022), DIAMOND Alonso et al. (2024), and DreamerV4 Hafner et al. (2025). More recently, Joint-Embedding Predictive Architectures (JEPAs) have emerged as a reconstruction-free alternative, learning predictive dynamics directly in a compact latent space without explicit pixel reconstruction. Within action-conditioned latent world modeling, JEPA-based methods differ in how representations are learned. Methods based on frozen pretrained encoders, such as DINO-WM Zhou et al. (2024), constrain the learned state space through pre-trained visual representations. End-to-end approaches such as PLDM Sobal et al. (2022; 2025), while capable of jointly learning representations and dynamic information from raw observation data, may face training instability issues. LeWM Maes et al. (2026) addresses this challenge with a stable end-to-end framework that combines latent prediction with representation regularization. Our method is based on the end-to-end JEPA framework and investigates how to leverage the learned dynamic model to improve planning.

Candidate evaluation. Beyond improving learned representations and dynamics models, another line of work improves planning by refining the objective used to evaluate candidate trajectories. Fast-LeWM Gao & Xu (2026) introduces a self-consistency score during candidate selection to filter out dynamical hallucinations, while RC-aux Li et al. (2026b) and TRM Li et al. (2026a) augment the terminal cost with reachability-aware metrics or temporal distance constraints. These approaches primarily improve the evaluation and ranking of generated candidates, while leaving the proposal distribution unchanged. Consequently, when the search budget is limited or the task requires substantial exploration, the quality of generated candidate set may become a bottleneck that cannot be resolved solely through improved candidate evaluation.

Inverse dynamics and action priors. Learned inverse dynamics and action priors have also been explored for visual planning. GC-IDM Nguyen et al. (2026), for example, uses inverse dynamics to directly generate goal-conditioned actions, making planning dependent on the accuracy of the learned action predictor. PRISM Wang et al. (2026) learns a goal-conditioned Gaussian prior separately from a frozen world model, leaving the action prior decoupled from the learned dynamics. INTACT Sun et al. (2026) incorporates learned intent-to-action mappings into CEM-based Rubinstein & Kroese (2004) planning, primarily using the learned mapping for direct action generation or local search verification. In contrast, DIAL jointly learns inverse dynamics with the latent world model and uses the resulting dynamics-informed action sequence to initialize the CEM proposal mean, while retaining CEM for subsequent candidate sampling and refinement.

3 Preliminaries

Latent world model. Following LeWM Maes et al. (2026), we consider a reconstruction-free latent world model consisting of a visual encoder $E_\phi : \mathcal{O} \to \mathbb{R}^D$ and an action-conditioned latent dynamics model F_θ. The encoder maps observations to latent states:

$$z_t = E_\phi(o_t). \tag{1}$$

The dynamics model predicts the next latent state from a temporal context of latent states and actions:

$$\hat{z}_{t+1} = F_\theta(z_{t-h+1:t}, a_{t-h+1:t}), \tag{2}$$

where h denotes the context length and a_t is the current action used to predict $\hat{z}_{t+1}$. Following the reconstruction-free formulation of LeWM, the model is trained to predict future latent states from observed trajectories together with an anti-collapse regularizer. At planning time, the learned dynamics model is applied autoregressively to predict the consequences of candidate action sequences.

Goal-conditioned planning. We consider finite-horizon goal-conditioned control from visual observations. At time t, the agent observes o_t and is given a goal observation o_g, with corresponding latent states z_t and z_g. Given the current latent context and goal latent, planning seeks an action

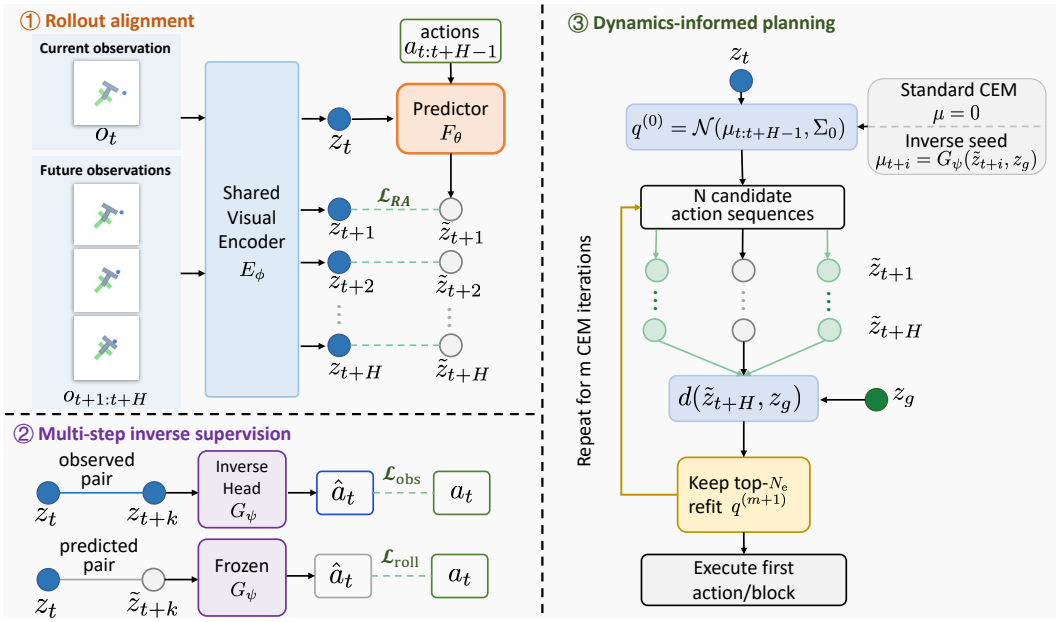

Figure 2: Overview of DIAL. **(1) Rollout alignment** trains the latent dynamics predictor F_θ on autoregressively generated latent contexts to reduce the mismatch between training and planning. **(2) Multi-step inverse supervision** trains the inverse-dynamics head G_ψ on observed latent state pairs and uses a frozen inverse head $G_{\bar{\psi}}$ to encourage model-generated states to preserve action-relevant information. **(3) Dynamics-informed planning** autoregressively couples G_ψ and F_θ to generate a goal-directed action sequence, which initializes only the mean of the CEM proposal.

sequence whose predicted terminal latent is closest to the goal:

$$a^*_{t:t+H-1} \in \arg\min_{a_{t:t+H-1}} d(\tilde{z}_{t+H}, z_g)\,, \qquad \tilde{z}_{t+H} = F_\theta^{(H)}(z_{t-h+1:t}, a_{t-h+1:t+H-1})\,, \tag{3}$$

where $d(\cdot,\cdot)$ denotes the latent goal-matching cost and $F_\theta^{(H)}$ denotes the autoregressive H-step rollout of F_θ. The historical actions $a_{t-h+1:t-1}$ are fixed context, while the future actions $a_{t:t+H-1}$ are optimization variables. Planning is typically implemented using sampling-based Model Predictive Control methods, such as the Cross-Entropy Method (CEM) Rubinstein & Kroese (2004). CEM samples candidate action sequences from a Gaussian proposal distribution, evaluates their terminal latent costs, selects the top N_e elite sequences, and refits the proposal distribution using these elites. This process is repeated for a fixed number of iterations, after which the resulting action sequence is executed in a receding-horizon manner.

4 Method

4.1 Overview

We propose Dynamics-Informed Action Learning (DIAL) for latent world models, a framework that aligns latent dynamics learning with autoregressive planning and leverages learned inverse dynamics to initialize action search. As illustrated in Fig. 2, DIAL comprises a visual encoder E_ϕ, an action-conditioned latent predictor F_θ, and a lightweight inverse dynamics head G_ψ. The encoder maps observations to D-dimensional latent states, while the predictor F_θ forecasts future latent states based on the current latent state and action context. The proposed inverse dynamics head G_ψ is responsible for estimating the action connecting a pair of latent states.

Building on the base world-model objective, DIAL introduces three complementary training terms. First, rollout alignment trains F_θ on autoregressively generated latent contexts, reducing the mismatch between teacher-forced training and planning-time rollouts. Second, multi-step inverse supervision trains G_ψ to recover actions from observed latent state pairs. Third, inverse supervision on predicted states applies a frozen copy of the inverse head to model-generated latent pairs, encouraging autoregressive rollouts to preserve action-relevant information. At planning time, DIAL uses

G_ψ and F_θ autoregressively to construct a goal-directed action sequence. Starting from the current latent state, G_ψ estimates an action toward the goal from the current latent state and goal latent, and F_θ advances the latent state under the predicted action. The resulting sequence initializes only the mean of the CEM proposal distribution, while CEM retains its standard sampling, evaluation, elite selection, proposal refitting, and search budget.

4.2 Rollout-Aligned Latent Training

Rollout alignment. The base prediction objective trains F_θ with teacher forcing, where each prediction is conditioned on latent states encoded from observed frames. During planning, however, F_θ is applied autoregressively, so its own predictions become inputs to subsequent steps. To align these two input regimes, we train F_θ over an autoregressive rollout of depth L. Starting from the observed latent context $\tilde{z}_{t-h+1:t} = z_{t-h+1:t}$, we generate an autoregressive rollout by feeding each predicted latent state back into the predictor. Specifically, the j-th predicted state is obtained as:

$$\tilde{z}_{t+j} = F_\theta(\tilde{z}_{t+j-h:t+j-1}, a_{t+j-h:t+j-1}), \qquad j = 1, \dots, L, \tag{4}$$

The first prediction is therefore conditioned entirely on observed latents, while subsequent input windows progressively replace observed states with model predictions. For $j > h$, the predictor is conditioned entirely on model-generated latents. We then minimize the rollout alignment loss as:

$$\mathcal{L}_{\mathrm{RA}} = \frac{1}{L} \sum_{j=1}^{L} \|\tilde{z}_{t+j} - \mathrm{sg}[z_{t+j}]\|_2^2, \tag{5}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. It is applied only to the target latents, while gradients propagate through the complete autoregressive recursion. Thus, $\mathcal{L}_{\mathrm{RA}}$ directly trains F_θ on the model-generated inputs encountered during planning.

Multi-step inverse supervision. Rollout alignment improves the reliability of autoregressive states, but does not explicitly require these states to preserve information about the actions that caused them. We therefore introduce a lightweight inverse dynamics head G_ψ to recover the initiating action from a latent-state transition. For an ordered transition from z_t to z_{t+k}, we predict

$$\hat{a}_t = G_\psi(z_t, z_{t+k}), \qquad k = 1, \dots, K_{\mathrm{obs}}, \tag{6}$$

where the supervision target is always the first action a_t. Thus, for $k > 1$, the head recovers the action initiating the transition rather than the complete intervening action sequence. We train G_ψ on latent-state pairs encoded from observed trajectories:

$$\mathcal{L}_{\mathrm{obs}} = \sum_{k=1}^{K_{\mathrm{obs}}} \omega_k \, \|G_\psi(z_t, z_{t+k}) - a_t\|_2^2, \tag{7}$$

where $\omega_k \propto K_{\mathrm{obs}} - k + 1$ and $\sum_k \omega_k = 1$. This objective trains the inverse mapping while encouraging the shared latent representation to retain information about the initiating action.

We further impose the same recoverability constraint on autoregressively predicted states. Let $\tilde{z}_{t+k}$ denote a state generated by the rollout in Eq. (4). We reuse G_ψ on these model-generated pairs while holding its parameters fixed:

$$\mathcal{L}_{\mathrm{roll}} = \frac{1}{K_{\mathrm{roll}}} \sum_{k=1}^{K_{\mathrm{roll}}} \|G_\psi(\mathrm{sg}[z_t], \tilde{z}_{t+k}) - a_t\|_2^2. \tag{8}$$

During this loss, the parameters of G_ψ are held fixed, so gradients flow through its latent inputs and the autoregressive rollout rather than through the inverse mapping itself. The anchor latent is also detached to prevent this loss from directly updating the first input of G_ψ. Consequently, $\mathcal{L}_{\mathrm{obs}}$ learns action recovery from observed transitions, while $\mathcal{L}_{\mathrm{roll}}$ encourages model-generated states to preserve the same action-relevant information.

Overall training objective. In addition to the objectives above, we retain the reconstruction-free latent prediction objective of LeWM Maes et al. (2026):

$$\mathcal{L}_{\text{pred}} = \|F_\theta(z_{t-\ell+1:t}, a_{t-\ell+1:t}) - z_{t+1}\|_2^2 . \tag{9}$$

The predictor accepts variable-length contexts, and this objective is additionally averaged over all valid context lengths $1 \le \ell \le h$ rather than supervising only the longest one. Following LeWM, we additionally employ the Sketch Isotropic Gaussian Regularizer (SIGReg) to prevent representation collapse:

$$\mathcal{L}_{\text{WM}} = \mathcal{L}_{\text{pred}} + \lambda_{\text{sig}}\mathcal{L}_{\text{SIG}}. \tag{10}$$

Combining this objective with the DIAL-specific terms yields the full training objective:

$$\mathcal{L}_{\text{DIAL}} = \mathcal{L}_{\text{WM}} + \beta_{\text{RA}}\mathcal{L}_{\text{RA}} + \beta_{\text{obs}}\mathcal{L}_{\text{obs}} + \beta_{\text{roll}}\mathcal{L}_{\text{roll}}, \tag{11}$$

where β_{RA}, β_{obs}, and β_{roll} control the weights of the three DIAL-specific training terms.

4.3 Dynamics-Informed Planning

Standard CEM initializes its proposal with a goal-agnostic Gaussian distribution $q^{(0)} = \mathcal{N}(\mathbf{0}, \Sigma_0)$. At planning time, DIAL uses the learned inverse dynamics head to construct a dynamics-informed proposal sequence. Starting from the current latent state $\tilde{z}_t = z_t$, the inverse head recursively predicts an action conditioned on the current predicted state and the goal latent, and the forward model advances the latent state under the predicted action:

$$\begin{aligned} \mu_{t+i} &= G_\psi(\tilde{z}_{t+i}, z_g)\,, \\ \tilde{z}_{t+i+1} &= F_\theta(\tilde{z}_{t+i}, \mu_{t+i})\,, \qquad i = 0, \dots, H-1. \end{aligned} \tag{12}$$

For notational simplicity, Eq. (12) writes F_θ with a single state and action, while in practice the predictor retains the most recent h latent states and actions, following the same convention as during training. Each action is therefore inferred from the latest predicted latent state, producing a goal-conditioned proposal sequence $\mu_{t:t+H-1}$. Because z_g may lie beyond the temporal offsets used to train G_ψ, this sequence is not assumed to be an accurate open-loop plan. This sequence is used only to initialize the mean of the CEM proposal:

$$q^{(0)} = \mathcal{N}(\mu_{t:t+H-1}, \Sigma_0)\,, \tag{13}$$

where the initial covariance Σ_0 remains unchanged. CEM then proceeds with its standard sampling, model rollout, elite selection, and proposal refitting steps. The inverse-decoded sequence only biases the initial search distribution and is never executed directly. Subsequent CEM iterations can therefore correct errors in the initialization.

5 Experiments

5.1 Experimental Setup

Environments and datasets. We follow the evaluation protocol of LeWM Maes et al. (2026) and use the same offline datasets, observation preprocessing, and goal-conditioned planning setup. The environments include TwoRoom Sobal et al. (2025), PushT Zhou et al. (2024), Reacher Tassa et al. (2018), and Cube Park et al. (2025). Each episode provides an initial observation and a goal image. We report success rate as the primary metric. We use a standard goal offset of 25 steps and additionally introduce an offset-50 setting to evaluate planning over a substantially larger temporal distance. Within each evaluation setting, all methods use the same interaction and evaluation protocol.

Planning protocol. Following prior work Maes et al. (2026); Gao & Xu (2026); Li et al. (2026b), we use an action block size of 5 and a planning horizon of $H = 5$, so each plan spans 25 primitive environment steps. An offset of 25 can therefore be addressed within a single plan, whereas an offset of 50 requires replanning while keeping the per-plan horizon unchanged. Importantly, the long-horizon setting does not increase the CEM search budget or planning horizon. The dynamics-informed sequence is generated once to initialize the CEM proposal mean and does not introduce an additional CEM iteration or candidate rollout. Unless otherwise stated, all methods use the same CEM sampling budget, update rule, and replanning schedule within each evaluation setting.

Table 1: **Success rates (%) at the standard 25-step and long-horizon 50-step goal offsets.** Offset-25 results are quoted from the corresponding papers unless otherwise noted. Offset-50 results are evaluated using a unified harness with identical episodes ($n = 50$) and the same CEM candidate budget across methods. Bold and underline indicate the best and second-best results.

Method	Offset 25					Offset 50				
	PushT	Cube	Reacher	TwoRoom	Macro	PushT	Cube	Reacher	TwoRoom	Macro
PLDM Sobal et al. (2022; 2025)	78.0	65.0	78.0	97.0	79.5	20.0	46.0	66.0	78.0	52.5
DINO-WM Zhou et al. (2024)	74.0	86.0	79.0	**100.0**	84.8	14.0	46.0	20.0	<u>94.0</u>	43.5
LeWM Maes et al. (2026)	<u>96.0</u>	74.0	86.0	87.0	85.8	60.0	64.0	72.0	56.0	63.0
RC-aux Li et al. (2026b)	90.8	76.0	87.2	<u>98.0</u>	88.0	**64.0**	54.0	82.0	54.0	63.5
Fast-LeWM Gao & Xu (2026)	<u>96.0</u>	80.0	<u>88.0</u>	<u>98.0</u>	90.5	46.0	58.0	80.0	50.0	58.5
Qantara Rakhimov et al. (2026)	90.1	93.7	80.9	**100.0**	91.2	40.0	<u>68.0</u>	78.0	84.0	67.5
Fast-LeWM-SC Gao & Xu (2026)	**98.0**	82.0	**90.0**	<u>98.0</u>	92.0	50.0	62.0	78.0	50.0	60.0
INTACT (Pure CEM) Sun et al. (2026)	88.4	68.4	83.7	82.9	80.9	54.0	46.0	72.0	38.0	52.5
INTACT (Actor CEM) Sun et al. (2026)	93.6	**96.9**	86.6	<u>98.0</u>	<u>93.8</u>	<u>62.0</u>	48.0	<u>86.0</u>	**96.0**	<u>73.0</u>
Ours	**98.0**	<u>96.0</u>	**90.0**	**100.0**	**96.0**	<u>62.0</u>	**70.0**	**88.0**	92.0	**78.0**

Baselines. We compare DIAL against prior methods that modify different components of the same goal-conditioned latent planning pipeline. Specifically, PLDM Sobal et al. (2022; 2025), DINO-WM Zhou et al. (2024), LeWM Maes et al. (2026), Fast-LeWM Gao & Xu (2026), and Qantara Rakhimov et al. (2026) modify the learned representations or the forward dynamics. RC-aux Li et al. (2026b) modifies the candidate re-scoring mechanism. INTACT Sun et al. (2026) replaces the action proposal distribution with learned action mappings or priors. Among these, LeWM is the closest point of comparison because it shares the same backbone architecture and planning protocol. Where official reports are available, baseline results are cited directly from the original papers. For settings not reported in the original publications, we evaluate methods using officially released checkpoints or retrain them using publicly available implementations, adhering strictly to the standardized evaluation protocol for each benchmark.

5.2 Main Results

Comparison at the standard 25-step offset. The left part of Table 1 summarizes performance at the standard 25-step offset. DIAL achieves the highest macro success rate of 96.0%. Compared with prior-free LeWM Maes et al. (2026), DIAL improves the macro score by 10.2 points, with particularly large gains of 22.0 and 13.0 points on Cube Park et al. (2025) and TwoRoom Sobal et al. (2025), respectively. Compared with Fast-LeWM Gao & Xu (2026) and RC-aux Li et al. (2026b), which improve candidate evaluation while retaining prior-free proposal initialization, DIAL further improves the macro score by 5.5 and 8.0 points, respectively. Compared with INTACT Sun et al. (2026), DIAL exceeds its pure CEM and actor CEM variants by 15.1 and 2.2 points, respectively. The former uses a generic Gaussian proposal, while the latter introduces a separately learned actor for action initialization. DIAL derives its initialization directly from the jointly trained forward and inverse dynamics, without introducing a separate policy training stage.

Comparison at the 50-step long-horizon offset. We further evaluate long-horizon planning by doubling the goal offset from 25 to 50 steps while keeping the evaluation and planning protocols unchanged, as shown in the right part of Table 1. DIAL achieves the highest macro success rate of 78.0%, outperforming Qantara Rakhimov et al. (2026) and LeWM Maes et al. (2026) by 10.5 and 15.0 points, respectively. The largest gain appears on TwoRoom, where DIAL improves over LeWM by 36.0 points. On Cube Park et al. (2025), DIAL also exceeds the INTACT actor CEM variant Sun et al. (2026) by 22.0 points. In contrast, the benefit of improving candidate evaluation alone is limited at this longer horizon. For example, Fast-LeWM-SC Gao & Xu (2026) improves over Fast-LeWM by only 1.5 points. These results indicate that as the temporal distance to the goal increases, providing CEM with a goal-conditioned initial proposal becomes increasingly important for effective search under a fixed planning budget.

Qualitative visualization. We provide a qualitative comparison of LeWM Maes et al. (2026), INTACT Sun et al. (2026), and DIAL in Fig. 3. The figure shows representative long-horizon rollouts on Cube Park et al. (2025) (left) and PushT Zhou et al. (2024) (right) under the 50-step goal-offset setting. For each task, the first column shows the initial observation, the intermediate columns show

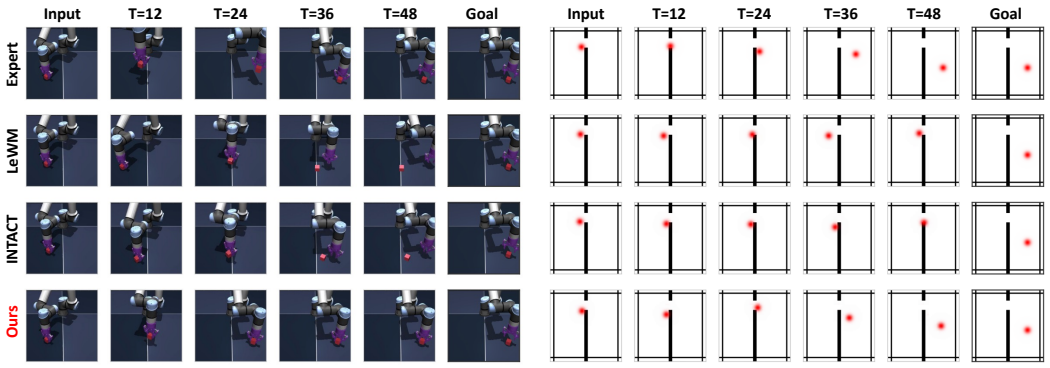

Figure 3: **Visualization of planning trajectories on Cube and PushT.** Compared with LeWM and INTACT, DIAL maintains object interaction and makes greater progress toward the goal, while the other methods either lose object interaction or terminate far from the target.

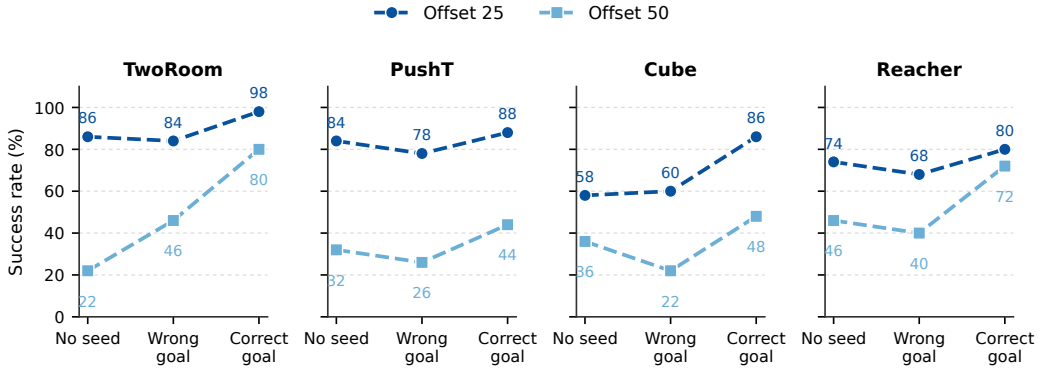

Figure 4: **Effect of goal consistency in proposal initialization.** Success rates (%) for uninitialized, mismatched-goal, and matched-goal proposals across environments and goal offsets under the constrained search budget (30×10, $n = 50$ episodes per setting).

the predicted rollout, and the final column shows the goal observation. LeWM and INTACT either lose effective object interaction or make insufficient progress toward the goal, resulting in rollouts that terminate far from the target. In contrast, DIAL maintains object interaction and makes greater progress toward the goal, consistent with its improved long-horizon performance in Table 1.

5.3 Ablation Study

Effect of goal consistency in proposal initialization. To determine whether the benefit of DIAL comes from goal-conditioned initialization rather than simply shifting the proposal distribution away from the default prior, we replace the target latent with one from a different episode while keeping the inverse dynamics module and CEM procedure unchanged. We evaluate this control under the constrained 30×10 search budget. As shown in Fig. 4, correctly matched goal proposals consistently outperform mismatched proposals across all eight environment and offset settings, with an average gap of 21.5 percentage points. In most settings, mismatched proposals provide little or no improvement over uninitialized CEM and can even degrade performance. These results indicate that the benefit of proposal initialization depends on its consistency with the target goal, rather than simply on shifting the proposal mean away from the default prior. The target latent therefore provides essential goal-specific information for constructing an informative initial proposal for CEM.

Dynamics-informed initialization under constrained budgets. We evaluate dynamics-informed proposal initialization across search budgets and goal offsets (Fig. 5). At the 25-step offset, initialization improves mean success by 13.5 percentage points under the low budget (30×10), compared with 3.5 points under the full budget (300×30), indicating that its benefit is more pronounced when CEM has fewer opportunities to recover through exploration. The effect grows substantially at the 50-step offset with the planning horizon fixed at $H = 5$, where the mean gains increase to 27.0 and

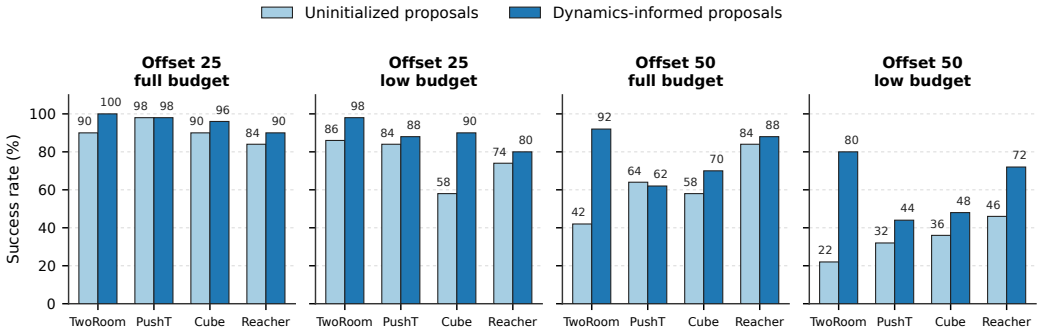

Figure 5: **Dynamics-informed initialization under constrained budgets.** Success rates (%) across environments for uninitialized and dynamics-informed proposals under different goal offsets and search budgets. Full and low budgets correspond to 300×30 and 30×10 candidate rollouts.

Table 2: **Loss ablation at the 25-step goal offset under unseeded CEM.** Each variant is trained from scratch with identical architecture and optimization schedule, and evaluated with the full 300×30 budget. Success rates (%) are reported for different combinations of training objectives.

$\mathcal{L}_{\text{WM}}$	$\mathcal{L}_{\text{RA}}$	$\mathcal{L}_{\text{obs}}$	$\mathcal{L}_{\text{roll}}$	PushT	Cube	Reacher	TwoRoom	Macro
✓				**96.0**	72.0	80.0	84.0	83.0
✓	✓			94.0	70.0	88.0	82.0	83.5
✓		✓		90.0	80.0	90.0	80.0	85.0
✓	✓	✓		94.0	**84.0**	82.0	**86.0**	86.5
✓	✓	✓	✓	**96.0**	**84.0**	**92.0**	**86.0**	**89.5**

16.0 points under the low and full budgets, respectively. In particular, TwoRoom improves from 22% to 80% under the low budget. These results indicate that dynamics-informed initialization becomes increasingly valuable when the search budget is constrained or the goal is temporally distant.

Loss ablation for model training. To assess the contribution of each training objective, Table 2 reports a controlled ablation at the 25-step goal offset. Starting from the teacher-forced baseline ($\mathcal{L}_{\text{WM}}$), adding rollout alignment ($\mathcal{L}_{\text{RA}}$) provides only a marginal improvement in macro success, from 83.0% to 83.5%. In contrast, multi-step inverse supervision ($\mathcal{L}_{\text{obs}}$) increases macro success to 85.0%, including an 8.0-point gain on Cube. Combining the two objectives further raises macro success to 86.5%. However, $\mathcal{L}_{\text{obs}}$ enforces action recoverability only on ground-truth transitions and does not directly constrain model-generated rollouts. $\mathcal{L}_{\text{roll}}$ addresses this limitation by applying the learned inverse head to autoregressively predicted latent pairs, encouraging model-generated transitions to preserve action-relevant information. Adding $\mathcal{L}_{\text{roll}}$ yields the highest macro success of 89.5%, with the best or tied-best performance on all four tasks: 96.0% on PushT, 84.0% on Cube, 92.0% on Reacher, and 86.0% on TwoRoom. The results show that preserving action recoverability on both ground-truth transitions and model-generated rollouts is important for effective planning.

6 Conclusion

We present Dynamics-Informed Action Learning (DIAL), a framework that connects latent dynamics learning with goal-conditioned planning through rollout-aligned training and learned inverse dynamics. DIAL trains the latent dynamics model on autoregressive rollouts while learning an inverse dynamics head to preserve action-relevant information in both observed and model-generated transitions. At planning time, the learned inverse and forward dynamics are coupled to construct a dynamics-informed action sequence that initializes the mean of the CEM proposal, while the subsequent search procedure remains unchanged. Across four visual planning benchmarks, DIAL improves goal-reaching performance at both standard 25-step and extended 50-step goal offsets, with particularly large gains under constrained search budgets. Overall, these results show that action-conditioned information learned during latent dynamics training can be transferred to guide goal-conditioned search, improving planning without modifying the CEM procedure.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and editing paper text, refactoring experimental analysis scripts, and preparing presentation materials. We have not used generative AI tools to invent experimental results, fabricate citations, or run unaudited evaluation pipelines. Generated text was manually checked against measured numbers and the literature cited in Sections 2–5; analysis code that produced reported tables was verified by comparing outputs to logged evaluation artefacts. We take responsibility for the final content of this work, including text, claims, and artefacts produced with the aid of generative AI.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 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.

Yuntian Gao and Xiangyu Xu. Fast leworldmodel. *arXiv preprint arXiv:2606.26217*, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

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*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Liangyu Li, Shengzhi Wang, and Qingwen Liu. Beyond euclidean proximity: Repairing latent world models with horizon-matched trajectory reachability metrics. *arXiv preprint arXiv:2605.22164*, 2026a.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: Rc-aux for latent world models. *arXiv preprint arXiv:2605.07278*, 2026b.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Hoang Nguyen, Xiaohao Xu, and Xiaonan Huang. Latent geometry beyond search: Amortizing planning in world models. *arXiv preprint arXiv:2605.08732*, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

Ruslan Rakhimov, George Bredis, Yuriy Maksyuta, and Daniil Gavrilov. Qantara: Bridge-flow training for multi-paradigm jepa control. *arXiv preprint arXiv:2607.04978*, 2026.

Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, volume 133. Springer, 2004.

Vlad Sobal, Jyothir SV, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features. *arXiv preprint arXiv:2211.10831*, 2022.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Stress-testing offline reward-free reinforcement learning: A case for planning with latent dynamics models. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

Junhan Sun, Hao Zhao, and Guofeng Zhang. Intact: Isomorphic intent-to-action learning for search-free world models. *arXiv preprint arXiv:2607.26056*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yuhai Wang, Jiawei Xia, Rongxuan Zhou, Xiao Hu, Yongliang Shi, Jing Du, and Yang Ye. Prism: Prior-guided imagination sampling in world models. *arXiv preprint arXiv:2606.07974*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

The appendix is organized as follows. Sec. A describes the model architecture, data preprocessing, optimization, loss configuration, and other implementation details for reproducing DIAL. Sec. B specifies the controlled evaluation protocols and search budgets. Sec. C provides additional qualitative comparisons, inverse-head diagnostics, rollout-depth analysis, and protocol sensitivity.

A Implementation Details

Architecture. DIAL is built on an action-conditioned LeWM backbone without a separate policy network. Each 224×224 image is encoded by a ViT-Tiny encoder with patch size 14, and its CLS token is projected to a $D = 192$-dimensional latent state. The action encoder maps each normalized action block to the same dimension using a 1×1 temporal convolution followed by a two-layer SiLU MLP with hidden width $4D = 768$. The latent predictor uses a causal context of at most $h = 3$ latent-action pairs. Its passive branch contains six transformer layers with 16 heads, head dimension 64, MLP width 2048, and dropout 0.1, while the action-conditioned residual branch contains three transformer layers with a rank-32 bottleneck.

The inverse dynamics head applies layer normalization to an ordered pair of latent states, followed by a 256-unit SiLU hidden layer and a linear projection to the action dimension. It contains 0.102M parameters. The predictor supports variable-length contexts up to $h = 3$ and predicts the next latent state at every valid position. All visual, action, dynamics, and inverse modules are optimized jointly.

Data preprocessing. Following prior work Maes et al. (2026); Gao & Xu (2026), we use the original offline datasets and success criteria of each benchmark. RGB observations are resized to 224×224 and normalized using ImageNet statistics. Actions are normalized using statistics computed from the corresponding dataset. Five consecutive primitive actions are concatenated into one action block, matching the frame skip of five. Unless otherwise specified, all experiments use clean observations without synthetic distractors.

Optimization and loss configuration. Table 3 summarizes the training configuration. Models are trained for 10 epochs on a single NVIDIA H200 GPU using AdamW with a learning rate of 5×10^{-5}, weight decay 10^{-3}, batch size 128, linear warmup followed by cosine annealing, and gradient clipping with maximum norm 1.0. We use a $90/10$ train/validation split. The normalized weights for the observed inverse loss are $(\omega_1, \omega_2, \omega_3) = (3, 2, 1)/6$. The predicted-transition inverse loss is disabled for the first three epochs and then applied with $k \leq K_{\text{roll}} = 2$. Unless otherwise specified, all DIAL-specific loss coefficients are set to $\beta_{\text{RA}} = \beta_{\text{obs}} = \beta_{\text{roll}} = 1$ across all environments.

Table 3: **DIAL training hyperparameters.** Values are shared across PushT, Cube, Reacher, and TwoRoom unless noted otherwise.

component	hyperparameter	value
input	image resolution / patch size	224×224 / 14
representation	latent dimension D	192
context	history h / rollout depth L	3 / 5
predictor	layers / heads / MLP width	6 / 16 / 2048
residual branch	layers / bottleneck rank	3 / 32
inverse head	depth / hidden width	2 / 256
optimization	epochs / batch size	10 / 128
optimization	optimizer / learning rate	AdamW / 5×10^{-5}
optimization	learning-rate schedule	linear warmup and cosine annealing
optimization	weight decay / gradient clip	10^{-3} / 1.0
optimization	precision / train split	bfloat16 / 0.9
SIGReg	weight / knots / projections	0.09 / 17 / 1024
rollout alignment	β_{RA} / L	1.0 / 5
observed inverse	β_{obs} / K_{obs}	1.0 / 3
predicted inverse	β_{roll} / K_{roll}	1.0 / 2
predicted inverse	warmup	first 3 epochs

B Comparison Protocol

The offset-25 results in Table 3 follow the reported planning protocols of the corresponding methods and therefore do not use a uniform search budget. We use these results as the standard-offset reference for comparison with prior work. For the controlled offset-50 evaluation, all search-based methods use the same CEM configuration, with 300 candidates per iteration, 30 iterations, 30 elites, and initial variance 1.0. All methods are evaluated on identical start-goal pairs under the same planning horizon and action block size. The low-budget setting used in the ablation studies reduces the search to 30 candidates per iteration, 10 iterations, and 3 elites. Apart from the search budget, the planning objective and evaluation protocol remain unchanged. This controlled setting isolates the effect of proposal initialization from differences in search budget. All methods use a planning horizon of $H = 5$, with each action block corresponding to five environment steps. The optimized sequence is executed before replanning. Unless otherwise specified, replanning therefore occurs every 25 environment steps.

C Additional Experimental Results

C.1 Qualitative Comparison

We provide additional qualitative comparisons under the 50-step goal-offset setting, including LeWM Maes et al. (2026), Fast-LeWM Gao & Xu (2026), Qantara Rakhimov et al. (2026), and INTACT (Pure CEM) Sun et al. (2026). As shown in Figs. 6 to 9, the first column shows the initial observation, the intermediate columns show the predicted rollout, and the final column shows the goal observation. Across the four environments, DIAL produces rollouts that remain more goal-directed over the long horizon, consistent with its improved performance under the 50-step offset.

C.2 Inverse-head Fidelity

Table 4 evaluates the inverse head G_ψ at the temporal gaps used to construct the dynamics-informed action sequence. At a five-step gap, the normalized ℓ_2 error ranges from 0.98 on TwoRoom to 4.61 on PushT, while R^2 is negative across all four environments. Increasing the gap to ten steps further degrades inverse prediction, with particularly low R^2 on Cube (-4.55) and PushT (-16.8). The predicted action magnitude also exceeds the normalized action bound in TwoRoom, Cube, and PushT, while remaining within the bound on Reacher. These results indicate that the inverse head is not a uniformly accurate action predictor, particularly at larger temporal gaps. Instead, its predictions provide a coarse goal-directed initialization that is subsequently refined by CEM.

Table 4: **Inverse-head fidelity at different temporal gaps.** Normalized ℓ_2 error, R^2, and maximum predicted action magnitude of G_ψ at five- and ten-step gaps across four environments.

	k=5			k=10		
env	nl2	R^2	$\|\hat{a}\|_\infty$	nl2	R^2	$\|\hat{a}\|_\infty$
TwoRoom	0.98	-0.02	1.3	1.07	-0.24	2.3
Reacher	1.09	-0.21	0.8	1.14	-0.32	0.8
Cube	1.94	-2.13	4.3	2.36	-4.55	5.2
PushT	4.61	-13.4	4.1	5.62	-16.8	4.5

C.3 Effect of Rollout Depth

We further examine the effect of the rollout-alignment depth L. Increasing L from 5 to 10 does not improve planning performance. On PushT at the 50-step goal offset, success decreases from 58% to 34%, approaching the performance of the model trained without rollout alignment. We therefore use $L = 5$ throughout the main experiments, matching the planning horizon.

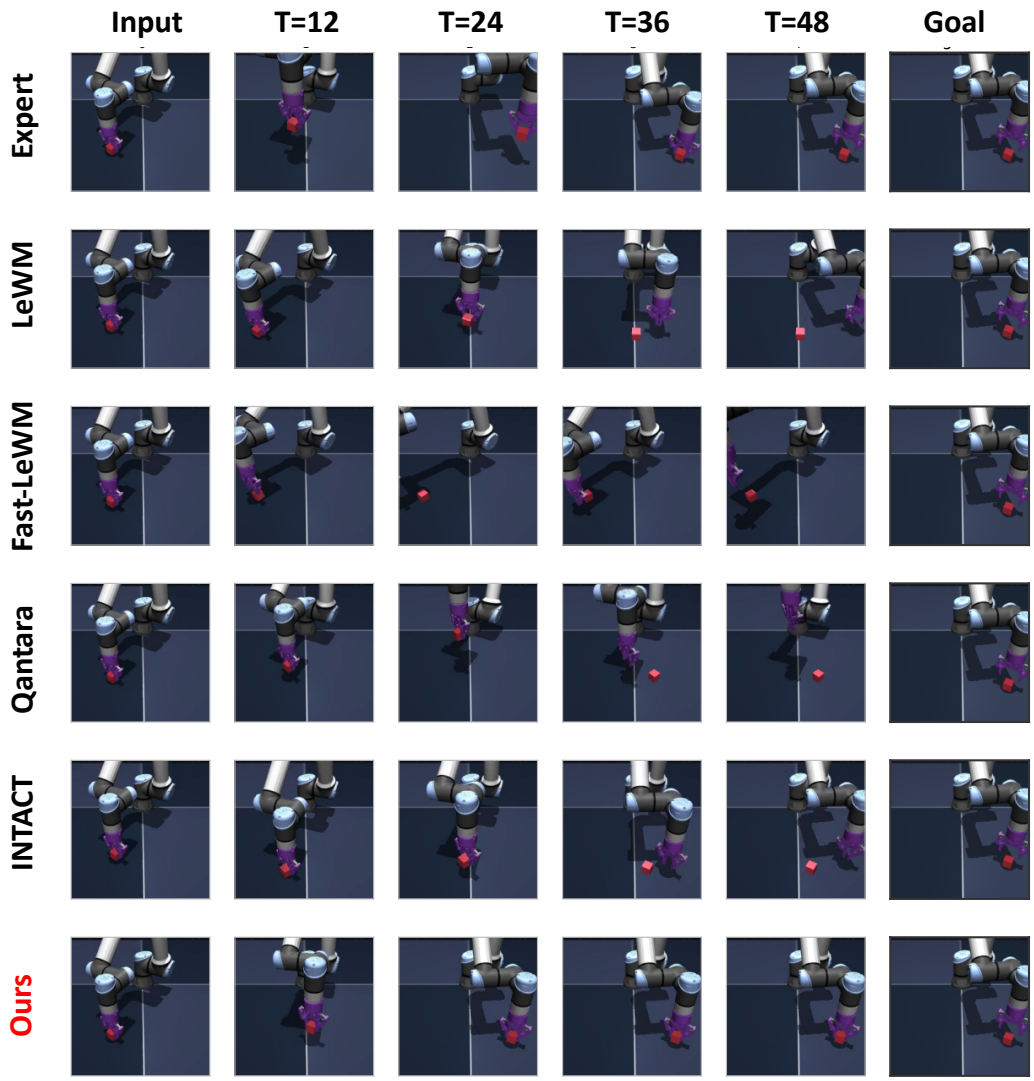

Figure 6: **Qualitative comparison on Cube under the 50-step goal offset.** The expert row shows the dataset trajectory from the same initial state to the given goal. LeWM, Fast-LeWM, Qantara, and INTACT (Pure CEM) fail to complete the required manipulation in this episode, while DIAL maintains contact with the cube throughout the rollout and reaches the target configuration.

Table 5: **Sensitivity to replanning interval and warm starting.** Macro success at the 50-step goal offset for standard CEM and dynamics-informed initialization under different replanning intervals. Entries are reported as standard CEM $\rightarrow$ dynamics-informed initialization.

r	warm start	TwoRoom	PushT	Cube	Reacher	macro
5	on	42 $\rightarrow$ 92	62 $\rightarrow$ 58	54 $\rightarrow$ 58	88 $\rightarrow$ 84	61.5 $\rightarrow$ 73.0
1	on	44 $\rightarrow$ 86	30 $\rightarrow$ 28	50 $\rightarrow$ 58	38 $\rightarrow$ 46	40.5 $\rightarrow$ 54.5
1	off	22 $\rightarrow$ 90	30 $\rightarrow$ 46	54 $\rightarrow$ 60	30 $\rightarrow$ 82	34.0 $\rightarrow$ 69.5

C.4 Protocol Sensitivity

Table 5 examines DIAL under different replanning intervals at the 50-step goal offset. With the default $r = 5$ setting and warm starting, macro success increases from 61.5% with standard CEM

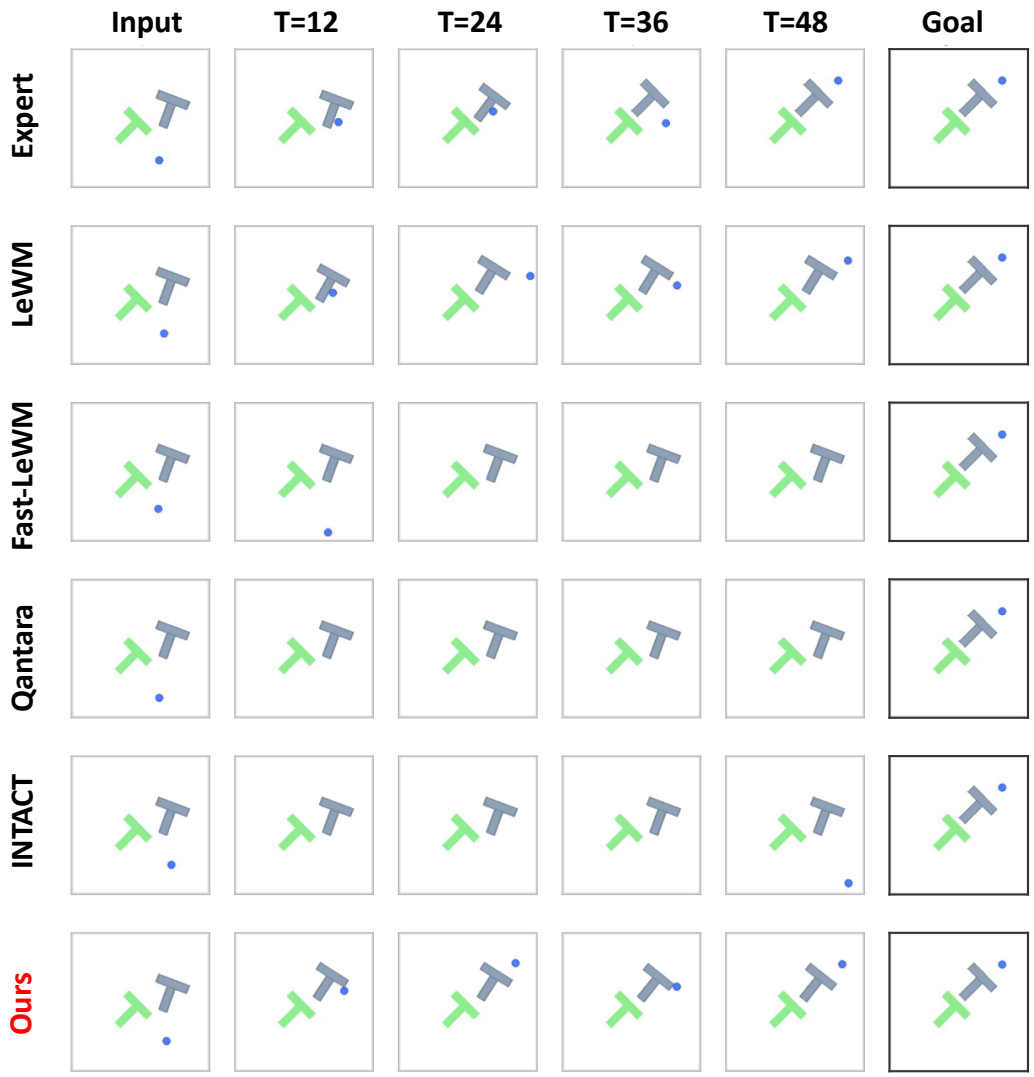

Figure 7: **Qualitative comparison on PushT under the 50-step goal offset.** The expert row shows the dataset trajectory for the same initial state and goal. The four external baselines fail to reach the target arrangement over the long-horizon rollout, whereas DIAL maintains effective object interaction and progressively moves the object toward the desired configuration.

to 73.0% with dynamics-informed initialization. Replanning after every action block ($r = 1$) reduces macro success to 40.5% with standard CEM and 54.5% with dynamics-informed initialization. When warm starting is disabled at $r = 1$, macro success further decreases to 34.0% with standard CEM, while dynamics-informed initialization achieves 69.5%. These results show that the benefit of dynamics-informed initialization remains substantial even without carrying the previous solution into the next planning step. They also indicate that more frequent replanning does not necessarily improve performance under a fixed planning budget, supporting the default $r = 5$ protocol used in our experiments.

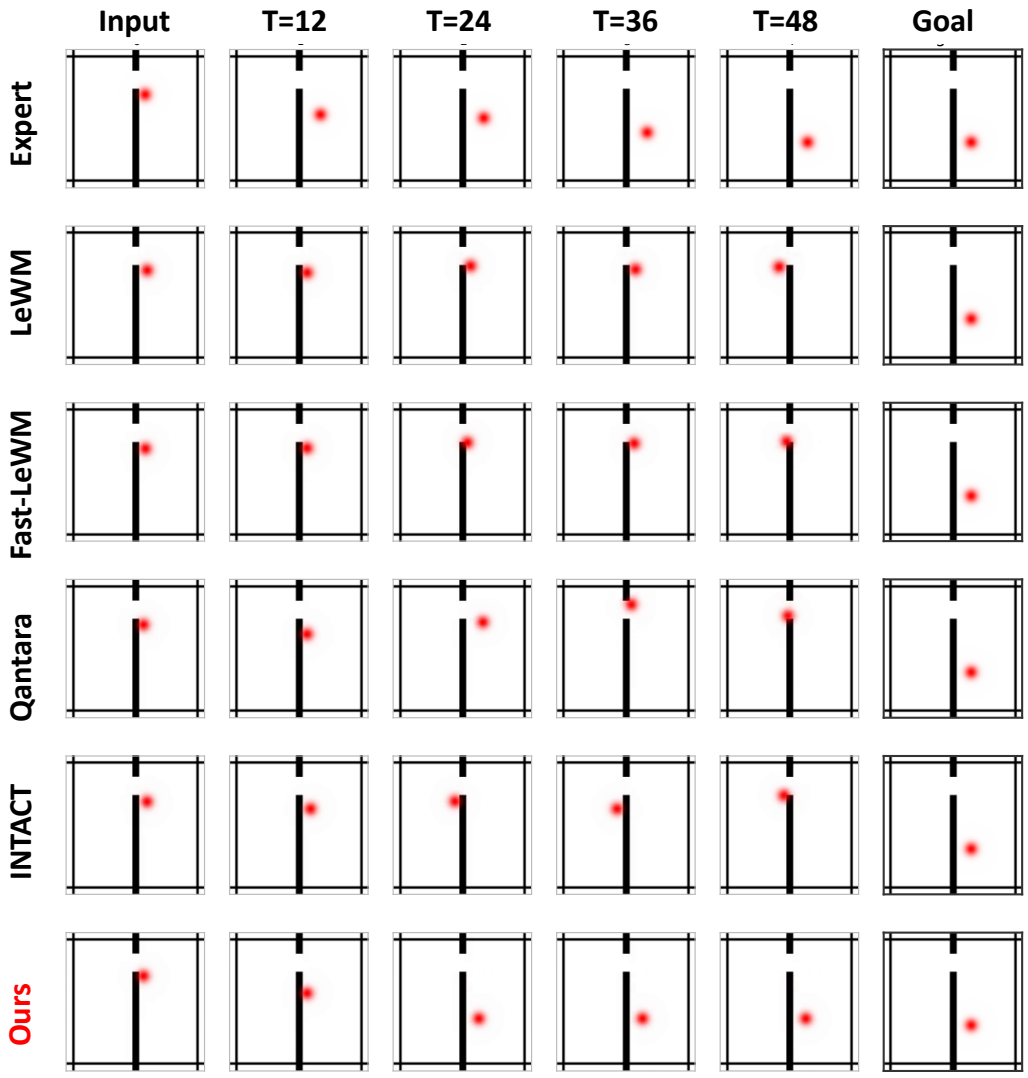

Figure 8: **Qualitative comparison on TwoRoom under the 50-step goal offset.** The expert row shows the dataset trajectory for the same initial state and goal. The external baselines remain in the starting room and fail to reach the goal region, whereas DIAL successfully passes through the doorway and continues toward the target room.

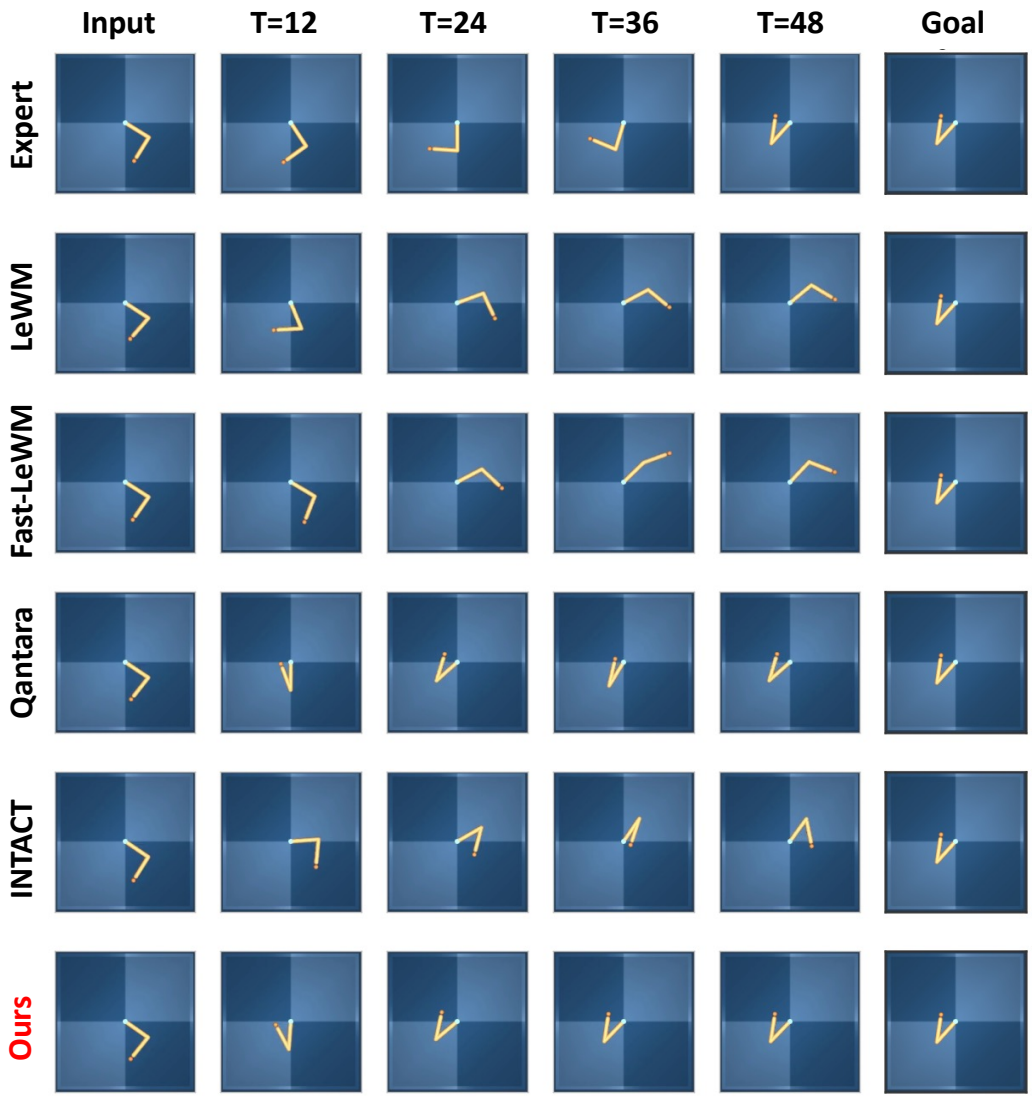

Figure 9: **Qualitative comparison on Reacher under the 50-step goal offset.** The expert row shows the dataset trajectory for the same initial state and goal. LeWM, Fast-LeWM, Qantara, and INTACT (Pure CEM) fail the success criterion on this episode, while DIAL produces a rollout that reaches the target pose.