PREDICTION IS NOT CONTROL: EQUIVARIANCE ENABLES PLANNING IN WEIGHT-SPACE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A world model lets an agent plan by imagining how its environment responds to actions. For reward-free, goal-conditioned planning, such models arrive by two routes: large models pretrained on web-scale data, which control well but are opaque, and from-scratch functional models whose state is the weights of an implicit neural representation, inspectable and pretraining-free but not yet shown to plan. We show the second route fails not on prediction but on controllability; over the same INR-weight state, non-equivariant and latent-code operators match an equivariant operator's rollout PSNR yet plan no better than doing nothing. We trace this to symmetry where only a permutation-equivariant operator learns an action map that survives the weight-space gauge and can be inverted by a planner. We distill this into an offline diagnostic, the action-response, that predicts controllability at $r{=}0.91$. Composed with a classical planner on occupancy read off its state, the equivariant model is competitive-to-ahead of a pretrained state-of-the-art world model in distribution and more than $2\times$ ahead out of distribution, from scratch and without reward. Controllability, not prediction, is what makes a world model useful for planning.

1 INTRODUCTION

A world model lets an agent plan by imagining how its environment will respond to actions. We study the *reward-free, goal-conditioned* setting, where the task is to reach an arbitrary target observation, not the reinforcement-learning sense of a model optimized for reward on a fixed task (Hafner et al., 2019; Schrittwieser et al., 2020; Hansen et al., 2024). Within this setting there are two roads to a planning world model. One is a large *pretrained latent* model: DINO-WM (Zhou et al., 2024b) plans to a goal over frozen DINOv2 features and controls well, but its state is opaque and it inherits its representation from web-scale pretraining. The other is a *from-scratch functional* model, whose state is the *weights* $\boldsymbol{\theta}$ of an implicit neural representation (INR) fit to each observation: it needs no pretraining and is *inspectable*, rendering back to the image and exposing structure such as occupancy. This second road is attractive but has not been established for planning.

The obstacle is not prediction quality. The natural instantiations of the functional road predict the observed frames about as well as any operator we test, yet plan no better than doing nothing. A world model is useful for planning only if it can be *controlled*: a planner interrogates it counterfactually ("what observation results under action a?") and then *inverts* it to recover the action that reaches a desired observation, so it depends on the model's response to *counterfactual* actions. Prediction quality (rollout PSNR, video-prediction fidelity), the field's default proxy, instead measures behavior under the *observed* actions, and the two can diverge; a model can predict well while responding to actions weakly or in the wrong direction. This separation of controllability from predictive fit is long-standing in control theory (Kalman et al., 1960; Åström & Murray, 2008). Nonetheless, learned world models are typically evaluated by prediction fidelity or downstream task success, and the model's controllability is rarely isolated as a criterion of its own. Our contribution is not this distinction, which is classical, but showing where and why it breaks in a concrete model class, and supplying a diagnostic and a planning method that fix it.

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.

Prediction accuracy does not guarantee a planner-ready world model, and the deciding factor is the symmetry of the operator. The functional road makes the "predict vs. control" question sharp and lets us pin down the mechanism. The weight state carries a large neuron-permutation *gauge* symmetry; two INRs of the same scene differ by a relabeling of neurons. A non-equivariant operator (or a latent-code operator over the same state) learns an action map that does not survive this gauge, reproducing the observed frames but resisting inversion by a planner. Holding the state representation fixed and varying *only* the operator, a permutation-equivariant operator $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + N(\boldsymbol{\theta}_t, a_t)$ is the only configuration we test that can be planned through; a non-equivariant transformer (FLAT), a NOVA-style additive operator (MLP), and a DINo/functa latent-code model (LATENT) all match its prediction PSNR yet plan at or below an identity floor (§4.1). We call the equivariant model the Implicit Neural Functional World Model, INFun-WM, and trace its advantage to the operator's action-to-displacement map. Only the equivariant operator induces motion in the commanded direction (§4.1).

Why weights, rather than a gauge-free latent code that would avoid the symmetry outright? Because the state is then a *function*: it renders back to the observation and exposes task-relevant structure an opaque latent does not. Our planner routes on occupancy (free vs. blocked cells) read directly off it (§4.4), controllability is diagnosable offline (§4.2), and structure as fine-grained as a safe-action set is readable from the weights (§4.3). A compact latent sidesteps the gauge symmetry but is not thereby controllable: our from-scratch LATENT baseline plans at the floor, while the one latent world model that controls well, DINO-WM, relies on web-scale pretraining. Equivariance is what makes the inspectable, from-scratch, full-fidelity state tractable. INFun-WM composed with a classical planner (A* on the occupancy it exposes) reaches routing control competitive with, and under matched planning ahead of, pretrained DINO-WM, from scratch and without reward, and diagnosable offline via an action-response predictor that tracks live control at $r = 0.91$.

Contributions. Beyond the two findings above, we contribute:

- **INFun-WM**: an INR-weight world model whose transition is a permutation-equivariant neural-functional operator (§3).
- **The action-response diagnostic**: an inexpensive, offline predictor of a model's controllability, validated against live control ($r = 0.91$), together with a scope criterion for when the operator is controllable (§4.2).
- **A composition method**: pairing the operator with a classical planner (A* on occupancy read off the state) for routing control on recognized layouts, competitive-to-ahead of a pretrained baseline (§4.4).

2 Related Work

World models and planning. Learning a dynamics model and acting through it is a long-standing recipe (Ha & Schmidhuber, 2018). In reinforcement learning, a "world model" typically means a holistic latent state learned from scratch and used for *reward-driven* control on a fixed task, either by planning through the model (Hafner et al., 2019; Schrittwieser et al., 2020; Hansen et al., 2024) or by learning a policy in imagination (Hafner et al., 2023; Micheli et al., 2022; Alonso et al., 2024). We use the term differently to mean reward-free, goal-conditioned planning to a target observation, where success is reaching an arbitrary goal rather than maximizing a task reward. The systems closest to this setting are DINO-WM (Zhou et al., 2024b), which plans zero-shot to a goal over frozen DINOv2 features, and Navigation World Models (Bar et al., 2025), a goal-conditioned video model; both are pretrained. Because DINO-WM is state of the art on the task family we study and shares our reward-free, goal-conditioned setting, we adopt it as our single external peer rather than surveying a leaderboard. Our central claim is a controlled ablation fixing the state representation and varying only the operator's symmetry, which a cross-system comparison would confound. This scopes two recurring points. Being *pretraining-free* distinguishes us only from DINO-WM, since the from-scratch systems above forgo pretraining too; and the from-scratch, same-setting comparison is supplied by our own FLAT and LATENT baselines, as any from-scratch *external* peer would be reward-driven.

Implicit representations as state for control. A parallel line uses implicit or neural scene representations as the state for control: NeRF-dy (Li et al., 2022) learns neural-scene-representation

dynamics for visuomotor control, ACID (Shen et al., 2022) is an action-conditional implicit dynamics model for deformable manipulation, and compositional neural radiance fields (Driess et al., 2023) model multi-object scenes with per-object fields. These place dynamics over a learned latent or scene representation; a closely related tradition keeps a compact per-frame code and rolls dynamics there: functa (Dupont et al., 2022) and DINo (Yin et al., 2022), the paradigm our LATENT baseline instantiates. We differ by taking the state to be the INR's own *weights* (we use a SIREN (Sitzmann et al., 2020)) and evolving them with an action-conditioned equivariant operator.

Weight-space dynamics. Closest to us are recent models that place dynamics directly in weight space. NOVA (Nzoyem & Comi, 2026) rolls INR weights under discovered latent actions with an additive-MLP transition for video generation and never tests plannability; our additive-operator baseline (MLP) shows that transition is uncontrollable and brittle to the neuron-permutation gauge. WARP (Nzoyem et al., 2026) and WeightCaster (Nzoyem, 2026) model weight-space sequences for forecasting and out-of-support generalization, and WeightFlow (Li et al., 2026) evolves a weight *density* with an equivariant operator; none conditions on actions or evaluates control. This is an active line; our contribution within it is to condition on ground-truth actions and to identify equivariance as what makes the resulting model *controllable*.

Neural functionals. The equivariant architectures we build on operate on the weights of other networks: NFN (Zhou et al., 2023a), neural functional transformers (Zhou et al., 2023b), DWSNet (Navon et al., 2023), graph metanetworks (Kofinas et al., 2024; Lim et al., 2023), and their generalizations (Zhou et al., 2024a; Kalogeropoulos et al., 2024). These have been used as *static* functionals (predicting or editing a trained network's properties) rather than as temporal operators. We repurpose one as an action-conditioned transition operator and find its equivariance is what renders the world model controllable; stronger operators from this family are natural drop-in replacements.

Prediction versus control. That prediction and control are distinct is foundational in control theory, where controllability is analyzed independently of predictive fit (Kalman et al., 1960; Åström & Murray, 2008); within learning, that a good predictor need not be a good basis for control is anticipated by the value-equivalence principle (Grimm et al., 2020) and by objective-mismatch and compounding-error analyses in model-based RL (Lambert et al., 2020; Janner et al., 2019), and is embodied by MuZero (Schrittwieser et al., 2020), whose model never reconstructs observations. Our contribution to this line is sharper and more mechanistic: we exhibit a matched-prediction pair of models whose controllability nonetheless diverges, and identify the operator's symmetry as the mechanism in a concrete representation class.

3 METHOD

We represent world state as the weights of an implicit neural representation and roll it forward with an equivariant operator conditioned on actions; we call the resulting world model INFun-WM (Figure 1).

State (INR). For each observed frame we fit a SIREN implicit neural representation (Sitzmann et al., 2020) by gradient descent and take its flattened weights $\boldsymbol{\theta} \in \mathbb{R}^d$ as the state. The INR renders to an image by evaluating its field $f_{\boldsymbol{\theta}}$ at the $H \times W$ pixel grid $\{x_i\}_{i=1}^{HW}$,

$$R(\boldsymbol{\theta}) = \left[\, f_{\boldsymbol{\theta}}(x_i) \,\right]_{i=1}^{HW}, \qquad f_{\boldsymbol{\theta}} : [0,1]^2 \to \mathbb{R} \text{ a SIREN.} \tag{1}$$

Fitting starts from an independent random initialization per trajectory, so two INRs encoding the same content are related by a neuron permutation π from the gauge group $\mathcal{S}$ (a relabeling of hidden units that leaves the rendered field unchanged); an operator must be invariant to this symmetry to generalize across trajectories.

Operator. We evolve the state autoregressively with a learned operator N,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + N(\boldsymbol{\theta}_t, a_t), \tag{2}$$

where N is a permutation-equivariant neural-functional transformer (Zhou et al., 2023b) that acts directly on the INR weights and is conditioned on the action a_t. It respects the gauge symmetry $\mathcal{S}$ above, and the render is invariant to it,

$$N(\pi \cdot \boldsymbol{\theta},\, a) = \pi \cdot N(\boldsymbol{\theta},\, a) \qquad \text{and} \qquad R(\pi \cdot \boldsymbol{\theta}) = R(\boldsymbol{\theta}), \qquad \forall\, \pi \in \mathcal{S}. \tag{3}$$

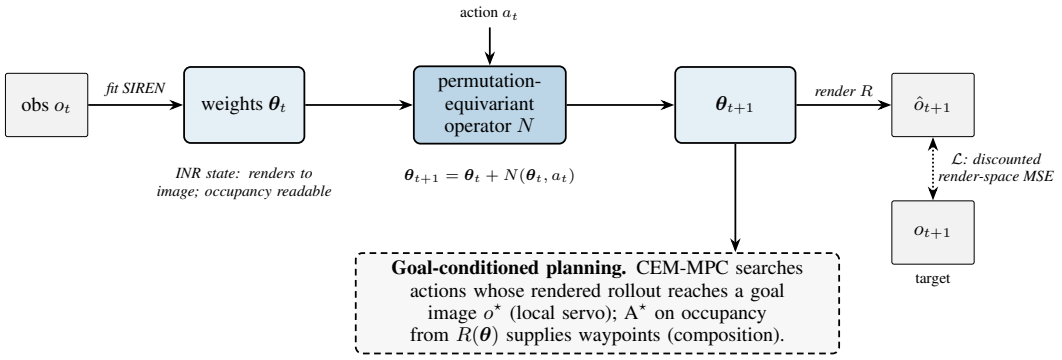


Figure 1: **INFun-WM.** World state is the weights $\boldsymbol{\theta}$ of a per-frame SIREN fit to the observation; a permutation-equivariant neural-functional operator N rolls the weights under actions ($\boldsymbol{\theta}_{t+1}{=}\boldsymbol{\theta}_t{+}N(\boldsymbol{\theta}_t, a_t)$), and the state renders back to an image $R(\boldsymbol{\theta})$. N is trained by a discounted *render-space* MSE to the observed frames (gauge-invariant). Because the state is an INR, occupancy is readable off the render for A* routing and controllability is diagnosable offline; at test time a planner reaches a goal image by CEM-MPC, composed with A* on recognized layouts.

This equivariance is the single property we vary in our study; it is what lets one operator generalize across gauges and, as we show, what makes the model controllable. We train N with a discounted *render-space* rollout loss: from a fitted start state we roll the operator forward, $\hat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_0$ and $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1}{+}N(\hat{\boldsymbol{\theta}}_{t-1}, a_{t-1})$, render each predicted state, and minimize the discounted image-space mean-squared error (MSE) to the ground-truth frames,

$$\mathcal{L} = \sum_{t=1}^{T} \gamma^t \, \tfrac{1}{HW} \big\| \, R(\hat{\boldsymbol{\theta}}_t) - R(\boldsymbol{\theta}_t) \, \big\|_2^2, \tag{4}$$

teacher-forced over short windows early in training and annealed to full free rollout. Because R is gauge-invariant, $\mathcal{L}$ depends on the states only through their rendered fields and so is well-defined across the neuron gauge, which is why we train in render space rather than weight space.

Planning. At control time we plan model-predictively: given a goal image, we sample action sequences, imagine the induced weight rollouts, render each to an image, score it against the goal, and execute the first action in the real environment (receding-horizon CEM-MPC). On recognized layouts we *compose* this local *servo* (a controller that drives the agent to a commanded target) with a classical planner: A* (classical shortest-path graph search) on occupancy extracted from the observation produces waypoints, and the operator reaches each in turn (§4.4).

4 Experiments

Baselines. Holding the INR-weight state fixed and varying only the operator, we compare INFun-WM against three non-equivariant or latent alternatives: FLAT, a non-equivariant transformer over the flattened weights; MLP, a NOVA-style additive MLP (Nzoyem & Comi, 2026); and LATENT, a from-scratch functa/DINo latent-code model (Dupont et al., 2022; Yin et al., 2022). The LATENT baseline is a shared base INR modulated by a small per-frame code fit to each observation, with dynamics an MLP over the code and decoding back to pixels; the compact code sidesteps the gauge symmetry but is capacity-capped by the shared base. As an external peer we also evaluate the pretrained DINO-WM (Zhou et al., 2024b), the pretrained-latent road of §1, in the composition comparison on PointMaze (§4.4). **Metrics.** We report *plan success* against an *identity floor* (the plan success of the identity operator $\boldsymbol{\theta}_{t+1}{=}\boldsymbol{\theta}_t$, whose state persists so the agent does not move) and a true-dynamics *oracle* ceiling, and *fresh-gauge rollout PSNR* (prediction quality from novel start states) as the prediction metric. **Environments.** Controllable sprites with $C{=}1$–4 objects; two egocentric environments, a walled maze and a landmark scene, each rendered from an agent-centered camera so the world scrolls as the agent moves; and the gymnasium/mujoco PointMaze. Figure 2 previews the two behaviors we establish below, goal-reaching control and composed routing.

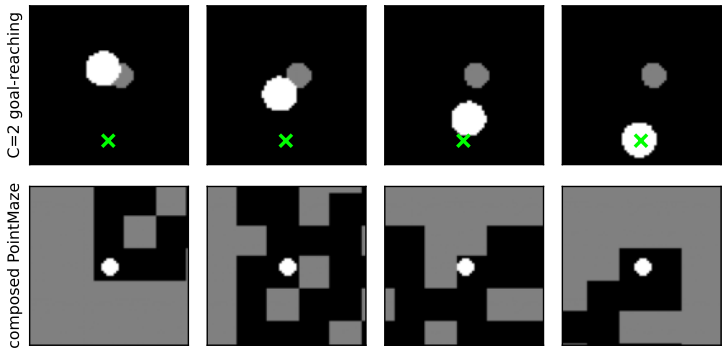


Figure 2: **INFun-WM controls, at a glance.** *Top*: the operator plans the agent onto the goal on the $C{=}2$ sprite scene (65% success, versus the $\leq 10\%$ identity floor of the non-equivariant and latent operators). *Bottom*: the same operator, composed with A* on the occupancy read off its state, routes clean-render PointMaze to the goal. All frames are genuine executed observations (§4).

Table 1: **Prediction does not imply control** (controllable sprites, 3 seeds). The non-equivariant (FLAT) and latent (LATENT) operators *match* INFun-WM's fresh-gauge rollout PSNR yet plan at or below the identity floor; only the equivariant INFun-WM controls. Oracle $= 100\%$ confirms the tasks are solvable. Per-operator action-response and per-seed values are in Table 3; controllability across all environments in Table 2.

	$C{=}1$ (single sprite)		$C{=}2$ (agent+distractor)	
operator	success%	PSNR	success%	PSNR
INFun-WM (equivariant)	**85±0**	13.90	**65±8**	15.21
FLAT (non-equiv. transformer)	12±5	13.12	10±4	15.11
LATENT (DINo/functa)	10±4	12.50	5±4	14.05
identity floor	15±0	–	10±0	–
oracle (true-dyn. ceiling)	100	–	100	–

4.1 The Predict-Control Gap and Its Mechanism

Table 1 reports the central phenomenon (3 seeds, $n_{\text{train}}{=}200$, 20 episodes). The non-equivariant and latent baselines *predict on par with the equivariant operator*, at fresh-gauge PSNR $13.1/15.1$ (FLAT) and $12.5/14.0$ (LATENT) versus INFun-WM's $13.9/15.2$, yet plan *at or below* the identity floor, while only INFun-WM exceeds it ($85\%/65\%$ vs. $15\%/10\%$). The oracle solves the task at 100%, so the baselines' failure is not task-impossibility: it is a failure of *controllability* that prediction accuracy does not reveal.

Three failure modes, one outcome. Holding the INR-weight state fixed and varying only the operator, the baselines fail planning in three *distinct* ways. (1) MLP (NOVA-style additive, fully non-equivariant) *fails gauge generalization*: fresh-gauge PSNR -107 dB, worsening to -218 dB at $3\times$ epochs, as it overfits training gauges and cannot generalize across the neuron-permutation symmetry, the documented failure of MLPs on flattened weights (Navon et al., 2023; Zhou et al., 2023b); its action-response overshoots wildly (3.6), far above the true scale. (2) FLAT *predicts but responds off-axis*: its PSNR ties INFun-WM and it does respond to the action (offline action-response 0.41), but near-orthogonally to the commanded direction ($92°$, Figure 3), so CEM cannot drive it to a goal. (3) LATENT *predicts but is misdirected*: it responds at nearly the true scale (action-response 0.77) yet points the wrong way ($84°$ off), so response magnitude alone would rate it controllable while its plan sits at the floor.

The mechanism: the action-to-displacement map. Magnitude alone cannot separate INFun-WM's accurate response from LATENT's misdirected one. We add a fidelity probe (Figure 3): sweep a grid of probe actions, read the agent's induced displacement from the rendered prediction versus the true environment, and report angular error, magnitude ratio, and R^2. Only INFun-WM has an *ac-*

curate map: induced displacement within 2–3° of the true action, near-unit magnitude, $R^2 \approx 0.98$. Every baseline is 74–92° off (near-orthogonal) with *negative* R^2 (worse than predicting the mean). LATENT does respond (even overshooting in magnitude) but in the wrong direction; the directional probe exposes what the magnitude metric hid. A CEM can control an operator only if its action map is accurate, and we attribute that accurate map to the operator's equivariance, the mechanistic explanation of the phenomenon in Table 1.

4.2 A Validated Predictor of Controllability

The mechanism suggests a cheap screen: an operator is controllable only if it responds to the action on the true scale. We therefore propose the *action-response* (§4.1) as an offline diagnostic of controllability, computed with no live environment: $\|R(\boldsymbol{\theta}{+}N(\boldsymbol{\theta}, a)) - R(\boldsymbol{\theta}{+}N(\boldsymbol{\theta}, 0))\| / \|R(\boldsymbol{\theta}_{t+1}) - R(\boldsymbol{\theta}_t)\|$ per held-out transition. That is, we use the fraction of the operator's predicted frame-to-frame change that the action drives (numerator: the render's response to a versus the null action; denominator: the total predicted change), which stays near 0 when the action barely moves the render and near 1 when it drives the transition. In control terms this is the operator's *control authority*. An operator whose action-response is near zero has a degenerate input map, so it is uncontrollable regardless of its predictive accuracy, the learned-model counterpart of an uncontrollable system (Kalman et al., 1960; Åström & Murray, 2008). It matches live control behavior and, across the environment suite, correlates with live plan-success at **Pearson** $r = 0.91$ (Figure 4).

The predictor does more than score individual transitions: it delimits the controllable *regime*. The equivariant operator is controllable exactly when the observation is a *clean, action-determined transform* of the state (an ego translation of a clean field), so from the observation structure alone one can say, before any control, whether it will work. A control confirms the criterion and separates two easily-confounded factors, entity size and contact dynamics: a *small* action-determined sprite of radius 4 at pusher scale, with the success threshold held at the standard 8px, keeps action-response 0.87 and controls at 52% (3.4× its identity floor; operator-specific, FLAT 15%, LATENT 10%), so a small-in-frame entity is controllable when its motion is action-determined. Where the criterion is violated, it is violated *predictably and traceably*: textured mujoco renders and contact-limited manipulation (PushT, where the block moves only by contact) break action-determinism, and the inspectable state locates each floor at its source, the render rather than the dynamics (§4.3), a boundary of *this pixel-INR operator*, not of any method, that points toward richer observation encodings.

4.3 Scaling and Breadth

Object count. Table 2 collects controllability across environments; with distinguishable objects, control scales from $C{=}1$ to 4. Controllability degrades *roughly linearly* (INFun-WM plan success 85%, 65%, 43%, 23%). Through $C{=}3$ the separation is decisive (over 5× the best foil); at $C{=}4$ INFun-WM still leads (23% vs. 10%) but its action-response has begun to drop (from 0.98 to 0.65), i.e. many-object entanglement degrades the action map. The oracle is 100% throughout, so the decline is a genuine capacity limit, not task difficulty: the equivariant operator is uniquely controllable but has a finite object budget.

Egocentric maze (works), and a PointMaze floor we trace to the render, not the method. The operator generalizes to an egocentric maze with walls and occlusion (INFun-WM 62% vs. floor 10%). On the faithful mujoco PointMaze it floors, but the inspectable state lets us localize the source rather than guess: prediction is intact (13 dB), so this is a *control* failure; it is not non-Markovian dynamics (a velocity/history-augmented operator does not help) nor obs-SNR (a clean 48 dB fit), but the *textured render* breaking the action-to-observation map. A controlled swap confirms it: re-rendering the *same* dynamics with a clean field restores action-determinism (R^2 from 0.07 to 0.99, action-response from 0.19 to 0.82) and local control, leaving only a *routing* gap that §4.4 closes. The floor is thus a property of the observation encoding, not the dynamics, task, or operator.

Breadth: the equivariance principle extends beyond the operator. The same permutation-equivariance that makes the transition controllable is the right prior for *any* function of INR weights, not only the dynamics. As a probe, we train an invariant neural-functional readout (Zhou et al., 2023a) to recover a non-affine safe-action set directly from a scene's INR weights, and test it on *fresh gauges* (random neuron permutations unseen at training): it reaches 81.8 **safe-set IoU** versus 68.4 for a canonicalize-then-MLP baseline (5 seeds), while a plain MLP on the raw weights col-

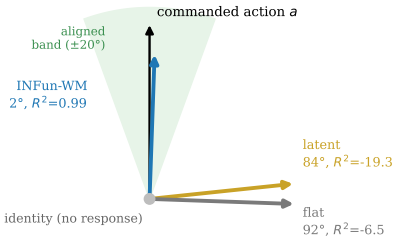


Figure 3: **The action-to-displacement map is the mechanism** (controllable sprites). Only the equivariant INFun-WM induces motion in the commanded direction (well within the shaded band); the non-equivariant and latent operators respond near-orthogonally (74–92°, negative R^2), with LATENT overshooting in magnitude while pointing the wrong way. The shaded band ($\pm 20°$) is illustrative, marking displacements closely aligned with the command. Mean angle/magnitude/R^2 (C=1): INFun-WM 2°/0.98/0.99, FLAT 92°/2.60/−6.5, LATENT 84°/4.13/−19.3.

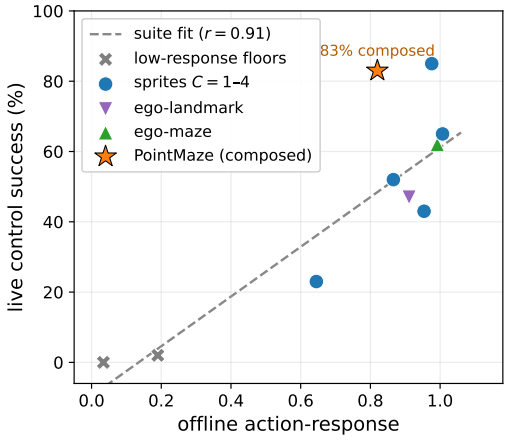


Figure 4: **Offline action-response predicts live controllability** (Pearson $r = 0.91$). High response yields control across sprites and egocentric mazes; low response floors (textured PointMaze, PushT). Clean-render PointMaze (star) is off-trend: controllable locally but routing-limited on its own, it reaches 83% once composed with A*, and is excluded from the fit (§4.4).

Table 2: **Controllability across environments and controllers** (plan success %, 3 seeds; *a-resp* = offline action-response diagnostic; *ctrl*: op = operator alone / open-loop CEM, A* = A*-composed; "–" = not run in that configuration). INFun-WM is the only operator that clears the identity *floor*, and its plan success tracks the offline action-response. FLAT/MLP/LATENT are the in-house operator ablation on the sprite and ego environments; DINO-WM is the pretrained external peer on PointMaze. On PointMaze the operator routes weakly on its own but, composed with A*, reaches 83% in-distribution and 37% on a never-seen layout (OOD), competitive-to-ahead of DINO-WM under the same A* routing. "C=1 small"=pusher-scale sprite; "ego-land."=egocentric landmarks; "ego-maze"=egocentric maze.

Environment	ctrl	INFun-WM	FLAT	MLP	LATENT	DINO-WM	floor	oracle	a-resp
C=1	op	**85**	12	13	10	–	15	100	0.98
C=2	op	**65**	10	8	5	–	10	100	1.01
C=3	op	**43**	3	8	8	–	5	100	0.95
C=4	op	**23**	10	5	10	–	0	100	0.65
C=1 small	op	**52**	15	7	10	–	15	100	0.87
ego-land.	op	**47**	0	8	0	–	5	100	0.91
ego-maze	op	**62**	5	5	5	–	10	100	0.99
PointMaze	op	33	–	–	–	53	0	100	0.82
PointMaze	A*	**83**	–	–	–	70	0	100	–
PointMaze (OOD)	A*	**37**	–	–	–	16.7	0	100	–

lapses under the gauge shift (to 23.7 IoU). The inspectable weight state is thus queryable by the same equivariant machinery for structure beyond occupancy, here a safety-relevant action set.

4.4 COMPOSITION: CONTROL ON A RECOGNIZED LAYOUT

The predictor flags clean-render PointMaze as locally controllable (0.82) yet weak on *global routing* on its own, the one case where action-response is high but open-loop end-to-end control still lags. This is exactly where a classical planner composes with the operator. We build a hierarchical controller: A* on the maze occupancy (extracted from the observation) produces a sequence of nearby world waypoints, and the operator reaches them one at a time, running the receding-horizon CEM

planner (§3) toward each waypoint's rendered image and advancing to the next. Because every hop is short, the operator only needs the local control it does well, while A* handles the global routing.

Composition makes routing reliable, and only for the equivariant operator (Table 2): composed with A* our operator reaches 83%, versus 17% and 17% for the non-equivariant and latent servos under the *same* A* routing, because only the equivariant operator has the accurate action map (§4.1) needed to execute the local hops. Neither paradigm alone suffices: the open-loop operator reaches only 33%, since it cannot route to distant goals, and A* needs a controllable local model to execute its hops; only their composition reaches 83%. Hierarchical subgoal planning is itself an established remedy for the long-horizon/local-minima collapse of flat world-model planning (Zhang et al., 2026). Our contribution is not the hierarchy but the low-level model it runs on: the equivariant operator is what makes the local servo controllable at all, since the same A* scaffold with a non-equivariant or latent servo reaches only 17%, and the high level can be a *classical* planner reading occupancy directly off the inspectable INR state rather than a learned latent hierarchy. Trained only on the Medium maze, the controller partially generalizes to a never-seen layout (37% vs. 83% in-distribution), so the local servo transfers but is not layout-invariant. *Scope*: actions are position deltas (Δpos), not raw forces, replayed by a wall-aware low-level servo that stops the ball at walls rather than pushing it through. The operator's dynamics are learned from faithful mujoco rollouts with real momentum and contact, not a simplified simulator.

Planner-matched comparison to DINO-WM. Is a pretrained state-of-the-art latent world model simply better? DINO-WM (Zhou et al., 2024b) (DINOv2 features + MPC) routes PointMaze *end-to-end* above our operator's 33% open-loop, so "latent world models cannot be controlled" is false in general: our from-scratch FLAT/LATENT baselines fail, but a pretrained latent world model (DINO-WM) is controllable, a limitation of those baselines, not of latent world models as a class. Its PointMaze performance, however, hinges on *proprioception*, the agent's ground-truth position and velocity $[x, y, \dot{x}, \dot{y}]$ handed to the planner as a privileged input alongside the image: with proprioception it routes end-to-end at $86\pm15\%$, but from vision alone (no proprioception, matching our goal-image operator) at only 53%. To compare fairly we run both *visual-only* (no proprioception, since our operator plans from goal images) and give DINO-WM the *same* A* scaffold, namely A* waypoints with DINO-WM's own CEM servo at matched budget and tolerance over 3 seeds: **DINO-WM composed** $= 70\pm16\%$, versus our operator's composed 83%. Two things follow. (1) Composition is *not* purely operator-specific: it lifts DINO-WM too, from 53% to 70%, as A* helps any servo with routing headroom. (2) It lifts our operator more, from 33% to 83% versus DINO-WM's 53% to 70%, because our operator is locally accurate with more routing headroom, while DINO-WM already routes end-to-end on its own. Crucially, *under matched routing our pretraining-free operator stays ahead of the web-scale-pretrained SOTA* (83% vs. 70%): the composition result is not an artifact of a scaffold withheld from the baseline. The advantage is, if anything, *larger* out of distribution: transferred to a never-seen layout with no retraining, our operator reaches 37% versus DINO-WM's 16.7% (both visual-only *held-out*, 3 seeds), more than a $2\times$ margin. DINO-WM's composed servo is high-variance (50–90%) and carries DINOv2 pretraining our from-scratch, 64px operator does not; we therefore claim *competitive-to-ahead under matched, visual-only routing, pretraining-free*. We deliberately compare under a *simple, identical* A* scaffold to isolate the local servo, i.e. which operator more accurately reaches a given waypoint, so the 83%-vs-70% gap is that servo difference. We do not address more sophisticated hierarchical planners here, because they act on an *orthogonal* axis: improving the high-level planner (which subgoals to chase), not the servo. Such planners are in fact *complementary*: a purpose-built latent hierarchy (Zhang et al., 2026) routes DINO-WM far better, but it still relies on a controllable low-level servo to reach its subgoals, which is exactly what our result provides: any hierarchy of this kind can take a controllable operator, such as ours, as its low-level model. One asymmetry the success numbers do not capture: INFun-WM's state renders to pixels and its action map is directly readable, whereas DINO-WM's DINOv2-feature dynamics are opaque, so INFun-WM is the more *diagnosable* of the two (§6).

5 LIMITATIONS

The controllability claim is scoped to discrete, distinguishable objects ($\lesssim 4$) and a kinematic, Markovian action model, and to observations that are clean action-determined transforms (§4.2): the current pixel-INR encoding floors on textured 3-D-perspective renders and contact-limited manipulation (PushT). A controlled ablation (§4.3) isolates the texture floor to the observation encoding, not

the dynamics, pointing to richer INR features rather than a fundamental barrier. The state is a per-frame INR fit (test-time optimization); *amortizing* it is a genuine open problem. The action-response diagnostic as defined reads out the observation change through the INR's decoder; extending it to *render-free* latents such as DINO-WM would require measuring the response in feature or position space rather than on pixels. *Future directions* include structured operators for more objects, generative (ego-video) rollout for first-sight navigation, layout-invariant operator training, 3-D egocentric observations, and this render-free extension of the diagnostic.

6 Conclusion

We showed that in a weight-space world model, prediction accuracy does not imply controllability, and that the symmetry of the dynamics operator is the deciding factor: a permutation-equivariant operator is uniquely controllable because it alone exposes an accurate action-to-displacement map. From this we distilled the *action-response*, a validated offline predictor of controllability, and a scope criterion for its controllable regime, satisfied when the observation is a clean, action-determined transform of the state. Composed with a classical planner, the operator reaches routing control competitive-to-ahead of a pretrained SOTA under matched planning, pretraining-free. Because INFun-WM's state renders to observations and its action map is directly measurable, it is uniquely *diagnosable*; its controllability can be predicted offline and its failures localized (§4.2, §4.3). Holistic latent world models, whose dynamics evolve in an opaque feature space, cannot do the same. The path to planner-ready world models therefore runs through controllability, a property to be measured and designed for, not inferred from predictive fit.

AI use statement

Generative AI tools (including an AI coding assistant) were used to help implement experiments, generate and refactor code, and draft and edit text. All AI-assisted code was executed and its numerical outputs verified by the authors; all AI-assisted text was reviewed and edited by the authors, who take responsibility for the final content, claims, and artifacts of this work.

Reproducibility statement

All operators, baselines, environments, and planners are specified in §3–4 and the appendix; every reported number is the mean over the stated seeds against the identity floor and oracle ceiling. Code and configuration to reproduce the tables and figures will be released upon publication.

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.

Karl Johan Åström and Richard M Murray. Feedback systems: an introduction for scientists and engineers. 2008.

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

Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Conference on robot learning*, pp. 1755–1768. PMLR, 2023.

Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in neural information processing systems*, 33:5541–5552, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 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*, pp. 2555–2565. PMLR, 2019.

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

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Rudolf E Kalman et al. On the general theory of control systems. *IFAC Proceedings Volumes*, 1(1): 491–502, 1960.

Ioannis Kalogeropoulos, Giorgos Bouritsas, and Yannis Panagakis. Scale equivariant graph metanetworks. *Advances in neural information processing systems*, 37:106800–106840, 2024.

Miltiadis Miltos Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J Burghouts, Efstratios Gavves, Cees G Snoek, and David Zhang. Graph neural networks for learning equivariant representations of neural networks. In *International Conference on Learning Representations*, volume 2024, pp. 45363–45381, 2024.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

Ruikun Li, Jiazhen Liu, Huandong Wang, Qingmin Liao, and Yong Li. Weightflow: Learning stochastic dynamics via evolving weight of neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 641–649, 2026.

Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pp. 112–123. PMLR, 2022.

Derek Lim, Haggai Maron, Marc T Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. *arXiv preprint arXiv:2312.04501*, 2023.

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

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, pp. 25790–25816. PMLR, 2023.

Roussel Nzoyem, Nawid Keshtmand, Enrique Crespo Fernandez, Idriss Tsayem, Raul Santos-Rodriguez, David Barton, and Tom Deakin. Weight-space linear recurrent neural networks. In *International Conference on Learning Representations*, volume 2026, pp. 62791–62830, 2026.

Roussel Desmond Nzoyem. Out-of-support generalisation via weight-space sequence modelling. *arXiv preprint arXiv:2602.13550*, 2026.

Roussel Desmond Nzoyem and Mauro Comi. Render, don't decode: Weight-space world models with latent structural disentanglement. *arXiv preprint arXiv:2605.06298*, 2026.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Bokui Shen, Zhenyu Jiang, Christopher Choy, Leonidas J Guibas, Silvio Savarese, Anima Anandkumar, and Yuke Zhu. Acid: Action-conditional implicit visual dynamics for deformable object manipulation. *arXiv preprint arXiv:2203.06856*, 2022.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Yuan Yin, Matthieu Kirchmeyer, Jean-Yves Franceschi, Alain Rakotomamonjy, and Patrick Gallinari. Continuous pde dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*, 2022.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, et al. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 36:24966–24992, 2023a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. *Advances in neural information processing systems*, 36:77485–77502, 2023b.

Allan Zhou, Chelsea Finn, and James Harrison. Universal neural functionals. *Advances in neural information processing systems*, 37:104754–104775, 2024a.

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*, 2024b.

A Appendix

We provide a per-seed breakdown of the predict-vs-control results (Table 3), the fresh-gauge safe-action-set read-out of §4.3 (Table 4), and qualitative goal-reaching and composition rollouts (Figures 5–10).

Qualitative results

Figures 5–7 show goal-reaching control contrasts on the discrete-object environments, and Figures 9–10 the A*-composed rollouts on clean-render PointMaze, in-distribution and on a held-out layout head-to-head with DINO-WM. In every goal-reaching panel each operator plans toward a fixed goal and the plan is executed in the real environment, so all frames are genuine observations and the discriminator is purely whether the agent reaches the goal. The shown episodes are representative, selected from the majority of held-out episodes with the same outcome, and the cited success rates are the multi-seed aggregates from §4.

Reproducibility. *State*: a SIREN (Sitzmann et al., 2020) with two hidden layers of width 64 (ω_0=30, scalar output, hardened $\sinh$ first layer) on 64×64 observations; each frame is fit by Adam (lr 5×10^{-4}), the first frame of a trajectory from an independent random initialization (2000 steps, which gives the per-trajectory gauge diversity) and each subsequent frame warm-started from the previous (200 steps). *Operator*: a permutation-equivariant neural-functional transformer (Zhou et al., 2023b) over the flattened SIREN weights, $\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+N(\boldsymbol{\theta}_t, a_t)$, trained with the discounted render-space rollout loss ($\gamma=0.9$; teacher-forced warmup annealed to free rollout) by Adam (lr 10^{-3}, batch 16, gradient-clip 1.0, 50–80 epochs). *Planning*: receding-horizon CEM-MPC with population 96, 5 CEM iterations, horizon 5, one action executed per replan, up to 70 steps per episode. *Composition*: A* on occupancy read from the render (recovered exactly, 0-cell mismatch versus the true maze), advancing to the next waypoint within 0.6 world units.

Table 3: Per-seed plan success (%) on the predict-vs-control task (3 seeds, n_{train}=200, 20 episodes), with mean fresh-gauge rollout PSNR and the offline action-response. Only INFun-WM exceeds the identity floor; FLAT/MLP/LATENT match INFun-WM's PSNR yet plan at or below it, and MLP fails gauge generalization (negative PSNR). Action-response is the offline diagnostic magnitude; the baselines respond (nonzero) but in the wrong direction (Figure 3), so only INFun-WM pairs a true-scale response with an accurate action map.

operator	plan success per seed (%)			PSNR (dB)	act-resp
	seed 0	seed 1	seed 2		
C=1 *(single object)*					
INFun-WM	85	85	85	13.9	0.98
FLAT	15	15	5	13.1	0.41
MLP	15	15	10	−107.7	3.56
LATENT	10	5	15	12.5	0.77
identity floor	15	15	15	10.8	0.00
oracle		100		–	–
C=2 *(two objects)*					
INFun-WM	65	55	75	15.2	1.01
FLAT	5	15	10	15.1	0.38
MLP	10	5	10	−113.7	3.70
LATENT	10	5	0	14.0	0.79
identity floor	10	10	10	13.7	0.00
oracle		100		–	–

Table 4: Fresh-gauge safe-action-set read-out (IoU, 5 seeds) on the KV affordance task of §4.3. The invariant neural-functional readout (INFun-WM) and canonicalize-then-MLP are gauge-invariant (in-gauge = fresh-gauge, zero drop); the raw-weight MLP fits well in-gauge but collapses under a fresh neuron permutation.

read-out	in-gauge IoU	fresh-gauge IoU	drop
invariant NFN (INFun-WM)	0.818	0.818 ± 0.008	0.000
canonicalize-then-MLP	0.684	0.684 ± 0.028	0.000
raw-weight MLP	0.876	0.237 ± 0.020	0.639

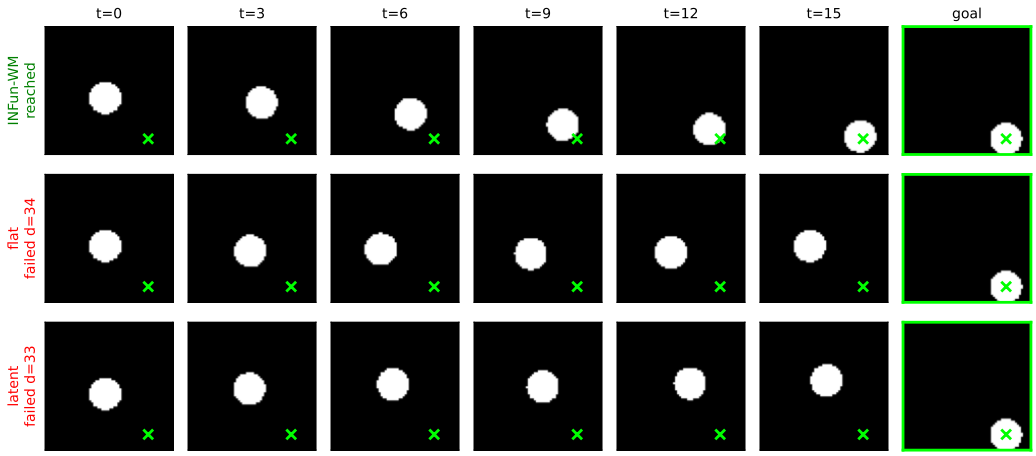


Figure 5: **Goal-reaching on the controllable sprite ($C{=}1$).** A representative held-out episode; each operator plans toward the goal (green $\times$) and the plan is executed in the real environment (all frames are genuine observations). Only the equivariant INFun-WM drives the agent onto the goal, and its final frame matches the goal panel; FLAT and LATENT move it the wrong way and stall far off (final distances annotated). Aggregate plan success: INFun-WM 85% versus FLAT/LATENT at the 10–15% identity floor (Table 1).

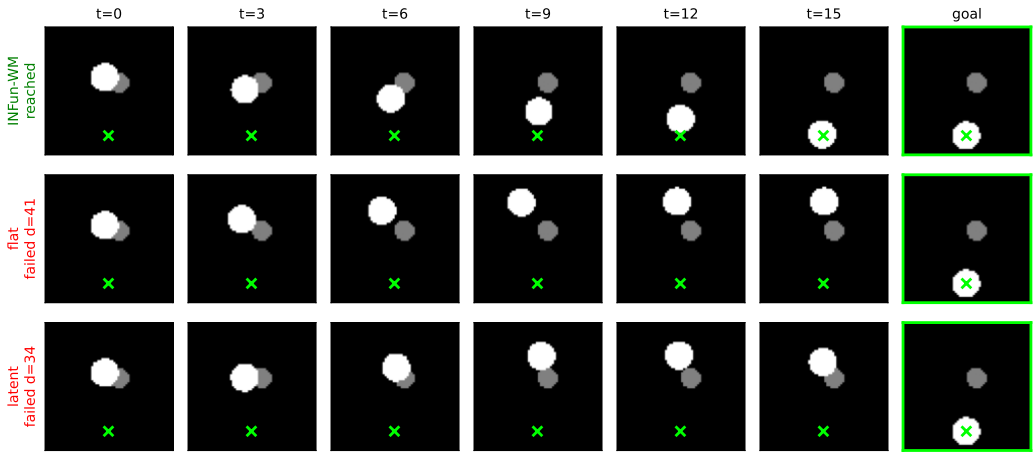


Figure 6: **Goal-reaching with an agent and a distractor ($C{=}2$).** As Figure 5, with a static gray distractor present. INFun-WM drives the *agent* (not the distractor) onto the goal, while FLAT sends it the wrong way and LATENT stalls. Aggregate plan success: INFun-WM 65% versus FLAT/LATENT $\leq 10\%$ (Table 1).

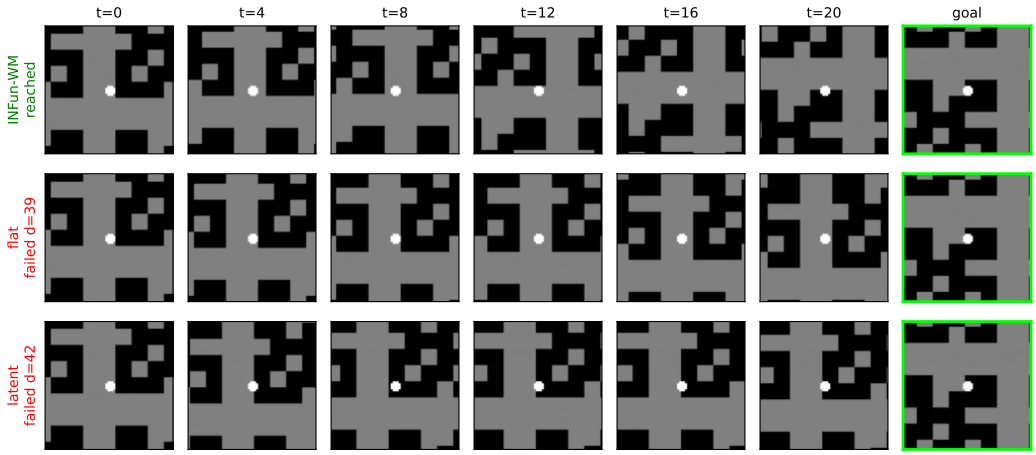


Figure 7: **Goal-reaching on the egocentric maze.** Observations are egocentric, so reaching the goal scrolls the wall-field into the goal configuration (rightmost panel). INFun-WM matches the goal configuration; FLAT barely moves and LATENT ends misaligned. Aggregate plan success: INFun-WM 62% versus the 10% identity floor (§4.3).

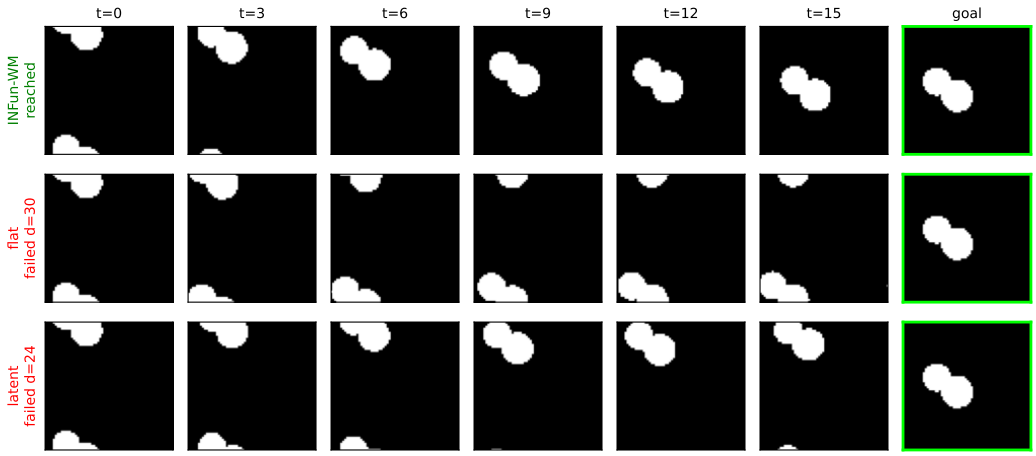


Figure 8: **Goal-reaching on the egocentric landmark scene.** Observations are egocentric over static landmarks, so reaching the goal scrolls the landmarks into the goal configuration (rightmost panel). INFun-WM matches the goal configuration and reaches; FLAT and LATENT end far off (failed). Aggregate plan success: INFun-WM 47% versus the 5% identity floor, with FLAT/LATENT at 0% (Table 2).

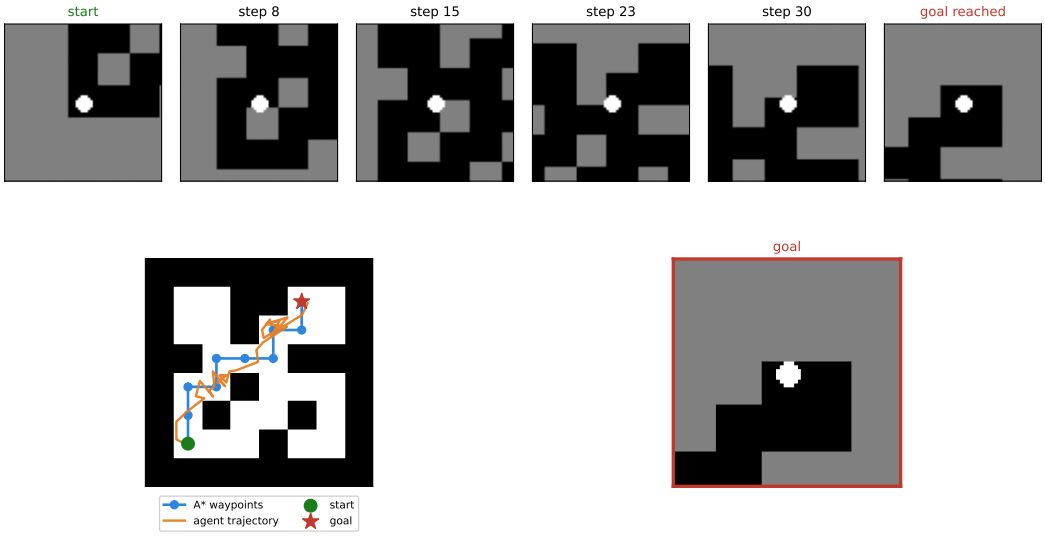


Figure 9: **Composition on clean-render PointMaze.** One successful composed episode (§4.4). *Top*: executed egocentric observations from start to goal-reached. *Bottom-left*: the maze occupancy read directly off the observation (walls black, free white; 0-cell mismatch versus the true maze) with the A* waypoints (blue), the executed trajectory (orange), start (green) and goal (red star). *Bottom-right*: the goal observation. A* supplies global routing while the equivariant operator's CEM servo reaches each waypoint locally; composition lifts end-to-end routing from 33% to 83% (§4.4).

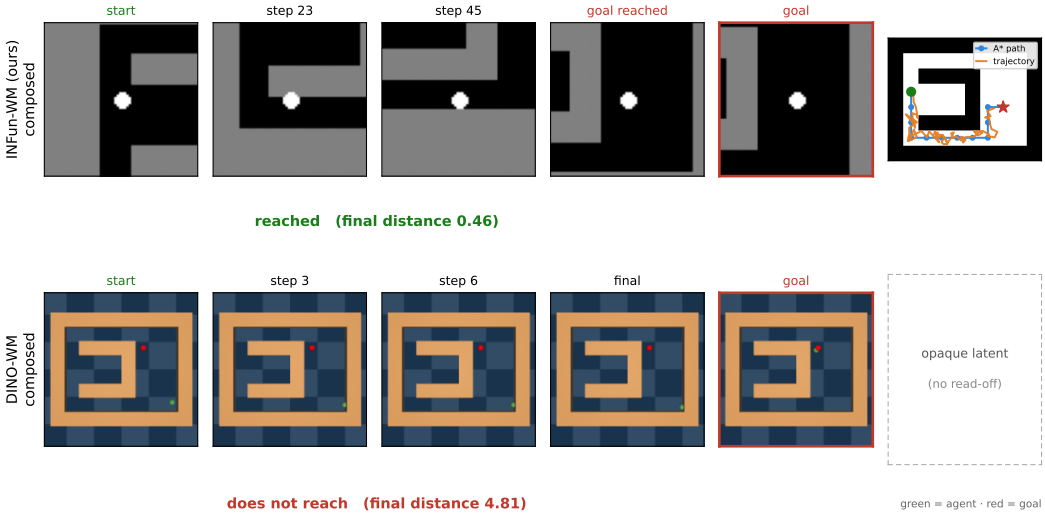


Figure 10: **Composed control on a held-out (never-seen) PointMaze layout.** Both models plan with the same A* routing and their own local servo (§4.4). *Top* (INFun-WM, ours): executed clean-render egocentric observations reach the goal (final distance 0.46), shown with the occupancy read directly off the observation, the A* path, and the executed trajectory. *Bottom* (DINO-WM): from its native allocentric RGB observation, the pretrained peer stalls short of the goal (final distance 4.81); its latent state is opaque, so there is no occupancy read-off. Over this never-seen layout, INFun-WM composes to 37% versus DINO-WM's 16.7% (Table 2), more than a 2× margin, from scratch.