ATM: WHY LATENT WORLD MODELS CAN FAIL TO PLAN

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models can achieve accurate latent prediction yet still differ substantially in downstream planning performance. We argue that a key source of this discrepancy lies in the structure of action-induced latent transitions. We formalize action-identifiability through Bayes inverse risk, characterizing how much uncertainty about an action remains after observing the transition it induces. Model-predicted transitions can become highly self-decodable while encoding a domain-specific action relationship that fails to transfer to real environment transitions. We characterize this mismatch through cross-domain inverse transfer, instantiated as the Action-Consistency Transfer Matrix (ATM), a 2×2 inverse-risk matrix over real and predicted transition domains. Across TwoRoom, PushT, and OGBench-Cube, true-transition action-identifiability tracks downstream planning substantially better than standard prediction loss, while the full ATM reveals highly self-decodable yet cross-domain-inconsistent predicted transitions. Controlled interventions on the two domains further produce distinct transition structures and planning outcomes, supporting this decomposition. The same diagnostics also support lightweight model screening, reaching 98.81% pairwise ranking accuracy for candidates separated by more than 5% success.

1 INTRODUCTION

Latent world models support goal-conditioned planning by rolling learned dynamics forward in representation space to evaluate candidate action sequences (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2022). Recent methods such as DINO-WM (Zhou et al., 2025), PLDM (Sobal et al., 2025), and LeWM (Maes et al., 2026) further show that such planning does not require explicit future-observation reconstruction. Yet accurate latent prediction does not necessarily translate into reliable planning: models with favorable prediction quality can still differ substantially in downstream performance (Grimm et al., 2020; Wang et al., 2026a). Existing evaluation largely focuses on prediction accuracy or endpoint success, leaving unclear what properties of the learned representation actually make it useful for action selection. We argue that a missing piece lies in the structure of action-induced latent transitions. Although latent planning is typically formulated as goal matching, candidate actions affect the objective through transitions induced by the learned dynamics.

A planning-oriented representation should therefore preserve action effects so that different candidate actions remain distinguishable in latent transition space (Pathak et al., 2017; Brandfonbrener et al., 2023; Cui et al., 2024; Han & Yilmaz, 2026; Zhang et al., 2026). We call this property *action-identifiability*: the action inducing a latent transition should remain recoverable to a meaningful extent. Recent concurrent work has likewise emphasized transition-level action sensitivity, using action recovery or transition separation as representation-learning principles (Boylan & Hokamp, 2026; Gan et al., 2026). If different actions induce weakly distinguishable or strongly entangled latent changes, the representation may provide an unreliable interface for action selection even when individual latent states are informative. We show that action-identifiability admits a principled interpretation through Bayes inverse risk, linking transition quality to the conditional uncertainty of the action given the latent transition.

Action-identifiability characterizes action structure within a transition domain. A latent world model naturally induces two such domains: real encoded transitions from environment evolution and model-predicted transitions generated by the learned dynamics. Both domains may contain decod-

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.

able action information yet encode actions through different relationships. In particular, predicted transitions can exhibit strong in-domain action consistency because the action is explicitly provided to the predictor; related planning-time methods test whether an inverse dynamics model can recover this conditioning action (Seo et al., 2026). This raises the question of whether this action–transition relationship is also preserved in real encoded transitions.

This motivates a cross-domain test of action structure. If real and predicted transitions preserve compatible action-conditioned relationships, an inverse rule learned in one domain should remain informative in the other; cross-domain degradation indicates that the action relationship changes between domains. We therefore introduce **cross-domain inverse transfer** and instantiate it as the **Action-Consistency Transfer Matrix (ATM)**, a 2×2 matrix formed by training inverse probes in each domain and evaluating them on both. Its diagonal entries characterize in-domain action-identifiability, while the off-diagonal entries measure cross-domain transfer of the action–transition relationship. Together, the four terms distinguish weak action structure from a qualitatively different failure mode: a domain-specific action relationship that is coherent within model rollouts but does not transfer to real transitions.

We validate this view on TwoRoom, PushT, and OGBench-Cube (Park et al., 2025). True-transition action-identifiability tracks planning substantially better than standard prediction loss, while the full ATM exposes models whose predicted transitions are highly self-decodable yet exhibit distinct cross-domain action structure. Controlled interventions further show that improving decodability in the two domains produces distinct transition structures and planning outcomes, supporting this distinction. Beyond diagnosis, the same quantities provide a lightweight signal for model screening. Together, these results show that latent planning failures reflect not only weak action information but also how it is represented and transferred between real and predicted transitions.

Our contributions are as follows:

1. We identify **action-induced transition structure** as a key factor in latent planning and formalize action-identifiability through Bayes inverse risk.
2. We show that predicted transitions can be highly self-decodable yet encode domain-specific action structure, and introduce **ATM** to characterize within-domain identifiability and cross-domain action transfer.
3. We validate this decomposition through controlled interventions across three planning tasks and demonstrate its utility for model diagnosis and screening.

2 Related Work

Latent world models for control and planning. Early work such as Ha and Schmidhuber's World Models (Ha & Schmidhuber, 2018) and PlaNet (Hafner et al., 2019) showed that compact latent dynamics can support control from high-dimensional observations, while Dreamer (Hafner et al., 2020; 2023) and TD-MPC-style methods (Hansen et al., 2022; 2024) further combined latent rollouts with policy optimization, value learning, or model predictive control. More recent reward-free or goal-conditioned methods, including DINO-WM (Zhou et al., 2025), PLDM (Sobal et al., 2025), and LeWM (Maes et al., 2026), plan directly in learned representation spaces with optimizers such as CEM (Rubinstein, 1999). Recent analyses suggest that latent planning depends on properties beyond predictive accuracy. In particular, decision-metric alignment asks whether the latent metric used by MPC preserves decision-relevant rankings of candidate plans (Wang et al., 2026a). ATM examines a complementary aspect of planning-oriented representations: the structure of action-induced latent transitions, including whether action effects are identifiable and whether their action–transition relationship transfers between real and predicted domains.

Evaluation and diagnostics for world models. Latent world models are commonly evaluated by downstream success, returns, or rollout-based planning performance (Hafner et al., 2020; Hansen et al., 2022; Zhou et al., 2025; Maes et al., 2026). Such endpoint evaluation directly measures task performance but is planner-coupled, simulator-dependent, and often costly. Recent works also study post-hoc repairs or verifiers, including reachability-based planning costs (Li et al., 2026), action-reachability or action-consistency constraints (Liu et al., 2026; Wang et al., 2026b), and future–action compatibility for test-time selection (Ruan et al., 2026). A recent planning-time method also

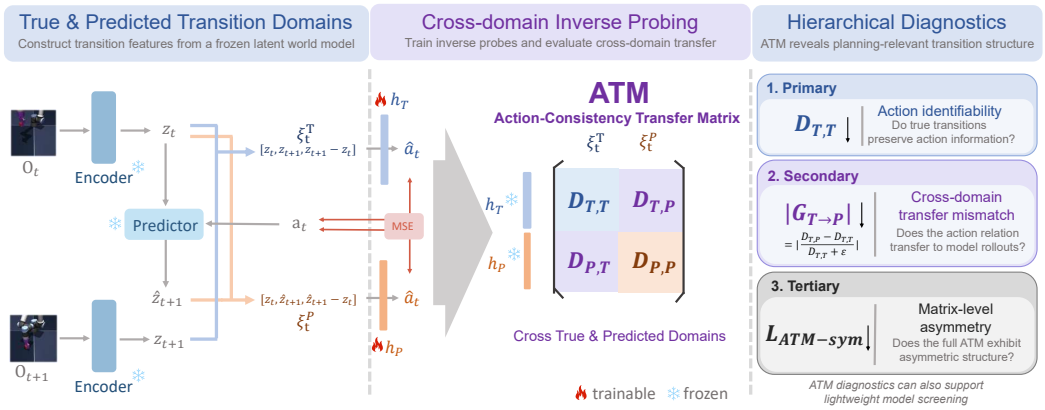

Figure 1: Overview of ATM. A frozen latent world model induces true and predicted transition domains, where inverse probes are trained and evaluated cross-domain to form the 2×2 Action-Consistency Transfer Matrix. The resulting hierarchy uses $D_{T,T}$ for action-identifiability, $|G_{T\to P}|$ for transfer mismatch, and $L_{\text{ATM-sym}}$ for matrix-level asymmetry. Snowflake and flame denote frozen and trainable modules, respectively.

uses inverse-dynamics consistency within predicted rollouts as an action-selection signal (Seo et al., 2026). ATM directly examines whether an action–transition relationship learned in one domain remains valid in the other. This yields a 2×2 cross-domain inverse-risk matrix without changing the planner or its objective.

Inverse dynamics and action-centric world modeling. Inverse dynamics has long been used to learn control-relevant representations, exploration signals, and action-conditioned features. Curiosity-driven exploration uses inverse dynamics to suppress uncontrollable visual variation (Pathak et al., 2017), while inverse-dynamics pretraining has been studied for imitation and visuomotor representation learning (Brandfonbrener et al., 2023). Contrastive inverse dynamics has been incorporated into JEPA-style world-model training, using action discrimination over latent transitions as an anti-collapse signal (Boylan & Hokamp, 2026). Recent action-sensitive world models encourage action recovery, action-conditioned transition separation, or action-aware dynamics (Zhang et al., 2026; Gan et al., 2026; Han & Yilmaz, 2026; Tian et al., 2026). These methods primarily investigate how to learn representations whose transitions remain sensitive to actions. ATM treats action-identifiability as a property of trained world models, formalizes it through Bayes inverse risk, and examines whether the action–transition relationship is preserved across real and predicted transition domains. AITS and AITS-P serve as controlled interventions to test this decomposition.

3 A TRANSITION-STRUCTURE VIEW OF LATENT PLANNING FAILURE

Figure 1 summarizes our transition-structure view of latent planning failure. We first formalize action-identifiability through Bayes inverse risk. We then distinguish real encoded from model-predicted transitions and ask whether their conditional action relationships transfer across domains. This yields a cross-domain inverse-risk formulation and the Action-Consistency Transfer Matrix (ATM). Finally, we introduce controlled interventions to manipulate these properties independently.

3.1 Latent Planning and Action-Identifiable Transitions

We consider goal-conditioned planning with an action-conditioned latent world model. An encoder E_θ maps observation o_t to $z_t = E_\theta(o_t)$, and a dynamics predictor produces $\hat{z}_{t+1} = F_\theta(z_t, a_t)$. For multi-step planning, an optimizer such as CEM (Rubinstein, 1999) rolls the predictor forward to find an action sequence whose predicted terminal state best matches the latent goal $z_g = E_\theta(o_g)$:

$$a^*_{t:t+H-1} = \arg\min_{a_{t:t+H-1}} d(\hat{z}_{t+H}, z_g), \qquad \hat{z}_{t+k+1} = F_\theta(\hat{z}_{t+k}, a_{t+k}). \tag{1}$$

Although the planning objective is expressed through the predicted future state, candidate actions affect it through latent transitions induced by the learned dynamics. The key question is whether different actions induce distinguishable structure in latent transition space. We formalize this as *action-identifiability*: how much information about an action remains recoverable from the latent transition it induces?

Consider an environment transition (o_t, a_t, o_{t+1}) and its latent feature $\xi_t = [z_t, z_{t+1}, z_{t+1} - z_t]$, where $z_{t+1} = E_\theta(o_{t+1})$. Let Ξ and A denote the corresponding random variables. We characterize action-identifiability by the minimum squared inverse-prediction risk. Under squared loss, the Bayes-optimal decoder is $h^*(\xi) = \mathbb{E}[A \mid \Xi = \xi]$, yielding

$$R^* = \inf_h \mathbb{E}\big[\|h(\Xi) - A\|_2^2\big] = \mathbb{E}[\mathrm{tr}\,\mathrm{Var}(A \mid \Xi)]\,. \tag{2}$$

Lower inverse risk means that the latent transition leaves less uncertainty about the inducing action, whereas higher risk indicates weaker action-identifiable structure. R^* is a property of the joint transition–action distribution and is independent of any particular inverse model. This gives action-identifiability a direct statistical interpretation. In practice, we estimate this quantity with the lightweight probes in Section 3.3.

Irreducible inverse ambiguity can arise when multiple actions induce equivalent transition consequences. In such cases, cross-domain preservation of the action–transition relationship is the relevant criterion.

Action-identifiability, however, is a within-domain property: it characterizes recoverable action information within a transition domain, but not whether the same action–transition relationship is shared across domains.

3.2 Real and Predicted Transition Domains

A latent world model induces two transition domains from the same tuple (o_t, a_t, o_{t+1}). The first is the *real encoded transition domain*, obtained by encoding the observed next state. The second is the *predicted transition domain*, obtained by applying the learned dynamics under the same current latent state and action. We represent them as

$$\xi_t^T = [\, z_t,\ z_{t+1},\ z_{t+1} - z_t\,], \qquad \xi_t^P = [\, z_t,\ \hat{z}_{t+1},\ \hat{z}_{t+1} - z_t\,], \tag{3}$$

where $z_{t+1} = E_\theta(o_{t+1})$ and $\hat{z}_{t+1} = F_\theta(z_t, a_t)$. The two features share the same current latent state and action, differing only in whether the next latent state comes from environment evolution or the learned predictor.

The two domains need not encode actions through the same relationship. Let $\mathcal{D}_T$ and $\mathcal{D}_P$ denote their joint transition–action distributions. They induce different conditional action relationships, $p_T(a \mid \xi^T)$ and $p_P(a \mid \xi^P)$. Under squared loss, their Bayes-optimal inverse rules are

$$h_T^*(\xi) = \mathbb{E}_{\mathcal{D}_T}[A \mid \Xi^T = \xi], \qquad h_P^*(\xi) = \mathbb{E}_{\mathcal{D}_P}[A \mid \Xi^P = \xi]. \tag{4}$$

By Eq. (2), the in-domain Bayes risks characterize action-identifiability within each domain. Low inverse risk in both domains, however, does not imply a shared action–transition relationship.

This distinction is particularly important for model-predicted transitions. Since the action is an explicit predictor input, predicted transitions may encode action information in a model-specific form that is highly decodable within the model's rollout space. Low predicted-domain inverse risk therefore shows only that actions are recoverable from model-generated transitions; it does not imply the same action–transition relationship in real encoded transitions.

This separates *in-domain action decodability* from *cross-domain action transfer*. If the two domains preserve compatible action-conditioned structure, an inverse relationship learned in one should remain informative in the other. Conversely, cross-domain degradation indicates that the action–transition relationship changes between real and predicted dynamics. We operationalize this transfer principle with the Action-Consistency Transfer Matrix in the next section.

3.3 Cross-Domain Inverse Transfer and ATM

We operationalize the domain-specific Bayes inverse rules from Section 3.2 with a shared lightweight probe family. For each world model, the encoder and predictor remain frozen; we train probes h_T and h_P on $\mathcal{D}_T$ and $\mathcal{D}_P$, respectively. Both probes use the same two-layer MLP with LayerNorm and GELU; token- or patch-level latents are mean-pooled before feature construction.

For $i, j \in \{T, P\}$, we define

$$\begin{aligned} h_i &\approx \arg\min_h \mathbb{E}_{(\xi,a)\sim\mathcal{D}_i}\left[\|h(\xi) - a\|_2^2\right], \\ D_{i,j} &= \mathbb{E}_{(\xi,a)\sim\mathcal{D}_j}\left[\|h_i(\xi) - a\|_2^2\right], \qquad \mathbf{A}_{\mathrm{ATM}} = \begin{bmatrix} D_{T,T} & D_{T,P} \\ D_{P,T} & D_{P,P} \end{bmatrix}. \end{aligned} \tag{5}$$

Rows of $\mathbf{A}_{\mathrm{ATM}}$ denote the training domain and columns the evaluation domain. Diagonal entries measure in-domain action decodability, while off-diagonal entries measure transfer of the learned action–transition relationship.

Specifically, $D_{T,T}$ approximates true-domain action-identifiability, while $D_{T,P}$ tests whether this inverse relationship transfers to predicted transitions. Conversely, $D_{P,P}$ measures predicted-domain self-decodability, while $D_{P,T}$ tests whether this relationship transfers to real transitions. Thus, $D_{P,P} \ll D_{P,T}$ exposes a *predicted-domain-specific action code*: actions are easy to recover in the model's rollout space, but the same decoding relationship fails on real transitions.

We summarize directional cross-domain transfer with the relative gaps

$$G_{T\to P} = \frac{D_{T,P} - D_{T,T}}{D_{T,T} + \epsilon}, \quad G_{P\to T} = \frac{D_{P,T} - D_{P,P}}{D_{P,P} + \epsilon}. \tag{6}$$

Here ϵ is a small stability constant. $G_{T\to P}$ measures the relative change when the true-domain inverse relationship transfers to predicted transitions, and $G_{P\to T}$ the reverse. Because cross-domain evaluation can occasionally improve inverse prediction, we use magnitudes to measure mismatch and signed values to analyze transfer direction.

ATM is thus an empirical cross-domain inverse-risk matrix whose diagonal entries quantify within-domain action-identifiability and off-diagonal entries cross-domain transfer. The full matrix can distinguish weak action information from domain-specific action coding—a distinction that no single inverse loss can reveal.

3.4 Matrix-Level Asymmetry

The directional gaps characterize directional transfer mismatch, while the full ATM may also exhibit asymmetry between the real and predicted domains. We summarize this behavior with a compact matrix-level statistic, retaining individual entries and directional gaps as the primary diagnostics.

We first measure the discrepancy between the two in-domain inverse risks as

$$\begin{aligned} I_{\mathrm{diag}} &= \frac{|D_{T,T} - D_{P,P}|}{\frac{1}{2}(D_{T,T} + D_{P,P}) + \epsilon}, \\ L_{\mathrm{raw}} &= I_{\mathrm{diag}} + |G_{T\to P} - G_{P\to T}|, \\ L_{\text{ATM-sym}} &= \frac{\log(1 + L_{\mathrm{raw}})}{1 + \log(1 + L_{\mathrm{raw}})}. \end{aligned} \tag{7}$$

Here, I_{diag} measures in-domain decodability discrepancy, while the second term measures directional transfer asymmetry. Logarithmic compression limits extreme relative gaps, and the final monotonic mapping bounds $L_{\text{ATM-sym}}$ to $[0, 1)$. Near-zero values indicate similar in-domain decodability and relatively symmetric transfer; larger values indicate stronger matrix-level asymmetry.

$L_{\text{ATM-sym}}$ is a tertiary rather than standalone diagnostic. Lower asymmetry is preferable when the primary ATM quantities are comparable, but should not override large differences in action-identifiability or directional transfer. We therefore use it as a tie-breaking summary, with $D_{T,T}$ and $|G_{T\to P}|$ as the primary diagnostics.

3.5 Controlled Interventions on Transition Structure

The diagnostics above characterize transition structure after training. To test whether the identified properties respond differently to targeted domain-specific changes, we introduce two training-time interventions. These interventions selectively increase action decodability in real or predicted transitions while leaving the planner unchanged.

Let h_ψ^T and h_ψ^P denote training-time inverse heads for real and predicted transition features. We define

$$\begin{aligned} \mathcal{L}_T &= \mathbb{E}\big[\|h_\psi^T(\xi_t^T) - a_t\|_2^2\big], \\ \mathcal{L}_P &= \mathbb{E}\big[\|h_\psi^P(\xi_t^P) - a_t\|_2^2\big], \\ \mathcal{L} &= \mathcal{L}_{\mathrm{WM}} + \alpha_T \mathcal{L}_T + \alpha_P \mathcal{L}_P. \end{aligned} \tag{8}$$

We call the true-transition intervention ($\alpha_T > 0, \alpha_P = 0$) *Action-Identifiable Transition Supervision* (AITS), and the predicted-transition intervention ($\alpha_T = 0, \alpha_P > 0$) AITS-P. We also consider their combination by enabling both terms. AITS encourages action-identifiable real transitions, whereas AITS-P encourages action decodability in the predicted transition space.

Neither intervention directly optimizes cross-domain transfer. Their comparison therefore tests whether increasing in-domain action decodability in different domains yields the same cross-domain structure. In particular, stronger predicted-domain decodability does not by construction require the learned inverse relationship to remain valid on real transitions.

The training-time inverse heads are discarded after training. For ATM evaluation, we freeze the encoder and predictor and train fresh probes h_T and h_P from scratch. This ensures that ATM reflects the learned world model rather than the auxiliary intervention heads.

4 Experiments

We organize our experiments around the diagnostic hierarchy above. First, we test whether true-transition action-identifiability captures planning-relevant variation beyond prediction loss and whether cross-domain quantities provide further refinement. Second, we inspect the full ATM for failure modes invisible to in-domain decodability alone. Third, controlled interventions test whether targeted changes in the real and predicted domains yield distinct cross-domain structures and planning outcomes. Finally, we evaluate the same diagnostics for lightweight model screening.

4.1 Experimental Setup

Tasks. We evaluate three goal-conditioned planning tasks: TwoRoom, PushT, and OGBench-Cube (Park et al., 2025). TwoRoom is 2D navigation, PushT continuous-control 2D manipulation, and OGBench-Cube more complex 3D manipulation. Together, they span increasingly challenging transition structures, from navigation to contact-rich manipulation and visually complex 3D control.

Models. Our controlled experiments use LeWM-style (Maes et al., 2026) latent world models. From a shared LeWM baseline, we compare AITS, AITS-P, and their combination, selectively intervening on action decodability in the real domain, predicted domain, or both. We also include DINO-WM (Zhou et al., 2025) checkpoints as an external family to test whether similar real–predicted transition structure arises beyond these interventions. All variants within each task use the same CEM planning protocol.

4.2 A Hierarchy of Planning-Relevant Transition Diagnostics

We first test whether action-identifiability captures planning-relevant variation beyond prediction loss and whether cross-domain transfer adds discrimination. We focus on OGBench-Cube, whose less-saturated success rates make it more suitable for studying continuous relationships between transition diagnostics and planning performance.

Table 1: Correlation with downstream planning success on OGBench-Cube. Higher is better. Here, λ is fitted within the same task on the candidate pool.

Score	Spearman ρ	Pearson r
$-\mathcal{L}_{\text{pred}}$	0.4983	0.3927
$-D_{T,T}$	0.8130	**0.7015**
$-D_{T,T} - \lambda\lvert G_{T\to P}\rvert$	**0.8239**	0.7006

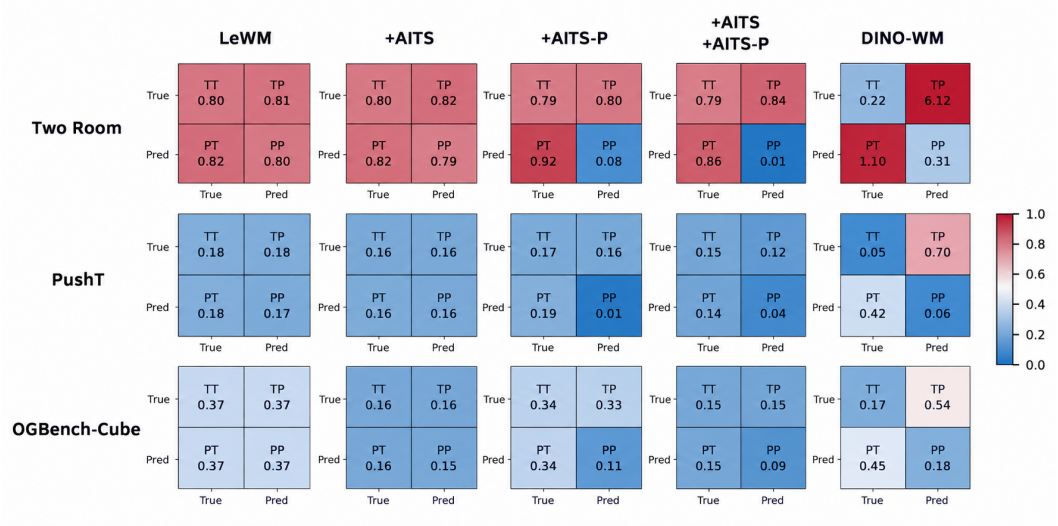

Figure 2: Full ATM matrices across tasks and model variants. Rows denote training domains and columns evaluation domains. Low diagonal loss indicates strong in-domain action decodability, while diagonal-to-cross-domain gaps reveal transfer mismatch. Predicted-transition supervision can yield low $D_{P,P}$ but much larger $D_{P,T}$, exposing a self-decodable yet domain-specific action code.

Table 1 compares latent prediction loss, true-transition action-identifiability, and a simple combination additionally penalizing true-to-predicted transfer mismatch. All quantities are oriented so higher values indicate better quality.

The largest gain comes from replacing prediction loss with true-transition action-identifiability: Spearman correlation increases from 0.4983 to 0.8130 and Pearson correlation from 0.3927 to 0.7015. This supports $D_{T,T}$ as the primary planning-relevant diagnostic. Adding the true-to-predicted transfer term further improves rank correlation to 0.8239, while leaving linear correlation nearly unchanged, suggesting that cross-domain structure provides secondary discrimination beyond in-domain identifiability.

This supports the hierarchy in Section 3: $D_{T,T}$ captures the dominant planning-relevant variation, while $|G_{T\to P}|$ adds discrimination when in-domain action-identifiability is insufficient. Higher-order matrix structure is analyzed next.

4.3 Cross-Domain Transfer Reveals Hidden Failure Modes

Having established true-transition action-identifiability as the primary diagnostic, we next examine what cross-domain structure reveals beyond in-domain decodability. Figure 2 shows the full ATM matrices across model variants, families, and all three tasks.

A recurring pattern is that strong in-domain decodability does not imply cross-domain transfer of the same action–transition relationship. Several predicted-transition-supervised models achieve low $D_{P,P}$ but substantially larger $D_{P,T}$: actions are highly decodable within predicted rollouts, yet the inverse relationship fails on real transitions. This is the predicted-domain-specific action structure characterized in Section 3.3.

Table 2: Controlled intervention analysis across three tasks. $D_{T,T}$ and $D_{P,P}$ measure in-domain action decodability, $|G_{T\to P}|$ measures true-to-predicted transfer mismatch, and $L_{\text{ATM-sym}}$ summarizes higher-order matrix asymmetry. All ATM diagnostic values are scaled by 100 for readability. Lower $L_{\text{ATM-sym}}$ is preferable as a tertiary criterion when the primary diagnostics are comparable. Success denotes downstream planning success (%).

Task	Variant	$D_{T,T}\downarrow$	$D_{P,P}\downarrow$	$\lvert G_{T\to P}\rvert\downarrow$	$L_{\text{ATM-sym}}\downarrow$	Success↑
TwoRoom	LeWM	80.26	80.00	1.28	0.95	87
TwoRoom	+ AITS	79.56	79.00	2.50	1.15	92
TwoRoom	+ AITS-P	79.36	8.00	0.68	72.43	94
TwoRoom	+ AITS + AITS-P	79.13	1.00	6.09	81.15	92
PushT	LeWM	18.29	17.00	3.72	16.94	96
PushT	+ AITS	16.23	16.00	1.22	7.80	100
PushT	+ AITS-P	17.34	1.00	10.56	72.83	100
PushT	+ AITS + AITS-P	15.34	4.00	24.99	62.88	98
OGBench-Cube	LeWM	36.65	37.00	0.45	1.07	74
OGBench-Cube	+ AITS	15.87	15.00	0.96	7.44	84
OGBench-Cube	+ AITS-P	33.77	11.00	1.60	58.13	78
OGBench-Cube	+ AITS + AITS-P	15.40	9.00	4.57	45.03	84

The full ATM therefore reveals distinctions invisible to either diagonal alone. Models with similar in-domain inverse risks can differ markedly in cross-domain transfer, while highly decodable predicted transitions can still rely on domain-specific action codes. Hence, cross-domain quantities provide secondary discrimination beyond true-transition action-identifiability.

This pattern appears across tasks and model configurations. DINO-WM (Zhou et al., 2025) checkpoints further show that real–predicted transition mismatch arises in an independently trained world-model family beyond our controlled transition-supervision objectives.

Together, these observations support ATM's central distinction: *action information can be highly decodable within a transition domain without the underlying action–transition relationship transferring across real and predicted dynamics.* We next test this distinction directly through controlled interventions.

4.4 Controlled Interventions Separate Decodability from Transfer

We next use the controlled interventions from Section 3.5 to test whether increasing action decodability in different domains yields the same learned structure. Neither AITS nor AITS-P directly optimizes cross-domain transfer: AITS targets real-transition decodability, whereas AITS-P targets predicted-transition decodability.

Table 2 shows distinct ATM signatures across the interventions. AITS improves true-transition action-identifiability and planning across all three tasks, increasing success from 87% to 92% on TwoRoom, 96% to 100% on PushT, and 74% to 84% on OGBench-Cube. On OGBench-Cube, $D_{T,T}$ decreases from 36.65 to 15.87.

AITS-P produces a different structure. It can sharply reduce $D_{P,P}$, making predicted transitions highly self-decodable without a correspondingly transferable action relationship. Figure 2 shows a large discrepancy between $D_{P,P}$ and $D_{P,T}$, revealing a predicted-domain-specific action code. Thus, predicted-domain decodability alone does not ensure compatibility with real transitions.

Planning results further clarify the hierarchy. AITS-P remains effective on the simpler TwoRoom and PushT tasks despite large matrix asymmetry, but on OGBench-Cube improves success only from 74% to 78%, versus 84% under AITS. Applying both interventions likewise does not consistently outperform AITS alone. These results support ATM's distinction: in-domain action decodability is useful, but where it arises—and whether the corresponding relationship transfers—can matter for planning, especially with more demanding transition structure.

Table 3: Pairwise model-ranking accuracy on OGBench-Cube using $S_{\mathrm{ATM}} = -D_{T,T} - \lambda|G_{T\to P}|$. Margins exclude pairs with only small differences in simulator-based success.

Success margin	n_{pairs}	Ranking acc.	Weighted acc.
All pairs	1865	82.84	92.61
$> 2\%$	1338	91.48	95.14
$> 5\%$	841	98.81	99.09
$> 10\%$	123	100.00	100.00

4.5 Practical Model Screening with ATM

Finally, we ask whether the transition-level quantities above can also support practical model selection. Following Section 4.2, we use a simple task-specific score combining true-transition action-identifiability with true-to-predicted transfer mismatch:

$$S_{\mathrm{ATM}} = -D_{T,T} - \lambda|G_{T\to P}|, \tag{9}$$

where λ is fitted within the same task. We use this score only for relative ranking rather than as a calibrated predictor of absolute planning success.

Table 3 reports pairwise ranking on OGBench-Cube, whose candidates span a broad range of planning performance. A pair is correct when S_{ATM} agrees with the simulator-based success ordering.

The score correctly ranks 82.84% of all pairs and becomes more reliable as the performance gap increases. For gaps above 5%, accuracy reaches 98.81%; all pairs above 10% are ranked correctly. Thus, the same diagnostics provide a lightweight signal for selecting meaningfully different candidate models.

We further compare LeWM-style and DINO-WM candidates across families. Transferring the coefficient fitted on LeWM-style models yields 68.54% accuracy on LeWM–DINO-WM pairs. Refitting the single coefficient on the mixed pool raises cross-family accuracy to 89.90%, with 83.10% over all pooled pairs. This suggests that the underlying ATM quantities remain informative across model families, while their relative scales may require lightweight calibration.

Model screening is a practical consequence rather than the defining purpose of ATM: once latent transitions are cached, ATM takes only 3–5 seconds per checkpoint in our setup, versus minutes to hours for simulator-based CEM evaluation. The core contribution remains separating in-domain action-identifiability from cross-domain transfer.

5 Conclusion

We study why latent world models with similar prediction quality can exhibit different planning performance. We identify action-induced transition structure as a key factor: true-transition action-identifiability provides the primary signal, while cross-domain transfer reveals failure modes. We formalize these properties through Bayes inverse risk and jointly diagnose them with the Action-Consistency Transfer Matrix (ATM). Across three tasks, action-identifiability tracks planning substantially better than prediction loss, while the full ATM exposes highly self-decodable yet domain-specific predicted transitions; controlled interventions produce distinct transition structures and planning outcomes. These results identify two sources of latent planning failure: weak action information and failure of predicted-transition action structure to transfer to real dynamics.

6 Discussion

ATM also provides a lens on desirable properties of planning-oriented latent representations. Our results suggest a hierarchy: strong true-transition action-identifiability is primary, while cross-domain transfer and matrix balance provide refinement when identifiability is comparable. World-model families exhibit different trade-offs along this hierarchy, suggesting that effective latent representations should first preserve clear action-induced structure in real transitions and then maintain compatible action relationships in predicted transitions.

AI USE STATEMENT

We used generative AI tools to provide feedback on research methodology and experimental presentation, assist in interpreting experimental results, support code implementation and debugging, and assist with manuscript drafting and editing. We additionally used generative AI for literature search and summarization, identifying relevant work, and refining figures and paper structure. All AI-assisted suggestions were independently reviewed by the authors; cited literature was verified against original sources, AI-assisted code was reviewed and tested against the intended experimental behavior, reported numerical results were checked against our experimental outputs, and all methodological decisions and final claims were made by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed experimental and implementation information in the appendix. Appendix A specifies the evaluated tasks and world models, ATM transition construction and probe protocol, and the controlled-intervention setup. The theoretical derivations underlying the Bayes inverse-risk formulation are provided in Appendix B. Additional screening and cross-family evaluation details are provided in Appendix C, and computational-cost measurements in Appendix D. Code will be released upon acceptance.

REFERENCES

Jack Boylan and Chris Hokamp. No gaussian required: Contrastive inverse dynamics for JEPA world models. *CoRR*, abs/2608.17542, 2026. doi: 10.48550/arXiv.2608.17542.

David Brandfonbrener, Ofir Nachum, and Joan Bruna. Inverse Dynamics Pretraining Learns Good Representations for Multitask Imitation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, December 2023. doi: 10.48550/ARXIV.2305.16985. URL http://papers.nips.cc/paper_files/paper/2023/hash/d36dfcdb14473a8526111c221660f2ab-Abstract-Conference.html.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-Domain Dynamics Pretraining for Visuo-Motor Control. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Annual Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, December 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3b8db54b629e00537b59cbc6612026d7-Abstract-Conference.html.

Zhenfeng Gan, ZiTong Zeng, Jiajun Cheng, Yeke Song, Yongyi Tang, and Xueqian Wang. ActSWM: Action-sensitive world models for long-horizon planning in open-world games. *CoRR*, abs/2607.26712, 2026. doi: 10.48550/arXiv.2607.26712.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552, 2020.

David Ha and Jürgen Schmidhuber. World Models. *CoRR*, abs/1803.10122, 2018. doi: 10.5281/zenodo.1207631. URL http://arxiv.org/abs/1803.10122. arXiv: 1803.10122.

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 2555–2565, Long Beach, California, USA, June 2019. PMLR. doi: 10.48550/ARXIV.1811.04551. URL http://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 2020. OpenReview.net. doi: 10.48550/ARXIV.1912.01603. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering Diverse Domains through World Models. *CoRR*, abs/2301.04104, 2023. doi: 10.48550/ARXIV.2301.04104. URL https://doi.org/10.48550/arXiv.2301.04104. arXiv: 2301.04104.

Yuci Han and Alper Yilmaz. Enhancing policy learning with world-action model. *CoRR*, abs/2603.28955, 2026. doi: 10.48550/arXiv.2603.28955.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Temporal Difference Learning for Model Predictive Control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 8387–8406, Baltimore, Maryland, USA, July 2022. PMLR. URL https://proceedings.mlr.press/v162/hansen22a.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, May 2024. OpenReview.net. doi: 10.48550/ARXIV.2310.16828. URL https://openreview.net/forum?id=Oxh5CstDJU.

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*, 2026. doi: 10.48550/ARXIV.2605.22164. URL https://doi.org/10.48550/arXiv.2605.22164.

Yuejiang Liu, Fan Feng, Lingjing Kong, Weifeng Lu, Jinzhou Tang, Kun Zhang, Kevin Murphy, Chelsea Finn, and Yilun Du. World Action Verifier: Self-Improving World Models via Forward-Inverse Asymmetry. *CoRR*, abs/2604.01985, 2026. doi: 10.48550/ARXIV.2604.01985. URL https://doi.org/10.48550/arXiv.2604.01985. arXiv: 2604.01985.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels. *CoRR*, abs/2603.19312, 2026. doi: 10.48550/ARXIV.2603.19312. URL https://doi.org/10.48550/arXiv.2603.19312. arXiv: 2603.19312.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking Offline Goal-Conditioned RL. In *International Conference on Learning Representations*, Singapore, April 2025. OpenReview.net. URL https://openreview.net/forum?id=M992mjgKzI.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction. In Doina Precup and Yee Whye Teh (eds.), *International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 2778–2787, Sydney, NSW, Australia, August 2017. PMLR. URL http://proceedings.mlr.press/v70/pathak17a.html.

Bo-Kai Ruan, Teng-Fang Hsiao, Ling Lo, and Hong-Han Shuai. Is the Future Compatible? Diagnosing Dynamic Consistency in World Action Models. *arXiv preprint arXiv:2605.07514*, May 2026. doi: 10.48550/arXiv.2605.07514. URL https://doi.org/10.48550/arXiv.2605.07514.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Gawon Seo, Dongwon Kim, and Suha Kwak. ACID: Action consistency via inverse dynamics for planning with world models. *CoRR*, abs/2607.02403, 2026. doi: 10.48550/arXiv.2607.02403.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from Reward-Free Offline Data: A Case for Planning with Latent Dynamics Models. *CoRR*, abs/2502.14819, 2025. doi: 10.48550/ARXIV.2502.14819. URL https://doi.org/10.48550/arXiv.2502.14819. arXiv: 2502.14819.

Yuxuan Tian, Yurun Jin, Bin Yu, Yukun Shi, Hao Wu, Chi Harold Liu, Kai Chen, and Cong Huang. STARRY: Spatial-Temporal Action-Centric World Modeling for Robotic Manipulation. *CoRR*, abs/2604.26848, 2026. doi: 10.48550/ARXIV.2604.26848. URL https://doi.org/10.48550/arXiv.2604.26848. arXiv: 2604.26848.

Jiawei Wang, Ke Rui, Yushen Zuo, Yichun Feng, and Minglei Li. Decision-metric alignment in latent world models: Diagnostics and action-conditioned objectives for MPC planning. *CoRR*, abs/2608.18746, 2026a. doi: 10.48550/arXiv.2608.18746.

Zijie Wang, Wei Zhang, Weiming Zhang, Fanqi Zhang, Xiao Tan, Yipeng Qin, and Guanbin Li. World Models as Group Actions. *arXiv preprint arXiv:2605.24578*, 2026b. doi: 10.48550/ARXIV.2605.24578. URL https://doi.org/10.48550/arXiv.2605.24578.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-JEPA: Learning action-sensitive world models via latent difference decoding. *CoRR*, abs/2606.31232, 2026. doi: 10.48550/arXiv.2606.31232.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vancouver, BC, Canada, July 2025. PMLR / OpenReview.net. URL https://proceedings.mlr.press/v267/zhou25t.html.

A Experimental and Implementation Details

A.1 Tasks and Evaluated World Models

We evaluate ATM on the same three goal-conditioned planning tasks used in the main experiments: TwoRoom, PushT, and OGBench-Cube (Park et al., 2025). TwoRoom is a 2D navigation task, PushT is a continuous-control 2D manipulation task, and OGBench-Cube is a visually more complex 3D manipulation task. Together, they span increasingly challenging transition structures, from navigation to contact-rich manipulation and 3D control.

Our controlled experiments use LeWM-style (Maes et al., 2026) latent world models. Starting from the same baseline, we consider AITS, AITS-P, and their combination, which intervene on action decodability in the true transition domain, the predicted transition domain, or both. We additionally include DINO-WM (Zhou et al., 2025) checkpoints as an independently trained world-model family. All variants within each task are evaluated under the same CEM planning protocol used for the corresponding comparisons in the main paper.

A.2 ATM Transition Features and Probe Protocol

For each offline transition tuple (o_t, a_t, o_{t+1}), the frozen world model produces the true and predicted transition features

$$\xi_t^T = [z_t, z_{t+1}, z_{t+1} - z_t], \tag{10}$$

$$\xi_t^P = [z_t, \hat{z}_{t+1}, \hat{z}_{t+1} - z_t], \tag{11}$$

where

$$z_t = E_\theta(o_t), \qquad z_{t+1} = E_\theta(o_{t+1}), \qquad \hat{z}_{t+1} = F_\theta(z_t, a_t).$$

The two transition domains are therefore constructed from the same underlying transition tuple and share the same current latent state and action; they differ only in whether the next latent state is obtained from actual environment evolution or from the learned predictor.

For vector-valued latent states, the latent representation is used directly. For token- or patch-level representations, we first apply mean pooling over tokens before constructing the transition feature. This gives a fixed-size representation across the evaluated world-model families.

Both ATM probes use the same lightweight two-layer MLP architecture with LayerNorm and GELU activation,

$$h(\xi) = W_2\,\sigma(\mathrm{LN}(W_1\xi + b_1)) + b_2.$$

For every evaluated world-model checkpoint, the encoder and dynamics predictor remain frozen, and the true-domain probe h_T and predicted-domain probe h_P are trained from scratch. Each probe is then evaluated on both transition domains to form

$$\mathbf{A}_{\mathrm{ATM}} = \begin{bmatrix} D_{T,T} & D_{T,P} \\ D_{P,T} & D_{P,P} \end{bmatrix}.$$

This separation ensures that ATM measures action information accessible in the frozen transition representation rather than the performance of a head trained jointly with the world model.

A.3 Controlled-Intervention Protocol

The controlled interventions used in the main paper modify the domain in which action decodability is explicitly encouraged. With training-time inverse heads h_ψ^T and h_ψ^P, we use

$$\mathcal{L}_T = \mathbb{E}\left[\|h_\psi^T(\xi_t^T) - a_t\|_2^2\right], \tag{12}$$

$$\mathcal{L}_P = \mathbb{E}\left[\|h_\psi^P(\xi_t^P) - a_t\|_2^2\right], \tag{13}$$

and

$$\mathcal{L} = \mathcal{L}_{\mathrm{WM}} + \alpha_T\mathcal{L}_T + \alpha_P\mathcal{L}_P.$$

The variants correspond to the following intervention structure:

Table 4: Controlled interventions used in the experiments. The table indicates which transition domain receives explicit action-decoding supervision.

Variant	α_T	α_P
LeWM baseline	0	0
AITS	> 0	0
AITS-P	0	> 0
AITS + AITS-P	> 0	> 0

The training-time inverse heads are discarded after world-model training. ATM evaluation always uses newly initialized post-hoc probes on the frozen world model. Consequently, neither AITS nor AITS-P directly optimizes the cross-domain ATM quantities evaluated after training.

B Additional Theoretical Details

B.1 From Latent Planning to Action-Induced Transitions

For goal-conditioned latent planning, a planner such as CEM searches for an action sequence by rolling the learned dynamics forward:

$$\hat{z}_{t+k+1} = F_\theta(\hat{z}_{t+k}, a_{t+k}), \qquad k = 0, \ldots, H-1,$$

and selecting

$$a^*_{t:t+H-1} = \arg\min_{a_{t:t+H-1}} d(\hat{z}_{t+H}, z_g).$$

Equivalently, the planning objective can be viewed as a function of the action sequence,

$$J_\theta(a_{t:t+H-1}) = d\Big(F_\theta^{(H)}(z_t, a_{t:t+H-1}), z_g\Big).$$

Thus, although the endpoint objective is expressed as latent goal matching, candidate actions affect that objective through the latent transitions induced by the learned dynamics. Locally, for a perturbation δa,

$$F_\theta(z_t, a_t + \delta a) \approx F_\theta(z_t, a_t) + J_a F_\theta(z_t, a_t)\delta a,$$

where $J_a F_\theta$ denotes the action Jacobian of the latent dynamics. This motivates examining whether action-induced latent transitions retain distinguishable action information.

ATM does not attempt to estimate this Jacobian directly, nor is inverse decodability an algebraic consequence of the CEM objective. Instead, it provides a model-agnostic diagnostic of the transition interface through which candidate actions affect latent planning.

B.2 Bayes Inverse Risk

Let Ξ denote a latent transition feature and A the corresponding action. For any inverse predictor h, the squared inverse risk is

$$R(h) = \mathbb{E}\left[\|A - h(\Xi)\|_2^2\right].$$

Conditioning on Ξ gives the standard decomposition

$$\begin{aligned}\mathbb{E}\left[\|A - h(\Xi)\|_2^2 \mid \Xi\right] &= \mathbb{E}\left[\|A - \mathbb{E}[A \mid \Xi]\|_2^2 \mid \Xi\right] \\ &\quad + \|\mathbb{E}[A \mid \Xi] - h(\Xi)\|_2^2. \end{aligned} \tag{14}$$

The second term is non-negative and vanishes for

$$h^*(\xi) = \mathbb{E}[A \mid \Xi = \xi].$$

Therefore,

$$R^* = \inf_h R(h) = \mathbb{E}\left[\operatorname{tr} \operatorname{Var}(A \mid \Xi)\right].$$

This identity gives action-identifiability a population-level interpretation. A lower Bayes inverse risk means that less uncertainty about the action remains after observing the induced transition. The lightweight ATM probes provide an empirical approximation to this quantity for a frozen world model.

Importantly, irreducible inverse ambiguity does not necessarily indicate a deficient representation. If multiple actions induce equivalent transition consequences, the corresponding action labels need not be uniquely recoverable. In such cases, the relevant question is whether the same action–transition relationship, including its intrinsic ambiguity, is preserved across the true and predicted transition domains.

B.3 Why Cross-Domain Transfer Is Needed

The true and predicted transition domains induce potentially different conditional action relationships,

$$p_T(a \mid \xi^T) \qquad \text{and} \qquad p_P(a \mid \xi^P),$$

with Bayes-optimal inverse rules

$$h_T^*(\xi) = \mathbb{E}_{D_T}[A \mid \Xi^T = \xi], \qquad h_P^*(\xi) = \mathbb{E}_{D_P}[A \mid \Xi^P = \xi].$$

A single in-domain inverse risk only determines whether action information is decodable within one transition domain. It cannot determine whether the learned inverse relationship remains valid in the other domain. ATM therefore evaluates each probe both in-domain and cross-domain:

$$D_{i,j} = \mathbb{E}_{(\xi,a)\sim D_j}\left[\|h_i(\xi) - a\|_2^2\right], \qquad i, j \in \{T, P\}.$$

In particular, the pattern

$$D_{P,P} \ll D_{P,T}$$

shows that actions are highly decodable inside the model-predicted transition domain while the corresponding inverse rule fails to transfer back to real encoded transitions. This is precisely the predicted-domain-specific action code discussed in the main paper.

Accordingly, ATM should be interpreted operationally as a cross-domain inverse transfer test rather than as a direct estimator of a distributional divergence. Its purpose is to determine whether an action–transition relationship learned in one domain remains informative in the other.

The directional gaps

$$G_{T\to P} = \frac{D_{T,P} - D_{T,T}}{D_{T,T} + \epsilon}, \qquad G_{P\to T} = \frac{D_{P,T} - D_{P,P}}{D_{P,P} + \epsilon}$$

summarize degradation in the two transfer directions. As in the main paper, we treat $D_{T,T}$ as the primary diagnostic, cross-domain transfer as a secondary diagnostic, and matrix-level asymmetry as a tertiary summary rather than a standalone quality score.

C Additional Screening and Cross-Family Analysis

C.1 Pairwise Ranking Protocol

For OGBench-Cube, the screening score used in the main experiments is

$$S_{\mathrm{ATM}} = -D_{T,T} - \lambda |G_{T\to P}|.$$

The score is used only for relative model ranking rather than as a calibrated predictor of absolute success.

For a pair of candidate models, ranking is counted as correct when the ordering induced by S_{ATM} agrees with the ordering induced by simulator-based planning success. Success-margin evaluation removes pairs whose simulator success rates differ by less than the specified margin. This avoids placing the same emphasis on nearly indistinguishable candidates and candidates whose downstream performance differs substantially.

C.2 Cross-Family Ranking

Table 5 provides the complete cross-family comparison between LeWM-style (Maes et al., 2026) and DINO-WM (Zhou et al., 2025) candidates on OGBench-Cube. The transferred setting uses the coefficient fitted only on LeWM-style candidates, whereas the refit setting calibrates the same single coefficient on the mixed candidate pool.

Table 5: Cross-family ranking on OGBench-Cube. DINO-only ranking is included as a stress case because it contains only 40 pairs over a relatively narrow success range. The LeWM–DINO pairs are the primary cross-family comparison.

Setting	Pair type	λ	n_{pairs}	Ranking acc.
Transferred	DINO-only	0.036	40	70.00
Transferred	LeWM–DINO	0.036	604	68.54
Transferred	All pooled	0.036	2509	78.24
Refit	DINO-only	0.005	40	32.50
Refit	LeWM–DINO	0.005	604	89.90
Refit	All pooled	0.005	2509	83.10

The transferred results show that direct score calibration does not completely transfer across world-model families. However, refitting only the single coefficient on the mixed candidate pool substantially improves LeWM–DINO ranking. This supports the interpretation in the main paper that the underlying ATM quantities remain informative across model families while their relative scales can differ.

C.3 Cross-Family Calibration Visualization

Figure 3 provides a complementary visualization of this analysis. A low-capacity spline is fitted on the pooled candidate set only for visualization; it is not used to define ATM or to produce the main pairwise ranking results.

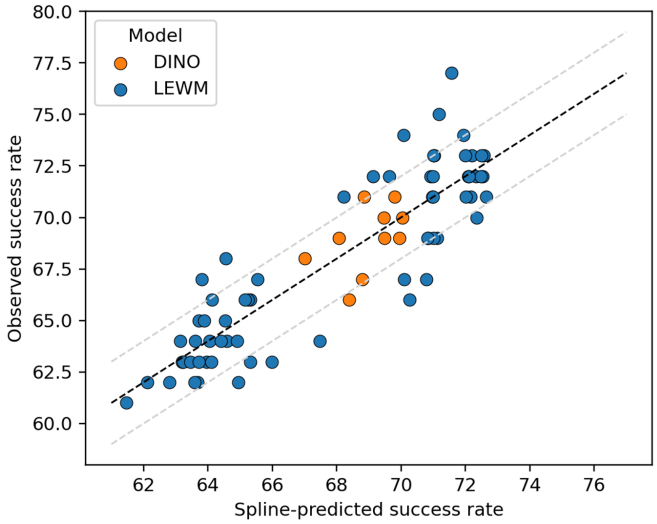

Figure 3: Cross-family calibration on OGBench-Cube. A low-capacity spline maps ATM diagnostics to downstream planning success for visualization. The calibration is not used as the primary screening score or for the main pairwise-ranking results.

D Computational Cost

The screening experiments also differ substantially from the simulator-based CEM evaluation used in this work in their wall-clock cost. Table 6 reports the measured evaluation time under our experimental setup. CEM evaluation requires repeated planner–model interaction over simulator episodes, whereas ATM operates on cached latent transitions and trains only the lightweight post-hoc probes.

Table 6: Wall-clock cost under the evaluation protocol used in this work. ATM timing is measured after latent transitions have been cached. These numbers characterize the practical screening cost in our setup rather than an intrinsic efficiency advantage over all possible planning methods.

Protocol	Simulator rollout?	50 episodes	100 episodes
LeWM CEM evaluation (Maes et al., 2026)	Yes	$\sim$ 3 min	$\sim$ 7 min
DINO-WM CEM evaluation (Zhou et al., 2025)	Yes	$\sim$ 1.4 h	$\sim$ 2.8 h
ATM diagnostics	No	3–5 s	

Once latent transitions are cached, ATM requires only approximately 3–5 seconds per checkpoint in our setup. In comparison, the simulator-based CEM evaluation used here requires minutes for LeWM-style models and hours for DINO-WM. We therefore view computational efficiency as a practical benefit of ATM-based screening under the present evaluation setting, rather than as the defining contribution of ATM.