TASK-CONDITIONED DECISION GEOMETRY FOR WORLD-MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

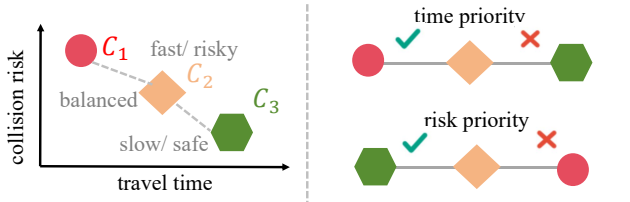
World-model planning turns predicted consequences into actions by comparing candidates for the current task. Yet standard world models learn to predict what each candidate will cause independently of the task, whereas planning depends on task-conditioned differences between candidates. Under a finite representation budget, predictive training can therefore discard a small but decision-critical difference while retaining low prediction error. We address this mismatch with Predictive Co-formation of Task-Induced Geometry (PC-TIG), which jointly shapes a shared representation through task-independent factual prediction and task-conditioned candidate comparison. PC-TIG combines a factual world model with a query-conditioned scorer and supports Joint and prediction-priority updates for both feedback actions and complete action segments. Across five-seed stochastic navigation and multi-objective HalfCheetah experiments, PC-TIG improves task-conditioned closed-loop control and segment selection over matched predictive, value-aware, and geometry-learning baselines while maintaining accurate dynamics. Our results show that effective world models must preserve not only predicted consequences, but also the task-dependent distinctions needed to choose among them.

1 INTRODUCTION

World models support model-based decision making by predicting the consequences of candidate actions and using learned rollouts for planning (Hafner et al., 2019b;a; Hansen et al., 2024; Zhou et al., 2024; Assran et al., 2025). Recent work has increasingly focused on whether the representations learned by world models are suitable not only for prediction but also for downstream decision making, through planning-oriented latent geometry, value-aware modeling, and task-sufficient representations (Voelcker et al., 2025; Wang et al., 2026b; Jiang et al.; Feng et al., 2026). At planning time, feedback control and trajectory optimization share a basic interface: propose candidate actions or action segments, predict their consequences, and rank them for the current task. This distinction becomes especially important when one predictive model is reused across different goals, preferences, or risk attitudes: prediction determines what each candidate is expected to cause, whereas action selection depends on how the current task distinguishes among those predicted consequences. A world model used for planning must therefore support not only accurate prediction, but also the task-dependent comparisons that determine which candidate is selected.

Accurate prediction does not guarantee accurate planning. A world model is trained to predict the consequence of each candidate, whereas a planner acts on differences between candidates. Predictive fit alone does not determine these differences: latent representations can preserve decoded predictions while changing their coordinate geometry (Arvanitidis et al., 2017; 2021), and the same predicted outcomes can reverse their preferred order as the task changes, as illustrated in Figure 1. This concern connects classical decision-aware model learning, which emphasizes prediction errors according to their downstream decision effects (Farahmand et al., 2017; Wilder et al., 2019; Grimm et al., 2020), with recent efforts to shape representations explicitly for planning (Wang et al., 2026b; Feng et al., 2026; Wang et al., 2026a). In particular, recent work has shown that planning-oriented geometry can improve latent planning and that predictive or decodable representations need not preserve the ranking of candidate plans (Wang et al., 2026b;a). These results expose a prediction-planning gap, but leave open the multi-query case central to our setting: the factual consequences of the candidates remain the same, while the candidate order itself can change with the task. Under

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.



(a) Same predicted outcomes (b) Task-conditioned geometry

Figure 1: Prediction and planning impose different requirements on a world-model representation. **(a)** The world model predicts the same physical outcomes for candidates c_1, c_2, c_3, regardless of the task. **(b)** The task changes how these outcomes should be compared: prioritizing travel time favors c_1, while prioritizing collision risk favors c_3.

a finite representation budget, predictive training can therefore discard a distinction that is small in prediction error yet large enough to change the selected action. For planning across tasks, the representation must preserve both candidate consequences and the task-dependent differences that determine the choice among them.

We address this gap with Predictive Co-formation of Task-Induced Geometry (PC-TIG), which learns factual prediction and task-conditioned candidate comparison through a shared representation. A task-independent world model preserves what each candidate causes, while a query-conditioned scorer learns the candidate order and cost gaps relevant to the current task; we refer to this task-dependent comparison structure as *task-conditioned decision geometry*. Our analysis formalizes why predictive fit alone leaves this structure underdetermined and establishes an action-gap condition connecting comparison accuracy to preservation of the optimal choice. PC-TIG co-forms the representation from factual and decision supervision while keeping the factual predictor task independent, allowing the same world model to support changing goals, preferences, and risk attitudes. We instantiate this principle for both closed-loop feedback actions and complete action segments, using direct Joint training or prediction-priority Feasible updates that protect factual prediction when the two objectives interact (Nocedal & Wright, 2006).

Across matched five-seed experiments, the results support this co-formation principle at the levels of prediction, representation, and decision making. In stochastic navigation, PC-TIG improves exact closed-loop task regret while maintaining accurate transition prediction, and fresh-head evaluation shows that the planning gain persists when scorer capacity and training are held fixed, isolating the contribution of the learned representation. In multi-objective HalfCheetah, PC-TIG improves complete-segment ordering and reduces preference regret by 74.5% relative to matched MAD (Steccanella et al., 2025), while its prediction-priority variant achieves the lowest prediction loss. The navigation gains further persist across changes in stochasticity and representation-training budget. Together, these results show that co-forming factual prediction and task-conditioned comparison improves planning across both feedback actions and complete action segments without sacrificing the predictive model. The contributions of this work are as follows:

- We identify a fundamental prediction–planning mismatch in world models: predictive accuracy does not by itself preserve the task-dependent candidate differences that determine planning decisions.
- We formulate task-conditioned decision geometry as the candidate order and cost gaps relevant to a planning task, and introduce PC-TIG to co-form this geometry with factual prediction in a shared world-model representation.
- We provide theoretical and empirical evidence that this co-formation improves task-conditioned planning while preserving predictive accuracy across stochastic feedback control and continuous-control segment selection.

2 Related Work

Decision-relevant world models. World-model learning has progressively moved from reconstructing observations toward learning predictive representations for control and planning (Ha & Schmid-

huber, 2018; Hafner et al., 2019b;a; Assran et al., 2023; Zhou et al., 2024; Assran et al., 2025; Maes et al., 2026). A complementary line of work asks which aspects of the environment a predictive representation must preserve for downstream decisions. DeepMDP and bisimulation retain reward and transition structure relevant to control (Gelada et al., 2019; Zhang et al., 2020), while value-aware, calibrated, and value-equivalent objectives prioritize model distinctions according to their effect on value computation (Farahmand et al., 2017; Voelcker et al., 2025; Grimm et al., 2020). More recent approaches align latent prediction with value information or seek task-sufficient world-model representations (Jiang et al.; Feng et al., 2026). PC-TIG is closest to this decision-aware line, but focuses on a multi-query planning setting in which a task-independent predictive model must support task-dependent comparisons among the same candidate consequences.

Planning-oriented representation geometry. A second line of work structures learned representations around relations useful for planning. Prior approaches learn graph or landmark structure, temporal and reachability distances, trajectory-straightened representations, Laplacian coordinates, anisotropic geometry, and asymmetric action distances (Yang et al., 2020; Zhang et al., 2021; Myers et al., 2024; Wang et al., 2026b; Shehmar et al., 2026; Hossain & Roy, 2026; Steccanella et al., 2025). Most directly related, Decision-Metric Alignment studies whether planner-facing latent costs preserve candidate-plan rankings and introduces action-conditioned objectives to improve this alignment (Wang et al., 2026a). PC-TIG differs in making the comparison itself task conditioned: rather than learning a single planner-facing geometry, it co-forms a reusable factual representation with the candidate order and cost gaps induced by the current query. This representation-learning problem is complementary to advances in candidate generation and model-based search (Psenka et al., 2026; Opryshko et al., 2025; Farebrother et al., 2026; Chang et al., 2026).

3 PC-TIG: Predictive Co-formation of Task-Induced Geometry

We now formalize task-conditioned candidate comparison and introduce PC-TIG. We first define the multi-query planning problem, then characterize what predictive fit alone leaves undetermined, and finally present the model and its co-formation updates.

3.1 Preliminaries: Multi-Query World-Model Planning

At state x, a planner compares a finite candidate set $\mathcal{C}(x)$, where each candidate c may be an action or an action segment, as in common world-model planning and model-predictive control formulations (Hansen et al., 2024; Zhou et al., 2024). Executing c induces a physical consequence $y(x, c)$, such as a successor-state distribution or trajectory. A task query q specifies how this consequence is valued, for example through a goal, preference, risk attitude, or horizon; this separation between predictive dynamics and task-dependent objectives is also central to recent task-oriented world-model learning (Feng et al., 2026). We write the resulting candidate cost as

$$J_q(c) \equiv J_q(y(x, c)), \tag{1}$$

and the planner selects

$$c_q^\star \in \arg\min_{c \in \mathcal{C}(x)} J_q(c). \tag{2}$$

The consequence $y(x, c)$ is task independent, whereas its valuation $J_q(c)$ depends on the query. Thus, the same candidate consequences can induce different orders across tasks. We call the query-dependent candidate order and pairwise cost gaps $J_q(c_i) - J_q(c_j)$ the *task-conditioned decision geometry*.

3.2 Why Prediction Alone Does Not Define Planning Geometry

Accurate prediction determines what each candidate will cause, but not how candidates should be compared for planning. Recent work has exposed a related gap between predictive representations and planner-facing candidate rankings (Wang et al., 2026a). In our multi-query setting, the same predicted consequences may additionally require different rankings under different queries.

Let $\widehat{J}_q(c)$ denote the learned task-conditioned cost, and let

$$\widehat{c}_q \in \arg\min_{c \in \mathcal{C}(x)} \widehat{J}_q(c). \tag{3}$$

We first ask whether predictive fit determines the geometry used for candidate comparison.

Proposition 1 (Prediction does not determine distance-based ranking)**.** *Two world models can produce identical decoded rollouts while inducing different Euclidean rankings of the same candidates in latent space.*

Thus, predictive equivalence does not imply planning equivalence when candidate comparison depends on latent coordinates. A separate ambiguity comes from the query: the same candidates may require opposite rankings under different tasks.

Proposition 2 (Conflicting tasks require query conditioning)**.** *Suppose $\mathcal{C}(x) = \{c_1, c_2\}$, query q_1 uniquely prefers c_1, and query q_2 uniquely prefers c_2. Any deterministic task-independent scorer incurs worst-query regret at least*

$$\min\{J_{q_1}(c_2) - J_{q_1}(c_1), J_{q_2}(c_1) - J_{q_2}(c_2)\} > 0. \tag{4}$$

Candidate comparison must therefore be learned beyond predictive fit and conditioned on the query. We next ask when the learned comparison preserves the optimal candidate. Since a common offset in all predicted costs does not affect their ordering, only the variation of cost error across candidates matters. For a query with a unique best candidate, define

$$\Delta_q = \min_{c \neq c_q^\star} \big[J_q(c) - J_q(c_q^\star)\big], \qquad \epsilon_q^\circ = \frac{1}{2}\Big(\max_c [\widehat{J}_q(c) - J_q(c)] - \min_c [\widehat{J}_q(c) - J_q(c)]\Big). \tag{5}$$

Here Δ_q is the true best-candidate margin, while $2\epsilon_q^\circ$ is the spread of cost error across candidates.

Theorem 1 (Candidate-order stability)**.** *The selected candidate satisfies*

$$J_q(\widehat{c}_q) - J_q(c_q^\star) \leq 2\epsilon_q^\circ.$$

Moreover, if $\Delta_q > 2\epsilon_q^\circ$, then $\widehat{c}_q = c_q^\star$.

These results motivate learning query-conditioned candidate comparison alongside factual prediction in the shared representation. Formal proofs are given in Appendix B.

3.3 PC-TIG: Predictive Co-formation

The preceding analysis suggests that a planning representation must satisfy two complementary requirements: it should preserve the factual consequences of candidates, while also supporting the query-dependent comparisons that determine their ordering. PC-TIG realizes these requirements through a shared encoder E_θ, a task-independent factual predictor G_θ, and a query-conditioned scorer S_ϕ.

The factual branch constrains the shared representation to remain predictive of what each candidate will cause. The decision branch complements this supervision by learning the query-dependent cost differences that determine candidate ordering, directly targeting the comparison errors characterized in Theorem 1. The query enters only S_ϕ, so the factual model remains reusable across tasks, while task supervision can shape the shared representation toward decision-relevant distinctions. We denote the corresponding objectives by $\mathcal{L}_{\text{fact}}$ and $\mathcal{L}_{\text{dec}}$; their joint influence on the shared encoder defines *predictive co-formation*. Figure 2 summarizes the architecture.

How these two objectives are instantiated depends on what the planner treats as a candidate. In feedback planning, a candidate is a single action whose value depends on its stochastic successor and subsequent replanning; task-dependent comparison is therefore naturally propagated through a Bellman recursion. In segment planning, a candidate is a complete action sequence whose trajectory and physical returns are predicted explicitly; its task value can therefore be supervised directly along the rollout. These give feedback and pathwise realizations of the same co-formation principle. The two realizations differ in what constitutes a candidate consequence and, consequently, how its task-conditioned comparison is supervised.

Feedback realization. For feedback planning, a candidate is a single action whose factual consequence is its successor distribution, while its task-dependent value also depends on subsequent replanning. The factual objective therefore preserves the successor law on which these comparisons are built:

$$\mathcal{L}_{\text{fact}}^{\text{B}} = \text{CE}_{\text{trans}} + \alpha_{\text{state}}\, \text{CE}_{\text{state}}, \tag{6}$$

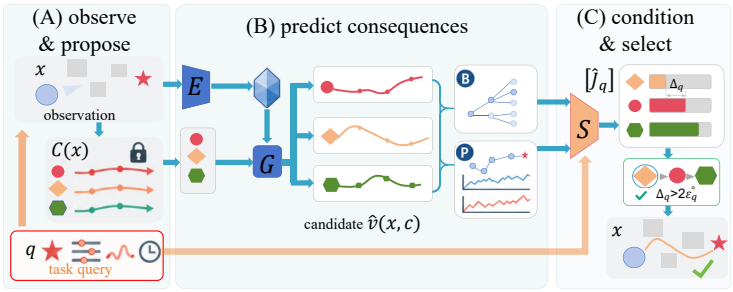


Figure 2: **Overview of PC-TIG.** A shared encoder feeds a task-independent factual branch and a query-conditioned scoring branch. Factual supervision constrains predicted consequences, while decision supervision shapes their task-conditioned comparison. The inset shows the stability condition $\Delta_q > 2\epsilon_q^\circ$ from Theorem 1.

where $\mathrm{CE}_{\mathrm{trans}}$ is categorical cross-entropy on the observed successor state, and $\mathrm{CE}_{\mathrm{state}}$ predicts the current physical state from its latent representation. The latter, weighted by $\alpha_{\mathrm{state}} > 0$, discourages state aliasing that could erase factual distinctions needed for downstream comparison.

The decision objective then learns how these consequences should be compared under query q. Because each action is followed by replanning, task-conditioned costs are propagated through a risk-sensitive finite-horizon Bellman recursion (Howard & Matheson, 1972):

$$\mathcal{L}_{\mathrm{dec}}^{\mathrm{B}} = \mathrm{SL}_1(\widehat{J}_q, B_q) + \alpha_{\mathrm{DP}}\,\mathrm{SL}_1(\widehat{J}_q, J_q^\star), \tag{7}$$

where B_q is the normalized Bellman target computed from the learned successor law and query-dependent risk criterion, and $J_q^\star$ is the corresponding exact dynamic-programming cost-to-go available in the controlled setting. The Bellman term propagates query-dependent comparison across future replanning steps, while the exact-DP anchor, weighted by $\alpha_{\mathrm{DP}} > 0$, fixes its boundary and scale. Planning selects the lowest-cost action and replans after each transition.

Pathwise realization. For segment planning, a candidate is a complete action sequence whose factual consequence is an explicit rollout. The factual objective therefore preserves the rollout quantities from which different queries form their utilities:

$$\mathcal{L}_{\mathrm{fact}}^{\mathrm{P}} = \mathcal{L}_{\mathrm{obs}} + \mathcal{L}_{\mathrm{reward}} + \mathcal{L}_{\mathrm{return}}. \tag{8}$$

Here $\mathcal{L}_{\mathrm{obs}}$ is the squared prediction error over the reconstructed initial observation and predicted rollout observations, $\mathcal{L}_{\mathrm{reward}}$ fits vector rewards, and $\mathcal{L}_{\mathrm{return}}$ fits cumulative vector returns; all use training-set normalization. Keeping rewards and returns vector valued preserves the factual information from which different queries can induce different candidate orderings.

For each nonempty prefix, let $R_{1:t}(c) = \sum_{j=0}^{t-1} r_j(c)$ denote its cumulative physical return. The query preference vector w_q defines the prefix utility $U_{q,t}(c) = w_q^\top R_{1:t}(c)$, and the scorer predicts $\widehat{U}_{q,t}(c)$. We write $U_q(c)$ and $\widehat{U}_q(c)$ for the corresponding utilities of the complete segment. To make utility errors comparable across queries, let $\sigma_q = \|w_q \odot s\|_2$, where s contains the training-set standard deviations of the physical return coordinates. The decision objective is

$$\mathcal{L}_{\mathrm{dec}}^{\mathrm{P}} = \mathbb{E}_{x,q,c,t}\left[\mathrm{SL}_{.25}\left(\frac{\widehat{U}_{q,t}(c) - U_{q,t}(c)}{\sigma_q}\right)\right] + \mathbb{E}_{x,q}\left[\max_{c\in\mathcal{C}(x)}\left|\frac{\widehat{U}_q(c) - U_q(c)}{\sigma_q}\right|\right]. \tag{9}$$

The first term learns query-conditioned valuation throughout the rollout. The second controls the largest final-candidate error within each planning instance and therefore upper-bounds the normalized error radius in Theorem 1. It directly targets the candidate-wise distortion that can change the selected segment, rather than only average regression accuracy. At test time, the planner ranks complete segments by $\widehat{U}_q(c)$, equivalently by the cost $\widehat{J}_q(c) = -\widehat{U}_q(c)$. Appendix H gives the complete normalization, target construction, and implementation details.

3.4 Co-formation Updates

With $\mathcal{L}_{\mathrm{fact}}$ and $\mathcal{L}_{\mathrm{dec}}$ specified for each realization, the remaining question is how they should jointly shape the shared encoder. Their gradients may be compatible, but they may also conflict: improving

Algorithm 1 PC-TIG co-formation training

Require: realization $r \in \{\mathrm{B}, \mathrm{P}\}$, update $u \in \{\textsc{Joint}, \textsc{Feasible}\}$, modules $E_\theta, G_\theta, S_\phi$
Pretrain the factual model (E_θ, G_θ)
if $(r, u) = (\mathrm{P}, \textsc{Feasible})$ **then**
    Fix a factual reference batch $\mathcal{B}_\mathrm{F}$
end if
for each training batch **do**
    Compute $\mathcal{L}^r_{\mathrm{fact}}$ and $\mathcal{L}^r_{\mathrm{dec}}$
    if $u = \textsc{Joint}$ **then**
        Update (θ, ϕ) with the Joint objective in Equation (10)
    else if $r = \mathrm{B}$ **then**
        Project the encoder decision gradient by Equation (11) and update (θ, ϕ)
    else
        Take a factual-decreasing step on $\mathcal{B}_\mathrm{F}$
        Project the decision step by Equation (12) and backtrack until Equation (13) holds
    end if
end for
Ensure: trained $E_\theta, G_\theta, S_\phi$

task-conditioned comparison can require representational changes that degrade factual prediction. Since co-formation requires decision supervision without losing the predictive basis on which comparison is built, we study two ways of resolving this trade-off. *Joint* treats factual and decision learning symmetrically and allows both objectives to shape the representation together. *Feasible* instead gives factual prediction priority, allowing task supervision to reshape the representation only when its update remains compatible with factual learning.

Joint. Without an explicit priority between the two objectives, the natural update is to optimize them jointly:

$$\min_{\theta,\phi} \ \mathcal{L}_{\mathrm{fact}}(\theta) + \lambda \mathcal{L}_{\mathrm{dec}}(\theta, \phi), \qquad \lambda > 0. \tag{10}$$

The scalar weight controls their relative influence but does not prevent the decision objective from opposing factual learning. The pathwise realization uses this form directly. The controlled feedback realization implements the same joint-update principle with an augmented-Lagrangian constraint around the predictive warm start; Appendix C gives the exact objective.

Prediction-priority Feasible. Feasible gives factual prediction priority by restricting task-driven updates that conflict with factual learning. For the feedback realization, let $g_{\mathrm{fact}} = \nabla_{\theta_E} \mathcal{L}^{\mathrm{B}}_{\mathrm{fact}}$ and $g_{\mathrm{dec}} = \nabla_{\theta_E} \mathcal{L}^{\mathrm{B}}_{\mathrm{dec}}$ denote the two gradients on the shared encoder. We project the decision gradient onto the half-space $\{g : \langle g, g_{\mathrm{fact}} \rangle \geq 0\}$:

$$\overline{g}_{\mathrm{dec}} = g_{\mathrm{dec}} + \frac{[-\langle g_{\mathrm{dec}}, g_{\mathrm{fact}} \rangle]_+}{\|g_{\mathrm{fact}}\|_2^2} g_{\mathrm{fact}}, \tag{11}$$

where $[u]_+ = \max(u, 0)$. This leaves compatible decision gradients unchanged and removes components that oppose factual learning. The encoder is updated with $g_{\mathrm{fact}} + \overline{g}_{\mathrm{dec}}$, yielding first-order factual compatibility.

For the pathwise realization, first-order compatibility alone may not preserve factual prediction after a finite optimizer step. We therefore project the proposed decision displacement δ_{dec} analogously,

$$\overline{\delta}_{\mathrm{dec}} = \delta_{\mathrm{dec}} - \frac{[\langle g_{\mathrm{fact}}, \delta_{\mathrm{dec}} \rangle]_+}{\|g_{\mathrm{fact}}\|_2^2} g_{\mathrm{fact}}, \tag{12}$$

and require the resulting parameters θ^+ to satisfy

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{fact}}(\theta^+; \mathcal{B}_\mathrm{F}) \leq \mathcal{L}^{\mathrm{P}}_{\mathrm{fact}}(\theta; \mathcal{B}_\mathrm{F}). \tag{13}$$

Thus, both realizations restrict decision-driven changes that conflict with factual prediction, with the pathwise variant additionally enforcing feasibility after the finite update. Algorithm 1 summarizes the training procedure; implementation details are given in Appendices C and H.

4 EXPERIMENTS

Experimental setup. We evaluate the method on controlled stochastic navigation and fixed-candidate continuous-action segment selection. Navigation follows the finite-state gridworld planning convention of Sutton et al. (1999) and the exponential risk-sensitive MDP criterion of Howard & Matheson (1972); the specific directed maze, rendered observations, stochastic shortcut, and task contexts are generated by our released code. Its known kernel provides exact dynamic-programming targets and evaluation for controlled representation attribution. For continuous-action segment selection, we use HalfCheetah from MO-Gymnasium (Felten et al., 2023), built on MuJoCo (Todorov et al., 2012). Six independently trained multi-objective policies provide three million replay transitions and twelve 25-step candidate segments for each of 6,144 planning instances. Supervision consists of the observed states, trajectories, and vector rewards. We fix the train, validation, and test split before training; Appendix H gives the environment identifier and collection details.

For a matched planning protocol, navigation scores all five actions and replans after each transition; HalfCheetah scores the same twelve segments. Every method receives the same states, task queries, candidate sets, prediction targets, and outer-update count. Thus the comparison changes only how the world-model representation and planner-facing score are learned. Evaluation uses exact kernel values for navigation and recorded segment returns for HalfCheetah.

Table 1: Benchmark and data scale; all counts are fixed across methods.

Benchmark	Obs.	Act.	H	$\lvert\mathcal{C}\rvert$	Replay	Train/Val./Test
Navigation	$24^2 \times 3$	5	16	5	250	$4^\dagger$/—/3,087
HalfCheetah	17	6	25	12	3M	4,096/1,024/1,024

† Four training task contexts; both benchmarks use five seeds.

We compare against matched adaptations of four closely related methods: MAD's directed minimum-action geometry (Steccanella et al., 2025), Temporal Straightening (Wang et al., 2026b), Calibrated VAML (Voelcker et al., 2025), and Latent-Space Value Alignment (Jiang et al.). Each receives the same data, model capacity, candidate sets, outer-update count, and planner interface as our method. In HalfCheetah, every adapted geometry baseline representation is paired with the same residual scorer used for representation comparison. The ablations isolate the source of improvement: *Prediction-only WM* uses prediction alone; *Frozen WM + scorer* learns only the task scorer; *Prediction-tuned WM* continues world-model training without decision gradients; and *Value-aware Joint* is a query-conditioned scalar value-aware objective motivated by VAML and value equivalence (Farahmand et al., 2017; Grimm et al., 2020). PC-TIG (Joint) directly combines prediction and decision learning; PC-TIG (Feasible) applies the prediction-priority update in Section 3.4. Appendices C and H specify the PC-TIG and matched MAD objectives, architectures, and training details.

Co-formation improves task-conditioned decisions. Across both planning interfaces, the main result is consistent: shaping the predictive representation with task-conditioned comparison improves the decisions made from that representation. In stochastic navigation, PC-TIG (Joint) achieves the lowest exact closed-loop regret, while Feasible also improves substantially over Frozen WM + scorer, MAD, and Value-aware Joint (Table 2). These gains occur at the policy level—closed-loop regret evaluates the complete sequence of replanned actions under the requested time, energy, and risk trade-off—rather than only the first selected action. Goal-hit probability remains a complementary task-agnostic reachability measure.

The same pattern appears when the planning unit changes from feedback actions to complete continuous-action segments. On HalfCheetah, PC-TIG (Joint) improves preference regret, best-segment recovery, pairwise ordering, and rank correlation over the matched geometry-learning baselines (Table 3). In particular, preference regret falls by 74.5% relative to matched MAD and by 61.5% relative to Latent-Space Value Alignment. Because every method ranks the same fixed candidate segments, these gains cannot be attributed to better candidate generation or a larger search budget; they reflect better task-conditioned comparison of the predicted consequences.

The gains are explained by better candidate ordering, not only lower average regret. Figure 3 examines the decision structure underlying the aggregate metrics. PC-TIG (Joint) shifts the query-level regret distribution toward zero and improves candidate ranks across the set: relative to Latent-Space Value Alignment, the fraction of exactly ranked candidates increases from 57.4% to 67.6% and mean rank error falls from .66 to .44. This is the behavior predicted by the decision-geometry

Table 2: Exact navigation feedback (five-seed mean ± SD). Closed-loop regret is the primary metric; goal-hit probability is a task-agnostic reachability statistic. Bold marks the best result.

Method	Closed-loop regret ↓	Goal-hit probability ↑	Trans. L_1 ↓
Matched adaptations			
MAD	22.0397 ± 1.0415	**0.6699 ± 0.0235**	0.009903 ± 0.001535
Temporal Straightening	18.2330 ± 1.6147	0.4923 ± 0.0826	0.002329 ± 0.000730
CVAML	19.1674 ± 3.2905	0.4702 ± 0.1036	0.001687 ± 0.001436
Latent-Space VA	19.0071 ± 2.4691	0.5123 ± 0.1029	0.003311 ± 0.001319
Ablations			
Frozen WM + scorer	21.5266 ± 1.0960	0.4085 ± 0.0771	0.018236 ± 0.001434
Value-aware Joint	18.3723 ± 2.8572	0.5263 ± 0.0408	0.003033 ± 0.001510
Ours			
PC-TIG (Joint)	**16.1225 ± 1.9107**	0.5014 ± 0.0902	0.004306 ± 0.001683
PC-TIG (Feasible)	16.2255 ± 1.5987	0.5169 ± 0.0388	**0.000585 ± 0.000091**

Table 3: HalfCheetah fixed-candidate results (five-seed mean ± SD; three decimals; best in bold).

Method	Prediction loss ↓	Preference regret ↓	Top-1 ↑	Pairwise ↑	Spearman ↑	Vel. regret ↓
Matched adaptations						
MAD geometry	0.537 ± 0.017	0.845 ± 0.037	0.608 ± 0.054	0.916 ± 0.002	0.924 ± 0.002	2.364 ± 0.125
Temporal Straightening	0.425 ± 0.019	0.678 ± 0.051	0.636 ± 0.043	0.926 ± 0.003	0.935 ± 0.003	1.796 ± 0.114
CVAML	0.232 ± 0.012	0.650 ± 0.034	0.627 ± 0.050	0.926 ± 0.003	0.937 ± 0.003	1.782 ± 0.115
Latent-Space VA	0.210 ± 0.007	0.560 ± 0.027	0.649 ± 0.046	0.931 ± 0.004	0.940 ± 0.004	1.565 ± 0.106
Ablations						
Prediction-only WM	0.495 ± 0.027	2.327 ± 0.278	0.507 ± 0.021	0.839 ± 0.013	0.788 ± 0.027	4.239 ± 1.152
Frozen WM + scorer	0.495 ± 0.027	0.885 ± 0.068	0.641 ± 0.026	0.914 ± 0.006	0.921 ± 0.007	2.361 ± 0.225
Prediction-tuned WM	0.234 ± 0.009	0.648 ± 0.038	0.629 ± 0.048	0.928 ± 0.003	0.938 ± 0.002	1.756 ± 0.149
Ours						
PC-TIG (Feasible)	**0.162 ± 0.004**	0.522 ± 0.044	0.704 ± 0.014	0.934 ± 0.003	0.944 ± 0.003	1.347 ± 0.132
PC-TIG (Joint)	0.192 ± 0.007	**0.216 ± 0.008**	**0.771 ± 0.041**	**0.955 ± 0.001**	**0.967 ± 0.002**	**0.581 ± 0.022**

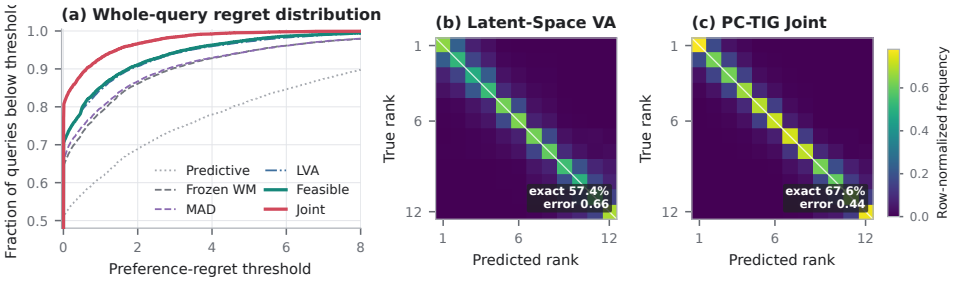


Figure 3: **How co-formation changes complete-segment decisions. (a)** Empirical cumulative distribution of preference regret over all held-out mixed-preference HalfCheetah queries; curves farther to the upper left are better. **(b–c)** Row-normalized true-rank versus predicted-rank matrices for Latent-Space Value Alignment and PC-TIG (Joint). Each query ranks the same twelve candidate segments. Brighter diagonal mass means that the learned geometry preserves more of the task-defined candidate order; insets report exact-rank accuracy and mean absolute rank error.

view: the representation better preserves the within-query order used to select among candidates, rather than obtaining its mean gain from a small number of unusually large improvements.

Fresh-head attribution localizes the gain to the representation. Improved decisions could otherwise arise from a better-trained or more expressive scorer. We therefore discard the trained task heads and fit identically initialized fresh heads on frozen Prediction-only WM, Value-aware Joint, and PC-TIG (Feasible) representations, using identical data order, objective, target updates, and training budget. The resulting closed-loop advantage persists on a denser grid of unseen task contexts, isolating the co-formed representation as the source of the planning improvement. Figure 4 further tests the predicted mechanism by evaluating cases in which increasing risk preference

changes the exact optimal action and measuring whether the learned representation recovers that task-induced change.

Prediction priority preserves the factual model. The decision gains do not require replacing factual accuracy with task fit. Feasible is designed specifically for this regime: in navigation it combines improved closed-loop regret with substantially lower transition error than the strongest geometry adaptation, while in HalfCheetah it attains the lowest prediction loss among the compared methods (Tables 2 and 3). The contrast with Joint is informative: Joint generally gives the strongest decision performance, whereas Feasible more strongly preserves factual prediction. This supports the intended distinction between unrestricted co-formation and prediction-priority co-formation.

Table 4: Equal-head attribution on the held-out 3×5 query grid (five-seed means).

Representation	Regret ↓	Goal-hit ↑	Trans. L_1 ↓	Switch ↑
Predictive	22.4751	0.3309	0.001547	0.5971
Value-aware	19.6150	0.4524	0.002050	0.5831
PC-TIG	**17.9909**	**0.4582**	**0.000607**	**0.6521**

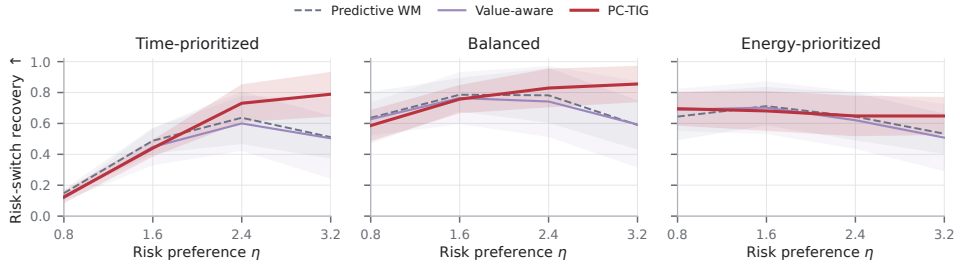


Figure 4: **Recovery of risk-induced planner switches.** Each panel fixes a time–energy preference and varies the unseen risk preference η. Among state–goal–horizon queries whose unique exact optimal action differs from its $\eta = 0$ optimum, the vertical axis is the fraction for which the learned planner selects the new optimum. Curves and shading show mean $\pm$ one standard deviation over five seeds after fitting the same fresh scorer to each frozen representation.

The effect persists across stochasticity and training budget. The navigation advantage is not confined to the default environment or training schedule. Across four shortcut-success probabilities and three representation-training budgets, PC-TIG (Feasible) has the lowest mean first-action regret in all seven evaluated conditions (Figure 5a–b). All paired seed-level comparisons against Frozen WM + scorer, MAD, and Value-aware Joint favor Feasible with positive 95% intervals. Thus, the benefit persists as both environmental uncertainty and the amount of representation training change.

Computational cost. Co-formation adds training cost but does not change the candidate set or planner-time search procedure. Measured wall-clock costs and hardware for all methods are reported in Appendix Table 15.

5 Conclusion

World-model planning requires not only accurate prediction, but also task-conditioned comparison of predicted consequences. We introduced PC-TIG, which co-forms factual prediction and task-conditioned decision geometry in a shared representation. Across feedback control and continuous-action segment selection, PC-TIG improves candidate ordering and planning decisions while maintaining accurate factual prediction. These results support treating task-dependent candidate comparison as an explicit representation objective for world-model planning.

Limitations and future work. Our experiments consider finite candidate sets and structured task queries. Extending co-formation to learned candidate generation, higher-dimensional observations, and more open-ended planning settings is an important direction for future work.

AI USE STATEMENT

Generative AI tools were used solely for language editing. The authors verified the final manuscript and take full responsibility for the submission.

ETHICS STATEMENT

All experiments run entirely in simulation on synthetic navigation and a standard control benchmark, involving only generated states and trajectories.

REPRODUCIBILITY STATEMENT

The supplementary artifact provides machine-readable numerical summaries and maps every reported table and figure to its source measurements. The appendix specifies environments, task contexts, training schedules, evaluation units, statistical procedures, and complete proofs.

REFERENCES

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

Georgios Arvanitidis, Miguel González-Duque, Alison Pouplin, Dimitris Kalatzis, and Søren Hauberg. Pulling back information geometry. *arXiv preprint arXiv:2106.05367*, 2021.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.21306.

Amir-Massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1486–1494, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Marc G. Bellemare, Alessandro Lazaric, and Ahmed Touati. Compositional planning with jumpy world models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.19634.

Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno Castro da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4aa8891583f07ae200ba07843954caeb-Abstract.html.

Fan Feng, Yujia Zheng, Minghao Fu, Yongqiang Chen, Guangyi Chen, Kevin Murphy, Biwei Huang, and Kun Zhang. Learning task-sufficient world models by synergizing agentic exploration and structured modeling. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2607.04409.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

Christopher Grimm, André Barreto, Satinder P. Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. World models. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/ha18a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

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

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

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.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Jumman Hossain and Nirmalya Roy. Learning anisotropic value geometry with finsler reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026.

Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972. doi: 10.1287/mnsc.18.7.356.

Xingyu Jiang, Yuheng Pan, Mukang You, Xiuhui Zhang, Ning Gao, Guanwei Yan, Hao Li, and Yue Deng. Boosting world models learning via latent-space value alignment. In *Forty-third International Conference on Machine Learning*.

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

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 37076–37096, 2024. URL https://proceedings.mlr.press/v235/myers24a.html.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

Evgenii Opryshko, Junwei Quan, Claas Voelcker, Yilun Du, and Igor Gilitschenski. Test-time graph search for goal-conditioned reinforcement learning. *arXiv preprint arXiv:2510.07257*, 2025.

Michael Psenka, Michael Rabbat, Aditi S. Krishnapriyan, Yann LeCun, and Amir Bar. Parallel stochastic gradient-based planning for world models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.00475.

Aidan Scannell, Mohammadreza Nakhaeinezhadfard, Kalle Kujanpää, Yi Zhao, Kevin Sebastian Luck, Arno Solin, and Joni Pajarinen. Discrete codebook world models for continuous control. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lfRYzd8ady.

Dikshant Shehmar, Matthew Schlegel, Matthew E. Taylor, and Marlos C. Machado. Laplacian representations for decision-time planning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.05031.

Lorenzo Steccanella, Joshua B Evans, Özgür Şimşek, and Anders Jonsson. Learning the minimum action distance. *arXiv preprint arXiv:2506.09276*, 2025.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. URL https://arxiv.org/abs/1801.00690.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Claas A. Voelcker, Anastasiia Pedan, Arash Ahmadian, Romina Abachi, Igor Gilitschenski, and Amir-Massoud Farahmand. Calibrated value-aware model learning with probabilistic environment models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 61745–61768, 2025. URL https://proceedings.mlr.press/v267/voelcker25a.html.

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. *arXiv preprint arXiv:2608.18746*, 2026a. URL https://arxiv.org/abs/2608.18746.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2603.12231.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/view/3982.

Ge Yang, Amy Zhang, Ari S. Morcos, Joelle Pineau, Pieter Abbeel, and Roberto Calandra. Plan2vec: Unsupervised representation learning by latent plans. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, 2020. URL https://proceedings.mlr.press/v120/yang20b.html.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Lunjun Zhang, Ge Yang, and Bradly C Stadie. World model as a graph: Learning latent landmarks for planning. In *International conference on machine learning*, pp. 12611–12620. PMLR, 2021.

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

A Standard Online Continuous-Control Benchmark

We report a development compatibility study for online control on Cheetah Run and Cartpole Swingup from the DeepMind Control Suite (DMControl) (Tassa et al., 2018). Agents receive proprioceptive states and act for 500 decisions per episode with action repeat two. Table 5 compares all methods after exactly 200,000 policy decisions using the undiscounted episodic return in $[0, 1000]$. SAC, DreamerV3, TD-MPC, and TD-MPC2 are transcribed at step 200,000 from the official TD-MPC2 release (Haarnoja et al., 2018; Hafner et al., 2023; Hansen et al., 2022; 2024); DC-MPC is transcribed from its official release (Scannell et al., 2025) at 400,000 native environment steps, which equals 200,000 policy decisions under action repeat two. Scalar Joint and PC-TIG use seed 1, the same TD-MPC2 backbone, replay capacity, optimizer, MPPI planner, and ten evaluation episodes; only the shared-parameter update differs. This study tests integration with the TD-MPC2 pipeline and complements the five-seed controlled evidence in the main paper.

Table 5: State-observation DMControl return after 200,000 policy decisions. Released entries are mean ± sample standard deviation over their official seeds; matched entries use one development seed and average ten complete evaluation episodes. "Avg." averages the two task means. Underlining marks the strongest released baseline and bold marks the stronger matched seed-1 result; Δ is PC-TIG minus Scalar Joint.

Method	Seeds	Cheetah Run	Cartpole Swingup	Avg.
SAC (Haarnoja et al., 2018)	3	573.7 ± 0.2	842.0 ± 3.4	707.8
DreamerV3 (Hafner et al., 2023)	3	353.4 ± 108.0	795.6 ± 19.5	574.5
TD-MPC (Hansen et al., 2022)	3	$\underline{771.1 \pm 54.9}$	870.6 ± 14.4	$\underline{820.9}$
TD-MPC2 (Hansen et al., 2024)	3	614.4 ± 94.3	$\underline{871.9 \pm 11.1}$	743.2
DC-MPC (Scannell et al., 2025)	5	605.4 ± 27.5	860.1 ± 7.5	732.8
Scalar Joint	1	700.2	367.0	533.6
PC-TIG (Feasible)	1	901.3	878.5	889.9
Δ	–	+201.0	+511.5	+356.3

B Proofs

B.1 Proof of Proposition 1

Take an exact encoder–transition–decoder model on a finite rollout family, with original chart $\mathbb{R}^2$, latent goal $z^g = (0, 0)$, and candidate endpoints $z^{c_1} = (1, 0)$ and $z^{c_2} = (0, 2)$. Coordinate-Euclidean scoring selects c_1. For $\gamma > 2$, let $\psi_\gamma(z_1, z_2) = (\gamma z_1, z_2)$ and define the conjugated model. For every rollout observation o and action a,

$$F'(E'(o), a) = \psi_\gamma(F(E(o), a)), \qquad C'(E'(o)) = C(E(o)), \tag{14}$$

so induction gives identical decoded predictions at every rollout step. The transformed endpoints are $(\gamma, 0)$ and $(0, 2)$, and coordinate-Euclidean scoring now selects c_2. Thus the predictive process is identical while the coordinate score changes. □

B.2 Proof of Proposition 2

Because $\mathcal{C}(x) = \{c_1, c_2\}$, a deterministic task-independent scorer gives one fixed order to c_1 and c_2. If it selects c_1, its regret under q_2 is

$$J_{q_2}(c_1) - J_{q_2}(c_2). \tag{15}$$

If it selects c_2, its regret under q_1 is

$$J_{q_1}(c_2) - J_{q_1}(c_1). \tag{16}$$

The same conclusion holds for a fixed tie-breaking rule. Taking the smaller of these two positive quantities proves Equation (4). □

B.3 Proof of Theorem 1

Write $e(c) = \widehat{J}_q(c) - J_q(c)$. Optimality of $\widehat{c}_q$ under the learned costs gives

$$J_q(\widehat{c}_q) - J_q(c_q^\star) \le e(c_q^\star) - e(\widehat{c}_q) \le \max_c e(c) - \min_c e(c) = 2\epsilon_q^\circ. \tag{17}$$

If $\widehat{c}_q \neq c_q^\star$, uniqueness of $c_q^\star$ makes the left-hand side at least Δ_q; this is impossible when $\Delta_q > 2\epsilon_q^\circ$. □

B.4 Proof of Proposition ??

If $\langle g_{\mathrm{dec}}, g_{\mathrm{fact}} \rangle \ge 0$, Equation (11) leaves g_{dec} unchanged. Otherwise, it subtracts exactly the component of g_{dec} opposite to g_{fact}, so $\langle \overline{g}_{\mathrm{dec}}, g_{\mathrm{fact}} \rangle = 0$. These are precisely the two cases of Euclidean projection onto the stated half-space. Hence

$$\langle g_{\mathrm{fact}}, g_{\mathrm{fact}} + \overline{g}_{\mathrm{dec}} \rangle \ge \|g_{\mathrm{fact}}\|_2^2, \tag{18}$$

and its negative is a first-order non-increasing direction for the factual loss with respect to the shared encoder. When $g_{\mathrm{fact}} = 0$, every direction belongs to the half-space and the zero-quotient convention returns g_{dec}. □

C Benchmark and Protocol Details

Environment. For this fully observed benchmark, x identifies a state in the finite set $\mathcal{S}$, which contains 49 grid states and one absorbing trap. Five actions move up, down, left, right, or invoke a shortcut. Coordinates are ordered as (column, row); the 21-state start and goal pools occupy columns 0–2 and 4–6, respectively. A vertical wall has upper and lower gaps with different energy profiles. The shortcut moves from $(1, 3)$ to $(5, 3)$ with scenario-dependent probability and otherwise enters the trap. RGB observations contain wall geometry, state position, and a predictable state-specific texture. We write $\mathcal{A}$ for this five-action set and $p^\star(x' \mid x, a)$ for the true probability of reaching x' from x after action a.

Costs and horizons. For navigation, a task query is $q = (g, \omega, \eta, h)$: goal g, time–energy weights ω, risk coefficient η, and remaining horizon h; the current state remains the separate planner input x. Let $\omega = (w_t, w_e)$ contain the nonnegative time and energy weights. The edge cost is $c_\omega = w_t c_t + w_e c_e$. Every step has $c_t = 1$, and $\mathbb{I}[\cdot]$ denotes an indicator. Let $f(x, a)$ denote the deterministic successor of a cardinal action and $y(f(x, a))$ its row. For a cardinal move, energy is $c_e(x, a) = .65 + 2.8\mathbb{I}[y(f(x, a)) \le 2] + .45\mathbb{I}[a \in \{\mathrm{up}, \mathrm{left}\}] + .8\mathbb{I}[f(x, a) = x]$. The shortcut costs .25 at its source; invoking it elsewhere costs 1.8, and every trap action costs 2. The terminal penalty is $22(w_t + 0.8w_e)$ and $H_{\mathrm{B}} = 16$. With p_{sc} denoting shortcut success probability, Composition uses a deterministic shortcut, while Risk and Bottleneck use $p_{\mathrm{sc}} = .55$ and .70.

Contexts. Every scenario contains four training contexts and seven disjoint held-out contexts. The following schedule defines the paired five-seed evaluation. The risk coefficient satisfies $\eta \ge 0$, with $\eta = 0$ giving expected cost. For a discrete successor law p and continuation costs v, define the entropic risk functional

$$\rho_\eta(p, v) = \begin{cases} \sum_{x'} p(x') v(x'), & \eta = 0, \\ \eta^{-1} \log \sum_{x'} p(x') \exp(\eta v(x')), & \eta > 0. \end{cases} \tag{19}$$

The exact finite-horizon dynamic program applies ρ_η after every transition; positive η increasingly emphasizes high-cost outcomes. This recursion defines the navigation instance cost J_q and supplies the exact anchor targets used in the navigation experiments. All contexts use $w_e = 1 - w_t$; the listed w_t and η entries are paired columnwise.

Scenario	Training (w_t, η)	Held-out contexts
Composition	$(1, 0), (.72, 0), (.38, 0), (.02, 0)$	$w_t = (91, 77, 63, 49, 35, 21, 7)/101$; $\eta = 0$
Risk	$(.97, .2), (.69, 1.0), (.41, 2.0), (.13, 3.1)$	w_t as above; $\eta = (17, 53, 89, 125, 161, 197, 233)/64$
Bottleneck	$(.94, .3), (.66, 1.2), (.39, 2.1), (.11, 3.0)$	w_t and η as in Risk

The stochasticity, budget, and layout studies use six held-out contexts paired columnwise: $w_t = (97, 81, 65, 49, 33, 17)/107$ and $\eta = (13, 47, 83, 127, 173, 223)/64$. The three layout specifications are: Upper/lower has wall-gap rows $(0, 4)$ and shortcut $(1, 2) \to (5, 4)$; Middle/lower has rows $(2, 6)$ and $(1, 4) \to (5, 2)$; Symmetric/narrow has rows $(2, 4)$ and $(1, 3) \to (5, 3)$. Each layout uses $p_{\mathrm{sc}} = .55$.

Choice of numerical settings. The planning problem and the results in Section 3.2 are independent of the numerical settings below. The Feasible projection coefficient is solved analytically by Equations (11) and (12); it is not tuned. Benchmark quantities such as the action set and horizon define the evaluation interface. Optimization quantities—loss normalizers, batch sizes, learning rates, schedules, and line-search depth—were selected on the validation split before evaluation and shared by all methods to which they apply. The budget sweep changes only the number of updates, making the role of that choice directly observable.

Optimization. The encoder has three stride-two convolutional blocks followed by a two-layer projection. The dynamics network combines the latent with a learned action embedding and predicts the categorical next state. The task network maps the three-dimensional context to a 32-dimensional embedding. The state and goal charts output 24-dimensional vectors, and a two-hidden-layer readout with three affine layers predicts the positive score. We use AdamW, learning rate 2.5×10^{-3}, weight decay 10^{-4}, prediction minibatches of 192, and Bellman and anchor minibatches of 64. These batch sizes determine stochastic gradient variance and have no planner-time role. Prediction batches sample state–action pairs uniformly. Bellman and anchor batches independently sample the source from $\mathcal{S}$, the goal from the 21-state goal pool, one of four training contexts, and $h \in \{1, \dots, H_{\mathrm{B}}\}$, all uniformly.

The sampled Bellman factual loss is $\mathcal{L}^{\mathrm{B}}_{\mathrm{fact}} = \mathrm{CE}_{\mathrm{trans}} + .35\,\mathrm{CE}_{\mathrm{state}}$; the auxiliary state term prevents state aliasing while leaving successor prediction as the principal term. Costs and cost-to-go targets are divided by the fixed environment scale 32 before the Smooth-L_1 losses are evaluated, placing one-step and horizon targets near unit scale. The decision loss is the Bellman residual plus 1.5 times the exact-DP anchor; the residual supplies temporal propagation and the anchor fixes its boundary and scale. The target network is synchronized every 50 steps to hold the continuation target fixed between updates.

The full training schedule has 350 factual warm-up and 1,600 representation steps; 25% and 50% budgets use 88/400 and 175/800 steps. PC-TIG (Feasible) computes its factual gradient over all 50×5 state–action pairs and all 50 state targets, so feasibility is defined on a fixed measure rather than a fresh minibatch. Its final scorer calibration matches the representation-stage budget—1,600, 400, or 800 steps—while freezing all factual parameters. Gradients are clipped to global norm 5 as an optimizer safeguard.

We write $p_\theta(x' \mid E_\theta(x), a)$ for the learned categorical successor law used by the Bellman realization. Let $D_{\theta,\phi}(x, g, \omega, \eta, h)$ be the scorer output in normalized cost units and let $\bar{D}$ denote its delayed target copy. With $C = 32$, define the normalized continuation

$$v_{h-1}(x') = \begin{cases} 0, & x' = g, \\ c_{\mathrm{term}}/C, & h = 1,\ x' \neq g, \\ \bar{D}(x', g, \omega, \eta, h-1), & h > 1, \end{cases} \tag{20}$$

where $c_{\mathrm{term}} = 22(w_t + .8w_e)$. The Bellman target is zero at $x = g$ and, otherwise,

$$b_q(x, h) = \min_{a \in \mathcal{A}} \left[\frac{c_\omega(x, a)}{C} + \rho_{\eta C}(p_\theta(\cdot \mid E_\theta(x), a), v_{h-1}) \right]. \tag{21}$$

Thus $\mathcal{L}^{\mathrm{B}}_{\mathrm{dec}} = \mathrm{SL}_1(D_{\theta,\phi}, b_q) + 1.5\,\mathrm{SL}_1(D_{\theta,\phi}, V^\star_q/C)$, with independent samples for the Bellman and exact-DP anchor terms; SL_1 denotes Smooth-L_1 loss with transition point one and $V^\star_q$ is the exact finite-horizon cost-to-go. The factor ηC is the unit conversion required when physical costs are divided by C.

For the Bellman Joint variant, let $\mathrm{CE}^{\mathrm{warm}}_{\mathrm{trans}}$ be its true-successor cross-entropy over all state–action pairs after warm-up and set $\tau = 1.08\,\mathrm{CE}^{\mathrm{warm}}_{\mathrm{trans}} + .01$. Tying τ to the achieved warm-start error makes the reference scale adapt to the fitted model; the relative and additive slack absorb minibatch

fluctuation around that reference. With $r = \mathcal{L}^{\mathrm{B}}_{\mathrm{fact}} - \tau$ and multiplier $\mu \geq 0$, the implemented augmented objective is

$$\mathcal{L}^{\mathrm{B}}_{\mathrm{dec}} + \mu r + 12[r]^2_+ + .05\mathcal{L}^{\mathrm{B}}_{\mathrm{fact}}, \qquad \mu \leftarrow \mathrm{clip}_{[0,50]}(\mu + .08r). \tag{22}$$

The quadratic term responds to positive excursions, the multiplier integrates persistent excursions, and the small factual term continues prediction descent inside the reference region. Their coefficients and multiplier cap are selected once on the validation split and then held fixed. In the Bellman Feasible implementation, the zero-gradient convention in Equation (11) is realized with a 10^{-20} squared-norm guard; this value and the norm clip affect floating-point evaluation only.

Comparator objectives. After the common warm-up, Prediction-only WM receives 1,600 further prediction updates and scores action a by $\sum_{x' \in \mathcal{S}} p_\theta(x' \mid E_\theta(x), a) \| E_\theta(x') - E_\theta(g) \|_2$. Frozen WM + scorer keeps the warm-start predictive representation fixed and minimizes only the exact-DP anchor component of $\mathcal{L}^{\mathrm{B}}_{\mathrm{dec}}$. Value-aware Joint adds that anchor to $\mathcal{L}^{\mathrm{B}}_{\mathrm{fact}}$ with weight $\lambda_s = 4$ in Composition and Risk and $\lambda_s = 7$ in Bottleneck. These development-set weights place factual and anchor gradients on comparable scales; the same scenario-specific value is reused for MAD so the two representation baselines receive matched task pressure. The MAD-style score is

$$\widehat{d}_{\mathrm{MAD}}(E_\theta(x), E_\theta(g)) = .9 \| [E_\theta(x) - E_\theta(g)]_+ \|_\infty + .1 \, \mathrm{mean}([E_\theta(x) - E_\theta(g)]_+). \tag{23}$$

Here mean averages the chart coordinates; the $.9/.1$ mixture is the Simple quasimetric in the released MAD implementation (Steccanella et al., 2025). The finite-horizon support target uses $d^{\mathrm{sup}}_0(x, g) = 0$ for $x = g$ and $H_{\mathrm{B}} + 4$ otherwise, and for $h \geq 1$,

$$d^{\mathrm{sup}}_h(x, g) = \begin{cases} 0, & x = g, \\ \min\limits_{a \in \mathcal{A}} \left[1 + \min\limits_{x' : p^\star(x' \mid x, a) > 0} d^{\mathrm{sup}}_{h-1}(x', g) \right], & x \neq g. \end{cases} \tag{24}$$

The value $H_{\mathrm{B}} + 4$ is an unreachable-state sentinel strictly beyond the largest in-horizon support distance. For independently uniform $x, g \in \mathcal{S}$ and $h \in \{1, \ldots, H_{\mathrm{B}}\}$, MAD minimizes $\mathcal{L}^{\mathrm{B}}_{\mathrm{fact}} + \lambda_s \mathcal{L}^{\mathrm{B}}_{\mathrm{MAD}}$, where

$$\mathcal{L}^{\mathrm{B}}_{\mathrm{MAD}} = \mathbb{E}_{x,g,h} \left[\mathrm{Huber}_1 \left(\widehat{d}_{\mathrm{MAD}}(E_\theta(x), E_\theta(g)) - d^{\mathrm{sup}}_h(x, g) \right) \right], \tag{25}$$

where the Huber transition is one action because the support target is measured in action steps. MAD uses the same scenario-specific λ_s. Its scorer is horizon-independent; h enters only through the sampled support-distance target. At planning time it assigns action a the minimum $\widehat{d}_{\mathrm{MAD}}(E_\theta(x'), E_\theta(g))$ over successors satisfying $p_\theta(x' \mid E_\theta(x), a) > .01$; this fixed numerical cutoff removes negligible softmax mass from the support minimum.

Evaluation. For each action-level seed and scenario, we evaluate 120 start–goal planning instances at horizon 16 under each of seven held-out contexts, for 840 instances. Exact dynamic programming supplies the true query-specific action cost $Q^\star_q$. For the selected first action $\widehat{a} = \widehat{c}_q$, first-action regret is

$$\mathrm{Reg}^{\mathrm{1st}}_q(\widehat{a}) = Q^\star_q(x, \widehat{a}) - \min_{a \in \mathcal{A}} Q^\star_q(x, a).$$

For closed-loop evaluation, we construct the learned finite-horizon feedback policy and evaluate its dynamic-entropic value under the true transition kernel for all $21 \times 21 \times 7 = 3{,}087$ start–goal–context planning instances per seed. Each robustness and topology condition uses six held-out contexts and 120 paired start–goal instances per context. For context index j and instance index k, these select starts at index $(7k+3j) \bmod 21$ in the start list and goals at index $(11k+5j) \bmod 21$ in the goal list, with both lists ordered first by column and then by row. This gives the same 720 planning instances to every method.

D Results Across Independent Runs

Table 7: Risk-sensitive first-action regret for each independent run.

Table 6: Unweighted first-hit reachability of learned feedback policies under the true transition kernel, averaged over five training seeds. This diagnostic is the probability of reaching the goal by horizon H_B.

Method	Goal-hit probability ↑
Frozen WM + scorer	.4085
MAD	**.6699**
Value-aware Joint	.5263
PC-TIG (Joint)	.5014
PC-TIG (Feasible)	.5169

Method	Run 1	Run 2	Run 3	Run 4	Run 5
Frozen WM + scorer	1.4363	1.3854	1.2130	1.5431	1.4388
MAD	2.0104	2.0283	2.0347	1.9688	1.6908
Value-aware Joint	.9887	1.2826	1.3201	1.3252	1.0418
PC-TIG (Feasible)	.6675	.5661	.5147	.6452	.7069

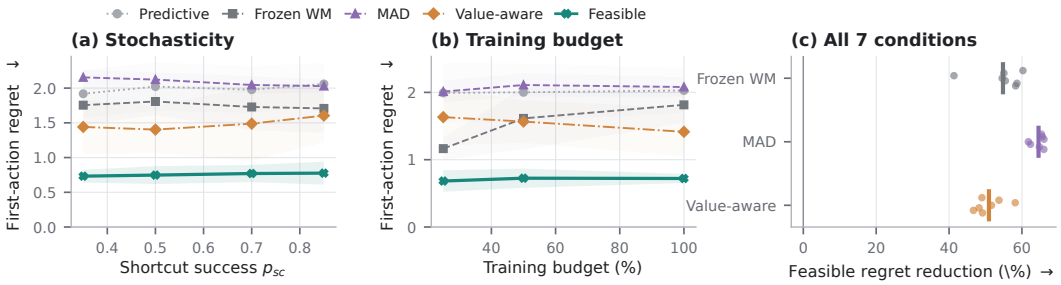


Figure 5: **Robustness of task-conditioned co-formation.** **(a–b)** Five-seed means and 95% confidence bands across dynamics stochasticity and representation-training budget. **(c)** Relative regret reduction of PC-TIG (Feasible) in every condition; all 21 paired confidence intervals against Frozen WM + scorer, MAD, and Value-aware Joint are positive.

Table 8: First-action robustness details for PC-TIG (Feasible). The three CI columns give the lower endpoint of the paired 95% regret-improvement interval over each control; every comparison improves in all five seeds. Switch columns report mean risk-switch accuracy for PC-TIG (Feasible) and Value-aware Joint.

Condition	Feasible first-action regret	CI low vs frozen WM	CI low vs MAD	CI low vs value-aware	Feasible switch	Value-aware switch
25% budget	.6830	.3196	1.2126	.4696	.6438	.5006
50% budget	.7253	.4349	1.2941	.4026	.6855	.5814
100% budget	.7210	.7643	1.1858	.2948	.6509	.4630
Shortcut $p_{sc} = .35$	.7323	.6725	1.3450	.2588	.7045	.4733
Shortcut $p_{sc} = .50$	.7478	.7964	1.1604	.3216	.6523	.4767
Shortcut $p_{sc} = .70$	.7702	.5775	1.1187	.1736	.5955	.4817
Shortcut $p_{sc} = .85$	.7758	.5039	1.0755	.3010	.6342	.5800

Table 9: Value-aware Joint and PC-TIG (Feasible) first-action regret under three separately trained layouts.

Topology	Method	Run 1	Run 2	Run 3	Run 4	Run 5
Upper/lower	Value-aware Joint	1.0818	1.2572	1.7809	1.2056	1.4995
	PC-TIG (Feasible)	.6619	.6164	.7221	.6249	.6023
Middle/lower	Value-aware Joint	1.6116	.9396	1.3823	.7061	.9144
	PC-TIG (Feasible)	.5445	.5175	.5611	.9434	.5438
Symmetric/narrow	Value-aware Joint	1.0236	.9028	1.5880	1.5968	1.1176
	PC-TIG (Feasible)	.6358	.6265	.7512	.6366	.7965

Table 11: Exact closed-loop task regret for each independent run.

Table 10: Mean first-action regret under each separately trained topology and the seed-wise macro-average. The final column gives the paired 95% interval for the macro-average regret improvement of PC-TIG (Feasible) over each control.

Method	Upper/lower	Middle/lower	Symmetric/narrow	Macro	Improvement CI
Prediction-only WM	2.0952	1.5825	2.0357	1.9045	$[1.0219, 1.4825]$
Frozen WM + scorer	1.3774	1.1229	1.6508	1.3837	$[.4630, .9998]$
MAD	2.4223	.6637	2.0638	1.7166	$[.9674, 1.1613]$
Value-aware Joint	1.3650	1.1108	1.2458	1.2405	$[.3484, .8281]$
PC-TIG (Feasible)	**.6455**	**.6221**	**.6893**	**.6523**	—

Method	Run 1	Run 2	Run 3	Run 4	Run 5
Frozen WM + scorer	21.6328	23.1507	21.5838	21.1599	20.1060
MAD	21.6320	20.4634	22.2126	23.0408	22.8498
Value-aware Joint	15.8835	22.5613	15.5087	19.0884	18.8196
PC-TIG (Joint)	15.6696	15.1114	13.8702	18.8163	17.1450
PC-TIG (Feasible)	18.4974	15.7047	17.1107	14.3917	15.4228

D.1 Projection and prediction diagnostics

Table 12 reports factual and update measurements for the five-seed PC-TIG (Feasible) action-level study. Transition L_1 and projection rate are means across seeds, with projection rate measuring the fraction of updates whose raw task and prediction gradients have negative inner product. State-classification and next-state entries are seed minima; projected inner product is the minimum across updates and seeds. Calibration drift is the maximum absolute change among frozen predictive parameters during score-head calibration. State accuracy is 50-state classification accuracy; next-state mode is the fraction of state–action pairs whose predicted modal successor matches the true mode; and transition L_1 averages probability-vector error over those pairs. The 10^{-10}-scale signed dot products are finite-precision residuals at the analytic zero boundary.

Table 12: Prediction and projection diagnostics across the three navigation scenarios.

Scenario	Trans. L_1	State acc.	Next-state mode	Projection rate	Min. projected dot	Calib. drift
Composition	.000446	1.000	1.000	.3521	-4.34×10^{-10}	0
Risk	.000588	1.000	.996	.4075	-1.31×10^{-10}	0
Bottleneck	.000616	1.000	1.000	.4325	-2.43×10^{-10}	0

E Equal-Head Representation Attribution

Using five independent training seeds, we compare Prediction-only WM, Value-aware Joint, and PC-TIG (Feasible) representations. Each branch has 350 warm-up and 1,600 representation steps. We then initialize three new score heads; within each seed they share identical parameters, minibatch streams, Bellman–anchor loss, target synchronization, and 1,600 frozen-representation steps. The held-out closed-loop evaluation grid is the Cartesian product

$$w_t \in \{81/101, 49/101, 21/101\}, \qquad \eta \in \{0, .8, 1.6, 2.4, 3.2\}. \tag{26}$$

This gives $15 \times 21 \times 21 = 6{,}615$ closed-loop planning instances per seed. Closed-loop values use exact true-kernel enumeration. Risk-switch accuracy is optimal-action accuracy at each positive η, restricted to the population of 50 states, 21 goals, 16 horizons, and 12 positive-risk comparisons whose unique exact dynamic-programming action differs from the $\eta = 0$ counterpart at the same time–energy weights. We average the four positive-risk comparisons at each of three time weights after requiring both exact candidate gaps to exceed 10^{-8}. Every representation uses the same contexts, planner settings, and score-head training schedule.

The PC-TIG (Feasible) representation obtains exact regret 17.9909, compared with 22.4751 for Prediction-only WM and 19.6150 for Value-aware Joint. Its paired improvement over Prediction-only WM is 4.4842 with a 95% interval $[2.0370, 6.9315]$ and lower regret in every paired seed.

Table 13: Equal-head representation attribution: five-seed means on the 3×5 held-out context grid.

Representation	Closed-loop regret ↓	Goal-hit probability ↑	Trans. L_1 ↓	Switch acc. ↑
Prediction-only WM	22.4751	.3309	.001547	.5971
Value-aware Joint	19.6150	.4524	.002050	.5831
PC-TIG (Feasible)	**17.9909**	**.4582**	**.000607**	**.6521**

At time weights $21/101$, $49/101$, and $81/101$, the paired improvements are respectively 3.9257 ($[.6004, 7.2510]$), 4.4777 ($[2.1627, 6.7926]$), and 5.0492 ($[2.9547, 7.1438]$); every mixture improves in all five paired seeds. Mean risk-switch accuracy is .6521 for the PC-TIG (Feasible) representation, .5971 for Prediction-only WM, and .5831 for Value-aware Joint. Calibration updates only score-head parameters in all 15 representation–run combinations.

F Evaluation Units and Statistics

Table 14 summarizes the independent runs and evaluation units behind each reported result.

Table 14: Training and evaluation units for each reported result.

Result	Independent runs	Evaluation scope	Instances per run
Held-out action order	5	3 scenarios, 7 queries	840
Exact closed loop	5	7 held-out queries	3,087
Equal-head attribution	5 paired	15 held-out queries	6,615
Stochasticity/budget	5 per condition	7 conditions, 5 methods	720
Topology	5 per layout	3 layouts, 5 methods	720
HalfCheetah	5	held-out segment sets	1,024

The supplementary material provides the per-run measurements and aggregate statistics. Navigation intervals use paired Student-t statistics over five runs. HalfCheetah intervals use hierarchical paired resampling over runs and held-out planning instances; table dispersions are sample standard deviations across runs. Each paired HalfCheetah interval uses 20,000 percentile-bootstrap replicates with seed 20270911: it resamples the five training seeds and then the 1,024 test instances within each sampled seed after averaging mixed preferences within an instance. Both methods share every sampled index; interval endpoints are the 2.5th and 97.5th percentiles. Appendix Tables 7, 9, and 11 show the run-level navigation results.

G Matched Prior-Method Specifications

All matched adaptations begin from the same seed-specific predictive model and use the same observations or replay, train–test split, task queries, task scorer, optimizer schedule, and evaluator as PC-TIG. The scorer receives the task loss with its input representation detached; hence each cited auxiliary objective determines the representation update while scorer capacity and supervision remain identical. Exact feedback uses five seeds, 350 predictive warm-up updates, 1,600 task updates, batch size 192, AdamW at 2.5×10^{-3}, and gradient norm 5. HalfCheetah uses five seeds, the shared 15,000-update warm start, 20,000 task updates, AdamW at 10^{-4}, gradient norm 5, and the common residual scorer over all twelve 25-step segments and every nonempty prefix.

Temporal Straightening. For three consecutive encoded observations z_t, z_{t+1}, z_{t+2}, let $d_t = z_{t+1} - z_t$. We add the cited curvature objective

$$\mathcal{L}_{\mathrm{TS}} = \mathbb{E}\left[1 - \frac{d_t^\top d_{t+1}}{\|d_t\|_2 \|d_{t+1}\|_2}\right]$$

with coefficient .1. The expectation uses consecutive true transitions; terms with a zero displacement are omitted. This adaptation introduces no additional network.

Calibrated VAML. In exact feedback, the full categorical kernels make the $(1, 0)$ estimator deterministic. For exact continuation value $v_{q,h-1}$, its representation objective is

$$\mathcal{L}_{\mathrm{CVAML}} = \mathbb{E}\left[\left(\sum_{x'} p_\theta(x' \mid x, a) v_{q,h-1}(x') - \sum_{x'} p^\star(x' \mid x, a) v_{q,h-1}(x')\right)^2\right].$$

On deterministic HalfCheetah segments, a target critic $\bar{V}(z, q, h)$ anchors the same specialization through $\mathbb{E}[(\bar{V}(\widehat{z}_{t+1}, q, h) - \bar{V}(z_{t+1}, q, h))^2]$. The critic is an MLP with hidden widths 256 and 128, is supervised by normalized factual suffix utility, and is copied to the target every 50 updates. Both feedback and pathwise objectives use coefficient 1.

Latent-Space Value Alignment. Let P_{prior} and P_{post} be 51-bin two-hot value distributions formed from predicted and encoded successors, respectively, and let sg denote stop-gradient. We implement the cited bidirectional alignment

$$.5\, D_{\mathrm{KL}}(\mathrm{sg}\, P_{\mathrm{post}} \| P_{\mathrm{prior}}) + .1\, D_{\mathrm{KL}}(\mathrm{sg}\, P_{\mathrm{prior}} \| P_{\mathrm{post}}).$$

The feedback value head has two 96-unit SiLU layers and is supervised by exact continuation mixtures. The HalfCheetah head has hidden widths 256 and 128, uses factual suffix returns, a training-return support enlarged by 1% at each endpoint, and a target copy every 50 updates. Alignment is activated after the shared predictive warm start with multiplier $1/\max(1, \mathcal{L}_{\mathrm{fact}})$. The supplementary artifact records the complete runner, protocol, checkpoint, and per-seed result hashes for every cell.

H Complete Continuous-Control Candidate Metrics

H.1 Decision structure and computational cost

Table 15: Measured task-phase wall-clock time (mean ± sample SD). Unmarked entries use A100; superscript B denotes RTX PRO 6000. HalfCheetah Temporal Straightening, CVAML, and Latent-Space VA aggregate 3, 4, and 4 A100 runs, respectively; all other entries aggregate five runs.

Method	Navigation (s)	HalfCheetah (s)
Frozen WM + scorer	$16.80 \pm .46$	543 ± 17
Prediction-tuned WM	—	$1{,}207 \pm 23$
MAD geometry	$24.83 \pm .80$	477 ± 24
Temporal Straightening	$63.68 \pm .41^{\mathrm{B}}$	935 ± 38
CVAML	$74.14 \pm .23^{\mathrm{B}}$	$1{,}000 \pm 54$
Latent-Space VA	$74.51 \pm .29^{\mathrm{B}}$	$1{,}018 \pm 52$
Value-aware Joint	$21.59 \pm .42$	—
PC-TIG (Joint)	$31.51 \pm .68$	$1{,}230 \pm 45$
PC-TIG (Feasible)	41.25 ± 1.20	$6{,}225 \pm 236$

Pathwise model and objectives. Let $c = (a_0, \ldots, a_{H_{\mathrm{P}}-1})$ be a complete action segment of length $H_{\mathrm{P}} = 25$ beginning at $x_0 = x$. The pathwise factual predictor G_θ comprises a latent transition F_θ, an observation decoder C_θ, and a reward head H^r_θ. Starting from $\widehat{z}_0 = E_\theta(x)$, it computes $\widehat{z}_{t+1} = F_\theta(\widehat{z}_t, a_t)$, $\widehat{\widetilde{x}}_t = C_\theta(\widehat{z}_t)$, and $\widehat{\widetilde{r}}_t = H^r_\theta(\widehat{z}_t, a_t)$. Tildes always denote componentwise training-set standardization. Inverse standardization gives $\widehat{r}_t$ and the predicted prefix return $\widehat{R}_{1:t} = \sum_{j=0}^{t-1} \widehat{r}_j$.

For factual vector rewards r_t, define $R_{1:t}(c) = \sum_{j=0}^{t-1} r_j(c)$. Let d_x and d_r be observation and reward dimensions and let $\sigma_R = \max(\mathrm{Std}_{\mathrm{train}}[R_{1:t}(c)], 10^{-4}\mathbf{1})$. Here $\mathrm{Std}_{\mathrm{train}}$ is the componentwise empirical standard deviation over all training instance–candidate–prefix triples, $\mathbf{1}$ is the all-ones vector, and the maximum acts componentwise. The frozen factual batch $\mathcal{B}_{\mathrm{F}}$ contains all candidates and

nonempty prefixes for 32 training instances. Its task-independent objective is

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{fact}} = \mathbb{E}_{(x,c)\sim\mathcal{B}_{\mathrm{F}}}\Bigg[\frac{\|\widehat{\widetilde{x}}_0 - \widetilde{x}_0\|_2^2}{d_x} + \frac{1}{H_{\mathrm{P}} d_x}\sum_{t=1}^{H_{\mathrm{P}}}\|\widehat{\widetilde{x}}_t - \widetilde{x}_t\|_2^2 + \frac{1}{H_{\mathrm{P}} d_r}\sum_{t=0}^{H_{\mathrm{P}}-1}\|\widehat{\widetilde{r}}_t - \widetilde{r}_t\|_2^2 + \frac{1}{H_{\mathrm{P}} d_r}\sum_{t=1}^{H_{\mathrm{P}}}\left\|\frac{\widehat{R}_{1:t} - R_{1:t}}{\sigma_R}\right\|_2^2\Bigg]. \tag{27}$$

A pathwise planning instance pairs an initial observation x with a task query q, which supplies the physical preference vector w_q. The true utility of a prefix is $U_{q,t}(c) = w_q^\top R_{1:t}(c)$. The score head combines the initial and predicted prefix latents, predicted return, query, and normalized prefix length; write its output as $\widehat{U}_{q,t}(c)$. Its final layer is initialized to zero around the factual base $w_q^\top \widehat{R}_{1:t}$, so task learning starts from the world model's physical-return estimate rather than an unrelated score. Let s contain the per-coordinate return standard deviations measured on the training set and define $\sigma_q = \|w_q \odot s\|_2$. Each coordinate of s has a 10^{-4} numerical floor for constant-return data. Let $\mathrm{SL}_{.25}$ denote Smooth-L_1 loss with transition point .25 in these standardized units; it is quadratic for small score errors and linear for large ones. The decision objective is

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{dec}} = \mathbb{E}_{x,q,c,t}\left[\mathrm{SL}_{.25}\left(\frac{\widehat{U}_{q,t}(c) - U_{q,t}(c)}{\sigma_q}\right)\right] + \mathbb{E}_{x,q}\left[\max_{c\in\mathcal{C}(x)}\left|\frac{\widehat{U}_{q,H_{\mathrm{P}}}(c) - U_{q,H_{\mathrm{P}}}(c)}{\sigma_q}\right|\right]. \tag{28}$$

For each query, the maximum-error term bounds the theorem's radius because $\epsilon^\circ_q/\sigma_q \le \max_c |\widehat{U}_{q,H_{\mathrm{P}}}(c) - U_{q,H_{\mathrm{P}}}(c)|/\sigma_q$. The first expectation teaches temporal credit on every nonempty prefix; the second penalizes the largest absolute final-candidate error. The common cost convention is $J_q(c) = -U_{q,H_{\mathrm{P}}}(c)$ and $\widehat{J}_q(c) = -\widehat{U}_{q,H_{\mathrm{P}}}(c)$.

Training setup. We use MO-Gymnasium HalfCheetah (environment identifier `mo-halfcheetah-v5`). The replay contains three million transitions collected while training a six-policy Multi-Objective Reinforcement Learning by Decomposition (MORL/D) population with Multi-Objective Soft Actor–Critic (MOSAC) (Felten et al., 2023). For each planning instance, a deterministic roll-in of length 1–200 under one frozen MOSAC actor supplies the initial state. From that state, one deterministic and one stochastic rollout from each of the six frozen actors produce twelve $H_{\mathrm{P}} = 25$ action segments, together with their trajectories and physical vector returns. A fixed instance-level split assigns the 6,144 instances to 4,096/1,024/1,024 train/validation/test sets. At evaluation, each model receives only the initial state, candidate action sequences, and query; Prediction-only WM ranks segments by the factual base $w_q^\top \widehat{R}(c)$. The MLP world model uses 17-dimensional observations, six-dimensional actions, two physical reward coordinates, latent width 128, and hidden width 256. For a replay transition (x, a, r, x^+), where x^+ is the successor observation, write $\widetilde{x}, \widetilde{x}^+, \widetilde{r}$ for the standardized versions of x, x^+, r, respectively. Let $\widehat{z}^+ = F_\theta(E_\theta(x), a)$, $\widehat{\widetilde{r}} = H^r_\theta(E_\theta(x), a)$, and let C_θ be the observation decoder. With sg denoting stop-gradient and MSE denoting mean squared error, the one-step replay objective is

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{warm}} = \mathbb{E}\,[\,\mathrm{MSE}(\widehat{z}^+, \mathrm{sg}(E_\theta(x^+))) + \mathrm{MSE}(C_\theta(E_\theta(x)), \widetilde{x}) + \mathrm{MSE}(C_\theta(\widehat{z}^+), \widetilde{x}^+) + \mathrm{MSE}(\widehat{\widetilde{r}}, \widetilde{r})]. \tag{29}$$

The expectation is over uniformly sampled transitions from the training replay. Each training run learns a prediction-only initialization with 15,000 AdamW updates to $\mathcal{L}^{\mathrm{P}}_{\mathrm{warm}}$, and every method in that run starts from the same model. Each method then receives 20,000 outer training iterations with learning rate 10^{-4} and the same planning-instance stream of batch size 16. Pathwise prediction updates evaluate all candidates and prefixes in the fixed 32-instance reference batch $\mathcal{B}_{\mathrm{F}}$; the matched MAD method uses replay prediction and MAD minibatches of 1,024. The four training preferences are $(1, .1), (.75, .25), (.5, .5), (.25, .75)$; evaluation uses $(1, 0), (.875, .125), (.625, .375), (.375, .625), (.125, .875)$. The 32 factual instances belong to the training split and are excluded from decision-loss minibatches.

This schedule gives every representation method the same predictive starting point and number of outer updates. Evaluating the 32 reference instances over all 12 candidates and all 25 prefixes defines the cross-query prediction measure independently of decision minibatches. Observation,

reward, and return standardization makes each prediction component dimensionless, so Joint uses the unit-weighted sum in Equation 10. The Smooth-L_1 transition, variance floor, batch sizes, and learning rate are selected on the validation split and held fixed for all five evaluation runs.

Matched MAD representation. The matched MAD representation samples same-episode state pairs separated by $k \in \{1, \dots, 25\}$ steps, one-step pairs, and random replay pairs. Its asymmetric chart distance is

$$d_{\mathrm{M}}(u, v) = .9\|[u - v]_+\|_\infty + .1\,\mathrm{mean}([u - v]_+). \tag{30}$$

Here mean averages chart coordinates. Let M be the learned MAD chart. For a k-step pair (x_i, x_{i+k}), an independently sampled one-step pair (x_j, x_{j+1}), and independently sampled replay states $x_{\mathrm{r}}, y_{\mathrm{r}}$, define

$$d_k = d_{\mathrm{M}}(M(E_\theta(x_i)), M(E_\theta(x_{i+k}))), \tag{31}$$

$$d_{\mathrm{one}} = d_{\mathrm{M}}(M(E_\theta(x_j)), M(E_\theta(x_{j+1}))), \tag{32}$$

$$d_{\mathrm{rand}} = d_{\mathrm{M}}(M(E_\theta(x_{\mathrm{r}})), M(E_\theta(y_{\mathrm{r}}))). \tag{33}$$

The matched MAD representation minimizes $\mathcal{L}^{\mathrm{P}}_{\mathrm{warm}} + \mathcal{L}^{\mathrm{P}}_{\mathrm{MAD}}$, where

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{MAD}} = .1\,\mathbb{E}_{k\text{-step}}\left[(d_k/k - 1)^2\right] + \mathbb{E}_{\mathrm{rand}}\left[[1 - d_{\mathrm{rand}}/25]^2_+\right] + .001\,\mathbb{E}_{\mathrm{one}}\left[[d_{\mathrm{one}} - 1]^2_+\right]. \tag{34}$$

The expectation subscripts identify the three pair-sampling schemes above. The three coefficients and the $.9/.1$ Simple distance follow the released MAD configuration (Steccanella et al., 2025). Division by k expresses distance per environment step, while 25 is both the largest sampled separation and the candidate horizon; the final term enforces the corresponding one-step local scale. $\mathcal{L}^{\mathrm{P}}_{\mathrm{warm}} + \mathcal{L}^{\mathrm{P}}_{\mathrm{MAD}}$ updates the encoder; $\mathcal{L}^{\mathrm{P}}_{\mathrm{warm}}$ also updates the pathwise predictor, and $\mathcal{L}^{\mathrm{P}}_{\mathrm{MAD}}$ updates the MAD chart. The identically initialized score head is trained, and only the head receives its decision gradients.

For the Feasible update, let L_0 be the loss on $\mathcal{B}_{\mathrm{F}}$ at iteration start. The factual AdamW proposal is backtracked over $\{1, 2^{-1}, \dots, 2^{-24}\}$ and accepted at the largest fraction yielding loss $< L_0$; if no fraction is accepted, the parameters return to their iteration-start values. The score head is then updated. At the resulting post-factual shared parameters, let g_{fact} be the gradient of the same factual loss and let δ_{dec} be the realized AdamW decision displacement. Define $\epsilon_{\mathrm{mach}} = \mathrm{finfo}(\mathrm{float32}).\epsilon$ and

$$m_{\mathrm{num}} = 16\epsilon_{\mathrm{mach}}\sqrt{\max(\|g_{\mathrm{fact}}\|_2^2\|\delta_{\mathrm{dec}}\|_2^2, 1)}. \tag{35}$$

In floating point, Equation (12) is realized by subtracting $(\langle g_{\mathrm{fact}}, \delta_{\mathrm{dec}}\rangle + m_{\mathrm{num}})g_{\mathrm{fact}}/\|g_{\mathrm{fact}}\|_2^2$ when the inner product is positive and the squared norm exceeds 10^{-20}; otherwise the proposal is unchanged. The margin places the rounded displacement inside the feasible half-space rather than exactly on its boundary. The projected proposal is backtracked over the same fractions and accepted at the largest fraction whose final factual loss is $\leq L_0$; if no fraction is accepted, the shared decision step returns to zero. The depth 24 resolves the line search down to the point at which an FP32 parameter displacement becomes ineffective; it changes the numerical search resolution, not the accepted-loss condition.

Table 16 complements Table 3 with preference-averaged pairwise accuracy, within-instance Spearman correlation, and velocity-only top-1 accuracy. Pairwise accuracy is the fraction of correctly ordered candidate pairs, and Spearman is the within-instance rank correlation, each averaged over the four mixed preferences. Velocity top-1 is exact-selection frequency at $w = (1, 0)$. Every method receives the same initial state, twelve 25-step action segments, task preferences, and planner budget.

Table 16: Supplementary HalfCheetah ranking metrics over five independent training seeds (mean $\pm$ sample standard deviation).

Method	Preference pairwise ↑	Preference Spearman ↑	Velocity top-1 ↑
Frozen WM + scorer	$.9142 \pm .0058$	$.9207 \pm .0074$	$.4461 \pm .0691$
Prediction-tuned WM	$.9281 \pm .0032$	$.9381 \pm .0021$	$.5117 \pm .0154$
MAD geometry	$.9156 \pm .0016$	$.9238 \pm .0024$	$.4549 \pm .0352$
PC-TIG (Feasible)	$.9343 \pm .0029$	$.9444 \pm .0032$	$.5641 \pm .0250$
PC-TIG (Joint)	$\mathbf{.9552 \pm .0013}$	$\mathbf{.9671 \pm .0020}$	$\mathbf{.6975 \pm .0198}$

Table 17: Paired advantage of PC-TIG (Joint) over Latent-Space VA. Positive values denote lower loss or regret and higher accuracy or correlation. Intervals use hierarchical resampling over training seeds and test planning instances.

Metric	Mean advantage	95% interval	Runs favoring Joint
Prediction loss	.0182	[.0095, .0295]	5/5
Preference regret	.3443	[.3187, .3709]	5/5
Preference top-1	.1229	[.0676, .1765]	5/5
Preference pairwise	.0245	[.0216, .0273]	5/5
Preference Spearman	.0266	[.0241, .0292]	5/5
Velocity regret	.9845	[.8883, 1.0948]	5/5
Velocity top-1	.1709	[.1385, .1979]	5/5

PC-TIG (Joint) leads all methods on the three additional ranking metrics. Together with Table 17, these results connect its lower candidate regret to more faithful pairwise, within-instance rank, and top-choice ordering.