HaM-World: Soft-Hamiltonian World Models with Selective Memory for Planning

Haoyun Tang[1*] **Haodong Cui**[2*] **Keyao Xu**[3] **Zhandong Mei**[1†] **Kun Wang**[4†]

[1]Xi'an Jiaotong University [2]Huazhong University of Science and Technology [3]Nankai University [4]Nanyang Technological University

NEURAL INFORMATION PROCESSING SYSTEMS

1 Why HaM-World?

Imagined rollouts drift with horizon and under dynamics shift, and an unconstrained predictor contracts as each predicted state is fed back. We add the two missing structures inside one shared latent state.

- **Input side.** With partial observability and action delay one frame is not a state; **Mamba memory** adds a history-conditioned input.
- **Output side.** A monolithic latent entangles pose, momentum and semantics; we split $z = [q, p, c]$ and add a **Soft-Hamiltonian prior** on (q, p).
- **Energy, not capacity.** One scalar H drives the (q, p) motion over energy level sets, and its field is divergence-free, so repeated prediction **does not shrink the latent distribution** to first order; energy change comes from action work.

2 Method: q/p/c latents, Soft-Hamiltonian pair & training

- **Encoder.** $z_t = E_\phi(o_t) = [q_t, p_t, c_t]$: a canonical pair (q, p) with learnable $H_\phi(q, p)$, plus context c for contact, friction and semantics.
- **Soft-Hamiltonian update.** The field $H_t = H_\phi(q_t, p_t)$ is mixed with a learned residual and a control port, both conditioned on the Mamba memory h_t:

$$\Delta q_t = (1-\alpha)\,\Delta q_t^{\text{net}} + \alpha\,\partial_p H_t, \qquad \Delta p_t = (1-\alpha)\,\Delta p_t^{\text{net}} - \alpha\,\partial_q H_t + G_t\,a_t$$

- **Soft, not strict.** α blends in the field: small action-free drift, and residual, contact and friction keep the prior **grounded, not idealistic**.
- **Energy-organized field.** Divergence-free, orthogonal to ∇H: **locally energy-consistent** rollouts; $G_t a_t$ adds action work.
- **Why Mamba.** h_t is read at every imagined step: a fixed-size state and $O(1)$ updates; a Transformer loses on 4/4 (panel 5).

Training objective & planning

- **Hamiltonian alignment**, **JEPA-style latent alignment** without reconstruction, and symlog two-hot **reward / value** heads (value with an EMA slow target); consistency and policy-prior terms complete the loss:

$$\mathcal{L}_{\text{ham}} = \left\|\Delta q^{\text{net}} - \partial_p H\right\|^2 + \left\|\Delta p^{\text{net}} + \partial_q H\right\|^2, \qquad \mathcal{L}_{\text{repr}} = \mathbb{E}_t \left\| g_\phi(\hat{z}_{t+1}) - \text{sg}\left(\bar{g}_{\bar{\phi}}(\bar{z}_{t+1})\right)\right\|^2,$$

$$\mathcal{L}_{\text{reward}} = \mathbb{E}_t\, \text{CE}_{\text{2hot}}\left(\hat{r}_\phi(\hat{z}_{t+1}),\, r_t\right), \qquad \mathcal{L}_{\text{value}} = \mathbb{E}_t\, \text{CE}_{\text{2hot}}\left(\hat{V}_\phi(z_t),\, R_t^\lambda\right)$$

- **Planner.** CEM over (z_t, h_t): $H = 6$, 128 × 6 candidates, re-planned per step.

Links & contact

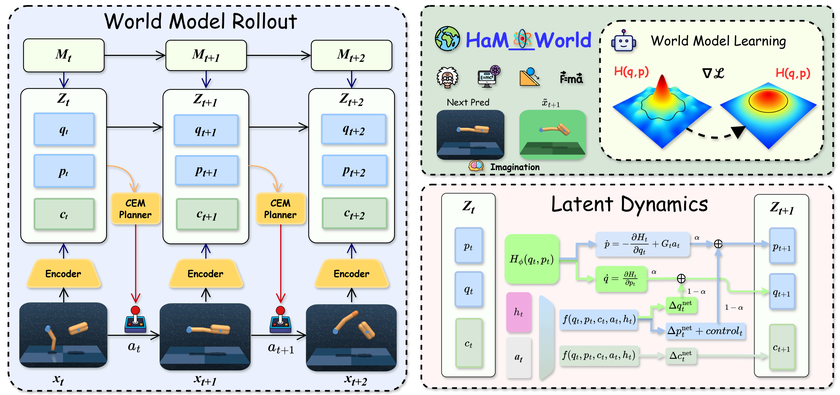
Paper · arXiv
arxiv.org/abs/2605.05951

Code · GitHub
github.com/HaoyunT/HaM_World

My homepage
haoyunt.github.io

Contact email: haoyuntang224@163.com

3 Architecture: one latent, two coupled stages



Left: Mamba memory M_t conditions one shared transition over (q, p, c), in which CEM plans directly. **Right:** one learned $H_\phi(q, p)$ drives the pair q, p (weight α), the network branch adds residual and control, and c updates separately.

4 Results: sample efficiency, rollout accuracy, OOD

Final return and six-task Avg. AUC (100k env steps, 3 seeds)

Final return @100k ↑	Finger Spin	Reacher Easy	Cheetah Run	Cartpole Swingup	Walker Run	Hopper Hop	Avg. AUC (6 tasks)
PPO	70.3±99.5	18.9±15.1	67.5±17.4	56.8±2.5	19.1±3.2	4.7±0.2	22.7
SAC	173.6±26.1	118.9±3.9	108.5±6.5	**65.8**±1.3	56.5±21.1	14.7±1.2	52.8
DreamerV3	72.3±102.3	18.4±7.9	**216.4**±17.0	55.8±0.4	104.6±22.9	11.3±0.3	51.7
TD-MPC2	232.2±8.2	105.1±52.3	155.7±17.4	15.0±17.0	101.9±29.7	14.0±7.8	83.3
HaM-World (ours)	**254.0**±14.1	**150.6**±7.5	184.4±8.4	58.9±3.8	**152.0**±33.3	**34.5**±8.9	**91.9**

k-step latent rollout error inside the planning range

k-step latent rollout MSE ↓	Finger Spin			Reacher Easy			Cheetah Run			Cartpole Swingup			Avg.
	k=3	k=5	k=7	k=3	k=5	k=7	k=3	k=5	k=7	k=3	k=5	k=7	
DreamerV3	7.20	9.29	11.51	10.46	14.26	18.22	13.29	18.17	24.63	7.25	8.59	10.34	12.77
TD-MPC2	3.48	3.58	3.61	3.34	3.38	**3.42**	4.28	4.40	4.47	4.66	4.79	4.87	4.02
HaM-World (ours)	**0.72**	**1.43**	**2.53**	**1.28**	**2.36**	3.68	**1.21**	**1.88**	**2.74**	**0.77**	**1.30**	**1.97**	**1.82**

Zero-shot OOD return: highest in all 12 perturbations

OOD return ↑ (avg. of 6 conditions)	HaM-World (ours)	TD-MPC2	SAC	DreamerV3	PPO
Finger Spin	**184.3**±4.0	167.3±7.8	118.4±13.1	55.2±78.0	47.3±66.9
Reacher Easy	**149.9**±5.6	132.0±37.2	92.4±8.7	9.3±3.5	12.9±10.9

- **Sample efficiency without sacrificing control.** Highest six-task Avg. AUC (**+10.3%** vs. TD-MPC2); **first on four of six tasks, second on two**; largest gains on Walker Run (+49%) and Hopper Hop (+147%).
- **Accurate where the planner looks. 11 / 12** MSE cells at k = 3, 5, 7 — the horizons the planners unroll (6 / 5 / 8) — and average error **45%** of TD-MPC2.
- **Robust zero-shot.** Highest return in **all 12** OOD perturbations (dynamics shifts, action delay, observation masking): **+10.2%** (Finger Spin) and **+13.6%** (Reacher Easy) over TD-MPC2, with far lower variance.

5 Ablations and design choices

Ablation: memory first, geometry adds headroom

Variant (3 seeds)	Cheetah ↑	Finger ↑	Reacher MSE@6 ↓ per-coordinate	Reacher OOD damp×2 ↑
Full HaM-World	**184.4**	**254.0**	**0.078**	**139.7**
A1: no q/p/c split, no geo. losses	169.8	247.7	0.085	131.9
A2: no memory	59.2	34.0	0.287	115.4
A3: Mamba → GRU	121.1	235.6	0.148	138.4
A4: pure Soft-Hamiltonian pair	181.7	251.3	0.080	136.7
A5: hard prior ($\alpha = 1$, no residual)	158.6	226.7	0.119	121.5
A6: no control port $G_t\,a_t$	153.6	219.3	0.129	115.4

- Removing memory (A2) hurts most; a GRU (A3) recovers return but not rollout accuracy: **input-selective** memory matters.
- Geometry (A1) adds a consistent margin on return and OOD; a hard $\alpha = 1$ prior (A5) over-constrains the dynamics, hence "**Soft**"; removing the control port (A6) hurts most, so actions need a direct path into the pair.

Memory swap: Mamba beats a Transformer on 4 / 4 tasks

Only the memory module swapped; final return @100k ↑	Finger Spin	Reacher Easy	Cheetah Run	Cartpole Swingup
Mamba (ours)	**254.0**	**150.6**	**184.4**	**58.9**
Transformer	91.9	144.7	60.7	51.3

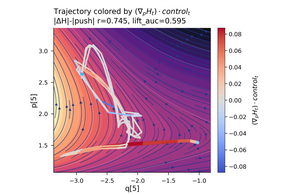
Planner swap: CEM beats MPPI on 6 / 6 measures

Retrained end-to-end per planner	Cheetah ↑	Finger ↑	Cartpole MSE@6 ↓	Reacher MSE@6 ↓	Reacher OOD mass×0.7 ↑	Reacher OOD damp×2 ↑
CEM (ours)	**184.4**	**254.0**	**0.009**	**0.078**	**158.5**	**139.7**
MPPI	166.1	231.7	0.019	0.104	143.2	124.6

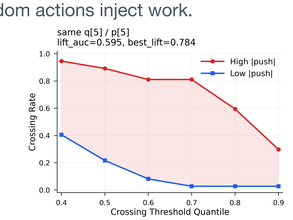
6 Mechanism diagnostics (Cheetah Run)



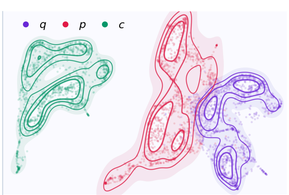
(a) Bounded action-free drift. H_t drifts by ≈1% over 200 zero-damping steps, while random actions inject work.



(b) Control couples to energy. On the slice (q_5, p_5), $\text{corr}(|\Delta H|, |\nabla_p H \cdot u|) = 0.745$ with u the control drive.



(c) Action work crosses energy layers. High-push steps cross equipotentials more often at every threshold (lift AUC 0.595).



(d) Separated q / p / c geometry. In the UMAP, q, p form elongated manifolds and c stays a compact cloud.