ML Lab Research Proposal

Name: Venkat - 12cx **Date:** 6/1/26 **Project:** SympScale — Do Structure-Preserving World Models Scale to High-Dimensional Chaos? **Target compute:** 8× AMD Instinct MI300X (single node), approx 2 weeks

---

1. Problem Statement

Learned world models (DreamerV3, PlaNet) predict the future with free-form recurrent cores (GRU/Transformer) that have no notion of physical conservation, so on chaotic conservative dynamics their rollouts drift off the constant-energy manifold and diverge by orders of magnitude past the Lyapunov horizon. We have already shown, at 4 dimensions (the circular restricted three-body problem), that a **capacity-matched** symplectic world model (a learned Hamiltonian integrated with 4th-order Yoshida) stays reliably bounded on every seed while the free-form GRU is bimodal — bounded on some seeds, divergent by 2–3 orders on most (10-seed locked result: Symp bounded 10/10, GRU divergent 6/10). The open and unanswered question is whether this **reliability advantage survives, vanishes, or strengthens as phase-space dimension grows into the hundreds**, and whether demonstrating it at scale *requires* differentiating through the full rollout rather than a truncated window. That is what we want to test.

2. Proposed Approach

We extend our existing structure-preserving core — `models/symp_rssm.py` (a separable learned Hamiltonian `H_θ(q,p,a)=T_θ(p)+V_θ(q)+Φ_θ(q,a)` integrated one step by `integrators/yoshida4.py`, with hard Noether/Casimir projection in `models/noether.py`) — from 4D toy systems to **high-dimensional Hamiltonian chaos**, and run a controlled scaling study against a capacity-matched free-form GRU (`models/gru_rssm.py`) plus RK4 and free-form-ODE ablations (`models/rk4_rssm.py`, `models/freeform_rssm.py`). Two coupled contributions:

1. **An empirical scaling law.** Reliability gap (bounded-vs-divergent rate, last-100-step rollout MSE, conservation residual) of the symplectic core vs. the capacity-matched GRU as a function of phase dimension $D \in \{4, 16, 64, 256, 1024\}$, measured across three system families that let us turn the dimension knob cleanly: the Fermi–Pasta–Ulam–Tsingou (FPUT) and Toda lattices (canonical scalable Hamiltonian chaos; Toda is near-integrable with O(D) conserved quantities, which stresses multi-invariant projection), planar N-body gravitation (conservative, chaotic, 6N-dim), and CR3BP (4D anchor, already locked). KS (`envs/ks.py`) is included as a dissipative-but-structured contrast for the projection machinery.
2. **A methodological result.** We test whether **untruncated, full-horizon backprop-through-time through the symplectic unroll** is necessary to learn the shadow Hamiltonian that controls long-horizon error — i.e., whether the BPTT truncation/checkpointing that an 80 GB GPU forces at high D measurably degrades the reliability advantage.

We pick this over a plain larger Transformer world model because the symplectic + hard-projection

stack gives an *interpretable, falsifiable* structural target (energy/Jacobi/Casimir residual to machine precision) and a clean capacity-matched control, so any reliability gap is attributable to structure, not parameter count.

Progress So Far (already done — in-repo, locked, reproducible)

This is not a from-scratch project; the 4D foundation is built, validated, and version-controlled. The MI300X ask is to **scale a working method**, not to discover whether it works at all.

- **Locked 4D reliability result (the seed of this proposal).** Capacity-matched (GRU 25,540 vs Symp 25,525 params, ratio 0.9994) 10-seed CR3BP sweep, committed and reproducible. Symp bounded on **10/10** seeds (last-100-step rollout MSE 4.39–14.48); the free-form GRU is bimodal, divergent by 2–3 orders on **6/10** seeds. Last-100 MSE ratio (GRU/Symp) median 81.8×, IQR [2.0×, 172.9×]. Figure + per-seed CSVs locked in `results/sweep/` and `figures/sweep/reliability_10seed.png`.
- **Hard Noether (Jacobi) projection working at the fp64 floor.** Newton projection drives the physical Jacobi-integral drift to mean **3.94e-11**, 95% CI [3.0e-11, 4.9e-11] over 5 seeds — ~5 orders of magnitude below the ≤1e-6 target, with rollout MSE unaffected. *This is precisely why the project is float64: the conservation signal we measure lives at the 1e-11 double-precision floor and does not exist in fp32.*
- **Validated reference integrators.** CR3BP DOP853 (`diffrax.Dopri8`), Jacobi drift 4.23e-11 over 100 periods; KS ETDRK4 pseudo-spectral, energy validated. These are the data generators we reuse at scale.
- **Differentiable symplectic core + projection, unit-tested.** `integrators/yoshida4.py`, `models/symp_rssm.py`, `models/noether.py`, with `tests/test_integrators.py` / `tests/test_noether.py` (energy conservation, straight-through projection gradients).
- **DreamerV3 integration de-risked.** SympRSSM is implemented as a drop-in RSSM in our DreamerV3 fork and smoke-tested in a full RL loop (trains stably, NaN-free). *That CUDA-pinned fork is out of scope for this MI300X work — the scaling experiments use only the pure-JAX training code, which ports to ROCm cleanly.*
- **Memory crux already measured, not estimated.** `scripts/profile_bptt_memory.py` profiles the compiled peak buffers of the real full-horizon BPTT program (float64). At our training batch (B=512, D=256, T=1000) it measures **108 GB**, crossing the 80 GB H100/H200 limit already at B=384 (81 GB) — confirming the 192 GB requirement empirically before we ever touch the hardware (table in §5).

3. Dataset

All data is **self-generated** by deterministic float64 reference integrators already in the repo and validated (CR3BP DOP853 via `diffrax.Dopri8`, Jacobi drift 4.23e-11 over 100 periods; KS ETDRK4 pseudo-spectral). No external download, no licensing, no PHI.

System	Generator	Phase dim D	Trajectories	Notes
CR3BP (anchor)	`envs/cr3bp.py` (DOP853)	4	50 × len 1,200	already locked, 10 seeds
FPUT lattice	new `envs/fput.py` (Yoshida-8 ref.)	16 / 64 / 256 / 1024	50 × len 1,200 per D	energy-conserving, scalable
Toda lattice	new `envs/toda.py` (DOP853 ref.)	64 / 256	50 × len 1,200 per D	O(D) invariants (Casimir test)
Planar N-body	new `envs/nbody.py` (DOP853 ref.)	6N (N up to ~40)	50 × len 1,200 per N	energy + ang. momentum
KS (contrast)	`envs/ks.py` (ETDRK4)	64 / 256 modes	50 × len 1,200	dissipative; mean-zero projection

Preprocessing: split on trajectories (train/test), z-score per phase coordinate, cache as memory-mapped float64 shards. On-disk footprint is small — approx **40–80 GB** of scratch for all systems and dimensions combined (trajectories are short and low-channel relative to the model state). Reference integration is the only data cost and is one-time.

4. Evaluation Metrics

Experiment	Metric
Reliability scaling (headline)	bounded/divergent count per D (threshold last-100 MSE), median + IQR of GRU/Symp MSE ratio (reported as median+IQR, not mean — ratios are bimodal)
Long-horizon rollout fidelity	last-100-step rollout MSE and prediction horizon (steps to normalized MSE > 1) vs. ground truth, per D
Conservation adherence	energy / Casimir residual `|C(s_t) − C(s_0)|` over 1,000-step rollout, with vs. without hard Noether projection; target ≤1e-6 projected
BPTT-horizon ablation (methodological)	reliability gap and conservation residual under full-horizon BPTT vs. truncated/checkpointed BPTT (window k ∈ {4, 32, 256, full}) — does truncation degrade the post-Lyapunov signal?
Capacity control	all comparisons param-matched to <0.1% (GRU vs Symp), as in the locked 4D result
Ablations	RK4-RSSM, free-form-ODE, no-projection, single- vs multi-invariant projection; 10-seed mean ± 95% CI

Seeds: 1–10 (data fixed, core/init/batch-order vary), matching the locked CR3BP protocol.

5. Compute Requirements

Estimates are scaled from measured CPU smoke runs of the existing 4D pipeline (`scripts/train_cr3bp_gru_baseline.py`, 40k steps, rollout 1,000) and padded ~25% since we have not yet profiled on MI300X.

Phase	**Hardware**	**Est. wall-clock**
Data generation (all systems × D, float64 reference integration)	1–8 MI300X / CPU	approx 0.5 day
Scaling sweep: 4 systems × {4,16,64,256,1024} × {Symp, GRU} × 10 seeds	8× MI300X	approx 4–5 days
BPTT-horizon ablation (full vs truncated, high-D systems only)	8× MI300X	approx 2 days
Method ablations (RK4, free-form ODE, single/multi-invariant projection)	1–8 MI300X	approx 2 days
Analysis + figures (`scripts/analyze_*`, `scripts/plot_reliability.py`)	1 MI300X / CPU	approx 0.5 day
Total	**8× MI300X node**	**approx 9–10 wall-clock days inside the 2-week window, with re-run buffer**

Why MI300X specifically — two independent cruxes.

(1) FP64 throughput. This project is **fundamentally float64** and cannot be done in fp32: the conservation signal we measure (Jacobi/energy/Casimir residual ~1e-11) lives at the double-precision floor and simply does not exist at single precision. MI300X is a scientific-compute part (CDNA3) with native, un-throttled fp64 — roughly **2–5× the FP64 throughput of H100/H200** depending on whether the matrix (MFMA) units are counted (MI300X ≈ 82 TFLOPS vector / 163 TFLOPS matrix FP64, vs ≈ 34 / 67 TFLOPS on H100/H200, which NVIDIA deliberately caps on datacenter parts). Our training loop is dense fp64 linear algebra — the Cholesky mass factorization in `T_θ(p)`, the Yoshida-4 force evaluations, and the Newton projection solves — so this gap maps almost directly to wall-clock. Note this is the *opposite* regime from MuJoCo-style RL: there is **no environment in the training loop** (data is generated offline), so nothing is CPU-bound; the loop is pure GPU fp64, exactly MI300X's strength.

(2) Memory for full-horizon BPTT. The memory driver is **reverse-mode differentiation through the full length-T symplectic unroll**. Yoshida-4 is 3 force evaluations per step; full-horizon BPTT over T=1000 steps, phase dim D=256, in float64, stores O(T·B·D·width) activations because the reverse `lax.scan` stacks per-step residuals. We **measured** the compiled peak buffer footprint (XLA ahead-of-time memory analysis of the actual gradient program —

`scripts/profile_bptt_memory.py`, float64, on the real `HamiltonianNet` + `yoshida4_step`):

T	batch B	D	width	measured peak	fits 80 GB H100?	fits 192 GB MI300X?
1000	256	256	512	54 GB	yes	yes
1000	384	256	512	**81 GB**	**NO**	yes
1000	512	256	512	**108 GB**	**NO**	yes
1000	768	256	512	**162 GB**	**NO**	yes
1000	1024	256	512	216 GB	NO	NO (needs sharding)

The crossover is sharp: full-horizon BPTT at our training batch (B=512, **108 GB**) **does not fit an 80 GB H100/H200** but sits comfortably on a single 192 GB MI300X; batch 768 (162 GB) runs to the MI300X ceiling. On the 80 GB part this forces either gradient checkpointing (≈3× recompute on top of 3 force evals/step) or BPTT truncation. Truncation is not merely slower — it **biases exactly the post-Lyapunov gradient signal** the method relies on, which would invalidate the headline claim. Quantifying that truncation effect is itself one of our deliverables. The near-integrable Toda/Casimir experiments add $O(D^2)$ constraint Jacobians per sample per step on top of these numbers. (Numbers are XLA AOT estimates compiled on CPU; we will confirm the exact device peak on the MI300X in week-1 bring-up via the same script's `--run` mode.)

Together: fp64 throughput makes each step cheaper, and 192 GB makes the full-horizon untruncated step *possible at all*. The 8-GPU node runs the system × dimension × seed grid in parallel within the window.

Software / dependencies. Pure JAX (the repo's training code is pure-JAX by convention; the CUDA-pinned DreamerV3 fork is **not** used here) → the official **JAX ROCm wheel**, no CUDA-only kernels. `diffrax` (data gen), `optax`, `equinox`, `numpy<2`, `matplotlib`. The differentiable integrator (`integrators/yoshida4.py`) and projection (`models/noether.py`) are isolated behind unit tests (`tests/test_integrators.py`, `tests/test_noether.py`) with a CPU fallback. Bring-up is one command on the node.

6. Expected Deliverables

By the end of the two-week window we aim (not guarantee) to have:

1. A **reliability-vs-dimension scaling figure** (bounded/divergent rate and MSE-ratio median+IQR for Symp vs capacity-matched GRU across $D \in \{4 \ldots 1024\}$, four systems).
2. A **conservation-residual-at-scale table** (energy/Casimir drift with vs. without hard projection, target ≤1e-6 projected, across D).
3. The **BPTT-horizon ablation**: quantified degradation of the reliability/conservation signal under truncated vs full-horizon backprop — the result that motivates the 192 GB requirement.
4. Method ablations (RK4, free-form ODE, single- vs multi-invariant projection), 10-seed

mean ± 95% CI.

5. A **workshop/NeurIPS-format draft** with all plots/tables generated end-to-end by repo scripts, per-seed JSON metrics in `results/`, and an **MI300X / ROCm reproducibility section** in the README.
6. New, reusable JAX environments (`envs/fput.py`, `envs/toda.py`, `envs/nbody.py`) added to the public MIT-licensed repo.

We are explicitly **not** promising "beats every baseline at every D" — the goal is a clean, reproducible test of whether structure preservation buys *dimension-robust reliability*, and whether full-horizon differentiation is necessary to show it.

7. Risk & Mitigation

Risk	Likelihood	Mitigation
JAX-ROCm op gap (FFT, `lax.scan`/while in the unroll, Cholesky) misbehaves on MI300X	Medium	Training code is pure JAX with no custom kernels; integrator + projection isolated behind unit tests with CPU fallback; validate on the node in **week 1** as the gating step before launching the full sweep.
Gradient explosion through the learned Hamiltonian at high D (a known failure mode of this stack)	Medium	Spectral-normed Hamiltonian layers (already in `models/hamiltonian_net.py`), gradient clipping, Leapfrog-2 → Yoshida-4 warmup, float64.
Capacity-matching is harder to hold exactly at high D	Low-med	Match parameter count to <0.1% per the existing locked protocol; report the exact ratio per D as we already do at 4D.
The scaling gap turns out flat (structure helps no more at high D than low D)	Medium	A flat curve is itself a publishable finding about dimension-independence; the **BPTT-horizon methodological result stands alone** as the paper's spine.
Multi-invariant (Casimir) projection too expensive even on 192 GB at D=1024	Low-med	Cap D at 256 for the multi-invariant Toda experiments; keep D=1024 to single-invariant FPUT where the activation set is the only memory driver.
Wall-clock overrun	Low-med	Sweep is embarrassingly parallel across 8 GPUs; can drop the D=1024 tier or trim to 5 seeds without losing the core

Risk	Likelihood	Mitigation
		scaling claim.

We will check in at the 1-week mark with the ROCm bring-up confirmed and intermediate scaling numbers, so scope can be trimmed or extended together.