Recoverable Low-Bit World-Dynamics Updates on a Laptop: Engineering Measurements and Negative Capability Controls

YiCheng Pan

Working draft, 25 October 2026 — not submitted

Evidence status. This report contains small synthetic-state and RGB/action experiments, a historical replicated coordinate-update candidate, and three actual full activated-gradient steps in an independently initialized native network with a 101.74B-parameter core execution path. Its held-out one-step errors do not improve after the second update. The report does not establish a capable trained 100B world model, real-world generality, or a new state-of-the-art result. Human author review and a substantive-contribution assessment remain pending.

Abstract

We implement recoverable disk-packed updates for native RGB/action dynamics on a laptop GPU, without a pretrained neural backbone. The current independently initialized model has 130.89B total parameters and a 101.74B-parameter core path per state slot. Actual two-bit matrices and fixed row scales occupy 32.77 GB. A full activated-gradient SGD step and a four-observation factored adaptive step complete with peak PyTorch-allocated CUDA memory below 1 GB at two cases per update. The second step changes 8.32 million quantized coordinates across 989 matrices, using 93.08 MB of row/column optimizer state. These are engineering measurements, not evidence of useful learned capacity. On 16 held-out synthetic action pairs, one-step RGB MSE increases from 0.001189584 to 0.001189768 after that update; action-effect error also increases. Earlier small-model experiments include distillation and native visual prediction with retained negative controls. No broad real-world result, general MoE advantage or trained 100B capability is established.

1 Research objective and scope

The intended long-term objective is an action-conditioned world model with more than 10^{11} unique parameters active in its dynamics prediction path. The present experiment addresses a smaller question: can a sparse prediction model, teacher-target cache and recoverable local training pipeline be implemented and evaluated on the available device?

Sparse expert routing has established precedents, including Switch Transformers [1]; our implementation uses top-2 routing rather than reproducing Switch's routing rule. Teacher-output transfer follows the general distillation idea [2], but uses continuous-state squared error. Latent prediction followed by visual decoding is a relevant broader direction; PAN's v1 implementation describes such a decomposition [4]. Our pilot does not implement PAN's video pipeline or establish algorithmic novelty.

2 Implementation

2.1 Sparse dynamics predictor

Each observation provides two bodies' positions and velocities, their masses and radii, together with a two-dimensional control acting on the first body. Four consecutive observation–action vectors form the input sequence. A causal transformer predicts the next eight-dimensional position–velocity increment. Static body attributes are retained during rollout.

For token representation h, a router selects the two largest expert logits. Selected logits are renormalized to weights $w_e(h)$:

$$y(h) = \sum_{e \in \mathrm{Top2}(r(h))} w_e(h) F_e(h).$$

Each F_e is a SwiGLU feedforward module. We include a load-balancing penalty with coefficient 0.01 and a router log-sum-exp penalty with coefficient 0.001. The student has two layers, width 64, four experts per layer and expert intermediate width 128. Its total and per-token active parameter counts are 231,616 and 133,312. A dense control uses intermediate width 256 and 132,800 active parameters.

2.2 Teacher and objectives

The dense teacher has three layers, width 128, intermediate width 512 and 790,144 parameters. It is fitted on 1,024 synthetic training episodes. The student input pool contains 256 separate episodes, while the small labeled subset contains 16 of those episodes. These are episode counts rather than independent state-window counts.

The distillation update combines 64 cached teacher-target examples and 64 true-label examples, with equal prediction-loss weights. Supervised students use 128 examples per update. All students receive 450 optimizer updates; the teacher receives 700. AdamW uses learning rate 4×10^{-4} and weight decay 0.01, with gradient-norm clipping at 1. The teacher's additional data, fitting time and target-generation time are charged to the distillation pipeline. Equal example counts do not guarantee equal runtime.

2.3 Sequential local training and tiling

A separate engineering test caches the dense teacher's normalized inputs and outputs for each feedforward layer. Each of 12 tiny experts is loaded, optimized, checkpointed and released in turn. Optimizer and sampling-generator states are preserved. The expert partitions are interleaved example indices; they are not claimed to represent semantic specialties. This test does not include a learned global router or end-to-end calibration.

For a local squared-error objective, write a feedforward output as a sum over intermediate-dimension tiles:

$$y = \sum_j y_j, \qquad L = \frac{\|y - t\|_2^2}{N}.$$

A first pass obtains y without storing all tile graphs. A second pass applies the same output derivative $g = 2(y - t)/N$ to each tile:

$$\nabla_{\theta_j} L = \left(\frac{\partial y_j}{\partial \theta_j}\right)^\top g.$$

This is a local vector-Jacobian-product identity, not a claim of a new training algorithm. It does not by itself provide full-transformer backpropagation through cached, fixed teacher features.

3 Experimental protocol

The original privileged-state pilot uses a simple two-dimensional two-body simulator with controlled acceleration, damping, wall reflections and pairwise collision impulses. The simulator is owned synthetic code and has not been validated as a realistic physical simulator. No images, robot trajectories or real-world measurements enter that original pilot; the later native diagnostics below use rendered synthetic images.

Training seeds are 11, 22 and 33. Episode-generation seeds are disjoint for teacher fitting, student fitting and evaluation. Each seed is tested on 96 independent IID episodes and 96 episodes with masses outside the training range. The primary metric is coordinate RMSE at rollout step 32, computed per episode and then averaged within each training seed. One-step, eight-step and action-effect errors are secondary metrics. No projection onto the true future state is applied during learned autoregressive rollouts.

All recorded runs use PyTorch 2.11.0 with CUDA 12.8 on an NVIDIA GeForce RTX 5070 Ti Laptop GPU. The initial main pilot completed in 289.60 seconds of measured total wall time; this is a tiny-model measurement and cannot be extrapolated into a large-model training estimate. Seed settings, data hashes, source hashes, raw outputs, checkpoints and measured times accompany the code.

4 Pilot results

Table 1: IID 32-step mean episode position RMSE in unit-box coordinates. Lower is better. Values are mean ± sample standard deviation across three training seeds; episodes are not counted as independent training replications.

Method	RMSE
Persistence	0.2630 ± 0.0045
Dense teacher	0.1798 ± 0.0085
MoE, small labeled subset	0.2680 ± 0.0059
MoE, mixed distillation	0.2186 ± 0.0206
MoE, full labels	0.1976 ± 0.0262
Dense, full labels	0.2064 ± 0.0213

Mixed distillation lowers the observed mean error by approximately 18.4% relative to the small-label control. It does not outperform the full-label MoE control in the aggregate. Because the simulator supplies ground truth directly, teacher inference is not automatically a cost-saving substitute for labels in this setting. The Python-dispatch sparse implementation is also slower than the dense control in the recorded training runs.

The paired seed-level distillation-minus-small-label difference is -0.04935 RMSE. An exploratory two-sided sign-flip test yields $p = 0.25$; Holm adjustment over the two declared primary contrasts gives $p = 0.50$. The second contrast is full-label MoE versus the approximately active-parameter-matched dense control. Three seeds cannot validate the sign-symmetry or normality assumptions.

Approximate Student-t intervals are reported in the statistical appendix as descriptive summaries, not confirmatory evidence.

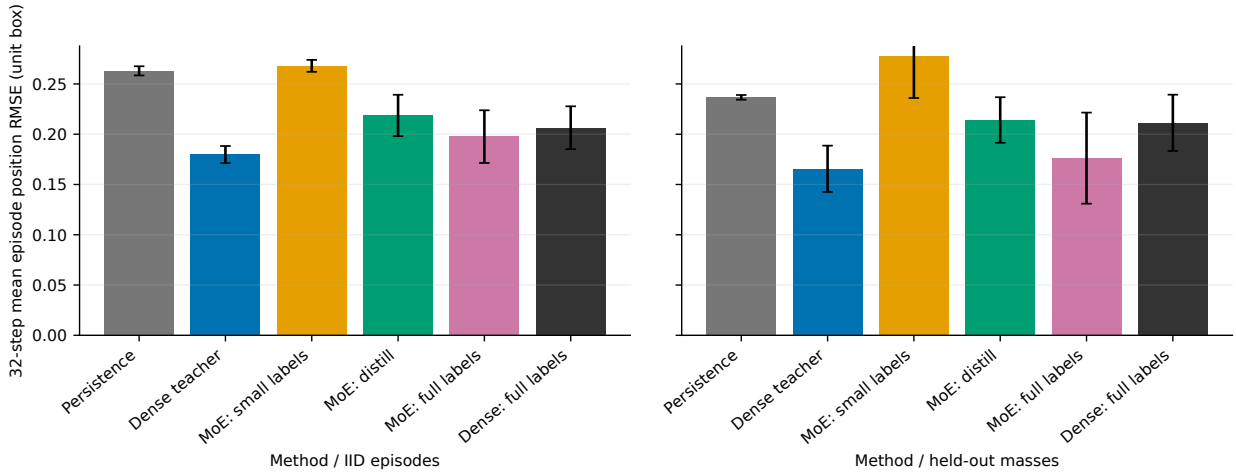


Figure 1: Actual pilot outputs for IID episodes and held-out masses. Bars are seed means; error bars are one across-seed sample standard deviation. The teacher and full-label controls show why improvement over a small-label baseline alone does not establish a more effective or cheaper general method.

5 Engineering checks

The locally trained dense teacher can be converted into independent expert copies while preserving its initial predictions to float32 tolerance: the observed maximum absolute difference is 3.58×10^{-6}. Three subsequent optimization updates produce unequal expert weights; this verifies that independent copies can change, not that useful specialization has emerged.

The disk-sequential test resumes all 12 experts from update 20 to update 40. A fresh uninterrupted 40-update control produces identical recorded expert weights and generator states. One control invocation encountered a transient Windows file-replacement error; bounded checkpoint-replacement retries were added before the successful control. Peak allocated CUDA memory for the resumed tiny local-feature test is approximately 21.6 MB; this excludes reserved driver memory and is not a large-model memory estimate.

A separate four-tile float64 local-SGD check matches the full-update weights with maximum difference 6.94×10^{-18}. This demonstrates the implemented local update identity on the tested tensors. It does not establish whole-model learning quality.

6 Large-architecture audit and laptop limits

A standalone top-2-of-8 shape specification uses width 8,192, 64 layers and expert intermediate width 28,672. Its meta-device parameter count is 377,962,553,344 total and 107,379,613,696 active per state-token. These counts refer to the dynamics backbone's unique parameters; repeated rollout calls are not added together. The specification uses privileged-state interfaces and has no trained visual grounding.

BF16 weights for that historical specification alone would occupy 755.93 GB, before gradients, optimizer state, activations or data. Splitting execution can reduce resident GPU memory, but cannot eliminate weight storage, transfer time or the need for training evidence. Language-backbone sizing explorations are excluded from the native-world-model objective. The following engineering candidate is a different, smaller-storage configuration, with no LLM backbone.

7 Subsequent native RGB/action diagnostics

The native predictor receives four RGB-derived 128-dimensional latents and two-dimensional actions; simulator states are used only by the evaluator. A learned convolutional codec maps 32×32 RGB observations to latents. A motion-residual model predicts a correction to observed constant-velocity latent extrapolation. A single-seed exploratory screen uses 2,048 training episodes, with inertial, gravity, spring and wind regimes. Its dense and sparse versions share the motion prior and data pool. On the development split, latent MSE is 0.02018 for MoE, 0.01956 for dense, and 0.06932 for constant velocity. This bundled prior/data change does not isolate either cause of improvement.

An independent diagnostic uses generation seeds 8401 and 9401, each with 64 episodes; the latter changes the mass range. The evaluator reconstructs the same hidden force and verifies exact equality with the original simulator transition before replacing only the current action with positive/negative x or y controls. It compares predicted and true differences between the paired future latents. IID action-effect MSE is 0.005996 for MoE, 0.004972 for dense and 0.010844 for zero action response. Under held-out masses, MoE's error is 0.003478, worse than zero response at 0.002931; dense obtains 0.002708. Positive directional agreement alone therefore conceals an effect-magnitude failure.

Starting from four observed frames and then using only predicted latents and known controls, 16-step IID rollout latent MSE is 3.2534 for MoE, 3.4257 for dense and 34.3936 for constant velocity. Corresponding foreground pixel MSE is 0.2166, 0.2224 and 0.2310. All are weak long-horizon image predictions; the codec is a further bottleneck. These one-seed synthetic diagnostics neither establish a general MoE advantage nor evaluate the large network below.

8 Actual large native execution and coordinate update

The distinct large engineering candidate has width 8,192, 96 causal Transformer layers, intermediate width 28,672 and 256 attention heads. Ten layers contain top-2-of-3 independent sparse experts; the remaining 86 use dense feedforward modules. It has 107,513,389,056 total parameters and 100,466,958,336 active parameters on the recorded token path. Consequently this hybrid is mostly dense and sparsity saves only a small fraction of active parameters. The codec is not included in those backbone counts. Repeated tokens or rollout invocations are not added to parameter counts.

Actual files store 2-bit matrices in newly expanded layers and 3-bit matrices in the learned prefix and prediction head, with FP32 per-row scales and a small FP32 resident state. The 541 packed matrices contain 107,509,448,704 scalar coordinates. Weight files, scales and resident state occupy 27,829,633,536 bytes. Initialization takes approximately 527 seconds. It widens our own small trained native RGB/action model by replication and adds residual blocks with a small nonzero gain. Although files are independent, the replicated initialization adds no learned knowledge or demonstrated useful capacity.

Tiled dequantization executes linear maps without storing a full-precision master copy. Backward rereads the old weights for input derivatives. One selected coordinate group receives stochas-

tic, fixed-grid projected SGD; this is an optimization approximation, not the derivative of rounding or an AdamW-equivalent update. A bounded write-ahead log coordinates packed writes with the small optimizer/RNG checkpoint. Separate tiny tests compare packed updates and input/resident gradients with materialized references and check interrupted-commit recovery. Those tests cannot establish large-scale convergence.

The first actual large run verifies every weight-file SHA, executes all 96 layers on a batch of four four-frame synthetic histories, and completes backward and a committed coordinate update. In the final sparse layer's selected expert, 562,196 quantized values change in its output matrix; other selected matrices have no nonzero grid displacement in that step. The same-batch latent loss changes from 0.02473319 to 0.02472999. Forward plus backward takes 176.81 seconds; hash verification, checkpointing and post-update forward bring total time to 287.44 seconds. Peak PyTorch-allocated CUDA memory is 513,459,200 bytes, excluding CUDA context, reserved memory and other applications. Counted matrix reads across forward, checkpoint recomputation, backward and post-update forward total 111.26 GB; this is logical read volume, not a measured physical SSD transfer count.

The recorded path proves parameter execution and limited native optimization. It does not prove that all active coordinates are independently learned, that a large batch or useful video context fits, that the expanded model improves on its teacher, or that 100B useful learned capacity exists. Most packed parameters have not been updated. Strong native pretraining or diverse training data, sustained optimization, independent capacity controls and broad real-world evaluation remain necessary.

9 Fresh-weight recurrent visual experiments

The current research route starts every neural component from fresh initialization, including the image encoder and decoder. It loads no pretrained checkpoint and uses no language-model backbone. Architectural precedents include separating state from visual readout and bounding recurrent context; this does not establish an original architecture or reproduce the performance of published systems.

The first 160-update RGB prototype erases small foreground objects. A second 480-update flow/content version retains more foreground but worsens development rollout pixel MSE from 0.01388 at initialization to 0.02580. These negative results are retained. A third version adds a local spatial motion/content branch, explicit attention projections, actual-observation-only memory writes, a self-rollout curriculum and four-step truncated image/state gradients. These changes are bundled, so the screen cannot attribute gains to any one mechanism.

The third version has 2,069,899 total parameters and completes 960 updates in approximately 788 seconds including data generation and evaluation. With four observed frames followed by self-generated inputs, its 24-episode development rollout pixel MSE improves from 0.01379 to 0.00995. The checkpoint still produces blurred and deformed objects. Removing memory at inference yields 0.00969, slightly better; removing the entire global state/MoE/memory branch yields 0.01048. These are inference interventions, not separately trained matched architecture controls.

A separate 32-episode diagnostic preserves the first five images exactly and keeps every camera pose identical between paired arms. Only subsequent forces on one body are reversed. Predicted image action-effect MSE is 0.002404, worse than the zero-response baseline at 0.001447. Thus ordinary rollout improvement does not demonstrate controllable physical prediction or useful memory. All experiments use one training seed and owned synthetic perspective scenes.

For the actual recurrent RGB/action architecture, a tiny independent eager oracle checks four-

step backpropagation through reused attention and expert matrices on CPU and CUDA. Gradients accumulate before any packed weight update; maximum matrix-gradient disagreement is below 6.5×10^{-10}, and all checked quantized updates match. This validates tested numerical execution, not convergence at scale. A new owned MuJoCo dataset contains 160 episodes and 2,720 frames across falling, collisions, rolling, stacking and pendulum motion. Repeated simulator states are exact; rendered RGB differs by at most one uint8 level. This dataset has not yet been used to establish learned transfer or broad generality.

10 Independent fresh 101.74B core: actual full updates

The active large route replaces replicated initialization with independent named random streams for every packed matrix. Width is 8,192, with 64 layers, intermediate width 17,920 and top-3-of-4 experts. The model has 130,886,421,899 total parameters; its core alone activates 101,743,329,280 unique parameters on one recorded state-slot path. Repeated tokens and the union of experts selected by other examples are not added to this active count. All 1,034 matrix files are actually allocated and independently hash-reread verified. Packed matrices and fixed FP32 row scales occupy 32,769,507,328 bytes; resident parameter tensors add 96,548,396 bytes. All neural components are freshly initialized; no LLM, pretrained visual codec or neural teacher is loaded.

Sparse byte-level undo records permit updates to all activated matrices without keeping a full model-sized old-weight copy. Input derivatives use old weights before direct stochastic projection to the two-bit grid. The initial large SGD step delivers backward to 992 matrices, changes 12,281 scalar codes across 819 matrices, and commits in 561.00 seconds including verification and the subsequent forward. Peak PyTorch-allocated CUDA memory is 674,634,752 bytes. These allocation figures exclude driver/context, reserved memory and other applications.

The next stage adds four observed RGB frames encoded together, relative time, poses and controls, followed by one state assimilation and one large dynamics call. No future image is an input. This interface avoids repeated matrix use in one backward; it is not backpropagation through earlier self-generated worlds. The packed optimizer uses an Adafactor-style row/column second moment, preconditioned-direction RMS clipping and parameter-RMS relative step size[1]. Gradient tiles are recomputed instead of retaining dense gradients or master weights. Four standalone matrix updates match the installed reference optimizer's nearest-projected codes on CPU and CUDA. In a full tiny native graph, raw gradients agree within the recorded tolerances, but a CUDA comparison differs in one packed byte near a rounding boundary; end-to-end bitwise equivalence is not claimed. Tests with identical supplied gradients isolate projection correctness from that numerical difference. Interrupted-write recovery is checked separately.

The actual four-frame factored update starts from step 1 and changes 8,323,946 codes in 989 matrices. Factor state occupies 93,079,552 bytes; preserved reverse undo occupies 41,615,505 bytes. Total time is 1,243.57 seconds, including file verification, checkpointing and post-update forward; peak PyTorch-allocated CUDA memory is 976,403,456 bytes. Same-batch pixel MSE falls from 0.0002326134 to 0.0002320187. This single-batch decrease does not imply generalization.

An owned diagnostic dataset has 80 paired MuJoCo scenes across five rigid-body families. Branches have identical observed histories and camera paths; only the current force on a visibly marked body differs. Pairs 0–63 are the training pool and pairs 64–79 remain outside training. The evaluator uses the same four-observation interface before and after step 2. A hash-verified read-only reverse-delta view reconstructs prior weights without duplicating 32 GB. On the 32 held-out cases, RGB MSE worsens from 0.0011895840 to 0.0011897683; copying the last image gives

[1] Adafactor precedent: https://arxiv.org/abs/1804.04235. We do not claim this optimizer mechanism as novel.

0.0011947927. Action-effect MSE worsens from 8.5528249×10^{-5} to 8.5536303×10^{-5}, compared with 8.5538333×10^{-5} for predicting zero action effect. The minute gap against zero response is not evidence of useful controllability. No inferential claim is made from these single-training-seed, one-step synthetic diagnostics.

A subsequent tested I/O implementation buffers at most one matrix's packed changes, durably writes all undo payloads and descriptors, and then flushes the matrix once. On a 16-tile test, synchronization calls fall from 48 to 3, with the same updated bytes and exact rollback at two simulated interruption points. This call-count reduction is not a measured whole-training speedup. Batched paired objectives have an independent NumPy loss/gradient oracle. The actual third large update combines grouped writes with 16 training pairs (32 cases) per traversal. It completes in 1,553.44 seconds, changing 7,581,605 codes across 1,022 matrices, with 2,973,071,360 bytes peak PyTorch CUDA allocation. Observed cases per end-to-end wall-time are 12.81 times those of the earlier two-case factored run. Batch and I/O changes are bundled, and cases, cache state and background verification activity differ; this is not an isolated causal speedup or a learning-efficiency claim. Same-batch pixel MSE increases from 0.002177281 to 0.002179202, so the engineering gain alone does not establish improved prediction.

The step-3 checkpoint is then evaluated once on the same fixed development cases; step-2 measurements are retained from the prior run rather than recomputed. Dataset hash, case range, history/target configuration and neural source match; the current checkpoint's parent SHA is verified. The older step-2 report predates explicit checkpoint-SHA metadata, so this is a provenance-linked cached comparison rather than a repeated paired execution. RGB MSE increases from 0.0011897683 to 0.0011910101. Action-effect MSE is 8.5543135×10^{-5}, slightly worse than zero response at 8.5538333×10^{-5}. These results do not show useful controllable prediction. They motivate a cheaper matched full-precision versus two-bit learning diagnostic before further large updates; they do not isolate a specific cause of failure.

11 Small learning controls before further large updates

The latest large updates do not demonstrate useful action response, so we run cheaper controls before extending expensive training. All controls start from the same independent small initializer: width 32, two layers, four experts and top-3 routing. They use materialized full-graph matrix gradients; these are learning controls, not measurements of the streamed 100B backend. Continuous FP32 matrices and direct stochastic two-bit matrices use the same factored preconditioner, RMS clipping and relative step rule. A third control freezes these matrices while training resident convolutions, biases, norms and routers. Resident AdamW, paired batches and RGB/action inputs match across arms.

With 200 updates and both learning rates 10^{-4}, development action-effect MSE is 8.542836×10^{-5} for continuous matrices, 8.542934×10^{-5} for direct two-bit updates and 8.542841×10^{-5} for frozen matrices. Zero response obtains 8.553833×10^{-5}. These small differences do not establish useful controllability or a trained-matrix advantage. State tensors are initially identical; GPU forward differences reach 1.79×10^{-7}. An earlier bitwise-forward assertion failed and is retained; the rerun protocol explicitly checks exact state tensors and a 10^{-6} forward tolerance. The continuous optimizer separately agrees with an installed reference to within 1.5×10^{-8} in the tested weights.

We next test an exploratory normalized paired-effect objective. For each pair, the weighted effect MSE is divided by weighted target-effect energy plus 10^{-5}, then averaged over pairs. Its coefficient is 0.05. Another variant also supervises the existing global readout with coefficient 0.05 for this normalized effect loss and 0.25 for RGB MSE weighted toward changed pixels. These are

training targets, never future-image inputs. The global image is reconstructed from the model's already computed flow/content fields; this adds no neural parameters. CPU/CUDA tests check loss derivatives against independent NumPy calculations, including a zero-effect pair, and recover the original global image exactly. These are numerical checks, not learning evidence.

At 1,000 updates on the original 64 training pairs, the unchanged objective's development action-effect error is 22.84% worse than zero response despite improving ordinary image MSE. Normalized main-output supervision alone is 3.14% worse; adding global supervision yields a 2.98% reduction. The raw global image still fails the action control. None passes the preregistered exploratory gate of a 20% reduction; that gate is not a statistical test.

An independently seeded larger corpus contains 800 pairs, 1,600 trajectories and 27,200 frames across the same five synthetic physics families. The split has 640 training pairs and 160 development pairs. With the same 1,000 updates and batch size, the original objective reduces development action error by 0.11% against zero response; combined supervision reduces it by 2.05%. Increasing both learning rates to 10^{-3} with combined supervision gives reductions of 4.83% for continuous matrices, 5.42% for two-bit matrices and 4.94% for frozen matrices. These are one-initialization screens, not evidence of a general quantized or MoE advantage. In particular, useful large-matrix learning remains unestablished. Concurrent rendering and changing system/cache activity preclude interpreting these screen runtimes as throughput benchmarks.

For a task-specific control, we fit eight ridge coefficients using training RGB gradients, current force and camera pose. It uses the benchmark's observed red-actuator convention and no simulator-state inputs. On development pairs, its clipped-image action effect reduces error by 5.42% against zero response; shuffling current forces makes error 6.27% worse than zero response. Its full-image MSE is 0.00550, worse than copying at 0.00176, so this component control does not improve general future-image prediction. The simple control and the frozen-matrix results are reasons to investigate representation and supervision before investing further large updates.

12 Independent takeover verification and measured feasibility

After this draft was prepared, an independent read-only audit re-verified each numeric claim against its result file and then ran new owned experiments on the same device. This section records the verification and the measured reasons why the >100B objective remains unmet. All artifacts cited here are retained under `verification/takeover-2026-10-25/` and in the run directories named in the text.

Asset and checkpoint verification. The current step-3 checkpoint in the 100B work root is 389,265,362 bytes with SHA256 `da1396...c463`, and `torch.load` reports `step=3`. The model has 130,886,421,899 total and 101,743,329,280 core-active parameters; 1,034 packed two-bit matrices occupy 32,769,507,328 bytes (0.250 B per matrix parameter); resident FP32 parameters are 24,137,099 (0.018% of total). The three updates are three single-step resumed processes: earlier records are replayed from the checkpoint, and only one step executed per process. Step 3 changed 7,581,605 codes and made its own batch 0.088% worse (0.00217728 to 0.00217920 pixel MSE). On 16 held-out synthetic pairs, action-effect MSE is 8.554314×10^{-5} versus a zero-response baseline of 8.553833×10^{-5} — 0.0056% worse than predicting no change. No broad capability, real-world generality or competitive evaluation exists.

Why three steps could not have trained this model. The packed update moves each weight by $\mathrm{lr} \cdot \mathrm{RMS}(W)$ against a grid spacing of 2 scale; for fresh two-bit weights this is 1.118 packed_lr of

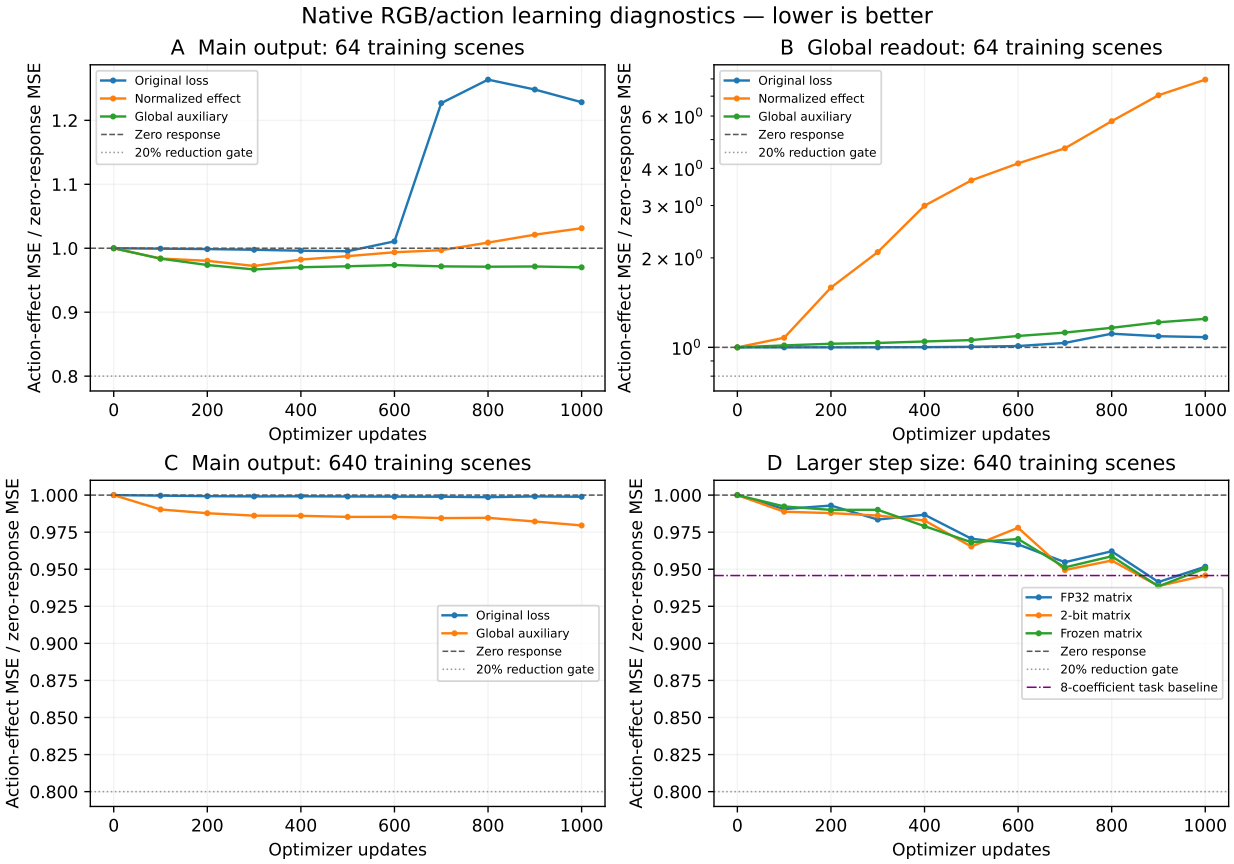


Figure 2: Actual small native-model development curves. Lower action-effect error ratio is better; 1 denotes zero action response and 0.8 is the exploratory 20% reduction gate. A–B use 64 training pairs, C–D use 640. D compares precision/frozen controls at the larger step size. One initialization per arm; there are no across-training-seed error bars or inferential claims. These are not the 100B model's capability curves.

one grid spacing per step, independent of matrix width. At `packed_lr`$= 10^{-4}$ only 5.79×10^{-5} of the 1.31×10^{11} matrix parameters changed in step 3. To test this mechanism directly, an isolated 32.77 GB copy of the checkpoint was trained for two steps at `packed_lr`$= 10^{-3}$ (SHA of the parent checkpoint verified; original large assets untouched): code changes rose from 7,581,605 to 66,805,973 (8.81×) and 56,659,388 (7.47×), as predicted, while a fixed batch's pixel MSE did not fall in one step (0.00236839 to 0.00236904). The learning-rate scale is validated; the step budget is the wall.

Readout and observation attention at 100B. A read-only diagnostic completed all 64 layers in 186 s with 992 consumed matrices hash-verified and no model file modified: observation attention is essentially uniform (normalized entropy 0.99999988; maximum probability 0.006976 versus uniform 0.006944), and the readout places 78.95% of its attention mass on 60 structurally zero memory tokens per case. The large readout is dominated by padding, consistent with the earlier small-model diagnostic.

Matched learning screens. A 5,000-step run on 640 training pairs (160 development pairs, stochastic two-bit updates, lr 10^{-3}) fits the paired action effect on its training split (+36.4% gain

by step 5,000) but development gain peaks at 5.01% at step 1,500 and falls to −19.2% by step 5,000 while development pixel MSE keeps improving; predicted effect energy is 0.23 of true energy. With 6,000 fresh pairs and 2,000 steps, development gain reaches 6.12% at step 2,000 (still rising) with training gain 7.11% and a 0.99 pp train–development gap, versus 7.6 pp at 640 pairs: data quantity improves generalisation, but a linear extrapolation of the 6,000-pair curve reaches the 20% gate only near 8.7×10^3 steps. A matched 1,500-step decomposition shows that the proxy's learning is carried by its resident FP32 parameters (4.84% development gain with packed matrices frozen, versus 5.01% with all parameters trainable); training only the packed matrices leaves the action effect unchanged at the 10^{-6} level because the zero-initialised resident output heads are frozen in that arm — this proxy therefore cannot by itself establish whether the 100B model's 99.98% matrix capacity can learn. A zero-memory-token readout mask changes the function (5.84% development gain at 1,500 steps versus 5.01% unmodified) but is not a step change and must be reinstalled when loading a checkpoint.

Matrix-dominated proxy and a first mechanism probe. To test whether the 325K proxy's 78% resident-parameter share was hiding matrix learning, a 64,454,539-parameter proxy was independently initialised with 138 packed 2-bit matrices (98.37% of parameters in matrices, resident FP32 1.63%; independent PCG64 streams; full-file SHA re-read verified; not trained). At 1,000 matched steps, all-parameter training reaches +5.08% development action-effect gain, resident-only training +4.51%, and matrix-only +0.03% (the matrix-only arm is confounded because the zero-initialised resident output head is frozen). Even at 98.4% matrix share the packed matrices add only 0.57 pp over the resident heads. We then proposed TLC (two-timescale low-bit learning): train the already-stored, previously frozen FP32 row scales continuously on the 2-bit grid ($dL/du_i = e^{u_i} \sum_j g_{ij}(2c_{ij} - 3)$, bounded per-step log change; about 145 MB of scale/Adam state at 100B scale versus 32.77 GB of matrices) and add a coarse 6×6-pooled counterfactual effect-field direction term. The first prototype (4 arms × 1,500 steps and a 3,000-step repeat) is *negative-to-partial*: the control's development gain peaks at 6.22% and then inverts to 0.61% at step 1,500 as its predicted-effect energy rises to 0.118; the field-alignment term keeps development at 4.84% with energy 0.014 and a 1.04 pp train–development gap (control 3.45 pp), i.e. it acts as a stability regulariser; scale training ends higher than the collapsed control (5.59% at step 1,500) but has a lower peak and was worse at 1,000–1,250 steps. Neither component approaches the 20% gate. The proposal and its pre-registered falsification criteria are retained in the artifact tree; no capability claim is made. Figure 5 shows the measured stability effect at 3,000 steps: the counterfactual field-alignment arm keeps a positive development gain (+2.51%) while the control inverts to -14.70%, with much smaller predicted-effect energy and train–development gap.

Privileged-state distillation (PIT). We also tested the complementary paradigm of training-time privileged supervision: the owned simulator's exact body state is used only as a teacher, while inference inputs remain RGB/actions/poses. Three variants ran at matched 3,000-step budgets with the same data, model and optimizer as their in-run controls: (i) a slot-state probe supervised by absolute body positions, (ii) a readout probe supervised by counterfactual position deltas, and (iii) distillation of an image-space flow field directly on the causal local-flow path. None passed the gate: best development gains were 4.83%, 5.02% and 5.42% versus controls at 5.17–5.26%, and all arms still inverted late. A read-only oracle diagnostic that replaced the predicted local flow with the privileged simulator-derived flow gave an oracle effect gain of -3.35%, and a flow-only parameterisation gave -0.25%; even perfect privileged centre-flow does not approach the gate. Inspecting the data explains why: in a representative pair the two branches differ in the 3-D cen-

tre of one body by about 0.0066 world units (at most 0.2 projected pixels) while the image-space effect is dominated by rigid-body rotation and shape change. Centre-derived teachers cannot represent that effect. We therefore extended the owned generator to record body quaternions (dataset `data/paired-rigid-v4`; actions/poses/positions identical to v2, RGB within the documented one-level renderer tolerance) and ran a fourth variant (iv) that distils a surface-point flow field: eight surface points per body are rotated by the exact body quaternions, so rigid-body rotation is expressed in the teacher. It also failed: best development gain 5.43% at step 1,250 and −11.90% at step 3,000, versus the in-run control's 5.79% and −8.61%. Under the pre-registered rule the PIT family as specified is **falsified**: privileged state supervision, even with full rigid-body orientation, does not produce the 20% counterfactual action-effect capability on this architecture, data and budget. No capability claim is made.

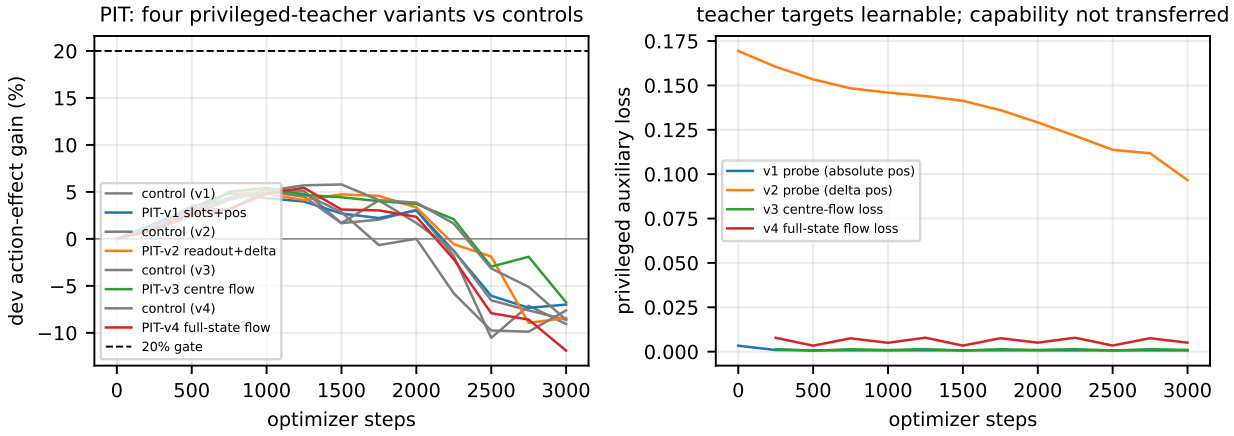


Figure 3: Privileged-state distillation (PIT) screens on the 325,003-parameter proxy at matched 3,000-step budgets. Left: four variants (slot probe on absolute positions, readout probe on counterfactual position deltas, centre-flow distillation, and full rigid-body-state surface-point flow distillation) all track or underperform their in-run controls and none approaches the 20% gate. Right: the privileged auxiliary losses do fall (the teachers are learnable), but the capability does not transfer; the oracle diagnostic shows this is at least partly a teacher-representation limit (rigid-body rotation dominates the paired image effect while body-centre displacement is about 0.2 pixels). Negative diagnostics, not capability results.

Measured feasibility. One 100B step costs 1,247.47 s of forward/backward and at least 32.77 GB of packed-weight reads (three passes in the current implementation). Table 2 converts this into training budgets; the best tiny-scale extrapolations to the 20% gate (∼5,900–8,700 steps) would need 85–126 days of uninterrupted compute at the measured step rate, and an illustrative compute-optimal bound for 10^{11} active parameters ($2{\times}10^{12}$ tokens) is 1.2×10^{24} FLOPs, i.e. 387–1,934 years at 100–20 sustained TFLOPS. Measured synthetic data generation (7.57 pairs/s) would also need millennia for that token count. The objective is infeasible on this hardware by two to three orders of magnitude for even the modest gate, and by more for general pretraining; it is not a tuning problem.

Measured/derived quantity	Value
100B forward+backward per step (step 3)	1,247.47 s
Packed weight bytes per step (one pass)	32.77 GB
Wall time for 1,000 steps	14.4 days
Wall time for 5,000 steps	72.2 days
Weight traffic for 5,000 steps (three passes)	491.5 TB
Tiny-proxy linear extrapolation to 20% gate (6,000 pairs)	∼8,700 steps
100B wall time for 8,700 steps	∼126 days
Illustrative 2×10^{12}-token compute bound	387–1,934 years

Table 2: Measured wall-clock and data floor for training the 101.74B-core-active model, computed only from result files in this artifact tree and explicitly labelled assumptions (see `verification/takeover-2026-10-25/feasibility.json`).

13 Limitations and next research gate

The main limitations are privileged observations, one simplified synthetic environment, three seeds, a small random-initialized student, lack of strong public visual world-model baselines, no multi-teacher evaluation and no large-scale training. The sequential test optimizes local teacher-feature reconstruction; replacing global optimization with local objectives can introduce composition errors. Router calibration and end-to-end evaluation are required before making a capability claim about assembled experts.

No checkpoint in this work meets the requested 100B-active trained-model objective. A claim of general world-model superiority would require a licensed trained checkpoint, active-path audit, broad held-out action-conditioned evaluation and evidence that improvements survive equal-resource comparisons. A novelty review and human author review are also needed before considering an arXiv submission. The independent takeover verification in Section 12 additionally shows that three optimization steps moved only 5.79×10^{-5} of the packed codes, that a ten-fold learning-rate increase moves the grid as predicted but still cannot reduce a fixed batch's loss in two steps, and that the measured step cost puts even the modest action gate at months of uninterrupted compute on this device; the objective is therefore infeasible here, not merely unfinished.

Reproducibility and assistance

The accompanying source includes the preregistered pilot protocol, actual metrics, checkpoints, statistical scripts, shape audits and local-training verification. Citation metadata was retrieved by DOI content negotiation and checked against arXiv records; failed scholarly API calls are preserved in provenance. Paper-lookup procedures from Scientific Agent Skills assisted reference retrieval [3]. An AI coding agent assisted implementation, experimentation and drafting. The named human author's review remains pending. No external compute was used and no arXiv submission has been made. The independent takeover audit, its measured feasibility calculations, the completed 100B read-only attention diagnostic and the new matched learning runs are retained under `verification/takeover-2026-10-25/` and the new run directories; repository-wide negative and anomalous results (including a completed packed update that changed zero bytes, and a readout that places 79% of its attention on structurally zero tokens) are preserved rather than removed. Company sharing could not be verified: the `gh` login tokens are invalid and the GitHub API returned

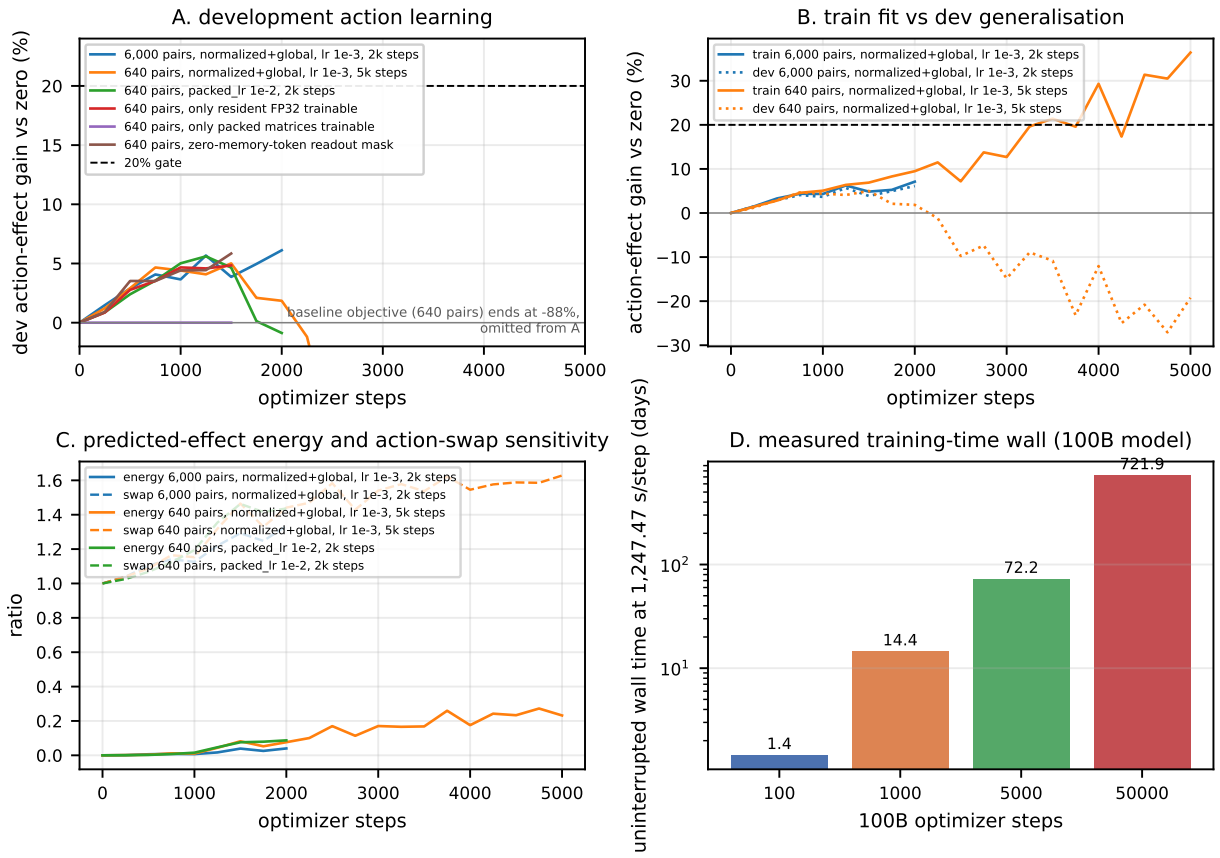


Figure 4: New matched takeover screens on the 325,003-parameter proxy. A: development action-effect gain against zero response (20% gate dashed); the baseline objective is omitted from A because it falls to −88%. B: training fit versus development for 640 and 6,000 pairs; the 6,000-pair run keeps a small gap while the 640-pair run overfits. C: predicted-effect energy ratio and action-swap ratio. D: measured 100B wall time for 100 to 50,000 optimizer steps at 1,247.47 s/step. One initialisation per arm; no inferential claims.

HTTP 403 (account suspended), so no shared task was created, claimed or updated.

References

[1] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[3] Timothy Kassis, Vinayak Agarwal, Yuhuan He, Darshil Patel, and Aubrey M. Brueckner. Scientific agent skills: A library of procedural knowledge for research agents, 2026.

[4] PAN Team, Zihan Liu, Yi Gu, Mingkai Deng, Guangyi Liu, Zeyu Feng, Qiyue Gao, Yiyan Hu, Benhao Huang, Yichi Yang, Kun Zhou, Jiannan Xiang, Zhiting Hu, Zhengzhong Liu, and

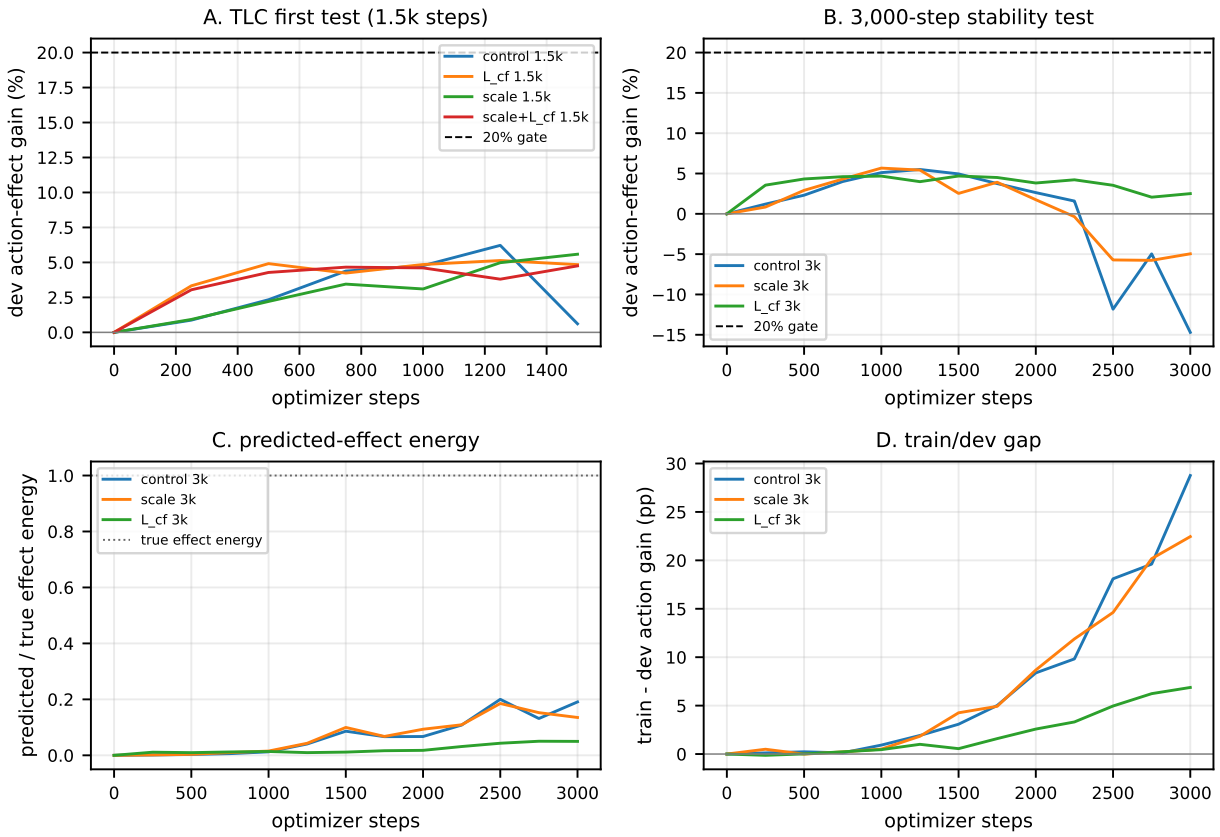


Figure 5: First TLC prototype on the 325,003-parameter proxy (single seed, 640 training pairs). A: four 1,500-step arms; the no-intervention control peaks then inverts while the counterfactual field-alignment (L_{cf}) arm stays positive. B: 3,000-step test; L_{cf} holds +2.51% development action-effect gain where the control reaches -14.70%. C: the control's predicted-effect energy grows to 0.19 of the true energy while L_{cf} stays near 0.05. D: train–development gap. These are negative-to-partial mechanism diagnostics, not capability results, and no arm passes the 20% gate.

Eric P. Xing. Pan: A world model for general, actionable, and long-horizon world simulation, 2025.