Verifying Digital-Twin Proxy Representations for Robust Sim2Real Locomotion Transfer

Chayanin Chamachot, Chulalongkorn University

***Abstract*— Sim2real locomotion pipelines increasingly embed learned internal dynamics representations—compact latent states that function as adaptive surrogates of the deployment environment—inside policies for online adaptation. For such digital-twin proxy representations to support monitoring, diagnosis, and uncertainty-aware deployment, their fidelity must be directly verifiable, not merely inferred from downstream reward.**

We present a systematic verification protocol applying six complementary analyses (probes, interventions, Mutual Information Gap (MIG), Disentanglement–Completeness–Informativeness (DCI), Separated Attribute Predictability (SAP), mutual information) to a factored auxiliary-supervised latent (DynaMITE) on a Unitree G1 humanoid in Isaac Lab. The internal surrogate fails every fidelity check: probes yield $R^2 \approx 0$, clamping interventions produce negligible behavioral change, and standard disentanglement metrics are near zero. An unsupervised long short-term memory (LSTM) hidden state scores higher on every readout. A 2×2 factorial (n=10) cleanly isolates the operative mechanism: a $\tanh$ information bottleneck—not the auxiliary supervision—drives observed robustness differences. A compound-mismatch transfer-readiness stress test (simultaneous friction, push, and delay perturbation, 10 seeds) reveals deployment-critical failure modes invisible to single-axis evaluation.

Our results establish that a common factor-supervision assumption for internal twin representations does not survive verification, that compression architectures may provide robustness benefits independent of semantic factorization, and that sim2real pipelines relying on learned adaptive surrogates need direct fidelity checks before transfer.

I. INTRODUCTION

Simulation-to-real (sim2real) locomotion pipelines increasingly embed compact learned dynamics representations inside policies—latent states trained to capture environment parameters such as friction, mass, and actuation characteristics from observation–action history [1], [2], [3]. These internal adaptive surrogates function as digital-twin proxies: compressed, online-updated models of the deployment environment that condition policy behavior for robust transfer. For such representations to support not only adaptation but also monitoring, failure diagnosis, and uncertainty-aware deployment, their fidelity must be directly verifiable. Yet in practice, fidelity is almost never checked—adaptation quality is assessed through downstream reward alone [4], [5].

This verification gap matters. If the internal surrogate is non-decodable—if standard readouts cannot recover the dynamics parameters it was trained to encode—then it cannot support online monitoring of deployment conditions, cannot enable interpretable failure diagnosis, and provides no direct evidence of transfer readiness beyond aggregate reward. Reward can improve for reasons unrelated to surrogate fidelity (e.g., implicit regularization from the bottleneck architecture), masking a non-functional internal model. For any sim2real system that relies on learned dynamics estimators as digital-twin proxies, this is a concrete risk: the internal surrogate is assumed to work as designed, but never directly verified.

This paper provides that direct verification. We instantiate the pattern as **DynaMITE** (Dynamics-Matching Inference via Transformer Encoding): a two-layer transformer encoder mapping an 8-step (160 ms) history to a factored 24-d latent with five factor subspaces (friction, mass, motor strength, contact stiffness, action delay), each trained with a dedicated auxiliary loss during proximal policy optimization (PPO) on a Unitree G1 humanoid in Isaac Lab. We apply six complementary verification analyses and find that the internal surrogate fails every fidelity check. A 2×2 factorial (n=10) isolates a $\tanh$ information bottleneck—not the auxiliary supervision—as the component driving observed robustness differences.

Negative results reported honestly are particularly valuable for the sim2real community: they warn that a common design pattern does not deliver what it promises [21]. We complement the verification analysis with a compound-mismatch transfer-readiness stress test—simultaneous friction, push, and action-delay perturbation—that reveals failure modes invisible to single-axis evaluation, directly relevant to the multi-dimensional dynamics mismatch characteristic of physical deployment. Our contributions:

1) **Verification protocol for internal twin fidelity.** Six complementary analyses (probes, interventions, MIG, DCI, SAP, mutual information (MI)) show that per-factor auxiliary supervision does not produce a decodable, functionally separable, or disentangled internal adaptive surrogate. An unsupervised long short-term memory (LSTM) hidden state achieves higher probe R^2 on every factor.
2) **Factorial isolation of the operative mechanism.** A 2×2 factorial (n=10) cleanly separates bottleneck compression from auxiliary supervision: compression drives observed differences; supervision has no measurable effect ($p > 0.66$).
3) **Compound-mismatch transfer-readiness evaluation.** A multi-axis stress test (simultaneous friction, push, and delay perturbation, 10 seeds) reveals deployment-critical failure modes invisible to single-axis testing. With n=10, 11 of 21 pairwise comparisons survive

1Chayanin Chamachot, Chulalongkorn University

Holm–Bonferroni correction.

4) **Implications for digital-twin-based sim2real pipelines.** Non-decodable internal surrogates cannot support monitoring or diagnosis; reward alone does not verify twin fidelity; compression may improve robustness without semantic factorization.

II. RELATED WORK

A. Domain Randomization and Adaptive Policies

Domain randomization (DR) trains policies under a distribution of simulator parameters to generalize beyond the training range [6], [7], and is standard practice for legged locomotion on quadrupeds [8], [9] and humanoids [10], [11]. However, DR provides no mechanism for online adaptation, and performance degrades outside the training range [12], [13]. Methods such as Rapid Motor Adaptation (RMA) address this by training a latent dynamics estimator conditioned on observation–action history [1], [2]. LSTM-based encoders accumulate evidence over time [14], [15]; transformer-based encoders process context in parallel [16], [10]. All evaluate adaptation quality through downstream reward only, without verifying what the latent encodes.

DynaMITE differs from RMA-family methods in two respects: it uses single-phase end-to-end training rather than two-phase distillation, and it imposes a factored latent with per-factor losses rather than a monolithic adaptation vector. Our analyses show that despite these design choices, the resulting representation is no more interpretable than an unsupervised LSTM hidden state—and that the component driving behavioral differences is the information bottleneck, not the auxiliary supervision.

B. Representation Verification in RL

Representation probing in deep reinforcement learning (RL) typically finds partial but unstructured encoding of task-relevant variables [4], [5]. Standard disentanglement metrics—MIG [17], DCI [18], SAP [19]—are rarely applied to RL representations, and the link between disentanglement and policy performance is poorly understood [20]. To our knowledge, direct probing and intervention on factored auxiliary-loss representations in locomotion has not been reported. Our analysis addresses this gap.

C. Digital Twins and Sim2Real Verification

Digital twins for robotics range from full-scope environment reconstruction and physics-aware simulation calibration to learned internal dynamics models for online adaptation [24]. Methods such as RMA [1], [2] train compact latent surrogates updated from sensory history, serving as internal twin proxies that enable policies to adapt to changing dynamics without explicit system identification. However, verification of whether these internal surrogates faithfully encode their target parameters has received little attention; runtime verification frameworks for digital twins exist [23], but evaluation of learned internal surrogates typically relies on downstream task reward alone [4], [5]. Our work addresses this complementary gap: directly verifying the fidelity of compact internal twin-like representations using probes, interventions, and disentanglement metrics.

D. Connection to Adaptive Digital Twins

The latent dynamics representation in DynaMITE—and in RMA-family methods generally—is designed to function as an internal adaptive surrogate: a compact, continuously updated model of the environment's dynamics parameters maintained inside the policy for online adaptation. This mirrors a core function of digital twins: maintaining a synchronized internal representation of the physical system to enable monitoring, prediction, and control.

We distinguish this from full-scope digital twins, which typically involve geometry-level scene reconstruction, multi-physics simulation, or generative modeling of the physical environment. The internal surrogate studied here is narrower: a low-dimensional latent vector intended to capture dynamics-relevant parameters from sensory history, without explicit geometric or visual modeling. Nonetheless, verifying the fidelity of such internal surrogates is directly relevant to digital-twin-enabled sim2real transfer: if the internal twin proxy cannot be read out, intervened on, or validated against ground truth, its utility for monitoring, diagnosis, and transfer assurance is limited.

III. METHOD

A. Architecture

DynaMITE (Fig. 1) uses a two-layer transformer encoder (4 heads, $d_{\text{model}}{=}128$) to process an 8-step observation–action history (160 ms at 50 Hz). At each control step t, the agent maintains a buffer of the $H{=}8$ most recent observation–action pairs $\{(o_{t-H+1}, a_{t-H+1}), \ldots, (o_t, a_t)\}$. Each pair is embedded via learned linear projections and summed with sinusoidal positional encodings. The resulting token sequence is processed by the encoder, and the output is mean-pooled to yield $\boldsymbol{h} \in \mathbb{R}^{128}$.

A linear projection followed by a $\tanh$ nonlinearity maps $\boldsymbol{h}$ to $\boldsymbol{z} \in \mathbb{R}^{24}$, partitioned into five contiguous factor subspaces: friction (dims 0–3), mass (4–9), motor strength (10–15), contact stiffness (16–19), and action delay (20–23). The partition is fixed before training; subspace sizes are proportional to expected factor complexity (e.g., mass covers more joints than scalar friction coefficient). The $\tanh$ bounds each dimension to $[-1, 1]$, creating an information bottleneck that compresses the 128-d encoder output to 24 bounded dimensions.

Each subspace $\boldsymbol{z}_f$ is trained with a dedicated auxiliary loss:

$$\mathcal{L}_{\text{aux},f} = \|g_f(\boldsymbol{z}_f) - \theta_f\|^2, \tag{1}$$

where g_f is a small multilayer perceptron (MLP) head and θ_f is the ground-truth dynamics parameter. The total training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{PPO}} + c_v \mathcal{L}_{\text{value}} + 0.1 \sum_f \mathcal{L}_{\text{aux},f}. \tag{2}$$

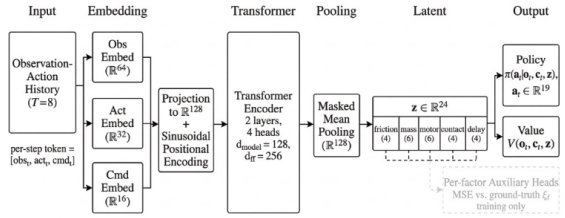

Fig. 1. DynaMITE architecture. A transformer encoder maps an 8-step history to a factored 24-d internal adaptive surrogate with per-factor auxiliary losses active during training only. The latent functions as a digital-twin proxy, intended to encode deployment-relevant dynamics parameters for online adaptation.

TABLE I
MODEL ARCHITECTURE SUMMARY.

Model	History	Latent	Aux	Params
MLP	None	No	No	266–362k
LSTM	Hidden	No	No	176–215k
Transformer	8 steps	No	No	330–342k
DynaMITE	8 steps	24-d	Yes	342–392k

Auxiliary losses are active only during training. The latent $\boldsymbol{z}$ is concatenated with the current observation o_t and fed to the policy $\pi(a \mid o_t, \boldsymbol{z})$ and value $V(o_t, \boldsymbol{z})$ heads.

B. Baselines

All four architectures share the same observation embedding, policy MLP, and value MLP (Table I), differing only in history encoding. The **MLP** receives only the current observation with no history. The **LSTM** processes observations sequentially and conditions the policy on its hidden state (128-d), with no auxiliary signal—any dynamics information it captures arises purely from the RL objective. The **Transformer** baseline uses the same encoder as DynaMITE but passes the 128-d mean-pooled context directly to the policy head, without the factored latent or auxiliary losses. This controls for encoder architecture but confounds the bottleneck with the auxiliary supervision; the 2×2 factorial (Sec. V-D) resolves this by independently toggling both components.

IV. VERIFICATION AND TRANSFER-READINESS PROTOCOL

A. Training and Evaluation

All models are trained with PPO (clipped objective, generalized advantage estimation (GAE)) [22] using 512 parallel Isaac Lab environments on a Unitree G1 humanoid across four tasks: *flat* (no perturbation), *push* (random pushes during training), *randomized* (full domain randomization over all five factors), and *terrain* (uneven ground with DR). Training spans 10M timesteps at 50 Hz (∼14 min on RTX 4060). Checkpoints are selected by training-time stochastic reward. Main comparisons use 5 seeds (42–46); factorial ablations and out-of-distribution (OOD) sweeps use 10 seeds (42–51).

All results use **deterministic evaluation** (mean-action rollouts, no sampling): 100 episodes for main comparison, 50 for OOD sweeps and push recovery. Environment resets fully randomize initial conditions. The evaluation seed is fixed at 42 for all models within a task, ensuring identical initial states.

B. Reward and Metrics

Reward is penalty-based (always negative); higher (less negative) is better. A difference of 0.30 reward units corresponds to ∼6% less penalty per step. **OOD sensitivity** $= \max(\bar{r}) - \min(\bar{r})$ across sweep levels; lower indicates a flatter reward curve but does not distinguish genuine robustness from uniformly poor performance (caveat: a model that performs badly everywhere also has low sensitivity). **Degradation** $= |\text{Severe} - \text{ID}|/|\text{ID}|$, where Severe Mean averages the two highest-severity levels.

C. Verification Analyses

Probes. Ridge regression (linear) and 1-hidden-layer MLP (64 units) probes predict ground-truth dynamics from frozen representations (∼36k samples, 5-fold cross-validation (CV)). Non-decodability is conditional on readout expressiveness.

Interventions. Each factor subspace is clamped to its training mean: 3 seeds × 5 factors × 3 DR levels = 90 conditions, 50 episodes each.

Disentanglement metrics. MIG [17], DCI [18], SAP [19] computed on frozen representations (5 seeds × ∼36k samples per model).

2×2 **Factorial.** Four variants independently toggle the $\tanh$ bottleneck and auxiliary losses, sharing identical transformer architecture. 10 seeds per cell.

Combined-shift stress test. Simultaneous friction (0.1–0.3), push (3–8 m/s), and action delay (3–5 steps) across five severity levels.

D. Statistical Reporting

We report mean ± standard deviation (SD), paired t-tests for matched-seed comparisons, and Holm–Bonferroni correction for OOD pairwise tests. With n=10 seeds, 11 of 21 pairwise OOD comparisons survive correction ($p_{\text{adj}} < 0.05$). All DynaMITE-vs-MLP comparisons are significant; DynaMITE-vs-LSTM reaches significance on friction, action delay, and push-task push magnitude. The ablation significance at $p = 0.048$ is marginal.

V. RESULTS

A. In-Distribution Comparison

Table II presents mean reward across five seeds and four tasks. LSTM achieves the best reward on all tasks with the lowest variance. All LSTM vs. DynaMITE paired t-tests are significant: flat t=8.49, p=0.001; push t=4.69, p=0.009; randomized t=3.34, p=0.029; terrain t=8.55, p=0.001. The reward gap between LSTM and DynaMITE (+0.30 to +0.87) is consistent across tasks despite varying task difficulty. DynaMITE ranks second on three of four tasks; the Transformer baseline exhibits the highest variance, suggesting sensitivity to random seed initialization.

TABLE II

IN-DISTRIBUTION REWARD (5 SEEDS, 100-EPISODE DETERMINISTIC EVAL). BOLD = BEST PER TASK. ALL LSTM VS. DYNAMITE PAIRED t-TESTS SIGNIFICANT ($p < 0.03$).

Method	Flat	Push	Rand.	Terrain
MLP	−4.83	−5.01	−5.32	−4.82
LSTM	**−4.01**	**−4.30**	**−4.18**	**−4.06**
Transformer	−5.02	−4.83	−4.77	−4.46
DynaMITE	−4.88	−4.60	−4.48	−4.49

TABLE III

LATENT PROBE R^2 (5 SEEDS, 5-FOLD CROSS-VALIDATION, ∼36K SAMPLES). BOLD = BEST.

Factor	DyM Lin	DyM MLP	LSTM Lin	LSTM MLP
Friction	−0.000	−0.001	0.026	**0.101**
Mass	0.000	−0.002	0.012	**0.018**
Motor	0.000	−0.002	0.010	**0.015**
Contact	0.000	−0.000	0.006	**0.045**
Delay	0.000	−0.000	0.011	**0.041**
Overall	**0.000**	**−0.001**	**0.013**	**0.044**

TABLE IV

INTERVENTION ANALYSIS: AVERAGE ABSOLUTE REWARD CHANGE WHEN CLAMPING EACH FACTOR SUBSPACE (3 SEEDS × 3 DR LEVELS, 50 EPISODES EACH).

Factor (dims)	Avg $\lvert\Delta r\rvert$	Interpretation
Friction (0–3)	0.007	Negligible
Mass (4–9)	0.012	Negligible
Motor (10–15)	0.021	Negligible
Contact (16–19)	0.020	Negligible
Delay (20–23)	0.020	Negligible

TABLE V

DISENTANGLEMENT METRICS (5 SEEDS). BOLD = BETTER. LOWER DCI-I (PREDICTION ERROR) IS BETTER.

Metric	DynaMITE	LSTM
MIG	0.001	0.001
DCI Disent.	0.020	**0.093**
DCI Compl.	0.017	**0.054**
DCI Inform.	**0.085**	0.140
SAP	0.000	0.001

B. Internal Surrogate Fidelity: No Decodable Factor Structure

Table III presents the fidelity assessment: probe R^2 for DynaMITE's internal adaptive surrogate vs. LSTM's hidden state. Despite per-factor auxiliary losses, DynaMITE shows $R^2 \approx 0$ for both linear and nonlinear probes across all five factors. LSTM's unsupervised hidden state achieves R^2 up to 0.101 (friction, MLP probe)—higher on every factor.

Causal fidelity assessment confirms the null: we clamp each factor subspace to its training mean across 3 seeds × 5 factors × 3 DR levels (90 conditions, 50 episodes each). Table IV reports per-factor results. All reward changes are negligible ($|\Delta r| < 0.05$). No factor subspace exerts selective behavioral control—the latent could be replaced by a constant vector with no measurable behavioral change.

Standard disentanglement metrics (Table V) reinforce this finding. MIG and SAP are near zero for both models (< 0.001), confirming that no individual latent dimension aligns with a single dynamics factor. On DCI disentanglement and completeness, LSTM scores ∼5× higher than DynaMITE (0.093 vs. 0.020; 0.054 vs. 0.017), consistent with its higher probe R^2. DynaMITE scores better only on DCI informativeness (lower prediction error: 0.085 vs. 0.140), reflecting its compressed representation rather than meaningful factor separation.

k-nearest-neighbor (KNN) estimated mutual information provides a complementary angle: DynaMITE's latent retains ∼8× more MI with dynamics parameters than LSTM's hidden state (0.23 vs. 0.03 nats), but both absolute values are low (< 0.25 nats). The coexistence of modestly higher MI with near-zero probe R^2 is consistent with dynamics information distributed across dimensions in a form that resists factor-wise decoding—precisely the opposite of the intended design.

The supervised latent is neither decodable, functionally separable, nor disentangled under our analyses.

Representation geometry corroborates the compression narrative. Table VI shows that DynaMITE's 24-d latent collapses to effective rank ∼5—only 20% of its nominal dimensionality—while LSTM's 128-d hidden state uses ∼32 effective dimensions (25%). The high condition number (316 vs. 108) indicates a highly anisotropic structure: a few dimensions dominate while most are nearly unused. Despite this compression, KNN-estimated mutual information shows DynaMITE retains ∼8× more information about dynamics parameters than LSTM (0.23 vs. 0.03 nats), but in a form not recoverable by standard probes—consistent with distributed, non-decodable encoding.

A training-dynamics observation reinforces the null: the total auxiliary mean squared error (MSE; summed across five factors) drops from 7.2 ± 0.7 at training start to 3.7 ± 0.1 at convergence (48% reduction, 5 seeds), but the residual per-factor MSE remains high (∼0.75). The auxiliary heads $g_f(\boldsymbol{z}_f)$ do not learn accurate predictions of their target parameters—consistent with the encoder not placing sufficient factor-specific information into the corresponding subspace. Per-factor gradient norms confirm that most factors' auxiliary gradients collapse near zero by end of training.

C. Component Ablation

Table VII reports ablation results with n=10 seeds on the randomized task. The Single Latent variant (unfactored 24-d) reaches marginal significance ($p = 0.048$), providing modest evidence that the factored partition contributes. No Aux Loss remains directional but not significant ($p = 0.641$). The Aux Only variant (auxiliary losses active but policy sees full 128-d features, no bottleneck) performs identically to No Latent (−4.64 vs. −4.64), providing initial evidence

TABLE VI

REPRESENTATION GEOMETRY AND INFORMATION CONTENT (5 SEEDS). BOLD = NOTABLE.

Metric	DynaMITE (24-d)	LSTM (128-d)
Effective rank	$\mathbf{4.78 \pm 0.72}$	32.20 ± 3.85
Participation ratio	$\mathbf{2.27 \pm 0.15}$	4.96 ± 1.52
Condition number	315.9 ± 204.1	$\mathbf{108.4 \pm 19.9}$
MI with dynamics (nats)	$\mathbf{0.233 \pm 0.052}$	0.028 ± 0.033

TABLE VII

ABLATION STUDY (RANDOMIZED TASK, n=10 SEEDS, PAIRED t-TEST VS. FULL DYNAMITE). BOLD $p < 0.05$.

Variant	Reward	Δ	p	Worse
Full DynaMITE	$\mathbf{-4.46 \pm 0.25}$	—	—	—
No Aux Loss	-4.51 ± 0.20	-0.05	0.641	6/10
No Latent	-4.64 ± 0.36	-0.18	0.225	9/10
Single Latent	-4.69 ± 0.18	-0.23	**0.048**	8/10
Aux Only	-4.64 ± 0.34	-0.18	0.251	7/10

that auxiliary gradient regularization without the bottleneck confers no benefit.

D. 2×2 Factorial: Compression vs. Supervision

The Transformer-vs.-DynaMITE comparison conflates two design choices introduced simultaneously: the 24-d $\tanh$ bottleneck (information compression) and the per-factor auxiliary losses (explicit supervision). To cleanly decompose these contributions, we construct a 2×2 factorial with four cells sharing identical transformer encoder architecture, differing only in whether the policy receives the 24-d compressed vector or the full 128-d features (**Bottleneck** vs. **No Bottleneck**), and whether auxiliary losses are active (**Aux Loss** vs. **No Aux Loss**). Each cell trains 10 seeds (42–51) on the randomized task.

The critical comparison is the **Aux Only** cell (No Bottleneck + Aux Loss): auxiliary losses are active, providing gradient regularization to the shared transformer, but the policy sees only the full 128-d features without the $\tanh$ compression. If auxiliary supervision contributes beyond its role as a regularizer, this cell should outperform No Latent (No Bottleneck + No Aux Loss). It does not: both achieve -4.64.

Table VIII presents in-distribution (ID) results; Table IX presents severe OOD results (combined-shift Levels 3–4).

Auxiliary losses show **no measurable effect** on ID reward ($+0.03$, $p = 0.732$) or OOD reward ($+0.03$, $p = 0.669$). The bottleneck shows a **consistent advantage**: ID $+0.16$ ($p = 0.207$), OOD $+0.10$ ($p = 0.208$). Interaction is negligible ($p > 0.78$). The Aux-Only cell matches No Latent exactly (-4.64), confirming that auxiliary gradient regularization without compression confers no benefit.

Table X completes the picture. Bottleneck models degrade *slightly more* than no-bottleneck variants (2.1% vs. 0.5%), ruling out a robustness-amplification mechanism: if the bottleneck specifically protected against distribution shift, we would expect *less* degradation in bottleneck cells, not more. Instead, the bottleneck provides a training-time representation benefit (better ID reward) that carries through proportionally to OOD. The auxiliary losses contribute nothing in either regime.

TABLE VIII

2×2 FACTORIAL: ID REWARD (10 SEEDS, RANDOMIZED TASK).

	No Aux Loss	Aux Loss
Bottleneck	-4.51 ± 0.20	$\mathbf{-4.46 \pm 0.25}$
No Bottleneck	-4.64 ± 0.36	-4.64 ± 0.34

TABLE IX

2×2 FACTORIAL: SEVERE COMBINED-SHIFT OOD REWARD (AVG. LEVELS 3–4, 10 SEEDS, 50 EPISODES PER LEVEL).

	No Aux Loss	Aux Loss
Bottleneck	-4.59 ± 0.21	$\mathbf{-4.55 \pm 0.22}$
No Bottleneck	-4.68 ± 0.28	-4.66 ± 0.23

This decomposition resolves the confound in the DynaMITE-vs.-LSTM comparison (Sec. V-E): DynaMITE degrades less than LSTM under combined shift (1.4% vs. 16.2%), but this reflects an architectural difference between the two model families—not a specific contribution of auxiliary dynamics supervision. The factorial confirms: within the same architecture, toggling auxiliary losses on or off has no measurable effect.

E. Compound-Mismatch Transfer-Readiness Test

To assess transfer readiness under compound dynamics mismatch characteristic of sim2real deployment, we simultaneously shift friction, push magnitude, and action delay across five severity levels. This protocol is designed to approximate the multi-dimensional nature of real dynamics mismatch, where friction, inertia, actuation, and latency all differ simultaneously from their simulated values. Table XI and Fig. 2 present results.

Among the two highest-performing ID models, LSTM degrades 16.2% from ID to severe OOD; DynaMITE degrades 1.4%. At level 4, DynaMITE's reward (-4.59) exceeds LSTM's (-5.10). Transformer and MLP also show low sensitivity—in MLP's case because it performs poorly everywhere, illustrating why sensitivity alone is insufficient as a robustness metric.

Note that Level 0 rewards (e.g., LSTM -3.58) differ from Table II ID rewards (LSTM -4.18): Level 0 pins all dynamics to nominal values, while ID evaluates under the full DR distribution. LSTM benefits disproportionately from nominal conditions ($\Delta = 0.60$) compared with DynaMITE ($\Delta = 0.12$), itself consistent with higher sensitivity to dynamics variation. The combined-shift result reveals a failure mode invisible to ID evaluation alone: compound perturbation exposes sensitivity that no single-axis test uncovers.

TABLE X

FACTORIAL DEGRADATION: ID VS. SEVERE OOD (10 SEEDS). BOTTLENECK ADVANTAGE PERSISTS BUT DOES NOT AMPLIFY.

Variant	ID	Severe	Δ	Degrad.
Full (B+A)	−4.46	−4.55	−0.09	2.1%
B only	−4.51	−4.59	−0.08	1.9%
Neither	−4.64	−4.68	−0.04	0.9%
A only	−4.64	−4.66	−0.02	0.5%

TABLE XI

COMBINED-SHIFT REWARD (RANDOMIZED TASK, 10 SEEDS, 50 EPISODES/LEVEL). BOLD = BEST AT EACH LEVEL.

Method	Lv 0	Lv 1	Lv 2	Lv 3	Lv 4
DynaMITE	−4.36	−4.43	−4.45	**−4.50**	**−4.59**
LSTM	**−3.58**	**−4.22**	**−4.40**	−4.62	−5.10
Transf.	−4.55	−4.58	−4.57	−4.61	−4.74
MLP	−5.43	−5.43	−5.39	−5.42	−5.48

F. Transfer-Readiness: Single-Axis Sweeps and Recovery

1) Single-Axis Robustness Sweeps: Table XII summarizes severe degradation across sweep types and tasks. LSTM degrades 13–21% at severe levels across push-magnitude sweeps while DynaMITE degrades 3–5% (sensitivity 1.48–1.58 vs. 0.40–0.44). Friction is a weak perturbation axis ($< 1.5\%$ degradation); action delay is not meaningful at 50 Hz (< 0.10 sensitivity). With n=10 seeds, 11 of 21 pairwise OOD comparisons survive Holm–Bonferroni correction ($p_{\text{adj}} < 0.05$).

2) Push Recovery: In a controlled push-recovery protocol (flat terrain, 7 magnitudes, 50 episodes × 5 seeds), DynaMITE returns below the tracking-error threshold (1.5) in ∼6 steps regardless of push magnitude (Table XIII). LSTM requires 9–20 steps, plateauing at large pushes. At pushes ≥3 m/s, DynaMITE crosses the threshold 3.4× faster.

However, the result is narrower than "better recovery": LSTM achieves lower peak tracking error at all magnitudes (e.g., 3.05 vs. 3.96 at 1 m/s) and substantially better post-push gait quality (−3.2 to −4.8 reward vs. −9.5 to −9.8). DynaMITE crosses the threshold fastest but with worse gait; LSTM achieves the best gait but crosses slowest. Aggregating across all OOD evaluations, no model dominates both ID reward and severe OOD robustness.

VI. DISCUSSION

A. The Supervision Signal Does Not Produce What It Promises

The central finding is unambiguous: under six complementary analyses—linear probes, nonlinear probes, clamping interventions, MIG, DCI, SAP, and KNN-estimated mutual information—the factored auxiliary supervision strategy does not produce a decodable, functionally separable, or disentangled latent representation. This is not a marginal failure: probe $R^2 \approx 0$ on all five factors, intervention effects below 0.05, disentanglement scores near zero. An unsupervised LSTM hidden state scores higher on every readout.

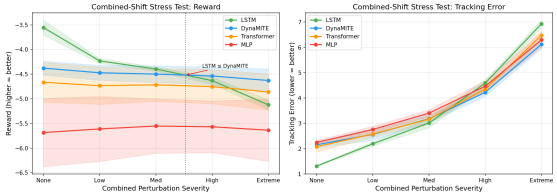

Fig. 2. Combined-shift stress test (randomized task, 10 seeds). LSTM leads at low severity but degrades steeply (16.2%); DynaMITE is more stable (1.4% degradation). Crossover at level 3.

TABLE XII

SEVERE OOD DEGRADATION ACROSS SWEEPS (10 SEEDS). DEGRADATION = |SEVERE − ID|/|ID|.

Sweep	Task	DyM	LSTM	Cross.
Combined	rand.	**1.4%**	16.2%	Lv 3
Push mag.	rand.	**4.5%**	16.9%	Lv 4
Push mag.	push	**2.8%**	12.7%	Lv 5
Push mag.	terr.	**5.1%**	20.7%	Lv 4
Friction	rand.	−1.0%	1.3%	—

For digital-twin and sim2real pipelines that rely on learned internal dynamics surrogates for monitoring, adaptation, or interpretability, this is a concrete warning: the representation may not encode what the loss function was designed to produce. Verification before transfer—not just downstream reward—should be standard practice.

B. Compression, Not Supervision, Is the Operative Mechanism

The 2×2 factorial cleanly separates the two components. Auxiliary losses show no measurable effect on either ID ($p = 0.732$) or OOD ($p = 0.669$) reward. The bottleneck shows a consistent advantage in both regimes (ID: +0.16; OOD: +0.10), and bottleneck models degrade *slightly more* (2.1% vs. 0.5%), confirming this is a training-time representation benefit, not a robustness mechanism.

Mechanistic observations are consistent with this account. Auxiliary and PPO gradients remain orthogonal throughout training ($|\cos| < 0.01$, 3 seeds), contributing 20–40% of gradient norm in orthogonal directions—regularization, not optimization synergy. The 24-d latent collapses to effective rank ∼5 (vs. LSTM's ∼32 of 128), constraining the representational capacity available for overfitting to the training distribution. Despite this compression, DynaMITE retains ∼8× more mutual information with dynamics than LSTM (0.23 vs. 0.03 nats), but in a form not recoverable by standard readouts—distributed, non-decodable encoding.

The practical recommendation is direct: invest in compression architectures (bottlenecks, information constraints) rather than relying on auxiliary supervision to produce structured representations. If the engineering goal is robustness under distribution shift, the bottleneck does more than the supervision.

TABLE XIII

STEPS TO RECOVER TRACKING AFTER PUSH (FLAT TASK, 5 SEEDS × 50 EPISODES). BOLD = FASTEST.

Push (m/s)	DyM	LSTM	Transf.	MLP
1.0	**5.6**	9.2	6.7	6.3
3.0	**6.0**	20.6	8.3	7.8
5.0	**6.1**	19.8	8.0	8.2
8.0	**6.2**	19.6	8.3	8.3

TABLE XIV

OBSERVED OPERATING REGIMES UNDER OUR VERIFICATION PROTOCOL. THESE ARE SIMULATION-SIDE OBSERVATIONS, NOT DEPLOYMENT RECOMMENDATIONS.

Scenario	Favored	Evidence
Maximize ID reward	LSTM	Confirmed ($p < 0.03$)
Moderate OOD	LSTM	Directional (n=10)
Severe multi-axis OOD	DynaMITE	11/21 significant (n=10)
Fast re-adaptation	DynaMITE	Directional (n=5)
Inspect latent dynamics	Neither	Confirmed ($R^2 \approx 0$)

C. Compound-Mismatch Evaluation Reveals Hidden Failure Modes

The compound-mismatch stress test reveals a failure mode invisible to ID or single-axis evaluation: LSTM loses its reward advantage under compound perturbation (16.2% degradation) but not under any individual axis in isolation. This pattern—compound sensitivity that is invisible to per-axis testing—is directly relevant to sim2real transfer, where dynamics mismatch is typically multi-dimensional. Single-axis robustness sweeps overestimate transfer readiness.

For digital-twin-enabled sim2real pipelines: compound perturbation stress tests should be a standard simulation-side checkpoint before deploying adaptive policies on hardware.

D. Practical Recommendations

For nominal reward: LSTM remains the strongest choice in our comparison ($p < 0.03$, all tasks). **For severe compound mismatch:** the bottleneck architecture degrades less, but the auxiliary supervision does not contribute; a bottleneck without auxiliary losses achieves the same benefit. **For interpretability:** auxiliary dynamics supervision does not produce representations usable for online monitoring or debugging in this setting. **For pipeline design:** verify representations before relying on them for transfer or monitoring—downstream reward alone is insufficient evidence that a learned estimator encodes what it was designed to encode.

Table XIV summarizes the observed operating regimes.

E. Scope and Non-Claims

This paper does not show that decodable or disentangled latent dynamics representations are impossible in locomotion—only that one specific supervision strategy did not produce them in this setup under the analyses we applied. It does not establish sim2real transfer. It does not claim that auxiliary losses are without value in all settings; they may act as useful regularizers even when they fail to produce interpretable latent structure. The negative results are conditional on our readout family, perturbation range, and platform.

F. Implications for Digital-Twin-Based Sim2Real Systems

Our results carry specific implications for sim2real pipelines that embed learned dynamics surrogates as internal digital-twin proxies.

Reward alone does not verify surrogate fidelity. DynaMITE achieves competitive reward while its internal surrogate encodes nothing decodable about the dynamics it was trained to represent. Any pipeline that uses reward as evidence that the internal twin model is functional risks deploying a non-verified surrogate.

Internal adaptive surrogates should be directly checked. The six-analysis verification protocol applied here—probes, interventions, and disentanglement metrics—provides a template for validating internal twin representations before transfer. This is analogous to validating a digital twin's calibration before using it for prediction or control.

Compression may improve robustness without semantic factorization. The 2×2 factorial shows that a $\tanh$ bottleneck drives observed robustness differences, not the factored supervision. The robustness benefit of internal-surrogate architectures may stem from implicit information constraints rather than explicit dynamics encoding.

Non-decodable surrogates limit monitoring and diagnosis. If the internal twin proxy cannot be read out by standard methods, it cannot support online monitoring of deployment conditions or interpretable failure diagnosis. A decodable internal surrogate could, in principle, be compared against expected ranges or integrated into uncertainty-aware decision-making; when the surrogate is non-decodable, this monitoring pathway is closed. This opacity matters most under large dynamics shifts where silent degradation is most likely. We do not propose formal uncertainty quantification here, but identify latent fidelity as a prerequisite for any monitoring-capable twin-based deployment system.

Compound-mismatch testing is essential for transfer readiness. Single-axis perturbation sweeps overestimate transfer readiness by missing failure modes that emerge only under simultaneous multi-dimensional mismatch—the typical condition during physical deployment. Twin-based sim2real pipelines should include compound stress tests as a standard evaluation checkpoint.

VII. LIMITATIONS AND FUTURE WORK

All experiments use Isaac Lab simulation. We argue simulation-side evaluation is a necessary antecedent to transfer, but our combined-shift protocol is a proxy for sim2real mismatch, not calibrated to physical hardware. Our probes are Ridge regression and small MLPs; non-decodability is conditional on readout expressiveness, and more expressive decoders (nonlinear independent component analysis (ICA),

manifold-aware probes) may recover structure invisible to our readout family.

The bottleneck main effect ($p \approx 0.2$) is consistent but does not reach conventional significance at n=10; expanding to n=20 with bootstrap confidence intervals would resolve this. With n=10 seeds, 11 of 21 OOD comparisons survive Holm–Bonferroni correction; DynaMITE-vs-LSTM on combined shift and randomized push magnitude remain directional. The factorial has not yet been applied to push recovery. The narrow reward spread between top models ($\sim$0.6 units) leaves practical significance for hardware deployment unresolved.

Hardware deployment on a Unitree G1 to test whether the OOD tradeoff transfers to physical dynamics mismatch is the natural next step.

a) Scope of digital-twin claims.: This paper does not build, construct, or generate a digital twin. It studies a compact internal adaptive dynamics surrogate—a 24-d latent trained with per-factor auxiliary losses—used for policy adaptation in simulated locomotion. We frame this latent as a digital-twin proxy to connect with the adaptive-twin function it is designed to serve, but the scope is narrower than full scene-level or generative digital twins. There is no real-robot validation. Claims should be interpreted as evidence about verification of internal adaptive twin representations in this specific setting, not as general statements about all digital twin methods or auxiliary loss designs.

VIII. CONCLUSIONS

We evaluated whether factor-wise auxiliary dynamics supervision produces an internal adaptive surrogate with verifiable fidelity for sim2real locomotion. Under six complementary analyses, it does not: probes yield $R^2 \approx 0$, clamping produces negligible behavioral change, and standard disentanglement metrics are near zero. An unsupervised LSTM hidden state achieves higher readout accuracy on every factor.

A 2×2 factorial (n=10) identifies the $\tanh$ information bottleneck—not the auxiliary losses—as the component driving observed differences: auxiliary losses show no measurable effect on either ID ($p = 0.732$) or OOD ($p = 0.669$) reward, while the bottleneck yields a consistent advantage that persists but does not amplify under severe perturbation.

For digital-twin-based sim2real systems, our findings suggest three operational principles: (1) internal adaptive surrogates should be verified directly—reward alone is insufficient evidence of twin fidelity; (2) compound-mismatch stress testing is essential for transfer-readiness assessment, as single-axis evaluations miss deployment-critical failure modes; and (3) compression architectures may provide robustness benefits independent of semantic factorization, informing how internal twin representations should be designed and validated before transfer.

ACKNOWLEDGMENT

During the preparation of this manuscript, the author used GitHub Copilot (powered by Claude, Anthropic) for drafting and editing. The author has reviewed and edited all output and takes full responsibility for the content.

REFERENCES

[1] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Proc. RSS*, 2021.

[2] A. Kumar *et al.*, "Adapting rapid motor adaptation for bipedal robots," in *Proc. IROS*, 2022, pp. 1161–1168.

[3] Y. Chebotar *et al.*, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. ICRA*, 2019, pp. 8973–8979.

[4] A. Anand *et al.*, "Unsupervised state representation learning in Atari," in *Proc. NeurIPS*, 2019, pp. 8766–8779.

[5] R. Agarwal, M. C. Machado, P. S. Castro, and M. G. Bellemare, "Contrastive behavioral similarity embeddings for generalization in RL," in *Proc. ICLR*, 2021.

[6] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IROS*, 2017, pp. 23–30.

[7] OpenAI *et al.*, "Solving Rubik's cube with a robot hand," *arXiv:1910.07113*, 2019.

[8] J. Hwangbo *et al.*, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, eaau5872, 2019.

[9] G. B. Margolis *et al.*, "Rapid locomotion via reinforcement learning," in *Proc. RSS*, 2022.

[10] I. Radosavovic *et al.*, "Learning humanoid locomotion with transformers," in *Proc. ICLR*, 2024.

[11] T. Haarnoja *et al.*, "Learning agile soccer skills for a bipedal robot with deep RL," *Sci. Robot.*, vol. 9, eadi8022, 2024.

[12] Q. Vuong *et al.*, "How to pick the domain randomization parameters for sim-to-real transfer of RL policies?" *arXiv:1903.11774*, 2019.

[13] F. Muratore *et al.*, "Robot learning from randomized simulations: A review," *Front. Robot. AI*, vol. 9, 799893, 2022.

[14] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. ICRA*, 2018, pp. 3803–3810.

[15] W. Yu, J. Tan, C. K. Liu, and G. Turk, "Preparing for the unknown: Learning a universal policy with online system identification," in *Proc. RSS*, 2017.

[16] E. Parisotto *et al.*, "Stabilizing transformers for reinforcement learning," in *Proc. ICML*, 2020, pp. 7487–7498.

[17] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in VAEs," in *Proc. NeurIPS*, 2018, pp. 2610–2620.

[18] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *Proc. ICLR*, 2018.

[19] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *Proc. ICLR*, 2018.

[20] I. Higgins *et al.*, "DARLA: Improving zero-shot transfer in RL," in *Proc. ICML*, 2017, pp. 1480–1490.

[21] P. Henderson *et al.*, "Deep reinforcement learning that matters," in *Proc. AAAI*, 2018, pp. 3207–3214.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[23] J. S. Betzer, J. Boudjadar, M. Frasheri, and P. Talasila, "Digital twin enabled runtime verification for autonomous mobile robots under uncertainty," in *Proc. DS-RT*, 2024.

[24] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen, "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence," *arXiv:2305.01263*, 2023.