PADÉKOOP: RATIONAL OBSERVABLE LEARNING FOR OUT-OF-DISTRIBUTION NONLINEAR DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Finite-dimensional Koopman models depend critically on the observables used to represent nonlinear dynamics. Generic neural encoders can fit in-distribution trajectories, but provide no structural preference for inverse-power, fractional, or saturation dynamics and can extrapolate poorly beyond the training region. We introduce PadéKoop, a Koopman framework that learns observables as ratios of independent numerator and strictly positive denominator networks. This Padé-style parameterization represents denominator-driven structure while keeping every observable well-defined; the resulting features augment the physical state and evolve through a linear Koopman operator. To separate representation error from latent amplification, PadéKoop combines rational observables with optional spectral control of the Koopman matrix. We derive well-posedness, rational-approximation, and rollout-error properties that clarify these complementary roles. Across four equation-generated physical systems and a multi-task enzyme benchmark, PadéKoop achieves the lowest mean horizon-100 out-of-distribution error among controlled MLP- and KAN-observable Koopman baselines. A parameter-matched enzyme-kinetics ablation shows 69% lower error than an MLP and 54% lower error than a constant-denominator control. On a public robotic square-pushing transfer task, an action-conditioned PadéODE reduces rollout MSE by 33.3% relative to NeuralODE, while a Padé-observable controlled Koopman model improves over the corresponding paper-style Koopman variants. Spectral-control experiments further show that hard projection can suppress catastrophic latent amplification, but does not consistently improve predictive accuracy. Together, these results support rational observables as an effective inductive bias for denominator-driven dynamics and spectral control as a complementary robustness mechanism, rather than a guarantee of universally accurate long-horizon prediction.

1 INTRODUCTION

Koopman operator theory represents a nonlinear dynamical system through the linear evolution of observables (Koopman, 1931; Mezic, 2013). This viewpoint has led to practical data-driven approximations based on dynamic mode decomposition (DMD), extended DMD (EDMD), and learned latent coordinates (Schmid, 2010; Williams et al., 2015; Lusch et al., 2018). Its appeal is particularly strong for long-horizon prediction: once a useful observable map has been found, repeated evolution is implemented by powers of a linear operator. The practical bottleneck is precisely that observable map. The exact Koopman operator is generally infinite-dimensional, and a finite-dimensional approximation can retain a one-step closure residual when its learned coordinates are inaccurate. Repeated application of the learned operator then propagates this residual, making the choice of observables especially important outside the training region.

This issue is particularly prominent when the functional form of the dynamics is known. Coulomb and Lennard–Jones interactions contain inverse powers (MacDonald et al., 1957; Koura & Matsumoto, 1991); Michaelis–Menten kinetics contains a saturating ratio (Srinivasan, 2022); and many biochemical rate laws combine numerator and denominator effects. A generic MLP can approximate these functions on a compact training domain, but universal approximation does not specify how the learned observable should extrapolate. An inaccurate observable can therefore produce both

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a large closure residual and a poor coordinate system for repeated linear evolution. We ask whether the observable itself should expose the rational structure of these systems.

Existing learned Koopman models address the observable bottleneck with generic neural encoders, autoencoder coordinates, or fixed dictionaries. These choices provide flexibility, but generic encoders do not explicitly prefer denominator-driven behavior, whereas a fixed physics dictionary can impose a law that is too specific to transfer across systems. To address this gap, we propose PadéKoop, a finite-dimensional Koopman model with Padé-style rational observables. PadéKoop adopts a dual neural-branch architecture to learn a numerator and a positive denominator, forming an element-wise ratio that is concatenated with the physical state for linear evolution. Unlike a fixed analytic force model, the architecture does not hard-code a particular force law; it restricts the observable family toward behavior that naturally includes inverse-power and saturation structure. The denominator and the linear operator play distinct roles in the model: the denominator controls the well-posedness and local scaling of the observable, whereas the operator governs repeated propagation in the lifted space. A positive denominator prevents poles in the encoder but does not stabilize the latent transition. Conversely, a bounded operator norm can suppress latent explosion but cannot make an inaccurate observable invariant. This separation motivates treating representation error and propagation error independently.

Our contributions are:

- We introduce PadéKoop, a finite-dimensional Koopman framework that combines rational observable learning with linear latent evolution and optional spectral control for denominator-driven nonlinear dynamics.
- PadéKoop employs independent numerator and strictly positive denominator networks to construct elementwise rational observables, which augment the physical state before Koopman evolution. We establish well-posedness, rational-approximation, and rollout-error results that clarify the roles and limitations of the rational representation and spectral control.
- We evaluate the method across five equation-generated physical and biochemical systems using OOD rollouts, sample-efficiency and horizon sweeps, noise stress tests, parameter-matched denominator ablations, spectral-control studies, and raw failure accounting. A separate public robotic square-pushing experiment tests transfer to measured action-conditioned dynamics. These experiments demonstrate that the learnable denominator improves rational extrapolation and that hard spectral projection can suppress catastrophic latent amplification, while also revealing its task-dependent accuracy cost.

2 Related Work

Data-driven Koopman models. DMD and EDMD estimate finite-dimensional linear operators from state measurements or fixed dictionaries (Schmid, 2010; Williams et al., 2015). Neural approaches learn observables or autoencoder coordinates jointly with the linear transition (Takeishi et al., 2017; Lusch et al., 2018; Yeung et al., 2019). Consistent Koopman autoencoders regularize forward and backward dynamics (Azencot et al., 2020), and controlled variants incorporate actions into latent linear systems (Shi & Meng, 2022). Recent work applies control-coherent Koopman models to contact-rich locomotion and manipulation, including dynamic planar pushing (O'Neill et al., 2026). These methods establish the value of learned coordinates, but typically use generic neural encoders. PadéKoop changes the function class of the observable while retaining the same finite-dimensional Koopman pipeline.

Structured dynamics learning. Neural ODEs learn continuous vector fields (Chen et al., 2018); SINDy selects sparse governing terms (Brunton et al., 2016); and Hamiltonian networks encode energy-based structure (Greydanus et al., 2019). Such models illustrate a broader principle: a useful inductive bias can improve identification when it matches the system. Our bias is neither conservation nor sparsity. It is a numerator–denominator decomposition intended for rational force and rate laws.

Rational neural approximation. Classical Padé approximants represent functions as polynomial ratios and often extrapolate differently from polynomial truncations (Baker & Graves-Morris, 1996).

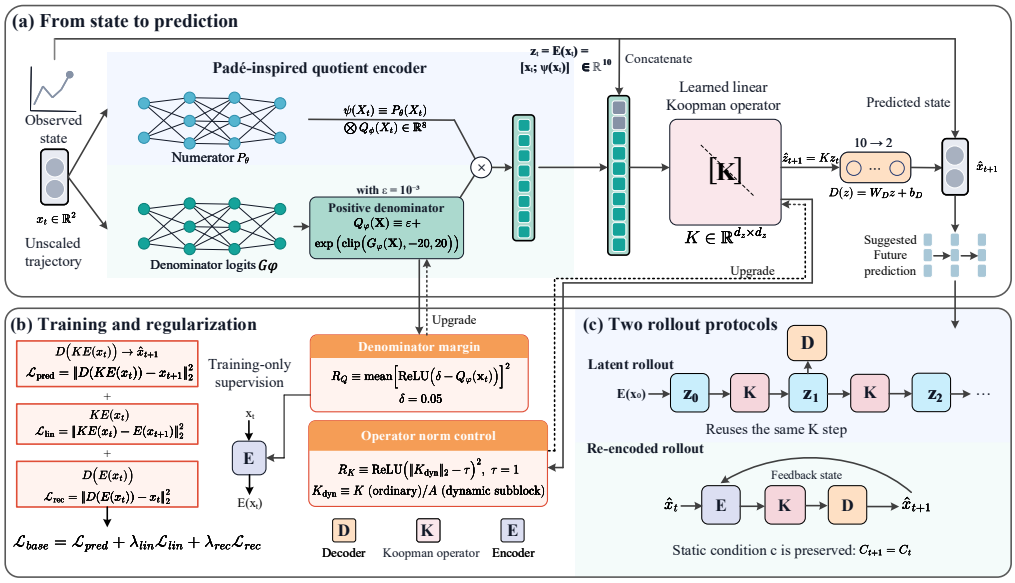

Figure 1: **PadéKoop architecture, training, and rollout.** **(a)** The observed state x_t is processed by independent numerator N_θ and denominator-logit G_ϕ networks. A positive transform produces $Q_\phi(x_t) > 0$, where $Q_\phi(x_t) = \epsilon + \exp(\mathrm{clip}(G_\phi(x_t), -20, 20))$ and $\epsilon = 10^{-3}$. Their elementwise quotient defines the rational observable $\psi(x_t) = N_\theta(x_t) \oslash Q_\phi(x_t)$. Concatenating x_t and $\psi(x_t)$ gives the lifted state $z_t = E(x_t) = [x_t; \psi(x_t)] \in \mathbb{R}^{d_z}$, which evolves through the learned linear Koopman operator K. The affine decoder $D(z) = W_D z + b_D$ maps the propagated latent state to $\hat{x}_{t+1} = D(Kz_t)$. **(b)** One-step prediction, latent Koopman consistency, and state reconstruction jointly train the shared encoder and decoder. The ground-truth next state is used as training-only supervision for $E(x_{t+1})$ and one-step prediction. The optional denominator-margin penalty discourages small denominators, while soft operator-norm regularization or hard spectral projection controls amplification in the dynamic Koopman block K_{dyn}. These configurable regularizers address distinct failure modes: small denominators in the rational encoder and amplification in the latent transition, respectively. **(c)** Latent rollout encodes the initial state once and repeatedly applies the same K, directly testing approximate Koopman closure. Re-encoded rollout feeds each decoded prediction back through the encoder before the next Koopman step and can partially correct off-manifold drift. Static conditioning variables are preserved with $c_{t+1} = c_t$ and are excluded from the dynamic-state error.

Rational neural networks and Padé activation units introduce trainable rational components into deep models (Boullé et al., 2020; Molina et al., 2019). We use the Padé principle at the level of a Koopman observable map: numerator and denominator are learned neural functions rather than fixed-degree polynomials. The name "Padé-style" is deliberate; our model is inspired by rational approximation, not a claim that every learned feature is a classical coefficient-matched Padé approximant.

3 Method

3.1 Finite-Dimensional Koopman Learning

A finite-dimensional Koopman approximation aims to identify coordinates in which nonlinear state evolution is represented by a linear operator. In practice, consider a discrete-time autonomous system

$$x_{t+1} = F(x_t), \qquad x_t \in \mathbb{R}^{d_x}. \tag{1}$$

Here t is the discrete time index, x_t is the observed state, d_x is its dimension, and $F : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ is the unknown state-transition map. For a controlled system, the corresponding notation is $x_{t+1} = F(x_t, u_t)$; the action-conditioned extension is used in the robotic transfer experiment. We do not assume that a globally invariant finite-dimensional subspace exists.

A learned observable $g : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ and matrix $K \in \mathbb{R}^{d_z \times d_z}$ seek to satisfy

$$z_t = g(x_t), \qquad z_{t+1} \approx K z_t. \tag{2}$$

Here $z_t \in \mathbb{R}^{d_z}$ is the lifted state, d_z is the lifted dimension, and K is the learned Koopman transition matrix. The decoder $D : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ maps the predicted lifted state back to the physical state as $\hat{x}_{t+1} = D(Kz_t)$. The approximation is local to the data distribution; no finite invariant subspace is assumed to exist globally.

3.2 Positive-Denominator Rational Observables

Given an input $x \in \mathbb{R}^{d_x}$, the proposed observable module uses two independent neural branches:

$$Q_\phi(x) = \epsilon + \exp(\mathrm{clip}(G_\phi(x), -20, 20)), \tag{3}$$

where $N_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{d_\psi}$ is the numerator network and $G_\phi : \mathbb{R}^{d_x} \to \mathbb{R}^{d_\psi}$ produces denominator logits. The rational observable is then formed coordinatewise:

$$\psi(x) = N_\theta(x) \oslash Q_\phi(x), \tag{4}$$

where $\oslash$ denotes elementwise division and $\epsilon = 10^{-3}$ is a numerical floor. Thus $Q_{\phi,i}(x) \geq \epsilon$ for every coordinate, so the observable has no zero-denominator singularity. The lifted state retains the physical coordinates,

$$g(x) = [x, \psi(x)], \qquad d_z = d_x + d_\psi. \tag{5}$$

The numerator can represent signed features, while the denominator controls feature scale and saturation. Unlike a fixed analytic force model, both branches remain data-driven. This explicit branch decomposition is the rational inductive bias tested against the generic MLP and KAN observable encoders.

Figure 1 shows the complete pipeline. The rational encoder addresses representation error; the linear layer exposes a finite-dimensional latent evolution; and spectral control acts directly on the repeated transition.

3.3 Objective and Spectral Control

We optimize one-step prediction, latent consistency, and reconstruction:

$$\mathcal{L}_{\mathrm{pred}} = \|D(Kg(x_t)) - x_{t+1}\|_2^2, \tag{6}$$

$$\mathcal{L}_{\mathrm{lin}} = \|Kg(x_t) - g(x_{t+1})\|_2^2, \tag{7}$$

$$\mathcal{L}_{\mathrm{rec}} = \|D(g(x_t)) - x_t\|_2^2. \tag{8}$$

Here $\mathcal{L}_{\mathrm{pred}}$ is the decoded one-step state-prediction loss, $\mathcal{L}_{\mathrm{lin}}$ is the Koopman latent-closure loss, and $\mathcal{L}_{\mathrm{rec}}$ is the state-reconstruction loss. The nonnegative coefficients λ_{lin} and λ_{rec} balance these terms. The base objective is

$$\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{lin}}\mathcal{L}_{\mathrm{lin}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}}. \tag{9}$$

When enabled, the optional denominator-margin term is

$$\mathcal{R}_Q = \mathrm{mean}_{t,i}\left[\mathrm{ReLU}\left(\delta - Q_{\phi,i}(x_t)\right)\right]^2, \qquad \delta = 0.05, \tag{10}$$

and it is added to the objective as $\lambda_Q \mathcal{R}_Q$ when used. For soft spectral control we add

$$\mathcal{L}_{\mathrm{spec}} = \lambda_{\mathrm{spec}}\left[\max\left(0, \|K_{\mathrm{dyn}}\|_2 - \tau\right)\right]^2, \tag{11}$$

where $\|K_{\mathrm{dyn}}\|_2$ is the induced operator norm of the dynamic Koopman block and $\tau > 0$ is the target threshold; this norm upper-bounds the spectral radius of that block. For ordinary systems $K_{\mathrm{dyn}} = K$, whereas for systems with static conditioning K_{dyn} denotes the dynamic subblock. For hard control, after each optimizer update we project

$$K_{\mathrm{dyn}} \leftarrow K_{\mathrm{dyn}} \min\left(1, \frac{\tau}{\|K_{\mathrm{dyn}}\|_2}\right). \tag{12}$$

The experiments use either no spectral constraint, the soft penalty, or the hard projection. We do not use the denominator lower-bound penalty as a main contribution: the positive parameterization already ensures $Q_\phi(x) \geq \epsilon$, and in our diagnostic runs the additional penalty was usually inactive.

3.4 Properties and Scope

The following results formalize the intended mechanisms. Proofs are in Appendix B.

Proposition 1 (Well-defined and Lipschitz observables)**.** *Let Ω be compact, $Q_i(x) \geq q_{\min} > 0$ for every output coordinate, N and Q be Lipschitz on Ω with constants L_N, L_Q, and $\|N(x)\|_\infty \leq B_N$. Then $\psi = N \oslash Q$ is well-defined and*

$$\|\psi(x) - \psi(y)\|_\infty \leq \left(\frac{L_N}{q_{\min}} + \frac{B_N L_Q}{q_{\min}^2} \right) \|x - y\|_2. \tag{13}$$

Proposition 2 (Error for a rational target)**.** *Let $f = p/q$ on Ω, with $q(x) \geq q_0 > 0$. If $\|N - p\|_\infty \leq \delta_p$, $\|Q - q\|_\infty \leq \delta_q < q_0$, then*

$$\left\| \frac{N}{Q} - \frac{p}{q} \right\|_\infty \leq \frac{\delta_p}{q_0 - \delta_q} + \frac{\|p\|_\infty \delta_q}{q_0(q_0 - \delta_q)}. \tag{14}$$

Proposition 2 shows how branch errors compose when a target admits a denominator bounded away from zero. It does not establish lower sample complexity than an MLP; the empirical question is whether exposing this decomposition creates a useful bias.

Proposition 3 (Accumulation of latent model error)**.** *Suppose the one-step latent residual is bounded by η, so that $\|g(F(x)) - Kg(x)\|_2 \leq \eta$, and $\|K\|_2 \leq \rho$. If e_t is the latent rollout error, then*

$$e_t \leq \rho^t e_0 + \eta \sum_{j=0}^{t-1} \rho^j. \tag{15}$$

For $\rho < 1$, the residual term is at most $\eta/(1 - \rho)$.

The result separates two failure sources: the observable controls the residual η, whereas spectral control limits its amplification. A soft penalty need not enforce $\|K\|_2 \leq \tau$; a hard projection does. Neither guarantees small state-space error when the decoder, finite-dimensional closure, or phase dynamics are inaccurate.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We evaluate five equation-generated systems, with their governing equations and train–test ranges reported uniformly in Appendix Table 5. **Coulomb** uses an inverse-square interaction; **Lennard–Jones (LJ)** combines attractive and repulsive inverse powers; and **Argon LJ** uses the same reduced LJ law with Argon reference scales and a broader velocity initialization (MacDonald et al., 1957; Koura & Matsumoto, 1991). **Michaelis–Menten (MM)** uses a saturating substrate-rate law, while **multi-task enzyme kinetics** combines manually specified MM, inhibition, and Hill-type rate laws with task codes (Srinivasan, 2022). All five datasets are simulated trajectories rather than measured molecular or SABIO-RK data. The enzyme split tests concentration extrapolation on the same reaction tasks rather than transfer to unseen enzymes.

4.1.2 Evaluation Metrics

Training and OOD initial states are disjoint by construction. For a rollout of length H, with dynamic coordinates indexed by j, the primary metric is $\mathrm{MSE} = H^{-1} d_x^{-1} \sum_{t=1}^{H} \sum_j (\hat{x}_{t,j} - x_{t,j})^2$. We report raw predictions; divergent or non-finite trajectories are retained through a failure indicator rather than silently discarded. "Latent" rollout repeatedly applies K, while "re-encode" rollout decodes and re-encodes each predicted state. The former is the primary test of finite-dimensional linear evolution.

System	MLP	KAN	PadéKoop
Coulomb	7.13×10^{-3}	1.92×10^{-2}	$\mathbf{7.93\times10^{-5}}$
Lennard–Jones	3.97×10^{-1}	1.50	$\mathbf{7.33\times10^{-2}}$
Argon LJ	5.32×10^{-1}	2.55	$\mathbf{6.89\times10^{-2}}$
Michaelis–Menten	1.13×10^{-2}	1.58×10^{-2}	$\mathbf{6.77\times10^{-3}}$

Table 1: OOD rollout MSE at horizon 100 with 1,000 training transitions (mean $\pm$ sample standard deviation over five seeds; lower is better). Models share the Koopman pipeline and differ only in the observable encoder.

4.1.3 Baselines

The main controlled baselines are MLP-Obs and KAN-Obs (Liu et al., 2024): they use the same lifted dimension, Koopman transition, decoder, losses, and data, changing only the observable encoder. The mechanism study adds (i) a constant-denominator numerator network and (ii) a capacity-matched MLP. We also ran broader DMD, EDMD, autoencoder, ODE, SINDy, and physics-informed baselines during development, but do not use those heterogeneous comparisons to claim that PadéKoop universally dominates dynamics models.

4.1.4 Stress-Test Protocols

We additionally retain three controlled stress tests from the earlier four-system protocol because they answer distinct questions about the observable architecture. The sample-efficiency sweep uses $n \in \{200, 500, 1000, 4000\}$ at horizon 100; the horizon sweep uses $H \in \{50, 100, 200, 500\}$ at $n = 1000$; and the label-noise sweep uses standard deviations $\{0, 0.01, 0.05, 0.1\}$ at $n = 1000$ and $H = 100$. These results use the same five optimization seeds within each setting. We report normalized ratios only within a system and setting, and do not compare their absolute errors with the audited 4,000-sample mechanism and spectral studies.

4.1.5 Robotic Square-Pushing Transfer

We additionally evaluate measured action-conditioned dynamics using the public square-slider trajectories released with the control-coherent Koopman study of O'Neill et al. (2026). The data contain one pushing and one shoving hardware demonstration, each with 60 eight-dimensional states and 59 two-dimensional control inputs. For each trajectory, the first 70% of transitions are used for training and the contiguous final 30% for open-loop rollout testing. State and action normalization statistics are computed from the training prefix only. We compare residual MLP and capacity-near WideMLP transitions, NeuralODE and PadéODE vector fields, direct Padé residual dynamics, and paper-style controlled Koopman models with either an MLP or Padé observable. All reported values use the same temporal split and five optimization seeds. Because this small transfer study contains only two recorded trajectories and no matched OOD-range or horizon sweep, we report it separately from the primary synthetic benchmarks.

4.2 Comparison Results

4.2.1 Rational Observables Improve OOD Rollout

Table 1 isolates the observable encoder at horizon 100. PadéKoop obtains the lowest mean OOD error on all four systems. Relative to MLP-Obs, the reduction ranges from 40.3% on MM to 98.9% on Coulomb. The seed variance and unadjusted paired tests also show why means alone should not be overinterpreted: evidence is strong on LJ and Argon LJ, whereas the five-seed Coulomb and MM comparisons against MLP have wide uncertainty. Against KAN-Obs, PadéKoop wins all mean comparisons, with the Coulomb and MM paired tests below 0.01.

4.2.2 Sample Efficiency Across Rational Systems

Figure 2 tests whether the rational advantage persists as the number of training transitions changes. Panel (a) reports the geometric mean of each model's OOD MSE after normalization by MLP-Obs

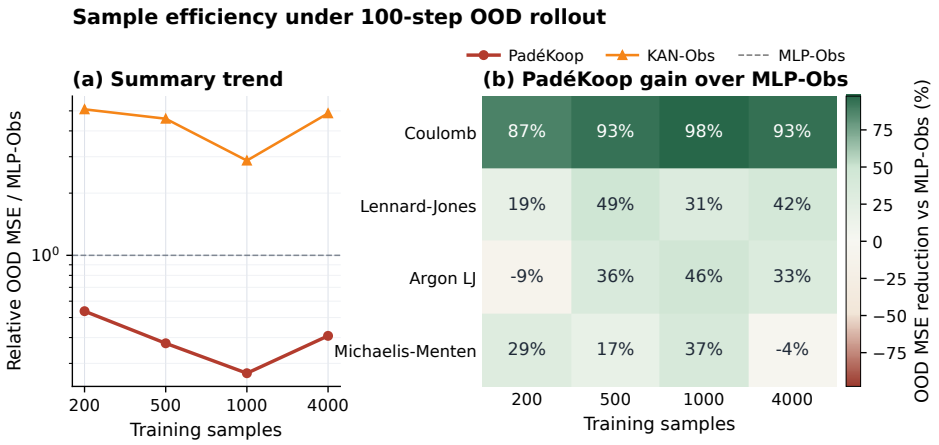

Figure 2: **Sample efficiency under 100-step OOD rollout.** (a) Geometric-mean OOD MSE relative to MLP-Obs across the four systems; lower is better. (b) Percentage reduction in MSE relative to MLP-Obs for each system and training size; positive values favor PadéKoop. Normalization is performed within each system and sample size, so the panel compares relative rather than absolute difficulty.

within every system and sample size. Panel (b) retains the system-wise results rather than hiding heterogeneity in one average. PadéKoop improves over MLP-Obs in 14 of the 16 system–sample-size cells. Coulomb shows reductions between 87% and 98%; LJ improves by 19–49%; and Argon LJ improves by 33–46% once at least 500 transitions are available. The exceptions are Argon LJ at $n = 200$ (-9%) and MM at $n = 4000$ (-4%). Thus the data support broad sample-efficiency gains, but not monotonic or universal superiority.

4.2.3 Horizon Dependence and the Boundary of the Bias

Figure 3 in Appendix E evaluates horizons from 50 to 500 with 1,000 training transitions. PadéKoop retains large gains on Coulomb (96–98%) and consistent gains on MM (36–43%). Argon LJ remains favorable on average, but its gain narrows from 57% at $H = 50$ to 4% at $H = 500$. LJ exposes the boundary most clearly: the rational encoder is better at $H = 50$ and 100, then becomes worse than MLP-Obs at $H = 200$ and 500. This crossover motivates treating representation error and repeated latent amplification separately; a compatible observable family does not by itself guarantee small error after many applications of K.

4.2.4 Noise Stress Test

The label-noise sweep provides a deliberately harder test. As detailed in Appendix E, PadéKoop is strongest without noise but loses its aggregate advantage at several nonzero noise levels. A force-level Padé module is more robust in this experiment, although that comparison changes the insertion point and therefore does not isolate the observable encoder. We use this negative result to bound the claim: rational observables improve OOD extrapolation in the studied clean and low-data regimes, but do not confer generic noise robustness.

4.3 Ablation Studies

4.3.1 Does the Learnable Denominator Matter?

Universal approximation makes a simple objection possible: perhaps PadéKoop only works because it has more parameters. Table 2 addresses this with the multi-task enzyme benchmark and the audited protocol. At horizon 100, the unconstrained rational encoder reduces OOD concentration error by 69.0% relative to the parameter-matched MLP and by 54.3% relative to the constant-denominator

Rollout	Horizon	Constant denominator	Matched MLP	PadéKoop
Trainable parameters	–	4,994	9,895	9,866
Latent, no spectral control	100	0.256 ± 0.132	0.377 ± 0.300	$\mathbf{0.117 \pm 0.054}$
Latent, soft spectral control	100	0.432 ± 0.050	0.476 ± 0.148	$\mathbf{0.174 \pm 0.050}$
Re-encode, soft spectral control	100	0.138 ± 0.081	0.237 ± 0.087	$\mathbf{0.081 \pm 0.018}$
Re-encode, soft spectral control	500	0.950 ± 0.428	1.386 ± 0.274	$\mathbf{0.631 \pm 0.092}$
Re-encode, soft spectral control	1000	1.093 ± 0.371	1.460 ± 0.271	$\mathbf{0.847 \pm 0.158}$

Table 2: Mechanism ablation on synthetic multi-task enzyme kinetics with 4,000 training transitions. Test concentrations are outside the training interval but use the same reaction tasks. Values are five-seed OOD MSE; parameter counts are included to distinguish the capacity-matched comparison from the smaller constant-denominator control.

System	No control	Soft penalty	Hard projection	Hard fail.
Coulomb	2.174 ± 2.920	1.527 ± 1.570	$\mathbf{0.827 \pm 0.451}$	0%
Lennard–Jones	$\mathbf{0.864 \pm 0.059}$	$1.432\times10^{3} \pm 3.200\times10^{3}$	1.123 ± 0.567	0%
Argon LJ	$1.26\times10^{10} \pm 2.78\times10^{10}$	$3.00\times10^{4} \pm 6.29\times10^{4}$	$\mathbf{1.785 \pm 0.342}$	0%
Michaelis–Menten	1.457 ± 2.380	1.457 ± 2.380	1.457 ± 2.380	0%
Enzyme kinetics	4.439 ± 3.020	4.439 ± 3.020	4.439 ± 3.020	0%

Table 3: Horizon-1000 OOD latent-rollout MSE with 4,000 training transitions and five seeds (mean $\pm$ standard deviation; lower is better). The unconstrained and soft-constrained Argon LJ models have failure rates of 59.8% and 48.6%, respectively, whereas hard projection reduces the observed failure rate to zero. A trajectory is classified as failed when its raw dynamic state is non-finite or has norm greater than 100 simulation units. Failed trajectories are counted rather than discarded.

control. It wins four of five paired seeds against each baseline. Under re-encoding, the rational model remains best from horizons 100 to 1,000, and soft spectral regularization further lowers its mean error.

The conclusion is narrower at very long pure-latent horizons. At horizon 1,000 without spectral control, the matched MLP has lower mean error than PadéKoop, while the constant-denominator model suffers failures. Thus the learnable denominator improves short- and medium-horizon concentration extrapolation, but does not alone control repeated linear propagation.

Inspection of the learned denominator confirms that it is input-dependent rather than numerically constant (coefficient of variation approximately one across the evaluated data). Its minimum remains above the numerical floor. Because the separate lower-bound penalty is zero at recorded checkpoints, these gains cannot be attributed to that inactive penalty; they support the architecture's learned denominator branch.

4.3.2 Spectral Control Prevents Amplification, Not Model Error

We next compare no constraint, the soft penalty in Eq. equation 11, and hard projection with $\tau = 1.05$. Table 3 reports the most diagnostic systems at horizon 1,000. Under the predefined failure criterion, the unconstrained Argon LJ models fail on 383 of 640 OOD latent rollouts (59.8%), whereas the hard-projected models fail on none. Soft regularization reduces but does not remove divergence. Hard projection also reduces the mean estimated $\|K^{1000}\|_2$ from 9.75×10^{7} to 1.11. However, the unconstrained seed-wise failure rates are highly heterogeneous (0%, 100%, 99.2%, 100%, and 0%), and four of five hard-projected runs select the initialization checkpoint. The result therefore demonstrates suppression of numerical amplification under this evaluation protocol, not successful learning of accurate stable dynamics in every run.

The result is not a universal accuracy gain. On LJ, the unconstrained model remains more accurate at horizon 1,000, and hard projection worsens horizon-100 error from 0.0139 to 0.0446. For MM and enzyme kinetics, the learned norm is already below the threshold, so projection is inactive and predictions are unchanged. Spectral control should therefore be understood as a robustness mechanism against latent amplification, with a task-dependent bias–accuracy tradeoff.

Model	Trainable parameters	Rollout MSE	Final-step MSE
MLP	5,384	0.1568 ± 0.0856	0.2729 ± 0.2240
WideMLP	10,109	0.1108 ± 0.0500	0.3930 ± 0.2965
NeuralODE	5,384	0.0229 ± 0.0114	0.0492 ± 0.0453
Padé residual	10,768	0.0468 ± 0.0378	0.1430 ± 0.2173
PadéODE	10,768	$\mathbf{0.0153 \pm 0.0007}$	$\mathbf{0.00285 \pm 0.00057}$
PaperKoopman	1,998	0.2168 ± 0.2218	0.0496 ± 0.0489
PaperKoopmanCtrl	2,034	5.9299 ± 11.3114	0.9602 ± 1.2723
PadéKoopmanCtrl	1,532	0.0263 ± 0.0101	0.0410 ± 0.0272

Table 4: Open-loop prediction on two held-out suffixes of public square-pushing hardware trajectories (mean $\pm$ sample standard deviation over five optimization seeds; lower is better). The experiment uses measured states and commanded controls, a 70/30 temporal split, and no OOD-range or horizon sweep.

Hard projection improves the horizon-1000 result on Coulomb and removes the observed catastrophic failures on Argon LJ. This robustness does not imply a universal accuracy gain: on Lennard–Jones, the unconstrained model remains more accurate, while the constraint is inactive on Michaelis–Menten and enzyme kinetics.

4.4 Additional Transfer Experiments

4.4.1 Transfer to Measured Contact-Rich Dynamics

Table 4 evaluates whether the rational parameterization transfers beyond the equation-generated benchmarks. PadéODE obtains the lowest mean rollout MSE, 0.0153 ± 0.0007, compared with 0.0229 ± 0.0114 for NeuralODE, a 33.3% reduction. The capacity-near WideMLP remains substantially worse at 0.1108 ± 0.0500, indicating that the gain is not explained by parameter count alone in this split. Within controlled Koopman models, PadéKoopmanCtrl reaches 0.0263 ± 0.0101, compared with 0.2168 ± 0.2218 for the autonomous paper-style Koopman model. The paper-style controlled variant is unstable in this training protocol, so we report its result but do not use it as the denominator of an effect-size claim.

The result is architecture-dependent rather than universal: the direct Padé residual model is worse than NeuralODE, whereas the PadéODE and Padé-observable controlled Koopman variants are competitive. Figure 6 shows representative planar rollouts on the two held-out trajectory suffixes. This evidence supports transfer of rational neural components to measured contact dynamics, but the two-trajectory temporal split is too small to establish broad robotic generalization or closed-loop control improvement.

4.4.2 Where Should Rational Structure Enter?

We also evaluated the numerator–denominator motif as an ODE vector field and force/rate module (Appendix E). Force-level insertion is strongest on Coulomb and MM, where the update exposes inverse-power or saturating structure, whereas learned-force baselines remain strongest on LJ and Argon LJ. These heterogeneous comparisons are diagnostic, not interchangeable architecture rankings: they show that the useful bias follows the rational mechanism's location, while the controlled MLP/KAN comparison supports the Koopman claim.

5 Conclusion

We introduced PadéKoop, a finite-dimensional Koopman model with positive-denominator rational observables. It improves controlled OOD rollouts across five equation-generated physical and biochemical systems, with denominator-matched ablations supporting the learnable denominator, and transfers to a public measured square-pushing task with action-conditioned dynamics. Spectral projection reduces catastrophic latent amplification but can trade accuracy for robustness and does not guarantee accurate long-horizon prediction.

AI USE STATEMENT

Generative AI tools were used to assist with code development, figure layout, language editing, and manuscript restructuring. All AI-assisted code and numerical summaries were checked against saved experiment artifacts, and the authors reviewed the final equations, claims, citations, figures, and prose. Generative AI was not treated as an experimental oracle or an evidence source. The authors take responsibility for the complete content of this submission.

ETHICS STATEMENT

The primary experiments use equation-generated trajectories. The robotic transfer experiment uses a previously released public dataset and does not involve new human-subject data collection. We identify synthetic and measured data explicitly and report negative and unstable outcomes rather than removing them from aggregate results.

REPRODUCIBILITY STATEMENT

Section 4 describes the datasets, evaluation protocols, metrics, and failure criterion. Appendix C specifies model sizes, optimization settings, checkpointing, failure accounting, and noise injection, while Appendix B gives complete proofs of the stated propositions. The supplementary artifact contains source snapshots, command-line configurations, random seeds, training histories, checkpoints, per-run results, and aggregation scripts for the audited experiments.

REFERENCES

Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael W. Mahoney. Forecasting sequential data using consistent Koopman autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 475–485, 2020.

Jr. Baker, George A. and Peter Graves-Morris. *Padé Approximants*. Cambridge University Press, 2 edition, 1996.

Nicolas Boullé, Yuji Nakatsukasa, and Alex Townsend. Rational neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14243–14253, 2020.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bernard O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Katsuhisa Koura and Hiroaki Matsumoto. Variable soft sphere molecular model for inverse-power-law or Lennard-Jones potential. *Physics of Fluids A: Fluid Dynamics*, 3(10):2459–2465, 1991.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

W. M. MacDonald, M. N. Rosenbluth, and Wong Chuck. Relaxation of a system of particles with coulomb interactions. *Physical Review*, 107:350–352, 1957.

Igor Mezic. Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Padé activation units: End-to-end learning of flexible activation functions in deep networks. *arXiv preprint arXiv:1907.06732*, 2019.

C. O'Neill, J. Terrones, and H. H. Asada. Koopman global linearization of contact dynamics for robot locomotion and manipulation enables elaborate control. *Nature Communications*, 2026. doi: 10.1038/s41467-026-72485-7.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

Haojie Shi and Max Q.-H. Meng. Deep Koopman operator with control for nonlinear systems. *arXiv preprint arXiv:2202.08004*, 2022.

B. Srinivasan. A guide to the Michaelis–Menten equation: Steady state and beyond. *FEBS Journal*, 289(20):6086–6098, 2022.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *2019 American Control Conference*, pp. 4832–4839, 2019.

A LIMITATIONS

All primary benchmarks are generated from known equations. This control is useful for testing structural match, but it is not evidence of performance on measured molecular or biochemical data. In particular, the multi-task enzyme benchmark uses manually specified kinetic parameters and should not be described as a SABIO-RK experiment. The robotic transfer study uses measured hardware trajectories, but contains only two demonstrations with a temporal prefix/suffix split; it is evidence of feasibility, not a comprehensive robot-learning benchmark or a closed-loop control evaluation. Evaluation on larger experimental datasets with independent trajectories, batch effects, missing variables, and irregular sampling remains necessary.

The method does not enforce conservation laws, symplectic structure, particle permutation symmetry, or equivariance. A specialized physical model may be preferable when those properties dominate. The Padé-style encoder also does not identify a unique numerator and denominator: multiplying both by the same nonzero function leaves their ratio unchanged, so branch values are not directly physical explanations.

Finally, neither positivity nor spectral control proves accurate global linearization. Hard projection bounds the full matrix norm at each update, but a threshold above one still permits exponential growth, and a threshold below one may erase neutral or oscillatory modes. Pilot multi-step experiments on LJ and Argon LJ frequently selected the initialization checkpoint, illustrating that non-expansive dynamics can still have poor phase accuracy. Better long-horizon objectives, invariant-aware constraints, and validation criteria are open problems.

B PROOFS

B.1 PROOF OF PROPOSITION 1

For each coordinate, add and subtract $N_i(y)/Q_i(x)$:

$$|\psi_i(x) - \psi_i(y)| \le \frac{|N_i(x) - N_i(y)|}{Q_i(x)} + |N_i(y)| \frac{|Q_i(y) - Q_i(x)|}{Q_i(x)Q_i(y)} \tag{16}$$

$$\le \left(\frac{L_N}{q_{\min}} + \frac{B_N L_Q}{q_{\min}^2} \right) \|x - y\|_2. \tag{17}$$

Taking the maximum over coordinates proves the result. Equation equation 4 supplies $q_{\min} = \epsilon$ for finite branch outputs.

B.2 PROOF OF PROPOSITION 2

Since $Q \ge q - \delta_q \ge q_0 - \delta_q > 0$,

$$\left| \frac{N}{Q} - \frac{p}{q} \right| \le \left| \frac{N - p}{Q} \right| + |p| \left| \frac{q - Q}{qQ} \right| \tag{18}$$

$$\le \frac{\delta_p}{q_0 - \delta_q} + \frac{\|p\|_\infty \delta_q}{q_0(q_0 - \delta_q)}. \tag{19}$$

Taking the supremum on Ω gives Eq. equation 14.

B.3 PROOF OF PROPOSITION 3

Let $r_t = g(x_{t+1}) - Kg(x_t)$, with $\|r_t\|_2 \le \eta$. The latent error obeys $e_{t+1} \le \|K\|_2 e_t + \|r_t\|_2 \le \rho e_t + \eta$. Unrolling this recursion gives Eq. equation 15. For $\rho < 1$, the geometric sum is bounded by $(1 - \rho)^{-1}$.

C ADDITIONAL EXPERIMENTAL DETAILS

Implementation details. Unless otherwise stated, all results are reported over five random seeds. The audited ICLR protocol uses 4,000 training transitions and trains each model for at most 6,000 epochs using Adam with learning rate 10^{-3} and gradient clipping at 1. The hidden width is 64, the lifted representation contains eight learned observables, and $\lambda_{\text{lin}} = \lambda_{\text{rec}} = 0.1$. Checkpoints are selected only by an independent in-distribution validation set with 32 sequences; all methods are then evaluated on 128 fixed test sequences. The soft spectral weight is 10^{-4}, and hard spectral projection uses $\tau = 1.05$.

Checkpointing. Validation trajectories are independent of training and test trajectories. The same in-distribution latent/raw horizon-100 criterion selects one checkpoint per training run; test horizons and OOD regions never select checkpoints. Early stopping checks every 100 epochs after a minimum of 1,000 epochs with patience 10 and relative improvement threshold 10^{-3}.

Failure accounting. A trajectory is marked failed when a raw dynamic coordinate is non-finite or its Euclidean norm exceeds 100 simulation units. Failure rates are reported separately, and failed outcomes are not deleted before aggregation. Clipped rollouts are diagnostic only; all main tables use raw outputs.

Noise. In the multi-task enzyme noise study, perturbations are applied only to dynamic substrate/reaction-output coordinates. The task code is categorical conditioning information and is not corrupted. This prevents label noise from silently changing the reaction identity.

Reproducibility. Each audited result directory stores the complete command-line configuration, model parameter counts, source snapshots and hashes, training histories, best and final checkpoints, per-trajectory metrics, aggregate summaries, and seed-wise comparisons. We will release these artifacts and the generation scripts with the anonymized submission.

System	Governing force/rate	Training range	OOD evaluation range(s)
Coulomb	$F(r) = r^{-2}$	$r \in [0.9, 3.0]$	near $[0.75, 0.89]$; far $[3.1, 5.0]$
LJ	$F(r) = 24(2r^{-13} - r^{-7})$	$r \in [0.9, 3.0]$	near $[0.75, 0.89]$; far $[3.1, 5.0]$
Argon LJ	reduced LJ; broader $v_0 \in [-1.2, 1.2]$	$r \in [0.9, 3.0]$	near $[0.75, 0.89]$; far $[3.1, 5.0]$
MM	$\dot{S} = -1.5S/(0.5 + S)$	$S \in [0.1, 2.0]$	low $[0.02, 0.09]$; high $[2.1, 5.0]$
Enzyme tasks	MM, inhibition, and Hill rate laws	$S \in [0.05, 2.0]$	same tasks, $S \in [2.1, 8.0]$

Table 5: Equation-generated systems and concentration/distance splits in the audited ICLR protocol. Force values and state coordinates are clipped to simulator safety ranges specified in the released configuration; main evaluations use unclipped model outputs.

Experiment	Main comparison	Regime	Claim role
Observable encoder benchmark	Padé vs. MLP vs. KAN observables	Four systems, $n = 1000$, $H = 100$, five seeds	Tests whether the rational observable improves a shared Koopman pipeline.
Sample-efficiency sweep	Padé, MLP, KAN observables	$n = 200, 500, 1000, 4000$, $H = 100$	Tests data dependence and exposes two settings without a Padé gain.
Horizon sweep	Padé, MLP, KAN observables	$H = 50, 100, 200, 500$, $n = 1000$	Tests error accumulation and documents the LJ long-horizon crossover.
Label-noise sweep	Observable and force-level models	Noise std. $0, 0.01, 0.05, 0.1$	Stress test; bounds rather than strengthens the generic robustness claim.
Enzyme mechanism ablation	Learnable denominator, constant denominator, matched MLP	Multi-task concentration OOD, $n = 4000$, five seeds	Separates denominator structure from parameter count.
Spectral-control study	None, soft penalty, hard projection	Five systems, $H = 1000$, five seeds	Separates suppression of amplification from predictive accuracy.
Plug-in study	Observable, ODE, and force/rate insertion	Four equation-generated systems	Tests where rational structure is useful; not used as an architecture-fair leaderboard.
Robotic transfer	Residual, ODE, and controlled Koopman models	Two measured square-pushing trajectories, five seeds	Tests feasibility on measured action-conditioned contact dynamics.

Table 6: Experiment inventory and the claim assigned to each study. Protocols with different training sizes, model assumptions, or data sources are interpreted separately.

System definitions and splits. All equation-generated systems use a discrete step of $\Delta t = 0.01$. For the force systems, $x_t = (r_t, v_t)$, $v_{t+1} = v_t + \Delta t\, F(r_t)$, and $r_{t+1} = r_t + \Delta t\, v_{t+1}$. MM evolves substrate concentration and stores a zero auxiliary coordinate. Multi-task enzyme kinetics evolves substrate concentration while preserving a normalized categorical task code. Table 5 gives the exact scalar ranges used by the audited protocol.

D Experiment Inventory and Claim Roles

Table 6 records the purpose of each experiment included in the paper. This separation is important because not every experiment supports the same claim: controlled encoder comparisons test the proposed observable family, whereas cross-architecture plug-in and robotic studies test transfer only.

E Additional Results and Diagnostics

Statistical scope. Five seeds characterize optimization variability but provide limited power. We therefore report means, sample standard deviations, and seed-wise wins, and avoid treating every unadjusted p value as a definitive hypothesis test. The strongest evidence comes from agreement

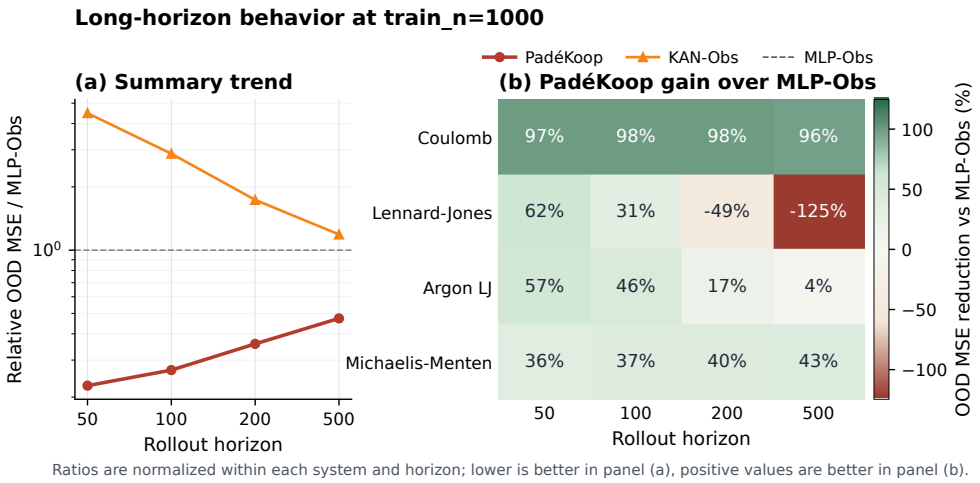

Figure 3: **Long-horizon OOD behavior at 1,000 training transitions.** (a) Geometric-mean OOD MSE relative to MLP-Obs. (b) System-wise MSE reduction relative to MLP-Obs. The LJ crossover at horizons 200 and 500 is a documented failure mode rather than an omitted outlier.

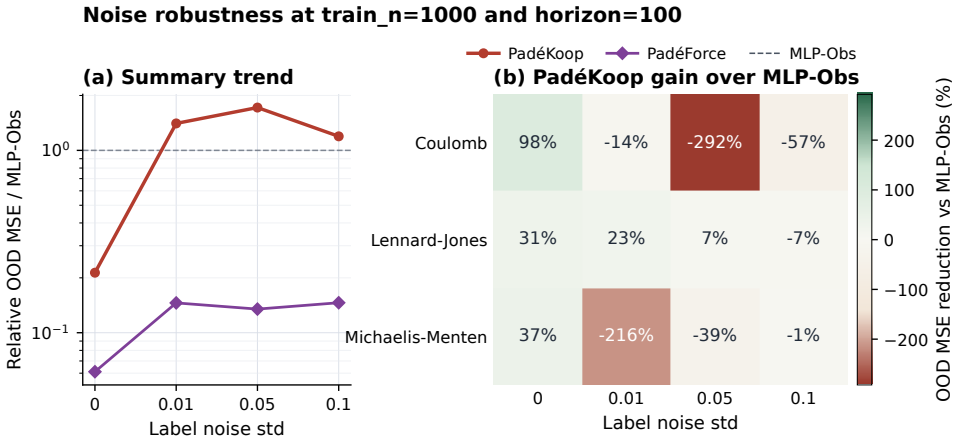

Figure 4: **Label-noise stress test at $n = 1000$ and $H = 100$.** (a) Geometric-mean OOD MSE relative to MLP-Obs. (b) System-wise reduction relative to MLP-Obs; positive values favor PadéKoop. PadéKoop is strong at zero noise but does not retain a uniform advantage as noise increases. PadéForce is shown as a diagnostic structured module, not as an encoder-matched baseline.

among controlled architecture comparisons, capacity-matched mechanism ablations, and stress tests rather than from a single significance threshold.

Why re-encoding differs from latent rollout. Pure latent rollout applies K^t and directly tests approximate Koopman closure. Re-encoding applies $g(D(\cdot))$ between steps and can correct off-manifold latent drift. Better re-encoding performance is useful operationally, but cannot substitute for evidence that the learned latent coordinates evolve linearly. We report both and use latent rollout for the central linearization claim.

Soft versus hard spectral control. The soft penalty only acts when the estimated norm exceeds the target and can be outweighed by prediction losses. Hard projection enforces the norm bound after every update, which explains its stronger effect on Argon LJ. For ordinary systems, the projection acts on the full lifted Koopman operator; for static-conditioned systems, it acts on the selected

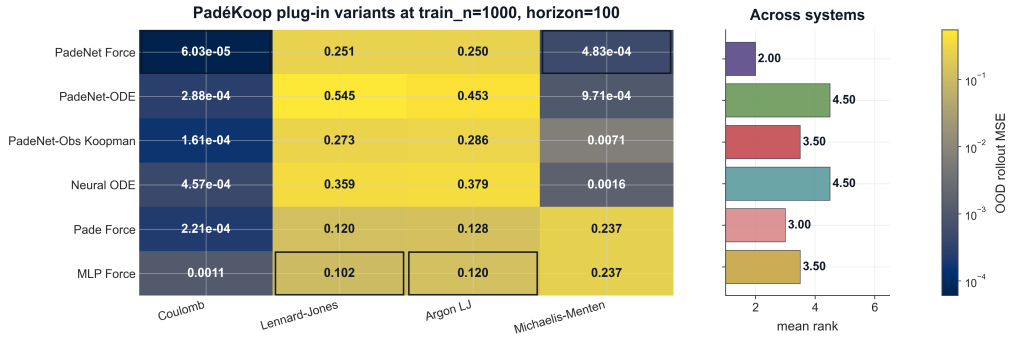

Figure 5: **Padé-style modules at different insertion points.** OOD rollout MSE at $n = 1000$ and $H = 100$ for observable, ODE, and force/rate variants. The heat map reports absolute MSE on a logarithmic color scale and the side panel reports mean rank. Highlighted cells are system-wise minima. Because the variants encode different assumptions, this figure diagnoses where rational structure helps and is not used to claim universal superiority of one architecture.

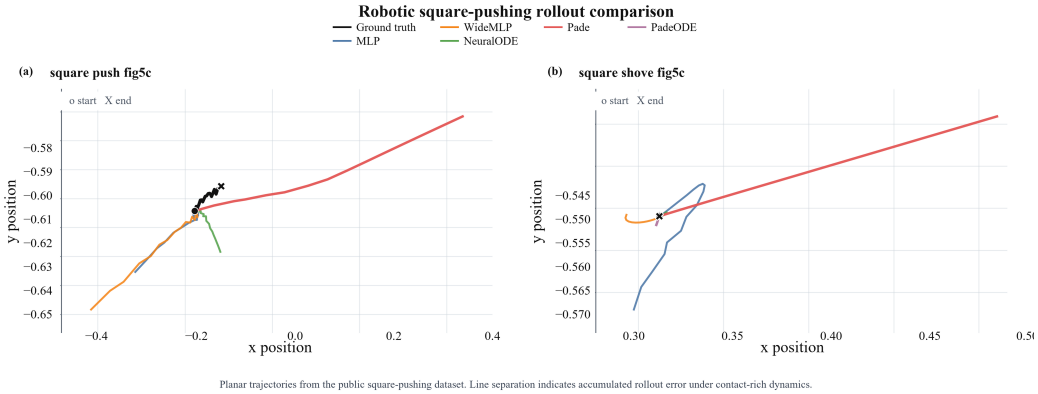

Figure 6: **Representative robotic square-pushing rollouts.** Planar x–y trajectories for the held-out suffixes of the pushing and shoving hardware demonstrations using the seed-0 models. The figure is qualitative; Table 4 reports the five-seed aggregate over all eight state coordinates. Start and end markers refer to the ground-truth suffix.

dynamic subblock K_{dyn}. A future block-structured constraint may preserve neutral physical modes while controlling only learned observables, but our pilot results do not yet support a positive claim for that variant.