Course Correcting Koopman Representations

Mahan Fathi   Clement Gehring   Jonathan Pilault   David Kanaa   Pierre-Luc Bacon   Ross Goroshin

Mila · Google DeepMind

arXiv

Koopman Autoencoders (n >> d)

Encoder
$\phi : x \in \mathbb{R}^d \rightarrow z \in \mathbb{R}^n$

Decoder
$\psi : z \in \mathbb{R}^n \rightarrow x \in \mathbb{R}^d$

Latent Dynamics
$\mathbf{K} \in \mathbb{R}^{n \times n} : z \rightarrow z$

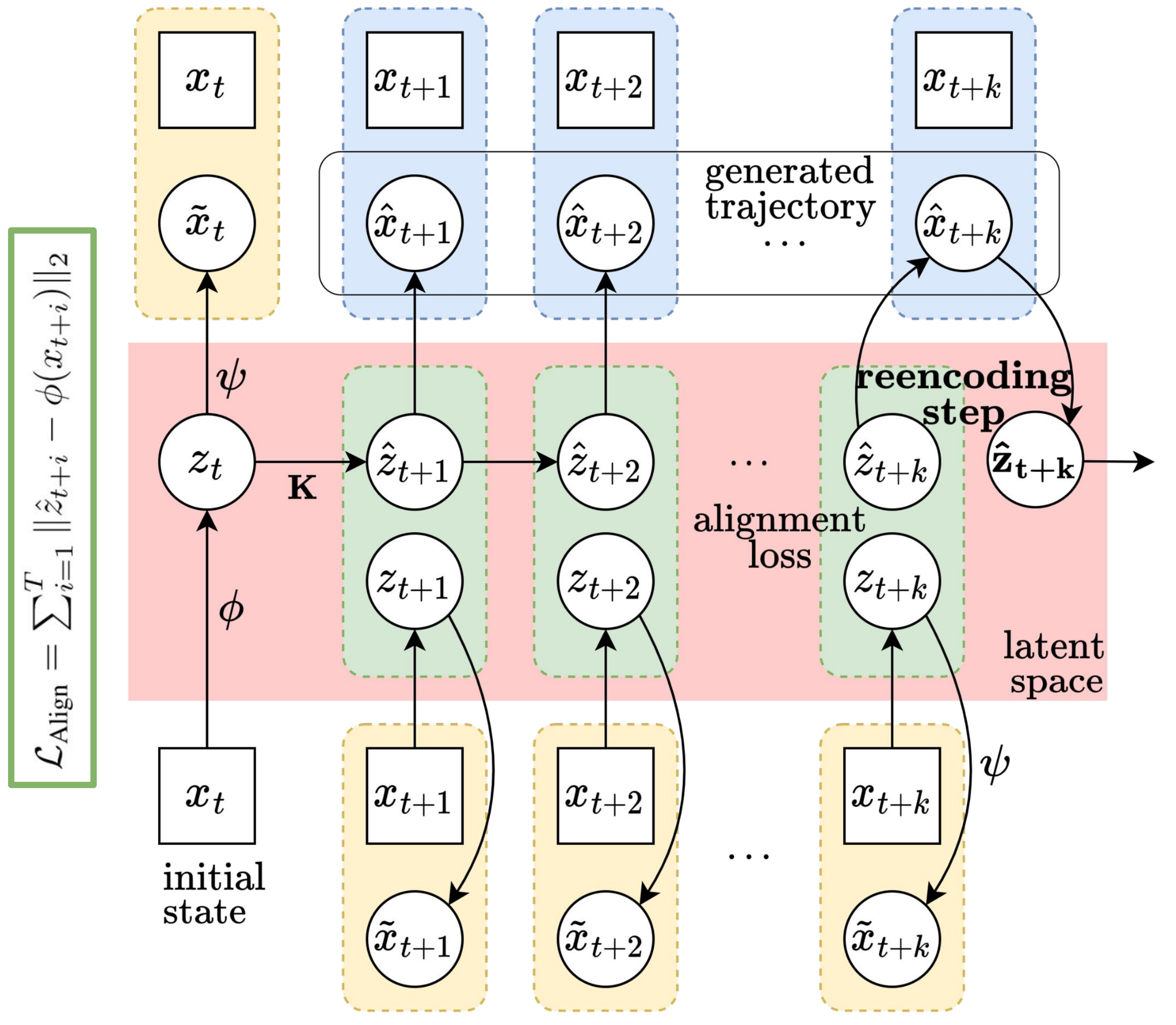
$$\phi(x_{t+1}) = \mathbf{K}\phi(x_t)$$

$$z_t \approx \phi(x_t), \quad z_{t+1} \approx \mathbf{K} z_t, \quad x_t \approx \psi(z_t)$$

$$\min_{\hat{\mathbf{K}},\phi,\psi} \sum_{\mathcal{D}} \|x_t - \psi(\phi(x_t))\|_2 + \lambda \cdot \|\phi(x_{t+1}) - \hat{\mathbf{K}}\phi(x_t)\|_2$$

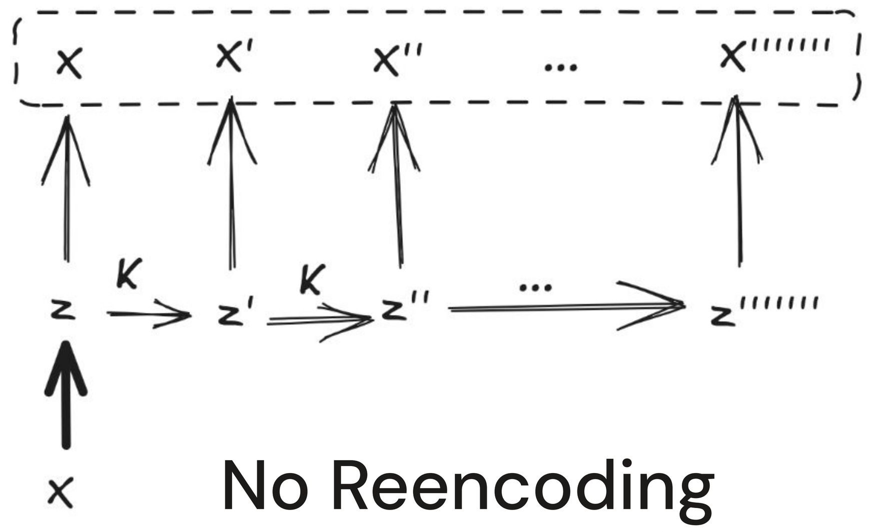
Multi-step Koopman

$$\mathcal{L}_{\text{Pred}} = \sum_{i=1}^{T} \|x_{t+i} - \psi(\hat{z}_{t+i})\|_2$$



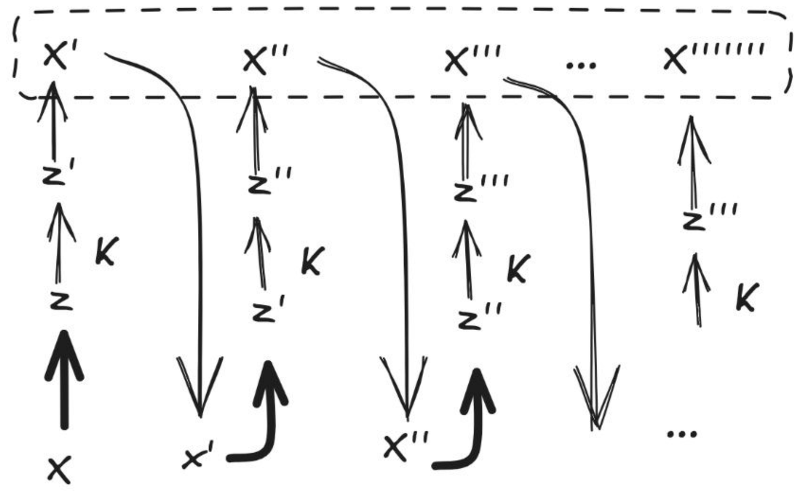
$$\mathcal{L}_{\text{Reconst}} = \sum_{i=0}^{T} \|x_{t+i} - \psi(z_{t+i})\|_2$$

The Reencoding Dilemma



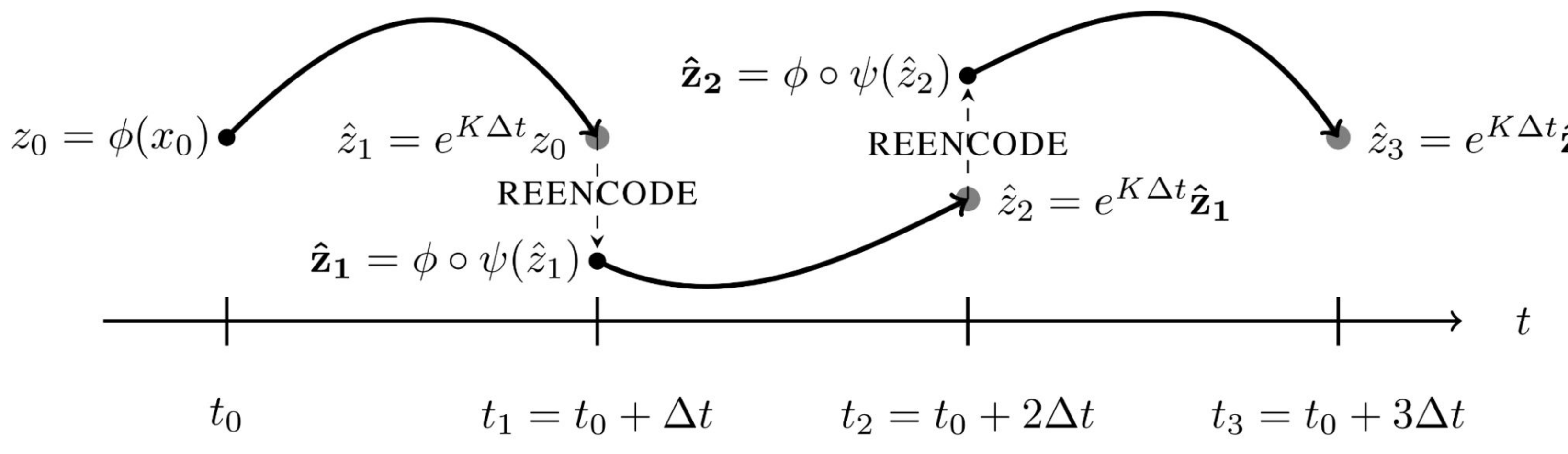
No Reencoding

- The latent trajectory is generated by repetitively hitting the first latent states by the Koopman matrix
- The system states are generated by decoding the entire latent trajectory, all at once.
- Unstable, but fast to run



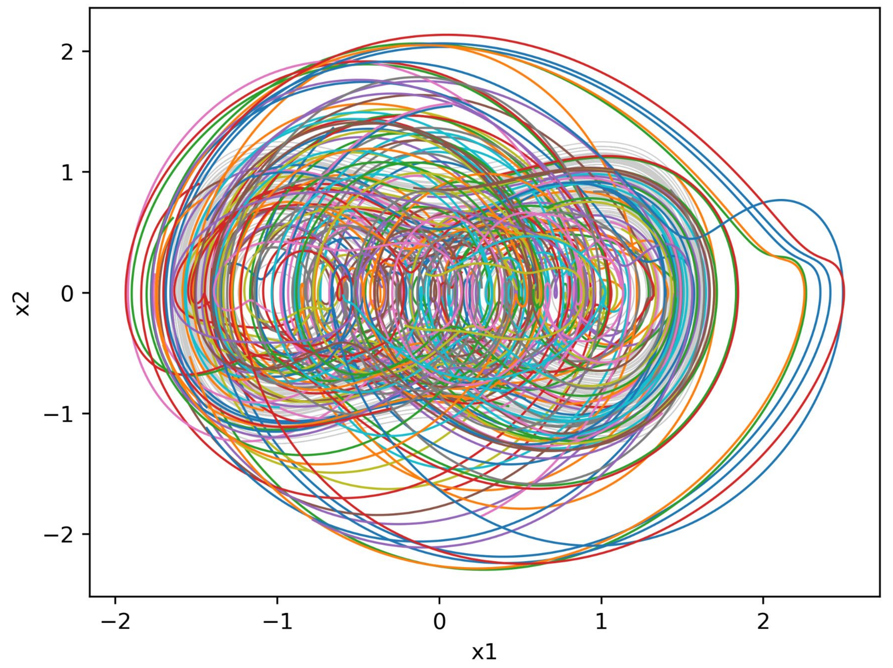
Reencoding every step

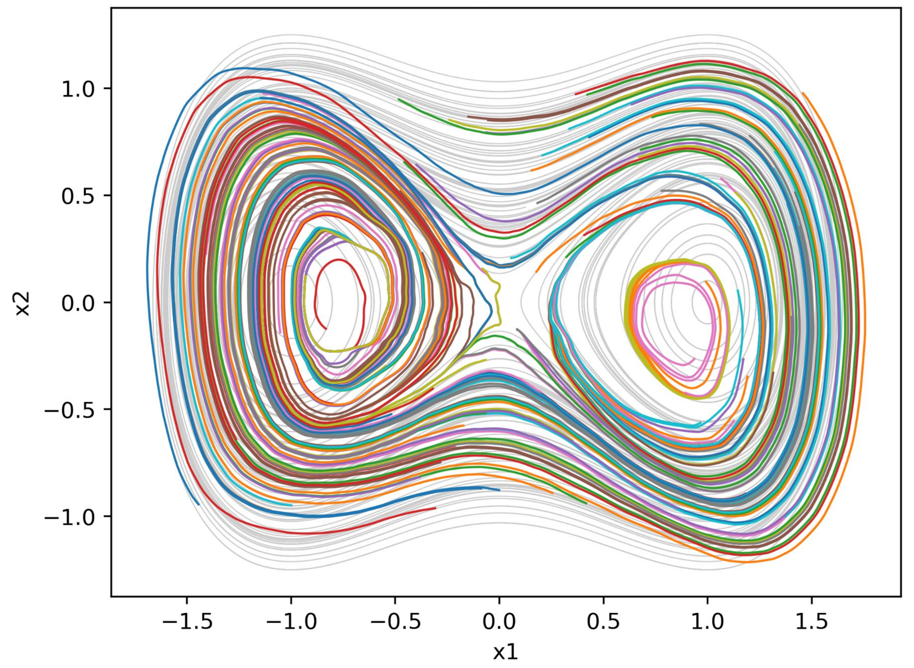
- The intermediate states are extracted from the intermediate **latent** states, on-the-fly, at ever step.
- Stable but slow to run

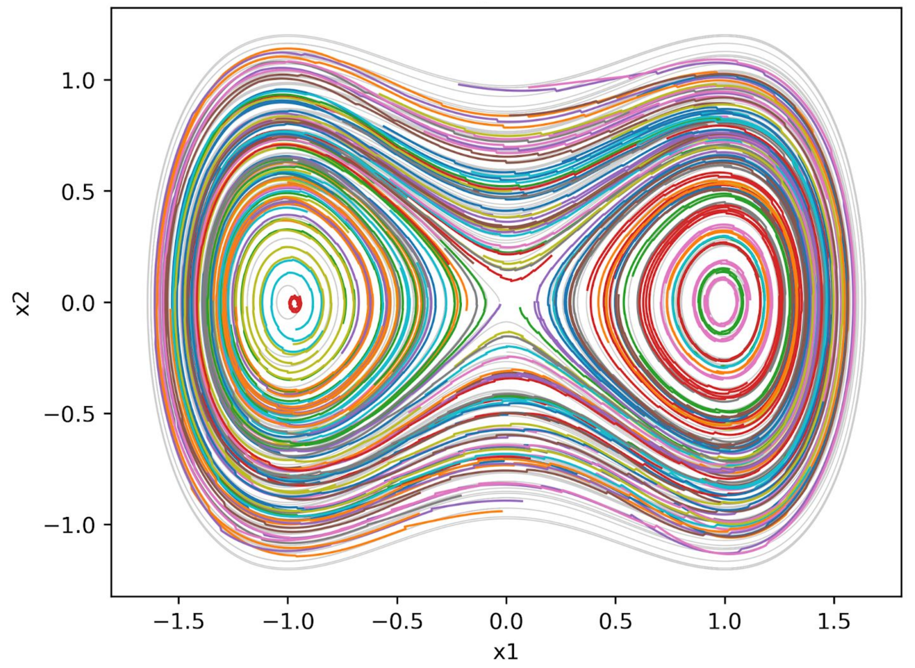


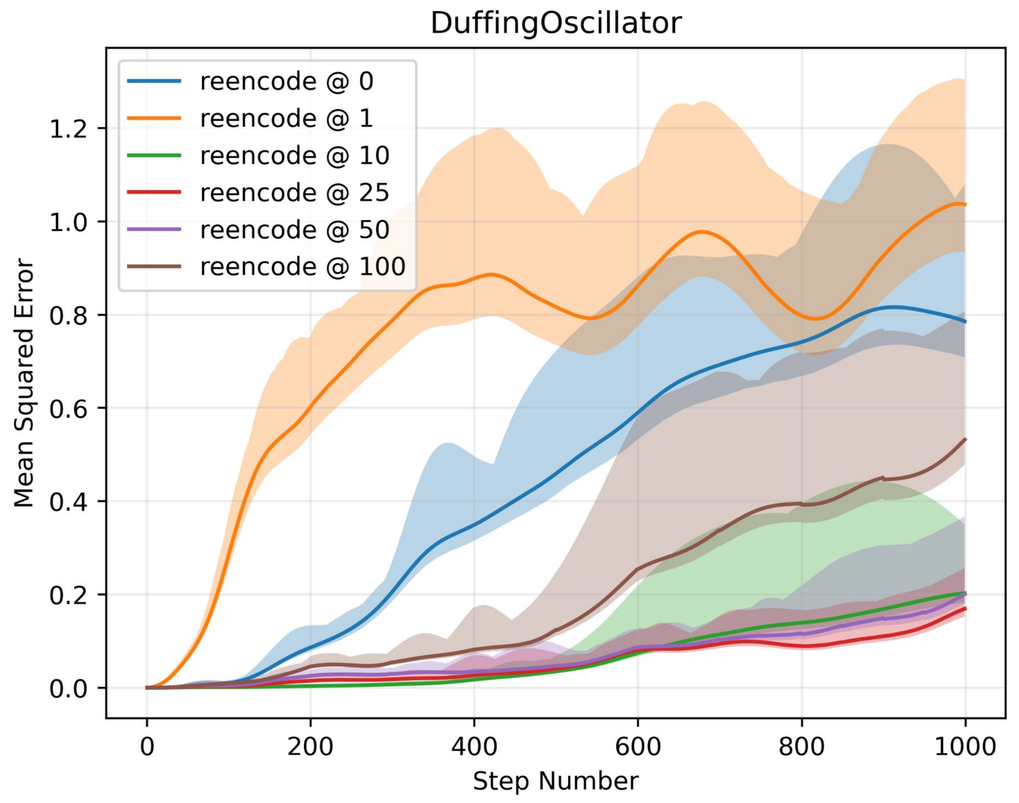
Periodic Reencoding

- Every so often, the latent state along the trajectory is decoded to the system state, and encoded back.
- Latent states are "course-corrected" periodically.
- Stable and fast to run, yay!









Open vs closed-loop and non-injective

$$\dot{z} = Kz|_{z_0=\phi(x_0)} \quad \text{and} \quad x = \psi(z)$$

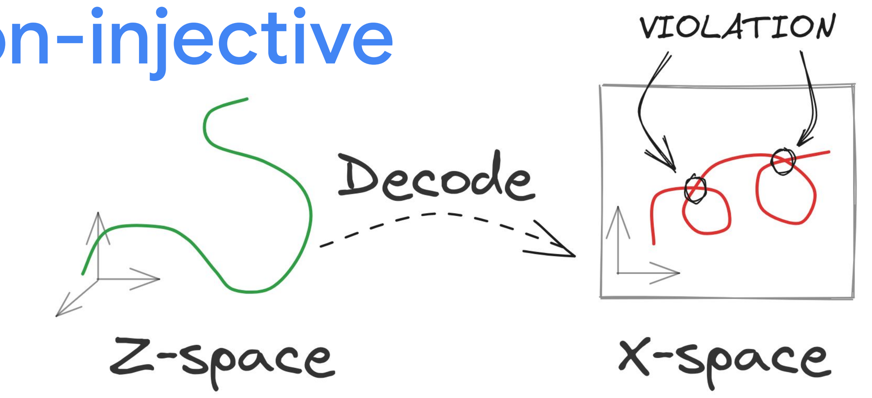
CURVE

$$x(t) = \psi(e^{Kt}\phi(x_0))$$

$$\dot{z} = Kz|_{z_0=\phi(x_0)}, \quad z = \phi(x), \quad \text{and} \quad x = \psi(z)$$

$$\dot{x} = J_\psi K\phi(x)$$

TRAJECTORY



Reencoding as a "Switch"

$$\mathbf{z} = [z_1, z_2, z_3, z_4]$$

$$\phi(x) = \{[z_1, z_2, 0, 0] \quad \text{if } \mathbf{x} \in R_1, \quad [0, 0, z_3, z_4] \quad \text{if } \mathbf{x} \in R_2\}$$

$$\dot{z} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} z$$

$$x = \left\{ z_1 \begin{bmatrix} | \\ w_1 \\ | \end{bmatrix} + z_2 \begin{bmatrix} | \\ w_2 \\ | \end{bmatrix} \text{if } x \in R_1, \quad z_3 \begin{bmatrix} | \\ w_3 \\ | \end{bmatrix} + z_4 \begin{bmatrix} | \\ w_4 \\ | \end{bmatrix} \text{if } x \in R_2 \right\}$$

(Lan & Mezić, 2013)

Results

Dynamical Systems

Model	Koopman (Linear Latent Dynamics)				NonLinear Latent Dynamics				MLP
Decoder Type	Linear		NonLinear		Linear		NonLinear		-
Periodic Reenc. (✗, ✓)	✗	✓	✗	✓	✗	✓	✗	✓	-
Environment	*MSE over **100** steps*								
Parabolic Attractor	**0.0205**	**0.0292**	0.1465	0.0758	0.0739	0.0496	0.0727	0.0547	0.2674
Pendulum	0.0512	<u>0.0042</u>	0.0648	0.0181	0.0288	**0.0025**	0.0242	0.0034	0.7442
Duffing Oscillator	0.1152	<u>0.0112</u>	0.1512	0.0512	0.0450	**0.0022**	0.0450	0.0112	0.4050
Lotka-Volterra	0.0113	<u>0.0072</u>	0.0145	0.0098	0.0128	**0.0040**	0.0112	0.0060	1.4450
Lorenz'63	✗	<u>11.162</u>	✗	12.569	18.985	**7.265**	19.051	**7.110**	88.565
Environment	*MSE over **1000** steps*								
Pendulum	9.2021	<u>0.1818</u>	14.5800	0.6612	10.7184	**0.0841**	10.5832	0.2964	55.281
Duffing Oscillator	20.5440	<u>1.0658</u>	15.0701	2.5312	5.7851	**0.5725**	9.2451	0.93845	22.445
Lotka-Volterra	1.6292	0.3961	1.6203	<u>0.2812</u>	0.7261	**0.2888**	0.7281	0.4324	83.205
Lorenz'63	✗	78.980	✗	<u>64.838</u>	✗	**59.262**	✗	**54.793**	133.509

Sparse Feedback D4RL

Model		Koopman Autoencoder		MLP	BC
State Feedback (✗, ✓)		✗	✗	✗	✓
Periodic Reencoding (✗, ✓)		✗	✓	-	-
Environment	Dataset	*total reward until termination*			
Hopper-v2	expert	18.40 ± 6.2	**53.5 ± 14.5**	18.9 ± 5.9	96.05 ± 4.3
HalfCheetah-v2	expert	15.06 ± 5.3	**64.2 ± 12.9**	19.5 ± 7.9	82.91 ± 5.9
Walker2d-v2	expert	18.46 ± 8.1	**61.9 ± 14.2**	11.4 ± 2.5	98.73 ± 1.5