A Additional Discussion

A.1 Contrastive feature learning vs. physics-informed optimal transport

Our two proposed approaches both aim to train neural operators to preserve the invariant measures of chaotic attractors. Ultimately, both methods have strong performance according to a variety of metrics, but the contrastive learning approach requires no prior physical knowledge and is often faster. Both approaches have significant advantages over the standard approach of minimizing RMSE, which fails to preserve important statistical characteristics of the system.

The primary conceptual difference between our approaches is that the optimal transport approach relies on prior domain knowledge, while the contrastive learning approach learns from the data alone. Specifically, the physics-informed optimal transport approach requires a choice of summary statistics, so it is much more dependent on our prior knowledge about the system and its chaotic attractor. Contrastive feature learning does not require a choice of statistics and instead learns useful invariant statistics on its own—although it may still be useful to have known relevant statistics to use as evaluation metrics for hyperparameter tuning.

The neural operators trained using the optimal transport loss perform the best on the L_1 histogram distance of the summary statistics $\mathbf{S}$ (Table 1 and 2), which is unsurprising given that the optimal transport loss specifically matches the distribution of $\mathbf{S}$. If we look at a different evaluation metric, e.g. the energy spectrum error or leading LE, the contrastive learning loss, in some cases, performs better even without prior knowledge. This suggests that contrastive feature learning may be capturing a wider range or different set of invariant statistics. We also see that the quality of the emulator degrades if we choose less informative statistics (Appendix B.3).

A.2 Interactions and trade-offs between short-term prediction and long-term statistics

Comparing our results in Tables 1 and 2, we find that emulators trained using our approaches perform significantly better on the long-term evaluation metrics (e.g. L_1 histogram distance and energy spectrum error) by trading off a bit of performance in terms of short-term RMSE (Tables 11 and 12). This suggests that training an emulator to model chaotic dynamics using purely RMSE can result in overfitting and generally a poor model of long-term dynamics. However, we still believe that RMSE is a useful part of the loss, even for long-term evaluation metrics, since it is the only term that is directly enforcing the short-term dynamics. In some cases, we find that the invariant statistics $\hat{S}_{\mathcal{A}}$ of trajectories generated by our trained neural operators are closer to the true statistics $S_{\mathcal{A}}$ than the statistics directly computed using the noisy training data $\tilde{S}_{\mathcal{A}}$, i.e., $|\hat{S}_{\mathcal{A}} - S_{\mathcal{A}}| < |\tilde{S}_{\mathcal{A}} - S_{\mathcal{A}}|$, despite being trained on the noisy data (Table 3). We attribute this result to the RMSE component of our losses, which provides additional regularization by enforcing the short-term dynamics, combined with the inductive biases of the neural operator architectures, which are designed for modeling PDEs.

r	Training	Histogram Error $\lvert\hat{S}_{\mathcal{A}} - S_{\mathcal{A}}\rvert \downarrow$	Histogram Error $\lvert\tilde{S}_{\mathcal{A}} - S_{\mathcal{A}}\rvert \downarrow$
0.1	ℓ_{RMSE}	0.056 (0.051, 0.062)	0.029 (0.023, 0.036)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.029** (0.027, 0.032)	0.029 (0.023, 0.036)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.033 (0.029, 0.037)	0.029 (0.023, 0.036)
0.2	ℓ_{RMSE}	0.130 (0.118, 0.142)	0.101 (0.086, 0.128)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.039** (0.035, 0.042)	0.101 (0.086, 0.128)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.073 (0.066, 0.080)	0.101 (0.086, 0.128)
0.3	ℓ_{RMSE}	0.215 (0.204, 0.234)	0.213 (0.190, 0.255)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.057** (0.052, 0.064)	0.213 (0.190, 0.255)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.132 (0.111, 0.151)	0.213 (0.190, 0.255)

Table 3: **Histogram error of neural operator predictions vs. noisy training data on Lorenz-96.** Averaging over 200 testing instances with varying $\phi^{(n)}$, we compared the L_1 histogram error of the predicted dynamics from the neural operator $|\hat{S}_{\mathcal{A}} - S_{\mathcal{A}}|$ with the histogram error of the raw noisy training data $|\tilde{S}_{\mathcal{A}} - S_{\mathcal{A}}|$. This shows that, even though the neural operator is trained on the noisy data, the statistics of the predicted dynamics are often better than those computed directly from the noisy data. Here, $S_{\mathcal{A}}$ is computed from the noiseless ground truth data.

B Additional Experiments and Evaluation Metrics

We have performed several additional experiments that act as points of comparison, help us better understand the behavior of our methods under a variety of conditions, and provide useful insights for future applications of our approaches.

B.1 Denoising with Gaussian blurring

Gaussian blurring, often used as a denoising technique for images, employs a Gaussian distribution to establish a convolution matrix that's applied to the original image. The fundamental idea involves substituting the noisy pixel with a weighted average of surrounding pixel values. A key hyperparameter in Gaussian blurring is the standard deviation of the Gaussian distribution. When the standard deviation approaches zero, it fundamentally indicates the absence of any blur. Under such circumstances, the Gaussian function collapses to a single point, leading to the elimination of the blur effect. In Table 4, we present the results from applying Gaussian blurring to noisy data during training based solely on RMSE. Despite the effectiveness of the widely adopted denoising approach, our findings indicate that Gaussian blurring may not be ideally suited for our purpose of emulating dynamics. This is primarily because significant invariant statistics might be strongly correlated with certain high-frequency signals that could be affected by the blurring preprocessing.

Training	Histogram Error ↓	Energy Spec. Error↓	Leading LE Error ↓
ℓ_{RMSE} ($\sigma_b = 0.1$)	**0.390** (0.326, 0.556)	**0.290** (0.226, 0.402)	**0.098** (0.069, 0.127)
ℓ_{RMSE} ($\sigma_b = 0.5$)	1.011 (0.788, 1.264)	0.493 (0.379, 0.623)	**0.098** (0.041, 0.427)
ℓ_{RMSE}	**0.390** (0.325, 0.556)	**0.290** (0.225, 0.402)	0.101 (0.069, 0.122)

Table 4: **Emulator performance with Gaussian blurring on Kuramoto–Sivashinsky data with noise scale** $r = 0.3$**.** Averaging over 200 testing instances with varying $\phi^{(n)}$, we show the performance of (1) the application of Gaussian blurring as a preliminary denoising effort with a small standard deviation ($\sigma_b = 0.1$); (2) the application of Gaussian blurring with a larger standard deviation ($\sigma_b = 0.5$); and (3) training purely on RMSE without any blurring preprocessing. The results suggest that the application of Gaussian blurring might further degrade the results, as the high-frequency signals associated with invariant statistics can be lost.

B.2 Sobolev norm baseline

We recognize that there are alternative methods that strive to capture high-frequency signals by modifying training objectives. For instance, the Sobolev norm, which combines data and its derivatives, has been found to be quite effective in capturing high-frequency signals [16, 49]. However, its effectiveness can be significantly curtailed in a noisy environment, especially when noise is introduced to a high-frequency domain, as minimizing the Sobolev norm then fails to accurately capture relevant statistics, as shown in Tables 5 and 6.

Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓	FD Error ↓
ℓ_{RMSE}	0.215 (0.204, 0.234)	0.291 (0.280, 0.305)	0.440 (0.425, 0.463)	3.580 (2.333, 4.866)
$\ell_{\mathrm{Sobolev}}+\ell_{\mathrm{dissaptive}}$	0.246 (0.235, 0.255)	0.325 (0.341, 0.307)	0.487 (0.456, 0.545)	4.602 (3.329, 6.327)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.057** (0.052, 0.064)	**0.123** (0.116, 0.135)	0.084 (0.062, 0.134)	3.453 (2.457, 4.782)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.132 (0.111, 0.151)	0.241 (0.208, 0.285)	**0.064** (0.045, 0.091)	**1.894** (0.942, 3.108)

Table 5: **Emulator performance (including Sobolev norm loss) on Lorenz-96 data with noise scale** $r = 0.3$**.** The median (25th, 75th percentile) of the evaluation metrics are computed on 200 Lorenz-96 test instances (each with 1500 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) Sobolev norm loss with dissipative regularization $\ell_{\mathrm{Sobolev}} + \ell_{\mathrm{dissaptive}}$; (3) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (4) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$.

Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓
ℓ_{RMSE}	0.390 (0.325, 0.556)	0.290 (0.225, 0.402)	0.101 (0.069, 0.122)
$\ell_{\mathrm{Sobolev}} + \ell_{\mathrm{dissipative}}$	0.427 (0.289, 0.616)	0.237 (0.204, 0.315)	**0.023** (0.012, 0.047)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.172** (0.146, 0.197)	0.211 (0.188, 0.250)	0.094 (0.041, 0.127)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.193 (0.148, 0.247)	**0.176** (0.130, 0.245)	0.108 (0.068, 0.132)

Table 6: **Emulator performance (including Sobolev norm loss) on Kuramoto–Sivashinsky data with noise scale** $r = 0.3$**.** The median (25th, 75th percentile) of the evaluation metrics are computed on 200 Kuramoto–Sivashinsky test instances (each with 1000 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) Sobolev norm loss with dissipative regularization $\ell_{\mathrm{Sobolev}} + \ell_{\mathrm{dissaptive}}$; (3) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (4) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$.

B.3 Optimal transport: reduced set of summary statistics

For our optimal transport approach, we test a reduced set of summary statistics, which shows how the quality of the summary statistic affects the performance of the method (Table 7). With an informative summary statistic, we find even a reduced set can still be helpful but, for a non-informative statistic, the optimal transport method fails as expected.

Training statistics	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓	FD Error ↓
$\mathbf{S}$ (full)	**0.057** (0.052, 0.064)	**0.123** (0.116, 0.135)	0.084 (0.062, 0.134)	3.453 (2.457, 4.782)
$\mathbf{S}_1$ (partial)	0.090 (0.084, 0.098)	0.198 (0.189, 0.208)	0.263 (0.217, 0.323)	3.992 (2.543, 5.440)
$\mathbf{S}_2$ (minimum)	0.221 (0.210, 0.234)	0.221 (0.210, 0.230)	0.276 (0.258, 0.291)	**3.204** (2.037, 4.679)

Table 7: **Emulator performance for different choices of summary statistics on Lorenz-96 data with noise scale** $r = 0.3$**.** Each neural operator was trained using the optimal transport and RMSE loss using (1) full statistics $\mathbf{S}(\mathbf{u}) := \{\frac{du_i}{dt}, (u_{i+1} - u_{i-2})u_{i-1}, u_i\}$; (2) partial statistics $\mathbf{S}_1(\mathbf{u}) := \{(u_{i+1} - u_{i-2})u_{i-1}\}$; or (3) minimum statistics $\mathbf{S}_2(\mathbf{u}) := \{\bar{\mathbf{u}}\}$, where $\bar{\mathbf{u}}$ is the spatial average.

B.4 Contrastive learning: reduced environment diversity

For our contrastive learning approach, we test a multi-environment setting with reduced data diversity and find that the contrastive method still performs well under the reduced conditions (Table 8), which demonstrates robustness.

Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓	FD Error ↓
ℓ_{RMSE}	0.255 (0.248, 0.263)	0.307 (0.302, 0.315)	0.459 (0.743, 2.746)	3.879 (2.456, 5.076)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.055** (0.050, 0.061)	**0.124** (0.116, 0.131)	0.080 (0.045, 0.109)	4.015 (2.401, 5.225)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.130 (0.111, 0.152)	0.193 (0.183, 0.200)	**0.031** (0.014, 0.053)	**1.747** (0.792, 2.939)

Table 8: **Emulator performance with reduced environment diversity (i.e. narrower parameter range) on Lorenz-96 data with noise level** $r = 0.3$**.** Averaging over 200 testing instances, we show the performance of training the neural operator with (1) only RMSE loss ℓ_{RMSE}; (2) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (3) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$. We shrink the parameter range for generating the dataset from $[10, 18]$ to $[16, 18]$.

B.5 Maximum mean discrepancy (MMD) vs. optimal transport loss

We also implement a variant of our optimal transport approach that uses maximum mean discrepancy (MMD) as a distributional distance rather than the Sinkhorn divergence. Using the same set of summary statistics, we find that MMD does not perform as well as our optimal transport loss for training emulators (Table 9).

Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓
ℓ_{RMSE}	0.390 (0.325, 0.556)	0.290 (0.225, 0.402)	0.101 (0.069, 0.122)
$\ell_{\mathrm{MMD}} + \ell_{\mathrm{RMSE}}$	0.245 (0.218, 0.334)	0.216 (0.186, 0.272)	0.101 (0.058, 0.125)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.172** (0.146, 0.197)	0.211 (0.188, 0.250)	**0.094** (0.041, 0.127)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.193 (0.148, 0.247)	**0.176** (0.130, 0.245)	0.108 (0.068, 0.132)

Table 9: **Emulator performance (including MMD loss) on Kuramoto–Sivashinsky data with noise scale $r = 0.3$.** The median (25th, 75th percentile) of the evaluation metrics are computed on 200 Kuramoto–Sivashinsky test instances (each with 1000 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) maximum mean discrepency (MMD) and RMSE loss $\ell_{\mathrm{MMD}} + \ell_{\mathrm{RMSE}}$; (3) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (4) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$.

B.6 Additional Lyapunov spectrum evaluation metrics

In the table 10, we evaluated the results of Lorenz 96 on Lyapunov spectrum error rates and the total number of positive Lyapunov exponents error rates. For the Lyapunov spectrum error, we report the sum of relative absolute errors across the full spectrum: $\sum_i^d |\hat{\lambda_i} - \lambda_i|/\lambda_i$, where λ_i is the i-th Lyapunov exponent and d is the dimension of the dynamical state. As suggested by [50], we also compare the number of positive Lyapunov exponents (LEs) as an additional statistic to measure the complexity of the chaotic dynamics. We compute the absolute error in the number of positive LEs $\sum_i^d |\mathbf{1}(\hat{\lambda_i} > 0 - \mathbf{1}(\lambda_i > 0)|$.

r	Training	Leading LE Error ↓	Lyapunov Spectrum Error ↓	Total number of positive LEs Error ↓
0.1	ℓ_{RMSE}	**0.013** (0.006, 0.021)	0.265 (0.110, 0.309)	0.500 (0.000, 1.000)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.050 (0.040, 0.059)	0.248 (0.168, 0.285)	**0.000** (0.000, 1.000)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.065 (0.058, 0.073)	**0.227** (0.164, 0.289)	**0.000** (0.000, 1.000)
0.2	ℓ_{RMSE}	0.170 (0.156, 0.191)	0.612 (0.522, 0.727)	4.000 (4.000, 5.000)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.016 (0.006, 0.030)	0.513 (0.122, 0.590)	**3.000** (2.000, 3.000)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	**0.012** (0.006, 0.018)	**0.459** (0.138, 0.568)	**3.000** (2.000, 3.000)
0.3	ℓ_{RMSE}	0.440 (0.425, 0.463)	0.760 (0.702, 0.939)	7.000 (7.000, 8.000)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.084 (0.062, 0.134)	**0.661** (0.572, 0.746)	**5.000** (4.000, 6.000)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	**0.064** (0.045, 0.091)	0.654 (0.558, 0.780)	6.000 (5.000, 6.000)

Table 10: **Emulator performance on Lyapunov spectrum metrics for Lorenz-96 data.** The median (25th, 75th percentile) of the Lyapunov spectrum metrics are computed on 200 Lorenz-96 test instances (each with 1500 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (3) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$. In the presence of high noise, OT and CL give lower relative errors on the leading Lyapunov exponent (LE). When evaluating the full Lyapunov spectrum, OT and CL show significant advantages than the baseline. In addition, the lower absolute errors of the total number of the positive Lyapunov exponents (LEs) suggest that OT and CL are able to match the complexity of the true chaotic dynamics.

B.7 1-step RMSE evaluation results

Evaluating on 1-step RMSE only shows short-term prediction performance and is not an informative evaluation metric for long-term behavior. Here, we report the 1-step RMSE (Tables 11 and 12) to show that training using our approaches retains similar 1-step RMSE results while significantly improving on long-term statistical metrics (Tables 1 and 2). See Appendix A.2 for additional discussion.

r	Training	RMSE ↓
0.1	ℓ_{RMSE}	0.107 (0.105, 0.109)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.108 (0.105, 0.110)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.109 (0.108, 0.113)
0.2	ℓ_{RMSE}	0.202 (0.197, 0.207)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.207 (0.202, 0.212)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.214 (0.203, 0.218)
0.3	ℓ_{RMSE}	0.288 (0.282, 0.296)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.301 (0.293, 0.307)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.312 (0.302,0.316)

Table 11: **1-step RMSE performance on Lorenz-96 data.** The median (25th, 75th percentile) of the Lyapunov spectrum metrics are computed on 200 Lorenz-96 test instances (each with 1500 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (3) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$. We see comparable performance on short-term 1-step RMSE.

Training	RMSE ↓
ℓ_{RMSE}	0.373 (0.336, 0.421)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.381 (0.344, 0.430)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.402 (0.364, 0.452)

Table 12: **1-step RMSE performance on Kuramoto–Sivashinsky data with noise scale $r = 0.3$.** The median (25th, 75th percentile) of the evaluation metrics are computed on 200 Kuramoto–Sivashinsky test instances (each with 1000 time steps) for the neural operator trained with (1) only RMSE loss ℓ_{RMSE}; (2) optimal transport (OT) and RMSE loss $\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$; and (3) contrastive learning (CL) and RMSE loss $\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$. We again see comparable performance on short-term 1-step RMSE.

C Implementation Details

C.1 Evaluation metrics

To evaluate the long-term statistical behavior of our trained neural operators, we run the neural operator in a recurrent fashion for 1500 (Lorenz-96) or 1000 (Kuramoto–Sivashinsky) time steps and then compute our long-term evaluation metrics on this autonomously generated time-series.

Histogram error. For distributional distance with a pre-specified statistics $\mathbf{S}$, we first compute the histogram of the invariant statistics $\mathbf{H}(\mathbf{S}) = \{(\mathbf{S}_1, c_1), (\mathbf{S}_2, c_2), \ldots, (\mathbf{S}_B, c_B)\}$, where $\mathbf{H}$ represents the histogram, and the values of the bins are denoted by $\mathbf{S}_b$ with their corresponding frequencies c_b. We then define the L_1 histogram error as:

$$\mathrm{Err}(\hat{\mathbf{H}}, \mathbf{H}) := \sum_{b=1}^{B} \|c_b - \hat{c}_b\|_1. \tag{21}$$

Note that for a fair comparison across all our experiments, we use the rule of thumb—the square root rule to decide the number of bins.

Energy spectrum error. We compute the relative mean absolute error of the energy spectrum—the squared norm of the spatial FFT $\mathcal{F}[\mathbf{u}_t]$—averaged over time:

$$\frac{1}{T} \sum_{\mathbf{u}_t, \hat{\mathbf{u}}_t \in \mathbf{U}_{1:T}, \hat{\mathbf{U}}_{1:T}} \frac{\| |\mathcal{F}[\mathbf{u}_t]|^2 - |\mathcal{F}[\hat{\mathbf{u}}_t]|^2 \|_1}{\| |\mathcal{F}[\mathbf{u}_t]|^2 \|_1}. \tag{22}$$

Leading LE error. The leading Lyapunov exponent (LE) is a dynamical invariant that measures how quickly the chaotic system becomes unpredictable. For the leading LE error, we report the relative absolute error $|\hat{\lambda} - \lambda|/|\lambda|$ between the model and the ground truth averaged over the test set. We adapted the Julia DynamicalSystem.jl package to calculate the leading LE.

FD error. The fractal dimension (FD) is a characterization of the dimension of the attractor. We report the absolute error $|\hat{D} - D|$ between the estimated FD of the model and the ground truth averaged over the test set. We use the Julia DynamicalSystem.jl package for calculating the fractal dimension.

RMSE. We use 1-step relative RMSE $\|\mathbf{u}_{t+\Delta t} - \hat{g}_\theta(\mathbf{u}_t, \phi)\|_2 / \|\mathbf{u}_{t+\Delta t}\|_2$ to measure short-term prediction accuracy.

C.2 Time complexity

The optimal transport approach relies on the Sinkhorn algorithm which scales as $\mathcal{O}(n^2 \log n)$ for comparing two distributions of n points each (Theorem 2 in [51]). In our experiments, we use $n = 6000$ to $n = 25600$ points with no issues, so this approach scales relatively well. The contrastive learning approach requires pretraining but is even faster during emulator training since it uses a fixed, pre-trained embedding network.

	Training	Encoder	Operator	Total
Lorenz-96	ℓ_{RMSE}	–	20	20
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	–	51	51
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	22	27	49
Kuramoto–Sivashinsky	ℓ_{RMSE}	–	55	55
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	–	262	262
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	26	56	82

Table 13: **Empirical training time.** Training time (minutes) with 4 GPUs for 60-dimensional Lorenz-96 and 256-dimensional Kuramoto–Sivashinsky.

C.3 Training of the encoder

Evaluation rule. A standard procedure for assessing the performance of an encoder trained with Noise Contrastive Estimation (InfoNCE) loss in an unsupervised manner [32–34], is employing the top-1 accuracy metric. This measures how effectively similar items are positioned closer to each other compared to dissimilar ones in the embedded space. While this evaluation measure is commonly employed in downstream tasks that focus on classification, it is suitably in line with our goals. We aim to learn an encoder that can differentiate whether sequences of trajectories are sampled from different attractors, each characterized by distinct invariant statistics. Therefore, the use of Top-1 accuracy to evaluate if two sequences originate from the same trajectory serves our purpose effectively. And we utilize this metric during our training evaluations to assess the performance of our encoder.

The formal definition of Top-1 accuracy requires us to initially define the softmax function σ, which is used to estimate the likelihood that two samples from different time windows originate from the same trajectory,

$$\sigma\left(f_\psi(\mathbf{U}^{(j)}_{J:J+K}); f_\psi(\mathbf{U}^{(n)}_{I:I+K})\right) = \frac{\exp\left(\langle f_\psi(\mathbf{U}^{(n)}_{I:I+K}), f_\psi(\mathbf{U}^{(j)}_{J:J+K})\rangle\right)}{\sum_{m=1}^{N}\left[\exp\left(\langle f_\psi(\mathbf{U}^{(n)}_{I:I+K}), f_\psi(\mathbf{U}^{(m)}_{H:H+K})\rangle\right)\right]}. \tag{23}$$

Using the softmax function to transform the encoder's output into a probability distribution, we then define Top-1 accuracy as:

$$\mathrm{Acc}_1 = \frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left\{n = \arg\max_j \sigma\left(f_\psi(\mathbf{U}^{(j)}_{J:J+K}); f_\psi(\mathbf{U}^{(n)}_{I:I+K})\right)\right\}, \tag{24}$$

where $\mathbf{1}(\cdot)$ is the indicator function representing whether two most close samples in the feature space comes from the same trajectory or not.

Training hyperparameters. We use the ResNet-34 as the backbone of the encoder, throughout all experiments, we train the encoder using the AdamW optimization algorithm [52], with a weight decay of 10^{-5}, and set the training duration to 2000 epochs.

For the temperature value τ balancing the weights of difficult and easy-to-distinguish samples in contrastive learning, we use the same warm-up strategy as in [10]. Initially, we start with a relatively low τ value (0.3 in our experiments) for the first 1000 epochs. This ensures that samples that are difficult to distinguish get large gradients. Subsequently, we incrementally increase τ up to a specified value (0.7 in our case), promoting the grouping of similar samples within the embedded space. From our empirical observations, we have found that this approach leads to an improvement in Top-1 accuracy in our experiments.

The length of the sequence directly influences the Top-1 accuracy. Considering that the sample length L of the training data $\{\mathbf{U}_{0:L}^{(n)}\}_{n=1}^{N}$ is finite, excessively increasing the crop length K can have both advantages and disadvantages. On one hand, it enables the encoder to encapsulate more information; on the other hand, it could lead to the failure of the encoder's training. This is likely to occur if two samples, i.e., $f_\psi(\mathbf{U}_{I:I+K}^{(n)})$ and $f_\psi(\mathbf{U}_{J:J+K}^{(n)})$, from the same trajectory overlap excessively, inhibiting the encoder from learning a meaningful feature space. With this consideration, we've empirically chosen the length K of subsequences for the encoder to handle to be approximately 5% of the total length T.

C.4 Lorenz-96

Data generation. To better align with realistic scenarios, we generate our training data with random initial conditions drawn from a normal distribution. For the purpose of multi-environment learning, we generate 2000 trajectories for training. Each of these trajectories has a value of $\phi^{(n)}$ randomly sampled from a uniform distribution ranging from 10.0 to 18.0. We discretize these training trajectories into time steps of $dt = 0.1$ over a total time of $t = 205s$, yielding 2050 discretized time steps. Moreover, in line with the setup used in [2, 11], we discard the initial 50 steps, which represent the states during an initial transient period.

Training hyperparameters. We determine the roll-out step during training (i.e., h and h_{RMSE}) via the grid search from the set of values $\{1, 2, 3, 4, 5\}$. Though it is shown in some cases that a larger roll-out number help improve the results [18], we have not observed that in our training. We hypothesized that this discrepancy may be due to our dataset being more chaotic compared to typical cases. Consequently, to ensure a fair comparison and optimal outcomes across all experiments, we have decided to set h and h_{RMSE} to 1.

In the case of the optimal transport loss[2], described in Section 3.1, the blur parameter γ governs the equilibrium between faster convergence speed and the precision of the Wasserstein distance approximation. We determined the value of γ through a grid search, examining values ranging from $\{0.01, 0.02, \ldots, 0.20\}$. Similarly, we decided the weights of the optimal transport loss α through a grid search, exploring values from $\{0.5, 1, 1.5, \ldots, 3.0\}$. For the experiments conducted using Lorenz-96, we selected $\alpha = 3$ and $\gamma = 0.02$.

In the context of the feature loss, the primary hyperparameter we need to consider is its weights, represented as λ. Given that we do not have knowledge of the invariant statistics $\mathbf{S}$ during the validation phase, we adjust λ according to a specific principle: our aim is to reduce the feature loss ℓ_{feature} (as defined in Eqn. 17) as long as it does not adversely affect the RMSE beyond a predetermined level (for instance, 10%) when compared with the baseline model, which is exclusively trained on RMSE. As illustrated in Figure 4, we adjust the values of λ in a systematic grid search from $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$. From our observations during the validation phase, we noted that the feature loss was at its lowest with an acceptable RMSE (which is lower than 110% compared to the baseline) when $\lambda = 0.8$. Therefore, we have reported our results with λ set at 0.8.

Results visualization. We present more results with varying noise levels in Figures 5, 6, 7.

[2] https://www.kernel-operations.io/geomloss/

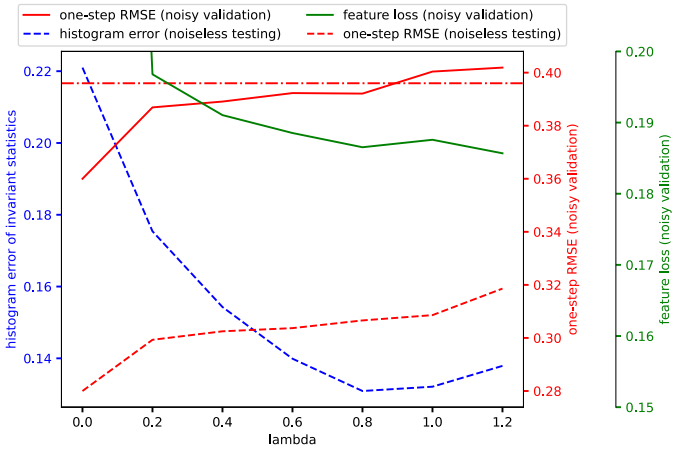
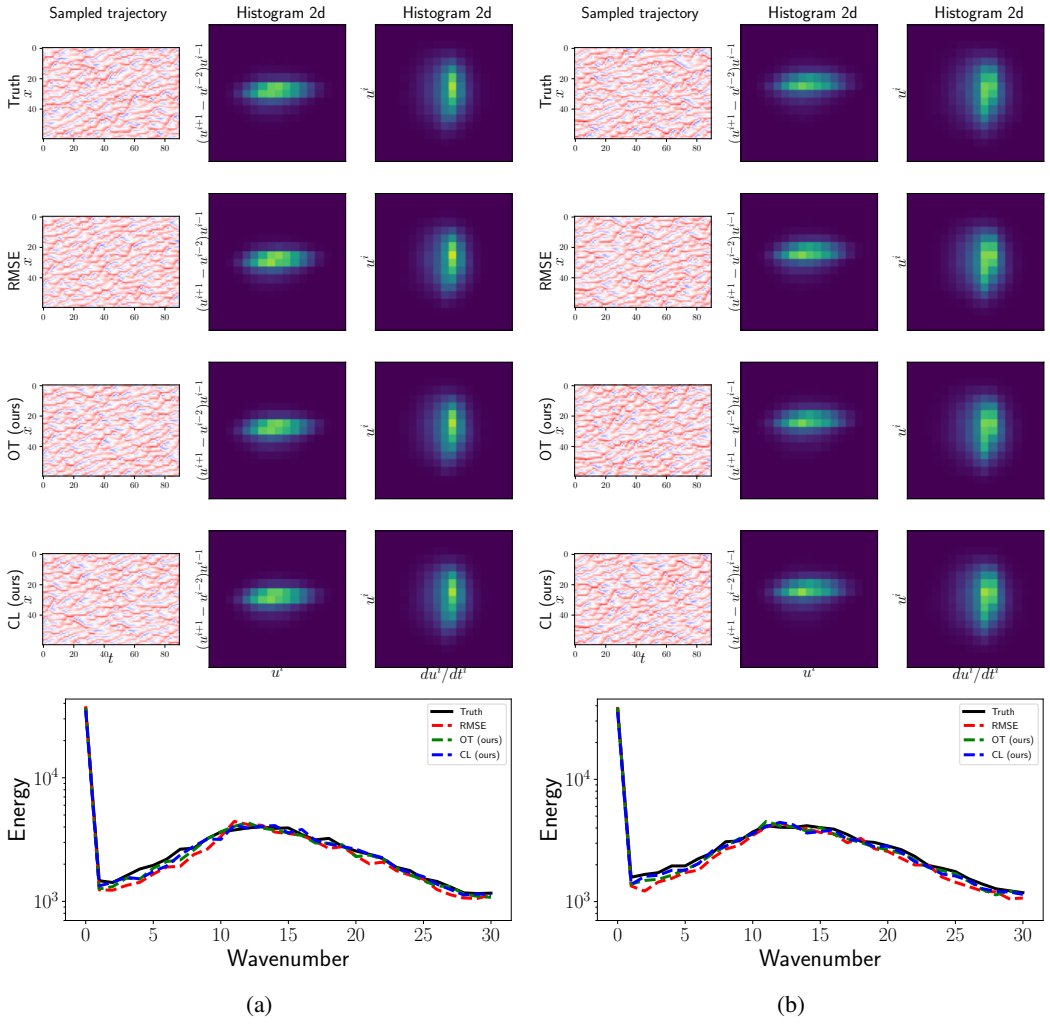


Figure 4: **The trend of feature loss with its weight λ when the scale of noise is $r = 0.3$.** The solid lines in the figure represent the evaluation metrics during the validation phase, comparing the outputs of the neural operator to the noisy data. In contrast, the dashed lines represent the actual metrics we are interested in, comparing the outputs of the neural operator to the clean data and calculating the error of the invariant statistics. In addition, the horizontal solid dashed line correspond to the bar we set for the RMSE, i.e., 110% of the RMSE when $\lambda = 0$. We observe that, (1) with the increase of λ from 0, the feature loss decreases until λ reaches 1.0. (2) The RMSE generally increases with the increase of λ. (3) The unseen statistical error generally decreases with the increase of λ. We reported the results when $\lambda = 0.8$ as our final result, since the further increase of λ does not bring further benefit in decreasing the feature loss, and the result remains in an acceptable range in terms of RMSE.

C.5 Kuramoto–Sivashinsky

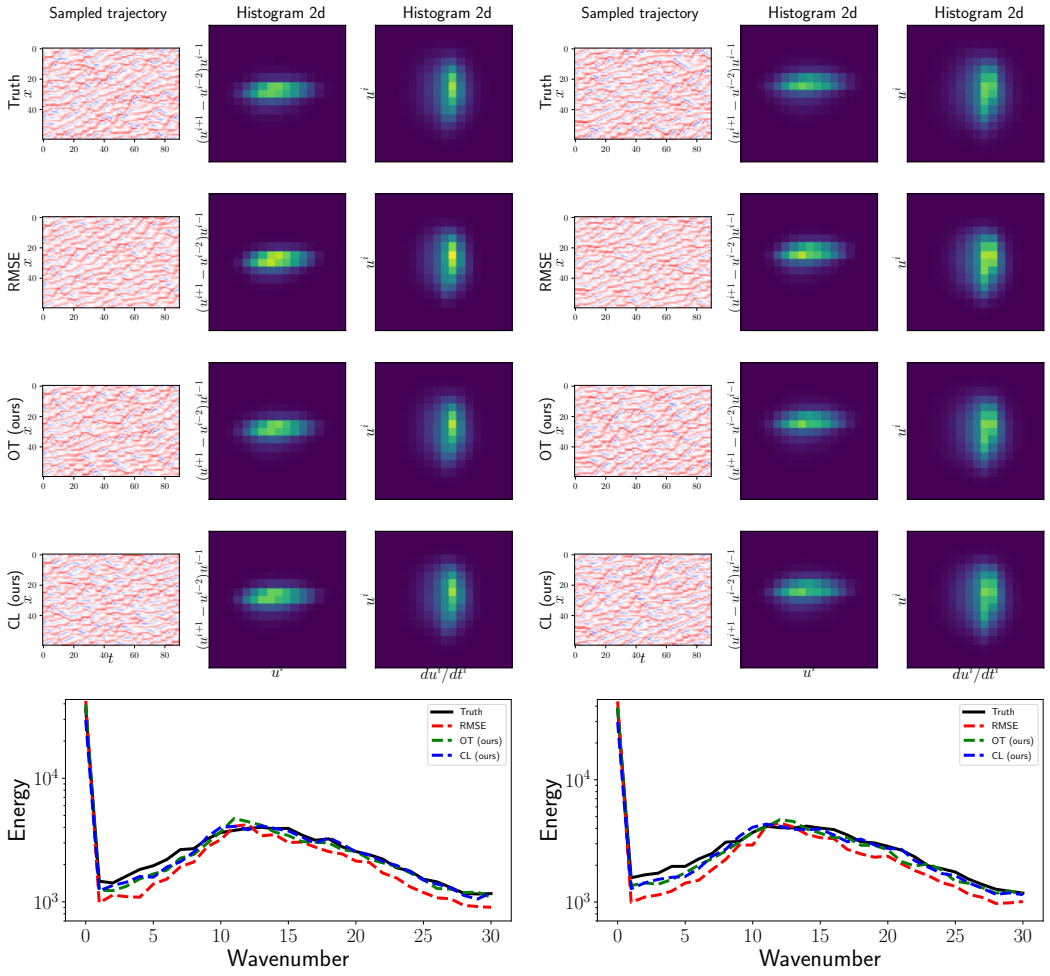
Data generation. In order to ascertain that we are operating within a chaotic regime, we set the domain size $L = 50$ and the spatial discretization grids number $d = 256$. We select initial conditions randomly from a uniform range between $[-\pi, \pi]$. A fourth-order Runge-Kutta method was utilized to perform all simulations. We generate 2000 trajectories for training, with each $\phi^{(n)}$ being randomly chosen from a uniform distribution within the interval $[1.0, 2.6]$.

Training hyperparameters. In the case of the optimal transport loss, similar to the discussion in C.4, we search the blur value γ and the weights of loss α from a grid search. And in our experiment, we set $\alpha = 3$ and $\gamma = 0.05$. In the case of the feature loss, we again adopt the same rule for deciding the value of λ, where we compare the trend of feature loss with the relative change of RMSE of the baseline and choose to report the results with the lowest feature loss and acceptable RMSE increase (110% compared to the baseline when $\lambda = 0$).



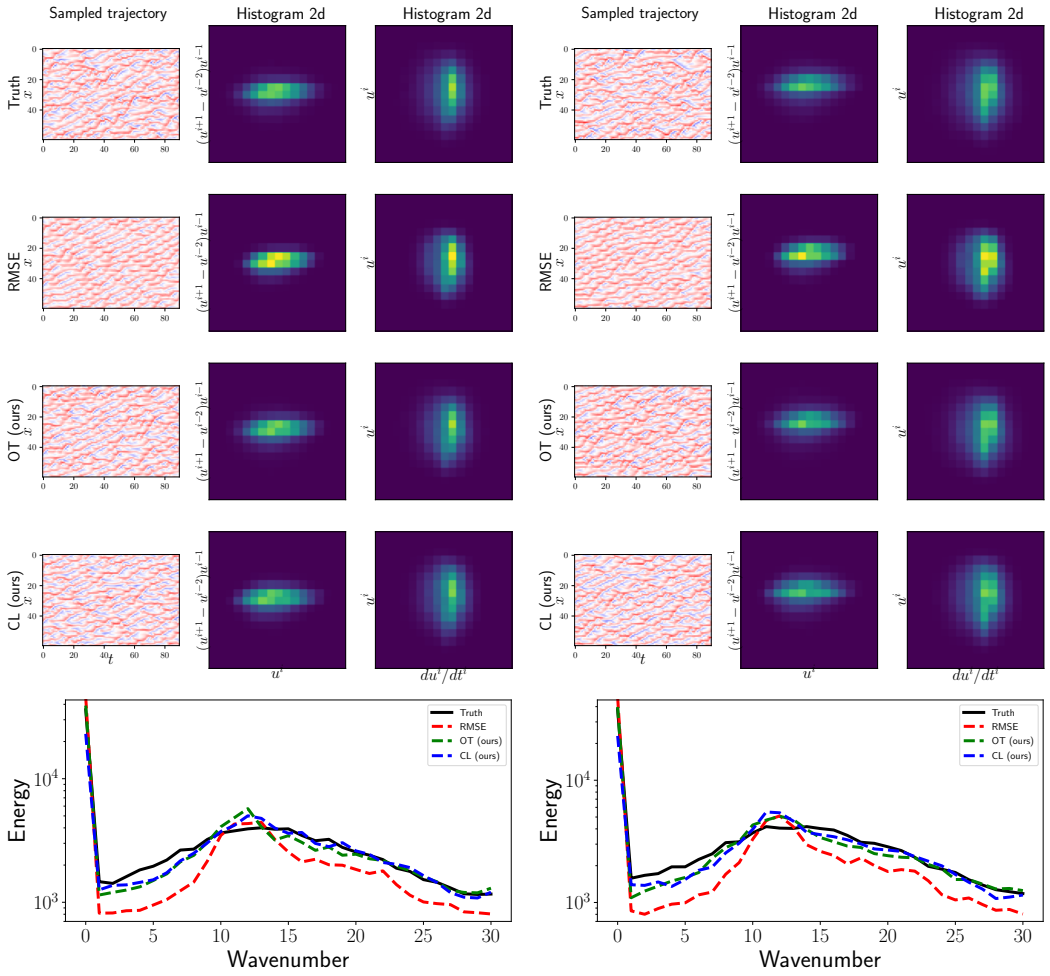
(a) (b)

Figure 5: **Visualization of the predictions when the noise Level** $r = 0.1$**.** We evaluate our method by comparing them to the baseline that is trained solely using RMSE. For two different instances (a) and (b), we visualize the visual comparison of the predicted dynamics (left), two-dimensional histograms of relevant statistics (middle and right). We notice that, with the minimal noise, the predictions obtained from all methods look statistically consistent to the true dynamics.



(a) (b)

Figure 6: **Visualization of the predictions when the noise level** $r = 0.2$**.** We evaluate our method by comparing them to the baseline that is trained solely using RMSE. For two different instances (a) and (b), we visualize the visual comparison of the predicted dynamics (left), two-dimensional histograms of relevant statistics (middle and right). We observe that as the noise level escalates, the degradation of performance in the results of RMSE is more rapid compared to our method, which employs optimal transport and feature loss during training.



(a) (b)

Figure 7: **Visualization of the predictions when the noise level** $r = 0.3$**.** We evaluate our method by comparing them to the baseline that is trained solely using RMSE. For two different instances (a) and (b), we visualize the visual comparison of the predicted dynamics (left), two-dimensional histograms of relevant statistics (middle and right). We find that, under higher levels of noise, the RMSE results exhibit fewer negative values, as indicated by the blue stripes in the predicted dynamics. This is further confirmed by the energy spectrum, which clearly shows that the RMSE results are significantly deficient in capturing high-frequency signals.