

ViT-Koop: Vision-Transformer–Koopman Operators for Efficient Time-Series Forecasting of Earth-Observation Data

Takayuki Shinohara, Hidetaka Saomoto
National Institute of Advanced Industrial Science and Technology
Tsukuba Japan

shinohara.takayuki@aist.go.jp, h-saomoto@aist.go.jp

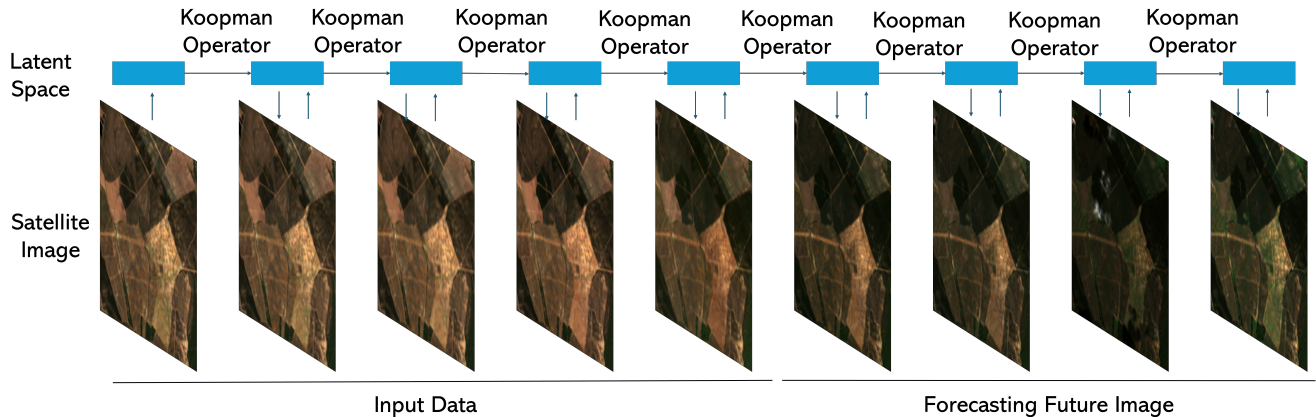


Figure 1. Example Satellite observation sequence from the EarthNet2021 dataset. The observation reflectance is mapped to pixel value of the range 0-255.

Abstract

Transformers can model the complex spatiotemporal dependencies present in satellite imagery, yet their quadratic computational cost limits real time, large scale applications such as climate monitoring and disaster response. We introduce ViTKoop, a lightweight framework that combines a Vision Transformer based autoencoder with a linear Koopman operator. The autoencoder compresses each image sequence into a compact latent state, and the Koopman operator advances this state linearly in time, greatly reducing computational complexity without sacrificing fidelity. On three benchmarks (ENSO, SEVIR, and EarthNet2021), ViTKoop matches or surpasses state of the art Transformer baselines while requiring only a small fraction of their floating point operations. This efficiency enables real time, high resolution forecasting on modest hardware and supports timely weather prediction as well as rapid, energy efficient Earth observation services that are vital for sustainable development.

1. Introduction

The escalating pace of climate change and alterations in environmental systems is resulting in more frequent and severe natural events such as droughts, wildfires, and hurri-

canes. These crises affect not only the environment but also social and economic systems, leading to disruptions in agriculture, loss of biodiversity, and extensive damage to infrastructure. In this critical context, computer vision technology has emerged as a powerful tool for social good, enabling more accurate and timely environmental forecasting. By harnessing high-resolution satellite imagery, computer vision can provide precise, global-scale insights into atmospheric, terrestrial, and oceanic dynamics. Traditional weather and climate forecasting models, which primarily rely on numerical simulations, often underutilize the rich visual data from modern Earth observation systems [16, 46]. However, advances in deep learning offer new opportunities to enhance data-driven forecasting methods [36, 38]. Developing robust computer vision models for predicting changes in satellite imagery is crucial for strengthening community resilience, improving disaster preparedness, and promoting sustainable environmental management in the face of accelerating global changes.

Deep learning (DL) has introduced powerful data-driven methods for satellite image forecasting, moving beyond explicit physics-based models [38, 42]. By learning directly from large-scale Earth observation data, DL models can often outperform traditional approaches [11], achieving strong results in tasks like precipitation nowcasting [9, 36] and ENSO prediction [19]. Yet, the chaotic,

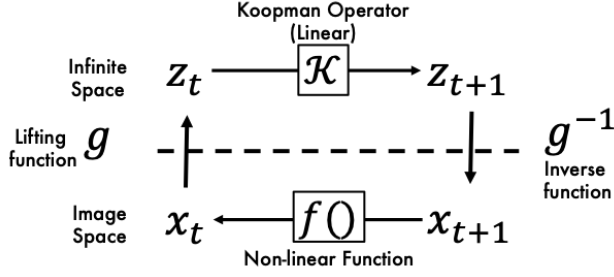


Figure 2. The Koopman operator framework. The top path shows the linear evolution of states in the infinite-dimensional space ($z_t \rightarrow z_{t+1}$) through the Koopman operator $\mathcal{K}$. The bottom path depicts the nonlinear dynamics of the system in the original state in image space ($x_t \rightarrow x_{t+1}$) via the nonlinear function $f(\cdot)$. The lifting function g maps the original state to the infinite-dimensional uplift space, while the g^{-1} performs the inverse mapping.

high-dimensional, and spatiotemporal nature of Earth systems presents major challenges. Earlier works combined RNNs and CNNs [18, 42, 43, 46, 49], while more recent Transformer-based models excel at capturing long-range dependencies for environmental forecasting [3, 13, 28]. Neural operator approaches like Fourier Neural Operator (FNO) [26, 34] are effective for PDE-governed physical systems but face limitations for optical satellite imagery, where smoothness, periodicity, and complete grids rarely hold. Real satellite data often include missing values from clouds and shadows, irregular sampling, and abrupt surface changes (e.g., wildfires, urbanization), leading to error accumulation and instability for FNO-based models. Our work thus focuses on architectures designed to robustly address these real-world challenges.

Despite yielding promising results, these Transformer-based methods face substantial computational challenges that limit their practical applicability. To address these limitations while maintaining forecasting accuracy, we propose integrating Physics-Constrained Learning (PCL) with efficient deep learning architectures. PCL algorithms embed physical consistency into vision models, thereby enhancing both interpretability and forecast accuracy while requiring fewer computational resources. The key insight behind our approach is that forecasting future states of satellite image sequences requires precise modeling of the underlying nonlinear dynamical systems. Koopman operator theory [21] provides an elegant mathematical framework that represents nonlinear dynamics via an infinite-dimensional linear operator, enabling more efficient modeling of stability analysis [31] and control applications [17, 35]. In practice, finite-dimensional approximations of the Koopman operator are necessary. Our deep learning approach addresses this by constructing a finite-dimensional Koopman-invariant subspace using Auto Encoder networks specifically designed for visual data (Fig. 1). These networks project high-dimensional satellite imagery into a latent space where the

dynamics can be approximated linearly through a finite-dimensional Koopman operator, implemented as a computationally efficient linear layer (Fig. 2). In this paper, we introduce a novel Koopman Operator-based Vision Transformer (ViTKoop) framework that leverages the computational efficiency of Koopman theory, specifically tailored to the challenges of satellite imagery forecasting.

2. Related Work

Deep Learning for Satellite Image Forecasting. Vision-based deep learning for satellite forecasting has evolved significantly, from U-Net architectures applied to precipitation nowcasting [46] and ENSO forecasting [19], to the integration of spatiotemporal dynamics with ConvLSTM [42]. Architectural innovations like PredRNN [49] with its spatiotemporal memory flow and PhyDNet [18] with PDE-constrained predictions have further advanced the field. Recently, Transformer-based models, including Rainformer [3] and Earthformer [13], have shown impressive accuracy by modeling global dependencies. However, these methods face substantial computational challenges due to the quadratic complexity of self-attention when processing high-resolution satellite imagery, which our work specifically addresses.

Koopman Operator Theory. Nonlinear dynamical systems are inherently complex, making direct analysis and long-term prediction challenging. Rather than modeling the evolution of the state $\mathbf{x}_n \in \mathcal{M}$ directly via a nonlinear map $\mathbf{f}$, the Koopman operator framework proposes lifting the system into a higher-dimensional space of observable functions.

Consider an observable function $g : \mathcal{M} \rightarrow \mathbb{C}$ that captures a specific measurement or feature of the state $\mathbf{x}_n$. Even though the underlying state evolution

$$\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n)$$

is nonlinear, the evolution of the observable is given by

$$g(\mathbf{x}_{n+1}) = g(\mathbf{f}(\mathbf{x}_n)).$$

This motivates the definition of the Koopman operator $\mathcal{K}$ as

$$(\mathcal{K}g)(\mathbf{x}) = g(\mathbf{f}(\mathbf{x})),$$

which is linear by construction:

$$\mathcal{K}(ag_1 + bg_2) = a\mathcal{K}g_1 + b\mathcal{K}g_2,$$

for any scalars a and b . The surprising and powerful aspect of this approach is that, although $\mathbf{f}$ is nonlinear, the operator $\mathcal{K}$ acts linearly on the space of observables.

By examining the spectral properties of $\mathcal{K}$, we can identify eigenfunctions ϕ_i and associated eigenvalues λ_i that al-

low us to decompose any observable g into a linear combination:

$$g(\mathbf{x}_n) = \sum_{i=1}^{\infty} \lambda_i^n \phi_i(\mathbf{x}_0) v_i^g,$$

where v_i^g are the Koopman modes corresponding to the observable g . This modal decomposition provides clear insights into the growth/decay rates and oscillatory behavior of the system, effectively linearizing its dynamics in the lifted space.

Koopman Auto Encoders. Koopman Auto Encoders (KAEs) embed nonlinear dynamical systems into linear Koopman spaces via deep autoencoders, enabling efficient analysis of complex visual phenomena. Recent theoretical advances by Wang et al. [2] and Yeung et al. [54] have strengthened these frameworks for visual data processing. KAEs have been successfully applied to fluid dynamics prediction [32, 53], autonomous vehicle modeling [52], and action recognition in video sequences [47]. Despite these advances, the application of Koopman theory to high-dimensional satellite imagery forecasting remains underexplored. Our work develops a specialized Koopman-based vision architecture that achieves both computational efficiency and high accuracy for satellite image sequence prediction.

3. Proposed Method

In this section, we introduce **ViTKoop**, our proposed framework for future frame forecasting in spatiotemporal satellite image sequences. ViTKoop combines a Vision Transformer (ViT)-based Auto Encoder with the Koopman operator to efficiently model temporal dynamics in the latent space. The Auto Encoder extracts a compact latent representation of the input sequences, while the Koopman operator enables linearized temporal evolution in this space, allowing for computationally efficient and accurate long-term forecasting.

3.1. Solving PDEs by the Koopman Operator

Partial differential equations (PDEs) are fundamental in modeling complex phenomena, yet many lack analytic solutions. We propose a data-driven approach leveraging the Koopman operator theory to transform nonlinear dynamics into linear evolution in a latent space.

Rather than directly predicting high-dimensional, nonlinear states $\gamma(x_t)$, we employ observation functions $\mathbf{g} : \Gamma \rightarrow \mathcal{G}$ where the time-dependent Koopman operator $\mathcal{K}_t^{t+\varepsilon}$ propagates observables linearly:

$$\mathbf{g}(\gamma_{t+\varepsilon}) = \mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) \quad (1)$$

We integrate this framework with an Auto Encoder to learn a nonlinear mapping from the original PDE solution to a

low-dimensional latent representation $\mathbf{z}_t \approx \mathbf{g}(\gamma_t)$ where:

$$\mathbf{z}_{t+\varepsilon} \approx \mathcal{K}_t^{t+\varepsilon} \mathbf{z}_t \quad (2)$$

This approach leverages the expressive power of deep learning to capture complex features while exploiting linear dynamics in the latent space for efficient prediction.

3.2. Koopman Operator Approximation

To operationalize this approach, we construct a Krylov sequence of observables with temporal step size ε :

$$\mathcal{R}_n = \left[\mathbf{g}(\gamma_0), \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0), \mathcal{K}_0^{2\varepsilon} \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0), \dots, \mathcal{K}_{(n-1)\varepsilon}^{n\varepsilon} \cdots \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0) \right] \quad (3)$$

These observables are arranged into a Hankel matrix $\mathcal{H}_{m \times n}$:

$$\mathcal{H}_{m \times n} = \begin{bmatrix} \mathbf{g}(\gamma_0) & \mathbf{g}(\gamma_\varepsilon) & \cdots & \mathbf{g}(\gamma_{n\varepsilon}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}(\gamma_{(m-1)\varepsilon}) & \mathbf{g}(\gamma_{m\varepsilon}) & \cdots & \mathbf{g}(\gamma_{(m+n-1)\varepsilon}) \end{bmatrix} \quad (4)$$

The columns of $\mathcal{H}_{m \times n}$ span a Krylov subspace $\mathbb{K} \subset \mathcal{G}(\mathbb{R}^{d_\gamma} \times T)$. We approximate the Koopman operator by projecting it onto $\mathbb{K}$ and enforce that the Hankel matrix evolves as:

$$\mathcal{H}_{m \times n}(k+1) = \hat{\mathcal{K}}_{k\varepsilon}^{(k+1)\varepsilon} \mathcal{H}_{m \times n}(k), \forall k = 1, \dots, n. \quad (5)$$

To reduce computational costs for time-dependent systems, we assume approximate temporal invariance for small ε and perform temporal averaging:

$$\bar{\mathcal{K}}_\varepsilon \simeq \underset{P \in \mathbb{R}^{d_\gamma+1}}{\operatorname{argmin}} \sum_{k=1}^n \|\mathcal{H}_{m \times n}(k+1) - P \mathcal{H}_{m \times n}(k)\|_F. \quad (6)$$

This formulation enables efficient offline approximation of the Koopman operator for long-term prediction of time-series satellite imagery. The combined architecture of the Koopman operator and Auto Encoder provides a coherent theoretical foundation for predicting complex spatiotemporal phenomena in satellite image sequences while maintaining computational feasibility.

3.3. ViTKoop

Our framework consists of three main components: an encoder, a Koopman operator, and a decoder. The encoder extracts latent representations from input frames, the Koopman operator performs time evolution in the encoded space, and the decoder reconstructs the future frames from the evolved latent representations (Fig. 3(a)).

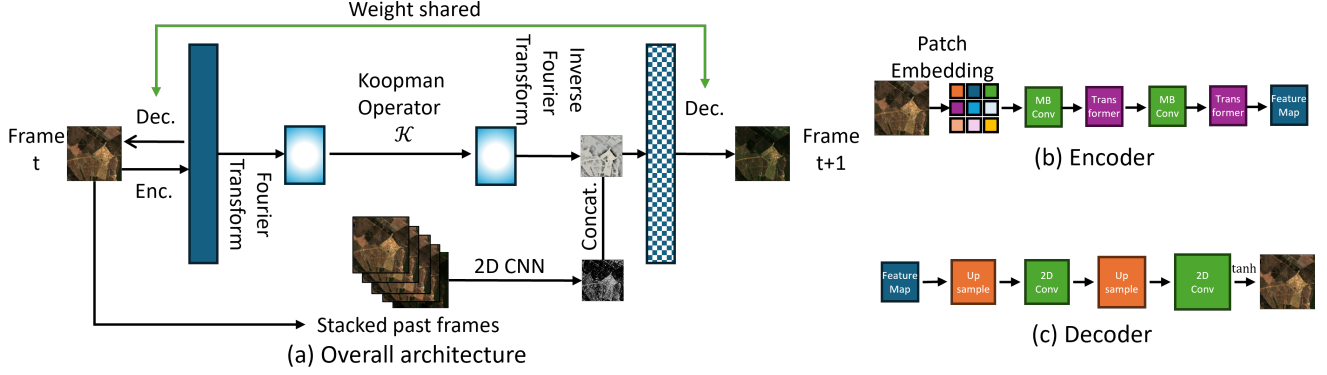


Figure 3. Architecture of the Koopman operator combined with an Auto Encoder for video frame prediction. (a) The overall architecture integrates past frames with a weight-shared encoder-decoder structure and the Koopman operator. The encoder transforms input frames into a feature space where the Koopman operator linearly predicts future states. (b) The encoder implementation uses a Tiny ViT (Vision Transformer) structure with MobileBlocks (MB Conv) and Transformer blocks to extract patch embeddings and create feature maps. (c) The decoder employs a simple 2D CNN with upsampling layers to reconstruct the predicted frame from the feature representation.

Koopman Operator. We construct a Hankel matrix $\hat{\mathcal{H}}_{m \times n}$ with embedding dimension m following Eqs. (25-26). Defining the Koopman operator as a Hankel matrix provides a linear approximation of complex nonlinear dynamics in a lifted latent space. Our 2D Koopman operator $\mathcal{K}$ is implemented as a learnable complex-valued transformation in the Fourier domain:

$$\mathcal{K} : \mathcal{F}(x_t) \rightarrow \mathcal{F}(x_{t+1}) \quad (7)$$

where $\mathcal{F}(\cdot)$ represents the 2D Fourier transform. The time evolution is performed as follows:

$$X_{t+1}^{\text{FT}} = \mathcal{K} \cdot X_t^{\text{FT}} \quad (8)$$

where X_t^{FT} is the Fourier-transformed latent representation of the input x_t . The inverse Fourier transform is then applied to obtain the updated representation in the spatial domain.

Auto Encoder Structure. The encoder is based on a Tiny ViT [50]-Base backbone that extracts spatial features from input frames (Fig. 3(b)). The decoder reconstructs latent features of frame t and calculates features using transposed convolution layers (Fig. 3(c)).

The Koopman operator acts as an intermediate transformation between encoding and decoding, enabling latent space time evolution. A simple 2D CNN is applied to the stacked past frames, which consist of a fixed number of input images determined by a hyperparameter, to extract contextual features. These extracted features are then concatenated with the transformed features from the Koopman operator. In each step of the Koopman layer, the transformed feature map is added to the original input, updating the representation over time. Two types of addition mechanisms are employed: linear and nonlinear. In the linear case, the transformed feature map is directly summed with the input, ensuring smooth updates. In the nonlinear case, a $\tanh$

activation function is applied before addition, allowing the model to learn more expressive transformations.

3.4. Implementation Details

Our model is implemented using PyTorch. The Koopman operator is parameterized with complex-valued weights and operates in the Fourier domain. The training is conducted for forecasting T_{out} frames from input T_{in} frames. The network is trained using the Adam optimizer with an initial learning rate of 10^{-4} . During training, input sequences are incrementally updated using predicted frames to enforce temporal consistency.

Following the formulation of Xiong *et al.* [53], our training objective jointly minimises an *image-reconstruction loss* (L_{rec}) and a *future-prediction loss* (L_{pred}). For an input sequence of length T_{in} and a prediction horizon of T_{out} , we define

$$L_{\text{rec}} = \sum_{t=1}^{T_{\text{in}}} \|\text{frame}_t - \text{rec}_t\|^2, L_{\text{pred}} = \sum_{t=1}^{T_{\text{out}}} \|\text{pred}_t - \text{gt}_t\|^2, \quad (9)$$

where $\text{frame}_t / \text{rec}_t$ denote the input and its reconstruction at time t , and $\text{gt}_t / \text{pred}_t$ denote the ground-truth and predicted future frame, respectively. To improve the *stability of the Koopman dynamics* in latent space, we introduce an additional *latent-consistency loss* (L_{latent}).

$$L_{\text{latent}} = \sum_{t=1}^{T_{\text{out}}} \|\hat{\mathbf{z}}_t - \mathbf{z}_t\|^2, \quad (10)$$

with $\hat{\mathbf{z}}_t$ the latent vector obtained by propagating the initial latent state through the learned Koopman operator, and $\mathbf{z}_t$ the latent vector produced by the encoder at the corresponding future step. The overall objective is the weighted

Dataset	Size			Len.		Size $H \times W$
	train	val	test	in	out	
SEVIR	35,718	9,060	12,159	13	12	384×384
ICAR-ENSO	5,205	334	1,667	12	14	24×48
EarthNet2021	8,100	900	1,000	10	20	64×64

Table 1. Statistics of the datasets used in the experiments.

sum

$$L_{\text{total}} = \lambda_1 L_{\text{pred}} + \lambda_2 L_{\text{recon}} + \lambda_3 L_{\text{latent}}, \quad (11)$$

in which $\lambda_1, \lambda_2, \lambda_3$ balance the contributions of prediction accuracy, reconstruction fidelity, and latent-space stability. Guided by pilot experiments on synthetic sequences (see Supplement 6.1), we empirically set $\lambda_1 = 1.0$, $\lambda_2 = 1.5$, $\lambda_3 = 0.1$.

4. Experiments

4.1. Forecasting Results

We figured out the quality of our ViTKoop and compared it with other recent state-of-the-art models on three real-world datasets: SEVIR [46], ICAR-ENSO¹ and EarthNet 2021[37]. The statistics of all the datasets used in the experiments are shown in Table 1. We normalized the data to the range $[0, 1]$ and trained all the models with the Mean-Squared Error (MSE) loss.

SEVIR. Storm EVent ImageRy (SEVIR) [46] is a spatiotemporally aligned dataset containing over 10,000 weather events. Each event consists of $384 \text{ km} \times 384 \text{ km}$ image sequences spanning 4 hours. Images in SEVIR were sampled and aligned across five different data types: three channels (C02, C09, C13) from the GOES-16 Advanced Baseline Imager, NEXRAD Vertically Integrated Liquid (VIL) mosaics, and GOES-16 Geostationary Lightning Mapper (GLM) flashes. The SEVIR benchmark supports scientific research on multiple meteorological applications including precipitation nowcasting, synthetic radar generation, and front detection. We adopt SEVIR for benchmarking precipitation nowcasting, i.e., predicting future VIL up to 60 minutes (12 frames) given 65 minutes of input VIL (13 frames). In the supplementary, Fig. 7 shows an example of VIL observation sequences in SEVIR.

The experimental results are listed in Table 2. Our ViTKoop consistently outperformed baselines on almost all metrics and provides significant performance gains, especially at high thresholds like CSI-219, which are more valued by the community. Table 2 presents the performance comparison of precipitation forecasting on the SEVIR dataset. Our proposed ViTKoop achieved results comparable to the Transformer-based Earthformer [13]

across all evaluation metrics, including Mean Squared Error (MSE) and Critical Success Index (CSI) at various precipitation thresholds. For details on these metrics, refer to the supp. 6.3. Notably, ViTKoop attained the same CSI scores as Earthformer while outperforming conventional CNN- and RNN-based approaches such as UNet [46], ConvLSTM [42], and PredRNN [49]. These results suggest that our method effectively captures the spatiotemporal dependencies of satellite imagery, achieving competitive performance with state-of-the-art Transformer models.

ICAR-ENSO. El Niño/Southern Oscillation (ENSO) has a wide range of associations with regional climate extremes and ecosystem impacts. ENSO sea surface temperature (SST) anomalies forecasting for lead times up to one year (12 steps) is a valuable and challenging problem. ICAR-ENSO consists of historical climate observation and simulation data provided by Institute for Climate and Application Research (ICAR). We forecasted the SST anomalies up to 14 steps (2 steps more than one year for calculating three-month-moving-average) given a context of 12 steps of SST anomaly observations. Table 3 compares the performance of our Earthformer with baselines on the ICAR-ENSO dataset.

We reported the mean correlation skill $C\text{-Nino3.4-M} = \frac{1}{K} \sum_k C_k^{\text{Nino3.4}}$ and the weighted mean correlation skill $C\text{-Nino3.4-WM} = \frac{1}{K} \sum_k a_k \cdot C_k^{\text{Nino3.4}}$ over $K = 12$ forecasting steps², as well as the MSE between the spatiotemporal SST anomalies prediction and the corresponding ground-truth. For details on the metrics, refer to the supp. 6.3. Table 3 presents the performance comparison of ENSO forecasting on the ICAR-ENSO dataset. Our proposed ViTKoop achieves state-of-the-art results, demonstrating identical performance to the Transformer-based Earthformer [13] across all evaluation metrics, including the mean correlation skill $C\text{-Nino3.4-M}$, the weighted mean correlation skill $C\text{-Nino3.4-WM}$, and the Mean Squared Error (MSE). Furthermore, ViTKoop consistently outperforms CNN- and RNN-based approaches such as UNet [46], ConvLSTM [42], and PredRNN [49]. These results indicate that ViTKoop effectively captures the temporal evolution of sea surface temperature (SST) anomalies, achieving competitive performance with state-of-the-art Transformer models.

EarthNet2021. EarthNet2021 data used here was provided as part of the EarthNet2021 Challenge and consists of 23,904 training datacubes located across Europe [39]. There are four evaluation tracks: Main (IID), Robustness (OOD), Extreme Summer, and Seasonal Cycle. The IID set contains about 4000 samples from the same regions as the training set, where one region corresponds to a Sentinel-2 tile (i.e., about $100 \times 100 \text{ km}$). However, if two sam-

¹Dataset available at <https://tianchi.aliyun.com/dataset/dataDetail?dataId=98942>

² $a_k = b_k \cdot \ln k$, where $b_k = 1.5$, for $k \leq 4$; $b_k = 2$, for $4 < k \leq 11$; $b_k = 3$, for $k > 11$.

Model	Metrics							
	CSI-M $\uparrow$	CSI-219 $\uparrow$	CSI-181 $\uparrow$	CSI-160 $\uparrow$	CSI-133 $\uparrow$	CSI-74 $\uparrow$	CSI-16 $\uparrow$	MSE (10^{-3}) $\downarrow$
Persistence	0.2613	0.0526	0.0969	0.1278	0.2155	0.4705	0.6047	11.5338
UNet [46]	0.3593	0.0577	0.1580	0.2157	0.3274	0.6531	0.7441	4.1119
ConvLSTM [42]	0.4185	0.1288	0.2482	0.2928	0.4052	0.6793	0.7569	3.7532
PredRNN [49]	0.4080	0.1312	0.2324	0.2767	0.3858	0.6713	0.7507	3.9014
PhyDNet [18]	0.3940	0.1288	0.2309	0.2708	0.3720	0.6556	0.7059	4.8165
E3D-LSTM [48]	0.4038	0.1239	0.2270	0.2675	0.3825	0.6645	0.7573	4.1702
Rainformer [3]	0.3661	0.0831	0.1670	0.2167	0.3438	0.6585	0.7277	4.0272
Earthformer [13]	0.4419	0.1791	0.2848	0.3232	0.4271	0.6860	0.7513	3.6957
Ours(ViTkoop)	0.4381	0.1673	0.2759	0.3174	0.4227	0.6728	0.7475	3.7140

Table 2. Performance comparison on SEVIR. We include Critical Success Index (CSI) besides MSE as evaluation metrics. The CSI, a.k.a intersection over union (IoU), is calculated at different precipitation thresholds and denoted as *CSI-thresh*.

Model	Metrics		
	<i>C</i> -Nino3.4-M $\uparrow$	<i>C</i> -Nino3.4-WM $\uparrow$	MSE (10^{-4}) $\downarrow$
Persistence	0.3221	0.447	4.581
UNet [46]	0.6926	2.102	2.868
ConvLSTM [42]	0.6955	2.107	2.657
PredRNN [49]	0.6492	1.910	3.044
PhyDNet [18]	0.6646	1.965	2.708
E3D-LSTM [48]	0.7040	2.125	3.095
Rainformer [3]	0.7106	2.153	3.043
Earthformer [13]	0.7329	2.259	2.546
Ours(ViTkoop)	0.7310	2.552	2.514

Table 3. Performance comparison on ICAR-ENSO. *C*-Nino3.4-M and *C*-Nino3.4-WM are the mean and the weighted mean of the correlation skill $C^{Nino3.4}$ over $K = 12$ forecasting steps. *C*-Nino3.4-WM assigns more weights to longer-term prediction scores. MSE is calculated between the spatiotemporal SST anomalies prediction and the corresponding ground-truth.

Model	IID				OOD			
	Metrics							
	MAD $\uparrow$	OLS $\uparrow$	EMD $\uparrow$	SSIM $\uparrow$	MAD $\uparrow$	OLS $\uparrow$	EMD $\uparrow$	SSIM $\uparrow$
Persistence [37]	0.2315	0.3239	0.2099	0.3265	0.2248	0.3236	0.2123	0.3390
Channel U-Net [37]	0.2482	0.3381	0.2336	0.3973	0.2402	0.3390	0.2371	0.3721
Arcon [37]	0.2414	0.3216	0.2258	0.3863	0.2314	0.3088	0.2177	0.3432
SGConvLSTM [20]	0.2589	0.3456	0.2533	0.5292	0.2512	0.3481	0.2597	0.4977
EarthFormer [13]	0.2638	0.3513	0.2623	0.5565	0.2533	0.3581	0.2732	0.5270
Ours(KoopamViT)	0.2596	0.3501	0.2582	0.5517	0.2517	0.3525	0.2764	0.5225

Table 4. Performance comparison on EarthNet2021 using the two different test tracks (iid, ood) of our models and baselines. MAD , OLS , EMD, and SSIM are calculated between the corresponding ground-truth frame.

ples capture exactly the same area, it was ensured that there is no temporal overlap between them. The OOD set contains a similar number of samples but from completely different regions, thereby additionally evaluating the model’s spatial generalization capability. For these two tracks, the input length is 10, while the prediction length is 20. EarthNet2021 is evaluated using Median Absolute Deviation (MAD), Ordinary Least Squares (OLS), Earth Mover’s Distance (EMD), and Structural Similarity Index (SSIM). For detailed definitions of these metrics, please refer to the supp. 6.3.

Table 4 presents the performance comparison of EarthNet2021 future satellite image prediction across both IID and OOD test tracks. Our proposed ViTKoop achieves comparable results to the Transformer-based EarthFormer [13], demonstrating state-of-the-art performance in key evaluation metrics, including Mean Absolute Deviation (MAD), Ordinary Least Squares (OLS), Earth Mover’s Distance (EMD), and Structural Similarity Index (SSIM). In particular, ViTKoop attains nearly identical scores to EarthFormer in the IID setting, highlighting its capability to capture spatiotemporal dependencies effectively. Moreover, our model

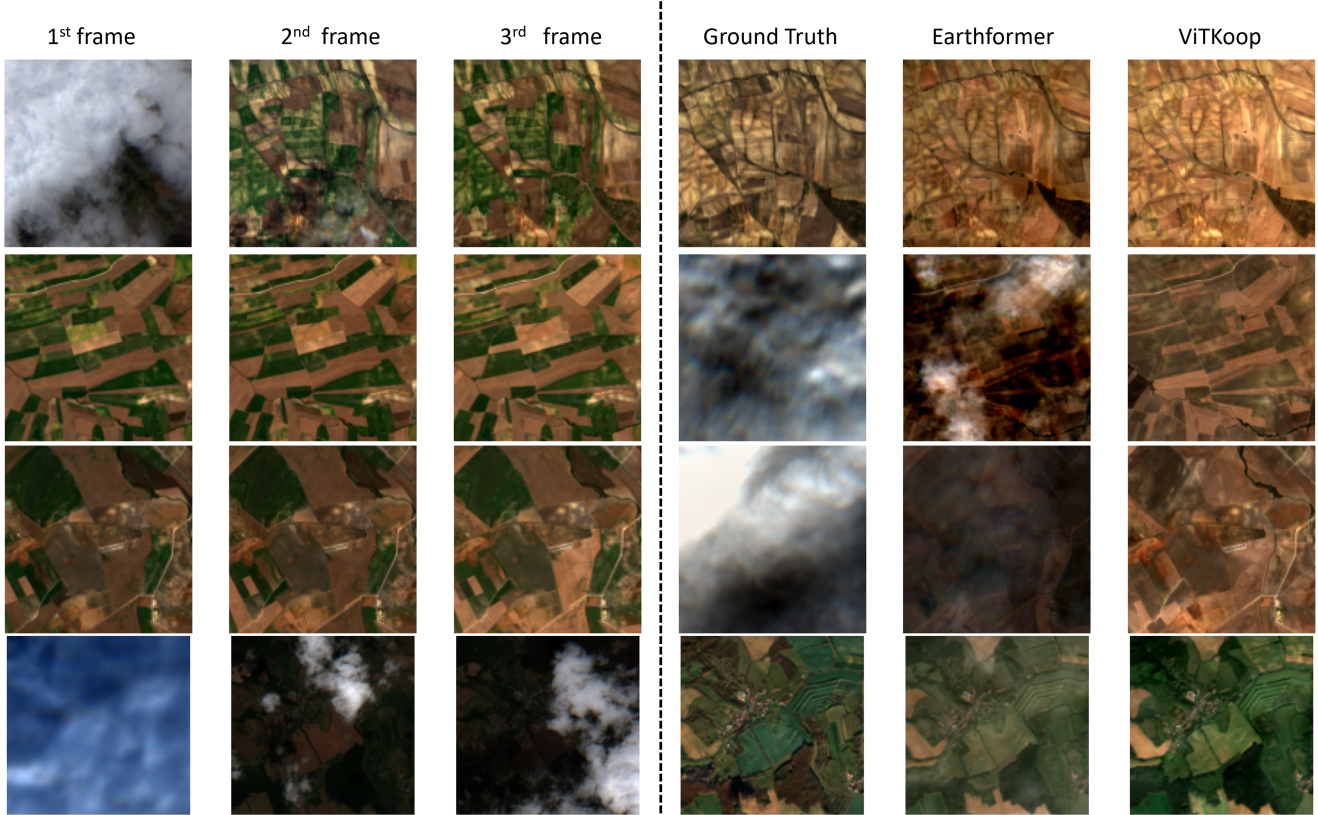


Figure 4. Forecasting results of our ViTKoop and the baseline Earthformer [13] on the EarthNet2021 dataset [37]. In this task, a time-series of 10 satellite images is provided as input to predict the subsequent 20 frames. The three images on the left correspond to the first three frames of the input sequence.

generalizes well to the OOD test set, showing robust performance under distribution shifts. These results confirm that ViTKoop is a strong alternative to existing Transformer-based methods for satellite image forecasting, leveraging Koopman operator theory to model complex temporal dynamics efficiently.

Additionally, Fig. 4 presents a qualitative comparison between predictions from our proposed ViTKoop and the Transformer-based EarthFormer. While EarthFormer demonstrates high fidelity in color reproduction, closely matching the ground-truth frames, ViTKoop also produces future images that resemble the ground truth. Notably, EarthFormer tends to generate images containing clouds, whereas ViTKoop generally yields images with fewer clouds. We hypothesize that this discrepancy arises from the inherent temporal dynamics of satellite imagery captured at sparse intervals (on the order of several weeks), where cloud information provides limited predictive value. Consequently, ViTKoop appears to discard cloud-related information to focus on more stable features of the scene.

4.2. Model Complexity

To evaluate the efficiency of our proposed method, we compare it with state-of-the-art approaches for satellite im-

age time-series forecasting. Fig. 5 presents the MSE and GFLOPS of various models on the SEVIR dataset, including baseline models such as UNet [46], ConvLSTM [42], and PredRNN [49], as well as more recent Transformer-based methods like Rainformer [3] and Earthformer [13]. Our proposed model, ViTKoop, achieves a favorable balance between predictive performance and computational efficiency, as demonstrated by its MSE of 3.7140 and 2.8 GFLOPS (Redpoint in Fig. 5). Compared to recurrent models such as PredRNN and E3D-LSTM, ViTKoop significantly reduces computational cost while maintaining competitive forecasting accuracy. Additionally, while Earthformer achieves a slightly lower MSE (3.6957) at the cost of higher computational complexity (23.9 GFLOPS), ViTKoop demonstrates similar predictive performance with substantially lower computational overhead. Other models, such as ConvLSTM and UNet, exhibit either higher MSE or lower efficiency trade-offs. These results highlight that ViTKoop effectively captures spatiotemporal dependencies, offering a well-balanced trade-off between prediction accuracy and computational efficiency.

Ablation Setting	MSE (Short-Term)	MSE (Mid-Term)	MSE (Long-Term)
ViT w/o Koopman	0.0452	0.0995	0.2723
ViT w/ Koopman	0.0387	0.0752	0.1345
MLP w/ Koopman	0.0528	0.0957	0.1704
ResNet w/ Koopman	0.0415	0.0823	0.1518
ViT /w Koopman	0.0387	0.0752	0.1345
ViT /w DMD Approximation	0.0401	0.0798	0.1457
ViT /w EDMD Approximation	0.0393	0.0775	0.1408
ViT /w Koopman Approximation	0.0387	0.0752	0.1345

Table 5. Ablation study results showing Mean Squared Error (MSE) for different model variations across short-term, mid-term, and long-term forecasting tasks of simulated non-linear data. The number of Input frames is 20 frames, and the number of output flames is 5 frames(short-term), 20 frames(mid-term), and 100 frames(long-term).

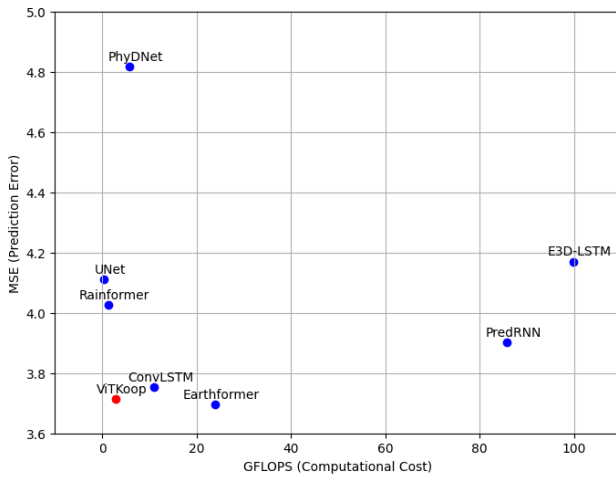


Figure 5. Comparison of model complexity in SEVIR dataset.

4.3. Ablation Study

To evaluate the effectiveness of the key components in our proposed ViTKoop framework, we conducted an ablation study focusing on four critical aspects: the presence of the Koopman operator, the choice of the Auto Encoder architecture, the approximation method for the Koopman operator, and the model’s long-term forecasting capability. The dataset comprises 1000 samples, each containing 200 frames of 32×32 pixel images depicting linear wave patterns. In these experiments, we set the number of input image frames to 20. These patterns are created by combining sinusoidal and cosine functions with randomly selected frequencies and speeds within specified ranges, generating diverse yet controlled wave dynamics and providing a rich temporal and spatial representation for potential model training. See supp. 6.2 for a detailed description of the ablation study settings. The results on simulated non-linear data are presented in Table 5.

Our ablation study confirms that each design modification in our proposed method positively contributes to performance improvement. In particular, when comparing

different approximation methods for the Koopman operator—namely Fourier-based, DMD, and EDMD—we observed that the Fourier-based approximation achieves the most accurate long-term predictions. This finding indicates that the Fourier-based Koopman operator not only captures the underlying dynamics more effectively but also maintains stability over extended forecasting horizons, making it especially suitable for long-term satellite image prediction tasks.

5. Conclusion

In this work, we propose a novel framework, ViTKoop, which integrates a ViT-based Auto Encoder with a Koopman operator for efficient and accurate forecasting of satellite image time series. By leveraging linear temporal evolution in the latent space, our method significantly reduces computational cost compared to transformer-based approaches, while experiments on real-world datasets (i.e., ENSO, SEVIR, and EarthNet2021) demonstrate comparable or superior performance. This balance between efficiency and accuracy suggests promising applications in environmental monitoring and disaster forecasting. Moreover, the Koopman operator-based framework offers enhanced interpretability. Rather than relying solely on a black-box approach, our method uses a physically motivated linear approximation in the latent space, allowing for a degree of interpretability in the learned representations and their temporal evolution, which benefits both understanding and further refinement of the model.

Key limitations remain. The Koopman step’s local-linear assumption can break under strong nonlinearities or abrupt regime shifts. Accuracy also depends on the autoencoder’s skill at embedding complex image patterns into a Koopman-compatible latent space. Moreover, evaluation on only ENSO, SEVIR, and EarthNet2021 leaves robustness across other resolutions, lighting, and regions untested.

Aknowlegement

We used ABCI 3.0 provided by AIST and AIST Solutions. We also thanks to data prvider of SEVIR, ENSO, and EarthNet2021.

References

- [1] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017. 5
- [2] Naveen Sharma Aswani, Saeed Jabari, and Muhammad Shafique. Representing neural network layers as linear operations via koopman operator theory. *arXiv preprint arXiv:2307.11208*, 2023. 3
- [3] Cong Bai, Feng Sun, Jinglin Zhang, Yi Song, and Shengyong Chen. Rainformer: Features extraction balanced network for radar-based precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 2, 6, 7
- [4] A Bashfield and A Keim. Continent-wide dem creation for the european union. In *34th international symposium on remote sensing of environment. the GEOSS era: Towards operational environmental monitoring. Sydney, Australia*, pages 10–15. Citeseer, 2011. 3
- [5] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eureka Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):19, 2017. 5
- [6] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. 7
- [7] Kathleen P Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019. 7
- [8] Nelida vCrnjarić-vZic, Senka Maćevsić, and Igor Mezić. Koopman operator spectrum for random dynamical systems. *Journal of Nonlinear Science*, 30:2007–2056, 2020. 5
- [9] Christian Schroeder de Witt, Catherine Tong, Valentina Zantedeschi, Daniele De Martini, Freddie Kalaitzis, Matthew Chantry, Duncan Watson-Parris, and Piotr Bilinski. Rain-Bench: towards global precipitation forecasting from satellite imagery. In *AAAI*, 2021. 1
- [10] Lokenath Debnath and Lokenath Debnath. *Nonlinear partial differential equations for scientists and engineers*. Springer, 2005. 4
- [11] Lasse Espeholt, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Jason Hickey, Aaron Bell, and Nal Kalchbrenner. Skillful twelve hour precipitation forecasts using large context neural networks. *arXiv preprint arXiv:2111.07470*, 2021. 1
- [12] Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904, 2022. 7
- [13] Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In *NeurIPS*, 2022. 2, 5, 6, 7
- [14] Ferran Gascon, Enrico Cadau, Olivier Colin, Bianca Hoersch, Claudia Isola, B López Fernández, and Philippe Martimort. Copernicus sentinel-2 mission: products, algorithms and cal/val. In *Earth observing systems XIX*, pages 455–463. SPIE, 2014. 3
- [15] Mark S Gockenbach. *Partial differential equations: analytical and numerical methods*. SIAM, 2010. 4
- [16] Steven J Goodman, Timothy J Schmit, Jaime Daniels, and Robert J Redmon. *The GOES-R series: a new generation of geostationary environmental satellites*. Elsevier, 2019. 1
- [17] Debdipta Goswami and Derek A. Paley. Bilinearization, reachability, and optimal control of control-affine nonlinear systems: A koopman spectral approach. *IEEE Transactions on Automatic Control*, 67(6):2715–2728, 2022. 2
- [18] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 2, 6
- [19] Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year ENSO forecasts. *Nature*, 573(7775): 568–572, 2019. 1, 2, 3
- [20] Klaus-Rudolf Kladny, Marco Milanta, Oto Mraz, Koen Hufkens, and Benjamin D. Stocker. Enhanced prediction of vegetation responses to extreme drought using deep learning and earth observation data. *Ecological Informatics*, 80: 102474, 2024. 6
- [21] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of National Academy of Sciences*, 17:315–318, 1931. 2
- [22] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28:687–710, 2018. 5
- [23] Munkhnasan Lamchin, Woo-Kyun Lee, Seong Woo Jeon, Sonam Wangyel Wang, Chul Hee Lim, Cholho Song, and Minjun Sung. Long-term trend and correlation between vegetation greenness and climate variables in asia based on satellite data. *Science of the Total Environment*, 618:1089–1095, 2018. 3
- [24] Jing Li, Ke Fan, and Liming Zhou. Satellite observations of el niño impacts on eurasian spring vegetation greenness during the period 1982–2015. *Remote Sensing*, 9(7):628, 2017. 3
- [25] Mengnan Li and Lijian Jiang. Data-driven reduced-order modeling for nonautonomous dynamical systems in multi-scale media. *Journal of Computational Physics*, 474:111799, 2023. 5
- [26] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2020. 2, 6
- [27] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and

- Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020. [4](#)
- [28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. [2](#)
- [29] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018. [7](#)
- [30] Robert MM Mattheij, Sjoerd W Rienstra, and JHM Ten Thijs Boonkamp. *Partial differential equations: modeling, analysis, computation*. SIAM, 2005. [4](#)
- [31] A. Mauroy and I. Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016. [2](#)
- [32] Di Meng, Yulin Zhu, Jing Wang, and Yijie Shi. Koopman neural operator approach to fast flow prediction of airfoil transonic buffet. *Physics of Fluids*, 35(9):095104, 2023. [3](#)
- [33] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. [6](#)
- [34] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022. [2](#)
- [35] Sebastian Peitz, Samuel E. Otto, and Clarence W. Rowley. Data-driven model predictive control using interpolated Koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3):2162–2193, 2020. [2](#)
- [36] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021. [1](#)
- [37] Konrad Reiche, Michael Schmitt, Yihao Wang, and Ribana Roscher. Earthnet2021: A large-scale dataset and challenge for earth surface forecasting. *Remote Sensing*, 13(21):4456, 2021. [5](#), [6](#), [7](#)
- [38] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019. [1](#)
- [39] Christian Requena-Mesa, Vitus Benson, Markus Reichstein, Jakob Runge, and Joachim Denzler. EarthNet2021: A large-scale dataset and challenge for earth surface forecasting as a guided video prediction task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1132–1142, 2021. [5](#), [3](#)
- [40] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011. [5](#)
- [41] Daniel Scheffler, André Hollstein, Hannes Diedrich, Karl Segl, and Patrick Hostert. Arosics: An automated and robust open-source image co-registration software for multi-sensor satellite data. *Remote sensing*, 9(7):676, 2017. [3](#)
- [42] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. [1](#), [2](#), [5](#), [6](#), [7](#)
- [43] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NeurIPS*, 2017. [2](#)
- [44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. [6](#)
- [45] Hiroki Tanabe. *Functional analytic methods for partial differential equations*. CRC Press, 2017. [4](#)
- [46] Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020. [1](#), [2](#), [5](#), [6](#), [7](#)
- [47] Xiaowen Wang, Xiaolong Xu, and Yong Mu. Neural koopman pooling: Control-inspired temporal dynamics encoding for skeleton-based action recognition. *arXiv preprint arXiv:2304.03508*, 2023. [3](#)
- [48] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *International conference on learning representations*, 2018. [6](#)
- [49] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip Yu, and Mingsheng Long. PredRNN: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [2](#), [5](#), [6](#), [7](#)
- [50] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision (ECCV)*, 2022. [4](#)
- [51] Yuzhen Wu, Guoping Tang, Hui Gu, Yonglin Liu, Muzhen Yang, and Lin Sun. The variation of vegetation greenness and underlying mechanisms in guangdong province of china during 2001–2013 based on modis data. *Science of the Total Environment*, 653:536–546, 2019. [3](#)
- [52] Yifan Xiao, Xin Zhang, Xiang Xu, Xiang Liu, and Max Q-H Meng. Deep neural networks with koopman operators for modeling and control of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7880–7891, 2023. [3](#)
- [53] Wei Xiong, Xiaomeng Huang, Ziyang Zhang, Ruixuan Deng, Pei Sun, and Yang Tian. Koopman neural operator as a mesh-free solver of non-linear partial differential equations. *Journal of Computational Physics*, page 113194, 2024. [3](#), [4](#)
- [54] Edgar Yeung, Soumik Kundu, and Nicholas Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. in *Proceedings of the American Control Conference (ACC)*, 2019. [3](#)

ViT-Koop: Vision-Transformer–Koopman Operators for Efficient Time-Series Forecasting of Earth-Observation Data

Supplementary Material

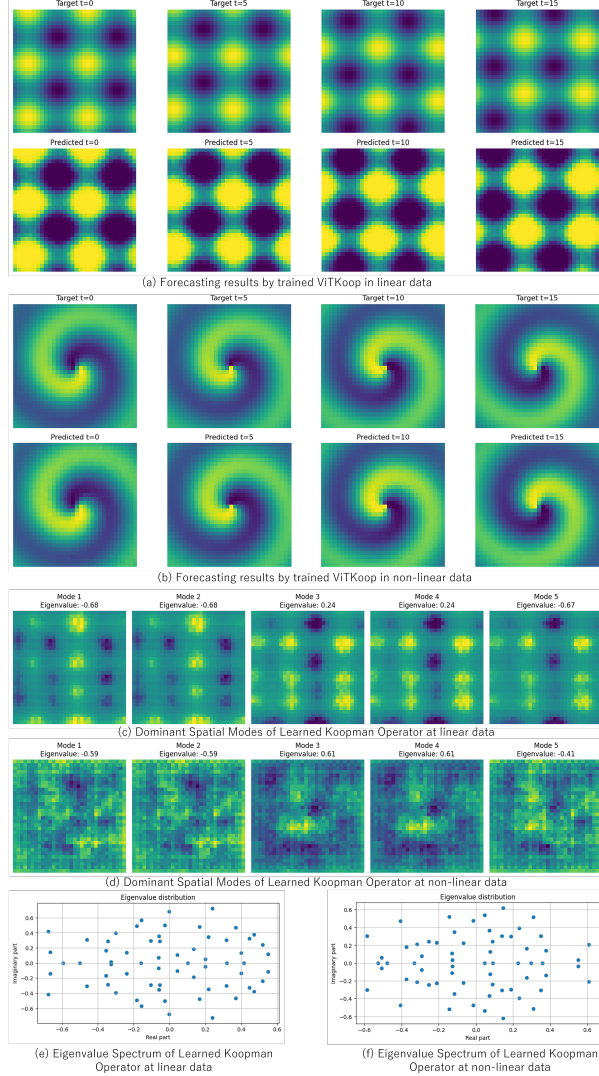


Figure 6. Visualized forecasting results and dominant spatial modes and eigenvalue spectrum of trained koopman operator.

6. Experiments

6.1. Simulated Data

In preparation for evaluating the proposed algorithm, we performed a series of preliminary experiments using a synthetic dataset. The primary goal was to verify the functionality and performance of our Koopman operator and Auto Encoder-based approach in predicting future frames in spatiotemporal sequences.

Linear Data. We generated sequences with specifically defined properties. In total, 1000 samples were created, each consisting of 200 frames of 32×32 pixel images. This setup ensures a comprehensive representation of temporal progression and spatial resolution. The dataset primarily focuses on linear wave patterns, generated through sinusoidal and cosine functions with randomly assigned frequencies and speeds. This approach enhances variability while maintaining controlled complexity, which is crucial for effective model training.

For linear data generation, the process involves selecting random frequencies freq_x and freq_y uniformly from the interval $[1, 3]$. These frequencies determine the periodic nature of the waves, while a random speed drawn from $[0.1, 0.3]$ influences their temporal evolution. At each time step t in the sequence, a meshgrid X, Y spanning from 0 to 2π is created over the 32×32 image size. The wave pattern Z is computed via the equation:

$$Z = \sin(\text{freq}_x \cdot X + \text{speed} \cdot t) + \cos(\text{freq}_y \cdot Y + \text{speed} \cdot t).$$

This computation combines sine and cosine waves to produce intricate spatiotemporal variations, emulating complex real-world phenomena.

The preliminary experiments aimed to confirm that the proposed algorithm accurately learns and predicts dynamics within the generated sequences (Fig. 6(a)). In our analysis, understanding the trained Koopman operator is crucial to deciphering the latent spatiotemporal dynamics captured by the model. A key focus is the distribution of eigenvalues derived from the Koopman operator, offering insights into our synthetic dataset’s structure (Fig. 6(c)). The eigenvalues exhibit notable characteristics. Their real parts are dispersed rather than concentrated near zero, indicating a lack of strongly attenuating modes (Fig. 6(e)). This suggests that the captured dynamics exhibit minimal rapid decay, consistent with systems that have weak damping or minimal attenuation. Moreover, the eigenvalues span a broad range along the imaginary axis, highlighting the presence of numerous oscillatory modes. This spread mirrors the wave-like nature inherent in the data, with various oscillations and frequency components. Unlike simple harmonic systems, where eigenvalues might align along a unit circle, the uniform distribution observed here indicates interactions among multiple temporal scales rather than a singular periodic pattern. Generated using a linear wave model, $Z = \sin(f_x X + vt) + \cos(f_y Y + vt)$, the dataset produces linear wave patterns with diverse frequencies and velocities.

The broad imaginary dispersion in eigenvalues is a direct outcome of this process. Randomized wave velocities and frequencies further ensure the absence of a simple circular eigenvalue pattern, reflecting the intricate interplay of these variables.

Non-linear Data. Additionally, we defined a nonlinear spatiotemporal function that produces a swirling pattern centered in an $N \times N$ grid. The pattern’s evolution follows:

$$Z_{i,j}(t) = \sin\left(\frac{r}{3} + \theta + 0.2t\right) \exp\left(-\frac{r}{20}\right),$$

where (i, j) are spatial coordinates, $r = \sqrt{(i - c_x)^2 + (j - c_y)^2}$ is the radial distance from the center (c_x, c_y) , and $\theta = \tan^{-1}((j - c_y)/(i - c_x))$ is the angular position. The sequence length is set to T , ensuring sufficient temporal variability.

We train a Koopman operator using an Auto Encoder-based framework. The encoder maps high-dimensional observations to a latent space where a linear Koopman operator approximates temporal evolution. The decoder reconstructs the original observations from these latent representations. After training, we extract Koopman eigenvalues and eigenfunctions, visualizing the learned modes.

Fig. 6(d) shows sample eigenfunctions corresponding to different eigenvalues, which are complex-valued. The real and imaginary components of the eigenvalues (Fig. 6(f)) indicate periodic dynamics, where eigenvalues near the unit circle correspond to persistent oscillatory modes. The obtained modes qualitatively align with the dominant swirling features in the original data, confirming that the Koopman operator effectively captures the system’s latent dynamics.

These results demonstrate the potential of Koopman-based representations for learning structured temporal patterns in complex spatiotemporal datasets. Further analysis—including robustness to noise and generalization across different parameter settings—remains an interesting direction for future work.

6.2. Detailed Ablation Study Setting

To evaluate the effectiveness of the key components in our proposed ViTKoop framework, we conducted an ablation study focusing on four critical aspects: the presence of the Koopman operator, the choice of the Auto Encoder architecture, the approximation method for the Koopman operator, and the model’s long-term forecasting capability. In these experiments, we set the number of input image frames to 20. The results of the simulated non-linear data (see supp. 6.1 for a detailed description of the ablation study settings) are presented in Table 5 and discussed below.

Effectiveness of the Koopman Operator. To assess the impact of the Koopman operator, we compared ViTKoop

with a baseline model that replaces the Koopman operator with a simple linear transformation in the latent space. As shown in Table 5, the absence of the Koopman operator leads to a significant degradation in prediction accuracy, particularly for long-term forecasts. This finding confirms that the Koopman operator effectively captures the underlying temporal dynamics, resulting in more accurate future state predictions.

Choice of Auto Encoder Architecture. We evaluated different Auto Encoder architectures, including a convolutional neural network (CNN)-based Auto Encoder, a ResNet-based Auto Encoder, and the Vision Transformer (ViT)-based Auto Encoder used in our proposed model. The results indicate that the ViT-based Auto Encoder achieves the highest accuracy, demonstrating its superior ability to extract spatiotemporal features from satellite image sequences. The CNN-based Auto Encoder struggles to capture long-range dependencies, while the ResNet-based model offers slightly better performance but still falls short of the ViT-based approach. The self-attention mechanism in ViT aggregates information across the entire image, enabling the model to capture global spatial dependencies that CNNs cannot easily model due to their limited receptive fields. This global information aggregation is crucial for satellite imagery, where phenomena often span large spatial extents and distant pixels may have significant correlations.

Approximation Method for the Koopman Operator.

We investigated various approaches for approximating the Koopman operator, including the Fourier-based approximation (our proposed method), Dynamic Mode Decomposition (DMD), and Extended DMD (EDMD). The results in Table 5 show that the Fourier-based approximation achieves the best balance between accuracy and computational efficiency. While EDMD provides competitive accuracy, it incurs a higher computational cost, making it less practical for large-scale satellite image forecasting.

Long-Term Forecasting Capability. We analyzed the model’s performance at different forecasting horizons: short-term (5 frames), mid-term (20 frames), and long-term (100 frames). The results reveal that ViTKoop maintains stable accuracy even in the long term, whereas the baseline models show significant error accumulation over time. The Koopman operator’s ability to model temporal dynamics in a linearized latent space contributes to this stability, making it particularly well-suited for long-term satellite image prediction tasks.

Summary of Ablation Study Results. Our ablation study confirms that each design choice in our proposed

method positively impacts performance. In particular, when comparing different approximation methods for the Koopman operator—namely Fourier-based, DMD, and EDMD—we observed that the Fourier-based approximation enables the most accurate long-term predictions. This indicates that the Fourier-based Koopman operator not only captures the underlying dynamics more effectively but also maintains stability over extended forecasting horizons, making it especially suitable for long-term satellite image prediction tasks.

6.3. Dataset Details

SEVIR. The Mean Squared Error used in optimization, we also incorporate the Critical Success Index (CSI), commonly used in precipitation nowcasting and defined as $CSI = \frac{\#Hits}{\#Hits + \#Misses + \#F.Alarms}$. To count $\#Hits$ (truth=1, pred=1), $\#Misses$ (truth=1, pred=0), and $\#F.Alarms$ (truth=0, pred=1), the prediction and ground truth are rescaled to the range $[0, 255]$ and then binarized at thresholds $\{16, 74, 133, 160, 181, 219\}$. We report CSI at each threshold and also their mean, CSI-M.

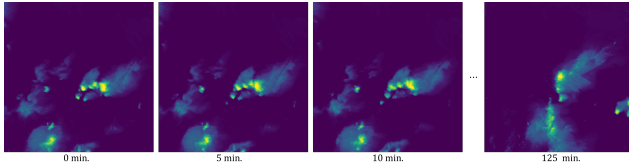


Figure 7. Samples of the SEVIR dataset.

ICAR-ENSO. To evaluate our model, we use the Nino3.4 index, which represents the area-averaged SST anomalies in the region $170^\circ - 120^\circ W$, $5^\circ S - 5^\circ N$, a crucial indicator of this climate event. We measure forecast quality using the correlation skill $C^{Nino3.4}$ [19] of the three-month moving average Nino3.4 index:

$$C^{Nino3.4} = \frac{\sum_N (\mathbf{X} - \bar{\mathbf{X}})(\mathbf{Y} - \bar{\mathbf{Y}})}{\sqrt{\sum_N (\mathbf{X} - \bar{\mathbf{X}})^2 \sum_N (\mathbf{Y} - \bar{\mathbf{Y}})^2}} \in \mathbb{R}^K,$$

calculated on the entire test set of size N . Here, $\mathbf{Y} \in \mathbb{R}^{N \times K}$ denotes the ground-truth of the Nino3.4 index at K forecast steps, and $\mathbf{X} \in \mathbb{R}^{N \times K}$ is the corresponding model prediction.

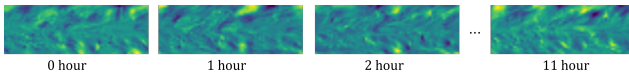


Figure 8. Samples of the ENSO dataset.

EarthNet2021. Remote sensing datacubes represent scenes from the Sentinel-2 mission [14], covering a spatial extent of 128×128 pixels at 20 m resolution

($2.56 \text{ km} \times 2.56 \text{ km}$ total area), and providing surface reflectance every 5 days in four wavelength bands (blue, green, red, and near-infrared, referred to as RGBI). Each datacube is complemented by a binary data quality mask indicating the presence of clouds. The EarthNet2021 Challenge also provides weather datacubes from E-OBS [41] for each day. These weather datacubes cover 80×80 pixels at a resolution of 1.28 km (i.e., $102.4 \text{ km} \times 102.4 \text{ km}$ total area) and include precipitation, air pressure at sea level, and daily mean, minimum, and maximum temperature. Prior work has shown these variables to be correlated with vegetation greenness [23, 24, 51], encoding information about plant stressors (e.g., low soil moisture, high vapor pressure deficit) and meteorological variables (e.g., wind speed). Additionally, time-invariant data layers of elevation from the EU-DEM digital elevation model [4] are provided at both high and low resolutions. Remote sensing and weather datacubes are spatially aligned so that multiple remote sensing datacubes fit within the geographical extent of a single weather datacube. A more detailed description of the data provided through EarthNet2021 is given by [39].

To prepare the data for our modeling, we applied additional preprocessing steps. The high-resolution elevation data were replicated for each time step, while the low-resolution elevation data (matching the meteorological data resolution) were not used. Daily meteorological data were aggregated into 5-day intervals to match the frequency of the remote sensing data. From the original daily mean temperature, daily total precipitation, and daily mean sea-level pressure, we computed their means over each 5-day interval. For daily minimum and maximum temperatures, we took the minimum and maximum values across the same 5-day period, respectively. We used only the subset of weather data matching the spatial extent of the remote sensing datacubes; data outside this domain were discarded. Due to persistent cloud coverage in some samples, data completeness within individual datacubes varied significantly. We removed three datacubes from the training set that were entirely cloud-contaminated during the context period.

The final evaluation metric, EarthNetScore (ENS), is the harmonic mean of four different metrics, computed only on non-masked pixels:

- **Median Absolute Deviation (MAD).** Computes the median absolute deviation between target pixels and predicted pixels for a robust measure of proximity.
- **Ordinary Least Squares (OLS).** Evaluates trend capture in vegetation by first computing NDVI maps for both target and predicted series, then fitting OLS models over time per pixel, and comparing fitted slopes.
- **Earth Mover’s Distance (EMD).** Similar to OLS but focuses on pixel distribution by computing the Wasserstein-1 distance between target and predicted NDVI values.

- **Structural Similarity Index (SSIM).** Captures perceptual similarity by averaging SSIM over channels and timesteps.

7. Implementation Details

7.1. Model Architecture

In our implementation, we employ a Tiny Vision Transformer (Tiny ViT) [50]-based Auto Encoder for feature extraction and reconstruction. The encoder and decoder are both designed as ViT models with the following configuration: an input resolution of (64, 64), a patch size of (2, 2), single input and output channels ($in_chans = 1$, $out_chans = 1$), 16 attention heads ($head_num = 16$), an embedding dimension of 768 ($embed_dim = 768$), and a depth of 16 transformer layers ($depth = 16$). The encoded latent representation is then processed by the Koopman operator module for time-series forecasting.

The Koopman operator is applied in the latent space, employing a spectral decomposition of the system dynamics to model temporal evolution. The forward propagation includes Fourier transforms for extracting dominant frequency components, followed by iterative updates with the learned Koopman matrix. The reconstructed output is obtained by applying the inverse Fourier transform and decoding the transformed latent representations.

7.2. Training

Our model is implemented using PyTorch. The Koopman operator is parameterized with complex-valued weights and operates in the Fourier domain. The training is conducted for $T_{out} = 12$ in SEVIR, $T_{out} = 14$ in ICAR-ENSO, and $T_{out} = 20$ in EarthNet2021 time steps with a batch size of B . The network is trained using the Adam optimizer with an initial learning rate of 10^{-4} . During training, input sequences are incrementally updated using the predicted frames to enforce temporal consistency.

7.3. Solving PDEs by the Koopman Operator

In many scientific and engineering applications, partial differential equations (PDEs) serve as fundamental tools to model complex phenomena such as fluid dynamics, quantum mechanics, and civil engineering problems [10]. Despite significant progress [45], many important PDEs (e.g., the Navier–Stokes equations) still lack analytic solutions [15], motivating the development of various numerical and data-driven methods for approximating their solutions [30].

Let $\Phi = \Phi(D; \mathbb{R}^{d_\phi})$ be a Banach space of inputs and $\Gamma = \Gamma(D; \mathbb{R}^{d_\gamma})$ a Banach space of solutions, both defined on a bounded open set $D \subset \mathbb{R}^d$. Traditional PDE solvers seek to approximate a solution operator

$$\mathcal{Q} : (\phi, \gamma_B, \gamma_I) \mapsto \gamma,$$

which reduces to $\mathcal{Q} : \phi \mapsto \gamma$ when the boundary and initial conditions γ_B and γ_I are fixed. In practice, a parametric version $\mathcal{Q}_\theta \simeq \mathcal{Q}$ is often considered to facilitate optimization problems [27].

The inherent nonlinearity and potential non-autonomy of these dynamic systems render long-term prediction a daunting challenge. A natural strategy to simplify the learning task is to transform the original nonlinear system into one that evolves linearly. In our framework, this is achieved via the Koopman operator.

For a suitably chosen observation function $\mathbf{g} : \Gamma \rightarrow \mathcal{G}(\mathbb{R}^{d_\gamma} \times T)$, the time-dependent Koopman operator $\mathcal{K}_t^{t+\varepsilon}$ propagates the observable in a linear fashion:

$$\mathbf{g}(\gamma_{t+\varepsilon}) = \mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t),$$

with the instantaneous evolution given by

$$\partial_t \mathbf{g}(\gamma_t) = \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) - \mathbf{g}(\gamma_t)}{\varepsilon}.$$

Intuitively, rather than directly predicting the high-dimensional, nonlinear state $\gamma(x_t)$, we lift the system into a latent space via $\mathbf{g}(\gamma_t)$, where its evolution is governed by a linear operator. This linearization is key to enabling efficient long-term predictions.

To operationalize this idea, we integrate the Koopman operator framework with an Auto Encoder. The Auto Encoder learns a nonlinear mapping from the original PDE solution $\gamma(x_t)$ to a low-dimensional latent representation $\mathbf{z}_t \approx \mathbf{g}(\gamma_t)$. In this latent space, the evolution is approximated by a linear operator:

$$\mathbf{z}_{t+\varepsilon} \approx \mathcal{K}_t^{t+\varepsilon} \mathbf{z}_t.$$

This approach leverages the expressive power of deep learning to capture the complex features of the PDE solution while exploiting the simplicity of linear dynamics in the latent space for prediction.

By unifying Koopman operator theory with the Auto Encoder framework, our method transforms the original nonlinear dynamics into a linear evolution in the latent space, thereby simplifying the prediction task and enhancing computational efficiency. This formulation is consistent with the derivations and notation presented in the next subsection, ensuring a coherent theoretical foundation for our approach.

7.4. Time-Dependent Koopman Operator.

Partial differential equations (PDEs) are fundamental in science and engineering but frequently lack closed-form solutions, necessitating computational approximations [10, 15]. Let $\Phi = \Phi(D; \mathbb{R}^{d_\phi})$ be a Banach space of inputs and $\Gamma = \Gamma(D; \mathbb{R}^{d_\gamma})$ be a Banach space of solutions, where both spaces are defined on a bounded open set $D \subset \mathbb{R}^d$. In

traditional PDE solvers, one seeks to approximate a solution operator $\mathcal{Q}$ that maps the input function ϕ (with fixed boundary and initial conditions γ_B, γ_I) to the solution γ :

$$\partial_t \gamma(x_t) = (\mathcal{L}_\phi \gamma)(x_t) + \eta(x_t), \quad x_t \in D \times T, \quad (12)$$

$$\gamma(x_t) = \gamma_B, \quad x_t \in \partial D \times T, \quad (13)$$

$$\gamma(x_0) = \gamma_I, \quad x_0 \in D \times \{0\}. \quad (14)$$

Here, $T = [0, \infty)$ denotes the time domain, $\mathcal{L}_\phi$ is a differential operator dependent on ϕ , and $\eta(\cdot)$ is a prescribed function in an appropriate function space.

Rather than directly solving for the high-dimensional state $\gamma(x_t)$, our approach shifts the focus to a set of observables $\mathbf{g} : \Gamma \rightarrow \mathcal{G}$, where $\mathbf{g}(\gamma_t)$ encapsulates the essential features of the state at time t . Although the state evolution governed by the PDEs is nonlinear, we posit that there exists a *time-dependent Koopman operator* $\mathcal{K}_t^{t+\varepsilon}$ that propagates these observables in a linear fashion:

$$\mathbf{g}(\gamma_{t+\varepsilon}) = \mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t). \quad (15)$$

Taking the limit as $\varepsilon \rightarrow 0$, we obtain the infinitesimal generator of the Koopman operator:

$$\partial_t \mathbf{g}(\gamma_t) = \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) - \mathbf{g}(\gamma_t)}{\varepsilon}. \quad (16)$$

This formulation offers an intuitive perspective: while $\gamma(x_t)$ evolves nonlinearly, the observable $\mathbf{g}(\gamma_t)$ is governed by a linear operator. This linearization is the cornerstone of our method and underlies our integration of Koopman theory with an Auto Encoder, which efficiently predicts the evolution of time-series satellite image data.

Furthermore, the connection to the Lax pair $(\mathcal{M}, \mathcal{N})$ in integrable PDEs—where $\mathcal{M}$ satisfies an eigenvalue problem and its evolution is determined by $\mathcal{N}$ via

$$\partial_t \mathcal{M} + [\mathcal{M}, \mathcal{N}] = 0, \quad (17)$$

and identifying

$$\mathcal{N} = \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) - \mathbf{g}(\gamma_t)}{\varepsilon}, \quad (18)$$

reinforces the rigorous foundation of the Koopman approach. This connection justifies employing Koopman operators for PDE-based time-series forecasting in our framework.

7.5. Koopman Operator Approximation.

Building upon the intuitive derivation above, we now describe a computational strategy for approximating the Koopman operator in a finite-dimensional setting. Inspired by dynamic mode decomposition (DMD) techniques—such as Hankel-DMD [1], sHankel-DMD [8], and HAVOK

[5]—we construct a Krylov sequence of observables with a temporal step size $\varepsilon \in [0, \infty)$:

$$\mathcal{R}_n = [\mathbf{g}(\gamma_0), \mathbf{g}(\gamma_\varepsilon), \mathbf{g}(\gamma_{2\varepsilon}), \dots, \mathbf{g}(\gamma_{n\varepsilon})], \quad (19)$$

$$\mathcal{R}_n = [\mathbf{g}(\gamma_0), \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0), \mathcal{K}_\varepsilon^{2\varepsilon} \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0), \dots, \mathcal{K}_{(n-1)\varepsilon}^{n\varepsilon} \dots \mathcal{K}_0^\varepsilon \mathbf{g}(\gamma_0)]. \quad (20)$$

This sequence is analogous to those used in Krylov subspace methods for eigenvalue computations [40]. To further capture the temporal dynamics, we arrange these observables into a structured Hankel matrix:

$$\mathcal{H}_{m \times n} = \begin{bmatrix} \mathbf{g}(\gamma_0) & \mathbf{g}(\gamma_\varepsilon) & \dots & \mathbf{g}(\gamma_{n\varepsilon}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}(\gamma_{(m-1)\varepsilon}) & \mathbf{g}(\gamma_{m\varepsilon}) & \dots & \mathbf{g}(\gamma_{(m+n-1)\varepsilon}) \end{bmatrix}. \quad (21)$$

Here, $m \in \mathbb{N}^+$ denotes the embedding dimension, and the columns of $\mathcal{H}_{m \times n}$ span a Krylov subspace $\mathbb{K} \subset \mathcal{G}(\mathbb{R}^{d_\gamma} \times T)$:

$$\mathbb{K} = \text{span}(\mathcal{R}_n) \simeq \text{span}(\mathcal{H}_{(m,n)}). \quad (22)$$

If $n \geq \dim(\mathbb{K}) - 1$, we approximate the Koopman operator by projecting it onto $\mathbb{K}$. Let $\hat{\mathcal{K}}_t^{t+\varepsilon} : \mathcal{G}(\mathbb{R}^{d_\gamma} \times T) \rightarrow \mathbb{K}$ be this Galerkin projection. For any test function $\mathbf{h} \in \mathcal{G}(\mathbb{R}^{d_\gamma} \times T)$, we enforce

$$\langle \hat{\mathcal{K}}_t^{t+\varepsilon} \mathbf{h}(\gamma_t), \mathbf{g}(\gamma_{i\varepsilon}) \rangle = \langle \mathcal{K}_t^{t+\varepsilon} \mathbf{h}(\gamma_t), \mathbf{g}(\gamma_{i\varepsilon}) \rangle \quad (23)$$

$$\forall i = 0, \dots, m,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Under appropriate conditions, as $m \rightarrow \infty$, the projected operator $\hat{\mathcal{K}}_t^{t+\varepsilon}$ converges to the true Koopman operator $\mathcal{K}_t^{t+\varepsilon}$ [22]:

$$\lim_{m \rightarrow \infty} \int_{\mathcal{G}(\mathbb{R}^{d_\gamma} \times T)} \|\hat{\mathcal{K}}_t^{t+\varepsilon} \mathbf{h}(\gamma_t) - \mathcal{K}_t^{t+\varepsilon} \mathbf{h}(\gamma_t)\|_F d\mu = 0, \quad \forall \mathbf{h}. \quad (24)$$

In practice, we approximate the Koopman operator by enforcing that the Hankel matrix evolves as

$$\mathcal{H}_{m \times n}(k+1) = \hat{\mathcal{K}}_{k\varepsilon}^{(k+1)\varepsilon} \mathcal{H}_{m \times n}(k), \quad \forall k = 1, \dots, n, \quad (25)$$

where $\mathcal{H}_{m \times n}(k)$ denotes the k -th column of $\mathcal{H}_{m \times n}$.

A major challenge in applying these ideas to time-dependent systems is the computational cost of online optimization for capturing temporal evolution. To alleviate this, following [25], we assume that for sufficiently small ε , the Koopman operator exhibits approximate temporal invariance. This assumption allows us to perform an offline

temporal averaging:

$$\begin{aligned}\bar{\mathcal{K}}_\varepsilon &= \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{g}(\gamma_\tau)^{-1} \mathbf{g}(\gamma_{\tau+\varepsilon}) d\tau \\ &\simeq \operatorname{argmin}_{P \in \mathbb{R}^{d_\gamma+1}} \sum_{k=1}^n \|\mathcal{H}_{m \times n}(k+1) - P \mathcal{H}_{m \times n}(k)\|_F.\end{aligned}\quad (26)$$

For a fixed ε , this averaged operator $\bar{\mathcal{K}}_\varepsilon : \mathcal{G}(\mathbb{R}^{d_\gamma} \times T) \rightarrow \mathbb{K}$ provides a computationally efficient offline approximation of the Koopman operator, suitable for long-term prediction tasks in PDE applications.

The successful implementation of this framework depends critically on two factors: achieving high temporal resolution via offline optimization of the Koopman operator, and designing an observation function $\mathbf{g}(\cdot)$ that satisfies ergodicity conditions within acceptable error bounds. Although these challenges are significant, our exploration of neural network architectures—specifically, Koopman Auto Encoders—offers a promising pathway to effectively integrate Koopman theory into the prediction of time-series satellite imagery.

7.6. Neural Approximation of Koopman Operators

A key challenge in implementing this approach lies in designing an effective neural architecture to enable Koopman-operator-driven PDE solutions. Here, we present the ViTKoop with koopman operator-based multi-step forecasting framework, which comprises six integrated components:

- (1) **Latent Encoding:** The initial PDE input $\phi_t = \phi(D \times t)$ from Eqs. (12-14) undergoes transformation through an encoder network to produce $\mathbf{g}(\hat{\gamma}_t)$ in space $\mathcal{G}(\mathbb{R}^{d_{\hat{\gamma}}} \times T)$. This encoder implements the observation function $\mathbf{g}(\cdot)$ using a single nonlinear layer with $\tanh(\cdot)$ activation. Refer to Figure 3 for visual representation.
- (2) **Fourier Transformation:** A Fourier transform maps $\mathbf{g}(\hat{\gamma}_t)$ to its frequency domain representation $\mathbf{g}_{\mathcal{F}}(\hat{\gamma}_t) = \mathcal{F} \circ \mathbf{g}(\hat{\gamma}_t)$. Following [26], we employ fast Fourier transform with frequency mode truncation at ω . This transformation serves two purposes:

First, it facilitates efficient computation of iterative updates:

$$\gamma(x_{t+\varepsilon}) = \int_{D \times t} \mathcal{J} \phi(xt, y_t) \eta(y_t) dy_t, \quad \forall; x_t \in D \times t. \quad (27)$$

Second, the frequency truncation naturally decomposes the system into low-frequency components, which typically exhibit greater stability and ergodicity, and high-frequency components requiring separate treatment. This decomposition enables focused Koopman operator learning on more stable modes while preserving high-frequency information through dedicated processing.

- (3) **Koopman Operator Learning:** Working with $\mathbf{g}_{\mathcal{F}}(\hat{\gamma}_t)$ at timesteps $t \in \varepsilon\mathbb{N}^+$, we construct a Hankel matrix $\hat{\mathcal{H}}_{m \times n}$ with embedding dimension m . The framework employs a $o \times o$ linear layer to learn $\bar{\mathcal{K}}_\varepsilon^r : \mathcal{G}(\mathbb{R}^{d_{\hat{\gamma}}} \times T) \rightarrow \mathbb{K}$ following Eqs. (25-26). This enables future state prediction through $\mathbf{g}_{\mathcal{F}}(\hat{\gamma}_{(m+n)\varepsilon}) = [\bar{\mathcal{K}}_\varepsilon^r \hat{\mathcal{H}}_{m \times n}(n)]^T(m)$. The parameter r provides temporal resolution control, allowing finer-grained evolution modeling.
- (4) **Inverse Spectral Transform:** Predicted states undergo inverse Fourier transformation: $\mathbf{g}(\hat{\gamma}_{(m+n)\varepsilon}) = \mathcal{F}^{-1} \circ \mathbf{g}_{\mathcal{F}}(\hat{\gamma}_{(m+n)\varepsilon})$. This reconstruction excludes previously filtered high-frequency components, which are handled separately.
- (5) **Low-level Component Processing:** Leveraging the property that shallow convolutional layers amplify low-level components [33], we implement network $\mathcal{C}$ to extract and predict low-level information $\mathbf{g}_{\mathcal{C}}(\hat{\gamma}_t)$. The prediction follows $[\mathbf{g}_{\mathcal{C}}(\hat{\gamma}_{(i+1)\varepsilon}), \dots, \mathbf{g}_{\mathcal{C}}(\hat{\gamma}_{(i+m+1)\varepsilon})]^T = \mathcal{C}[\mathbf{g}(\hat{\gamma}_{i\varepsilon}), \dots, \mathbf{g}(\hat{\gamma}_{(i+m)\varepsilon})]^T$. For complex PDEs, advanced architectures incorporating inception modules [44] may be employed.
- (6) **Image Reconstruction:** The framework combines predictions from both pathways through weighted averaging: $\mathbf{g}_{\mathcal{U}}(\hat{\gamma}_{(m+n)\varepsilon}) = (1-\lambda)\mathbf{g}(\hat{\gamma}_{(m+n)\varepsilon}) + \lambda\mathbf{g}_{\mathcal{C}}(\hat{\gamma}_{(m+n)\varepsilon})$, where $\lambda \in [0, 1]$ balances spectral components. A nonlinear decoder with $\tanh(\cdot)$ activation implements $\mathbf{g}^{-1}(\cdot) \simeq \mathbf{g}_{\mathcal{U}}^{-1}(\cdot)$, producing the final solution state $\hat{\gamma}_{(m+n)\varepsilon}$ in $\mathbb{R}^{d_{\hat{\gamma}}}$.

Based on the framework described in upper mechanisms, we propose an alternative iterative update approach that differs from Eq. (27). Given $t' > t \in \varepsilon\mathbb{N}$, our formulation is expressed as:

$$\begin{aligned}\hat{\gamma}_{t'} &= \left[\mathbf{g}^{-1} \left(\underbrace{\mathcal{F}^{-1} \circ \bar{\mathcal{K}}_\varepsilon^r \circ \mathcal{F} \circ \mathbf{g}(\hat{\gamma}_{[t-m\varepsilon, t]})}_{\text{Processing in component 1-4}} \right. \right. \\ &\quad \left. \left. + \underbrace{\mathcal{C} \circ \mathbf{g}(\hat{\gamma}_{[t-m\varepsilon, t]})}_{\text{Integration of component 1 and 5}} \right) \right]^T(m).\end{aligned}\quad (28)$$

Here, $\hat{\gamma}_{[t-m\varepsilon, t]}$ represents the vector $[\hat{\gamma}_{t-m\varepsilon}, \dots, \hat{\gamma}_t]$, where $m \in \mathbb{N}$ denotes the delay-embedding dimension. Drawing inspiration from the architecture of Fourier neural operators [26], a multi-unit koopman operator structure can be constructed by sequential replication of **component (2)-(5)** x times.

The objective function for our koopman operator framework is formulated as:

$$\begin{aligned}\mathcal{L} &= \alpha \|\hat{\gamma}_{t'} - \gamma_{t'}\|_F \\ &\quad + \beta \sum_{i=0}^m \|\mathbf{g}^{-1} \circ \mathbf{g}(\hat{\gamma}_{t-i\varepsilon}) - \gamma_{t-i\varepsilon}\|_F,\end{aligned}\quad (29)$$

where parameters $\alpha, \beta \in (0, \infty)$ determine the relative importance of prediction and reconstruction components. This approach of decomposing the loss function into distinct components aligns with methodologies employed in recent studies on nonlinear dynamic system identification [6, 7, 12]. While our framework shares conceptual elements with previous work on neural network parameterization of Koopman operators [29], it introduces novel elements specifically designed for PDE solutions, including frequency-based decomposition and adaptive temporal resolution. We now proceed to demonstrate the effectiveness of our approach through various computational experiments.

7.7. Forecasting Algorithm

Our forecasting algorithm predicts future frames in a satellite imagery time series using an iterative autoregressive approach. Algorithm 1 outlines the prediction procedure. The algorithm takes a batch of input data and outputs the predicted future images. First, we extract the input data and the corresponding mask from the batch (lines 1-2). The mask helps focus on relevant regions during prediction.

The core of our approach is the iterative prediction loop (lines 3-12), which generates predictions for each future time step from 0 to $T_{\text{out}} - 1$. At each iteration:

- We extract the ground-truth target frame for the current time step t for evaluation.
- We apply our kernel function to the current input sequence (x_t), which produces two outputs: the predicted next frame (`im_pred`) and a reconstruction of the input (`im_re`). The kernel implements a spatio-temporal model capturing both spatial patterns and temporal dynamics.
- We calculate the temporal prediction error between the predicted frame and the ground-truth frame to evaluate accuracy.
- For the first time step ($t = 0$), we initialize the prediction tensor; for subsequent steps, we concatenate the new prediction along the temporal dimension.
- Crucially, we update the input sequence with the current prediction (line 12), allowing the model to use its own outputs for subsequent forecasts. This autoregressive mechanism enables increasingly longer-term predictions.

Algorithm 1 Forecasting Algorithm.

Input: Input batch

Output: Predicted future images

```

1: Extract input from batch:  $x \leftarrow \text{batch}$ 
2: Extract mask from batch:  $\text{mask} \leftarrow \text{batch}$ 
3: for  $t = 0$  to  $T_{\text{out}} - 1$  do
4:   Get target frame  $x_t$  at time  $t$ 
5:   Apply kernel to  $x_t$  sequence:  $\text{im\_pred}, \text{im\_re} \leftarrow \text{kernel}(x_t)$ 
6:   Calculate temporal error:  $\text{t\_error} \leftarrow \text{loss}(\text{im\_pred}, y)$ 
7:   if  $t = 0$  then
8:     Initialize predictions:  $\text{pred} \leftarrow \text{im\_pred}[:, -1 :, :, :]$ 
9:   else
10:    Append new prediction:  $\text{pred} \leftarrow \text{torch.cat}((\text{pred}, \text{im\_pred}[:, -1 :, :, :]), 1)$ 
11:   end if
12:   Update input sequence with current prediction:  $x \leftarrow \text{im\_pred}$ 
13: end for
14: return  $\text{pred}$ 
```
