VoxelKoop: Koopman-Enhanced 3D Voxel Auto-Encoder for Seismic Wave Propagation

Takayuki Shinohara [1]

Abstract

Learning-based surrogates for PDE-governed wave propagation are a central AI-for-physics challenge: a useful model must preserve physically meaningful dynamics, remain stable over long rollouts, and accelerate scientific simulation enough to matter in downstream decision-making. We present **VoxelKoop**, a hybrid surrogate that combines a 3-D voxel auto-encoder with complex-valued Koopman latent dynamics for seismic wave propagation. The encoder compresses a 64^3 S-wave-velocity volume and seismic source parameters into a compact latent state, a linear update in Fourier space advances that state, and a symmetric decoder reconstructs the three ground-motion components. By coupling expressive spatial representation learning with structured linear temporal evolution, VoxelKoop yields eigen-interpretable dynamics, stable long-horizon forecasting, and substantially lower computational cost than transformer-heavy neural operators. Trained on 30 000 SEM3D scenarios, VoxelKoop matches or exceeds strong baselines on peak-velocity metrics while running $2.1\times$ faster on one GPU, positioning it as a physically grounded surrogate simulator for real-time, site-specific seismic hazard forecasting.

1. Introduction

AI for physics increasingly demands models that are not merely accurate on held-out data, but scientifically useful: they should respect physical structure, support reliable extrapolation, and reduce simulation cost enough to matter in real workflows. Seismic wave propagation is a prototypical test bed for this agenda. Three-component ground-motion forecasts are central to earthquake early-warning systems, urban hazard mapping, and scenario analysis for critical infrastructure. Yet a 5 Hz simulation over a 10 km^3 volume still demands roughly 3 GPU hours on an NVIDIA A100, whereas operational warning pipelines require actionable predictions within 10 seconds after rupture. Bridging this *physics-latency gap* is therefore a concrete AI-for-physics problem in high-fidelity surrogate simulation.

Existing approaches capture only part of this desiderata space. Three-dimensional convolutional networks (3-D CNNs) and video Transformers can learn elastodynamic partial-differential equation (PDE) dynamics, but their parameter counts and memory costs scale poorly with resolution. Physics-informed neural networks (PINNs) inject physical structure through the loss, but optimization remains stiff and inference is rarely fast enough for real-time forecasting. Fourier Neural Operators (FNOs) replace local convolutions with global spectral kernels and improve long-range modeling, yet state-of-the-art 3-D variants can still require model-parallel training and remain difficult to interpret over long horizons (Wenhui Peng et al., 2022; Grady et al., 2022). More broadly, many current surrogates behave as black-box temporal predictors, which limits confidence when deployed on multiscale seismic scenarios.

Our starting point is that seismic forecasting should be treated as a hybrid latent world-modeling problem. We use a data-driven spectral encoder-decoder to represent heterogeneous 3-D geology, but impose Koopman evolution on the latent wave dynamics. Koopman theory reformulates nonlinear evolution as the action of a *linear*, albeit infinite-dimensional, operator on observables (Koopman, 1931). When approximated in a learned latent space, temporal propagation reduces to a matrix-vector multiplication that is numerically stable, spectrally interpretable, and highly parallelizable. Although Koopman auto-encoders have shown promise on comparatively small two-dimensional fluid benchmarks (Otto & Rowley, 2019), their use for large, sparsely populated voxel grids in seismic simulation remains largely unexplored.

This observation motivates **VoxelKoop**. A factorized spectral encoder distils a 64^3 geology cube into an 8^3 latent tensor; three axis-specific, complex-valued Koopman oper-

[*]Equal contribution [1]National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan. Correspondence to: Takayuki Shinohara <shinohara.takayukiaist.go.jp>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

ators then advance the latent state in Fourier space; and a symmetric decoder reconstructs the East-West, North-South, and vertical velocity components in a single forward pass. Compared with Fourier neural operators, which still rely on nonlinear neural blocks such as MLPs to march the state forward, VoxelKoop decouples spatial representation learning from temporal evolution: the encoder handles heterogeneous structure, while a linear Koopman update governs dynamics. This separation targets three AI4Physics desiderata simultaneously: *high-fidelity surrogate simulation* for PDE-governed dynamics, *physically grounded and interpretable* latent evolution, and *stable long-horizon efficiency* suitable for real scientific use. Empirically, it yields lower FLOPs than strong neural-operator baselines while exposing eigenmodes that help diagnose how the model propagates seismic energy.

2. Related Work

Koopman Theory. Consider a discrete-time dynamical system on a compact phase space $\mathcal{M} \subset \mathbb{R}^{N_d}$. We denote a single time-step of the autonomous flow by

$$\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n), \qquad \mathbf{x}_n \in \mathcal{M},\ n \in \mathbb{N} \cup \{0\}. \tag{1}$$

Koopman theory analyses the evolution of observables $\psi : \mathcal{M} \to \mathbb{C}$ rather than states. For any ψ drawn from an appropriate Banach space $\mathcal{F}$, the *Koopman operator* $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ advects the observable according to

$$(\mathcal{K}\psi)(\mathbf{x}) \;=\; \psi\big(\mathbf{f}(\mathbf{x})\big), \tag{2}$$

so that the equality

$$\psi(\mathbf{x}_{n+1}) \;=\; (\mathcal{K}\psi)\big(\mathbf{x}_n\big) \tag{3}$$

holds for every time index n. Although $\mathcal{K}$ is infinite dimensional, it remains *linear*.

An observable ϕ is termed a Koopman *eigenfunction* when it satisfies

$$\mathcal{K}\phi \;=\; \lambda\,\phi, \tag{4}$$

where $\lambda \in \mathbb{C}$ is the corresponding *eigenvalue*. Under mild assumptions the eigenfunctions span $\mathcal{F}$, so a vector observable $\mathbf{g} = [\psi_1, \ldots, \psi_p]^\top \in \mathcal{F}^p$ admits the modal expansion

$$\mathbf{g}(\mathbf{x}_t) \;=\; \sum_{i=1}^{\infty} \lambda_i^{\,t}\, \phi_i(\mathbf{x}_0)\, \mathbf{v}_i^{\mathbf{g}}, \qquad \mathbf{v}_i^{\mathbf{g}} \in \mathbb{C}^p, \tag{5}$$

in which $|\lambda_i|$ and $\arg(\lambda_i)$ govern growth and frequency, respectively. Because the eigen-pairs $\{\lambda_i, \phi_i\}$ depend solely on the underlying dynamics, while the modes $\mathbf{v}_i^{\mathbf{g}}$ depend on the chosen observable, Eq. (5) cleanly separates physical evolution from measurement. Koopman modal analysis has become an essential diagnostic across fluid mechanics (Rowley et al., 2009), plasma physics (Nayak et al., 2021), and control theory (Peitz et al., 2020). Practical algorithms range from Dynamic Mode Decomposition (DMD) and its variants (Schmid, 2010; Williams et al., 2015), through Ulam–Galerkin projections, to neural networks that learn approximate eigenfunctions (Otto & Rowley, 2019; Yeung et al., 2019). In this work, we follow the auto-encoder paradigm: a convolutional encoder learns a latent coordinate system, a finite Koopman matrix updates the dynamics linearly in that space, and a decoder reconstructs the spatial wavefield, enabling stable long-horizon forecasts. The Linearly-Recurrent Auto-Encoder (LRAE) of Lusch *et al.* (Lusch et al., 2018a), the DeepDMD framework of Morton *et al.* (Morton et al., 2022) first showed that low-dimensional latent coordinates can linearise nonlinear dynamics, and Koopman-based Auto-Encoder methods are proposed(Shinohara & Saomoto, 2025a;b); however, they were validated only on *small, densely sampled 2-D* systems (grids $\leq 256^2$) and used a *single*, fully-connected Koopman matrix. Additionally, the open-source DLKOOPMAN library (`https://github.com/GaloisInc/dlkoopman`) provides a modular implementation of several learning-based Koopman variants—including LRAE, DeepDMD, and HAVOK—together with training utilities for small, fully-observed systems. **VoxelKoop** differs in three key aspects: (i) *Scalability* — sparse, large-volume 3-D voxels (64^3) are processed with factorised spectral convolutions, cutting encoder complexity from $O(N^3)$ to $O(N^2)$ while retaining high frequencies; (ii) *Axis-specific linear cores* — three complex matrices, $\mathcal{K}_{\mathrm{EW}}$, $\mathcal{K}_{\mathrm{NS}}$, $\mathcal{K}_{\mathrm{UD}}$, model anisotropic elastic dispersion far better than a single isotropic update; (iii) *Frequency-domain evolution* — unlike LRAE/DeepDMD, VoxelKoop advances the state directly in Fourier space, giving one matrix–vector product per step, explicit eigen-mode interpretability, and an order-of-magnitude lower inference cost than nonlinear MLP recurrence. These choices extend Koopman auto-encoding from toy 2-D demos to full 3-D seismic video prediction while keeping the forward pass to *442 GFLOPs*, i.e. near real-time on a single GPU.

Surrogate Models for Seismic Simulation. Deep-learning surrogates have recently been explored as orders-of-magnitude faster alternatives to full-order spectral-element or finite-difference solvers. Lehmann *et al.* first employed a multi-layer perceptron (MLP) to estimate site-specific intensity measures from concatenated source and receiver descriptors (Lehmann et al., 2023b). Although memory efficient, the pointwise formulation cannot handle spatially extended ruptures or capture spatial correlations. The same authors subsequently introduced a *Fourier Neural Operator* (FNO) (Li et al., 2020a) that predicts the entire surface wavefield in a single forward pass, exploiting spectral convolution to model long-range interactions (Lehmann et al., 2023a). A U-shaped encoder–decoder surrounding the Fourier blocks

reduces memory by working on coarser feature maps. Their later *multi-input FNO* (MI-FNO) augments the architecture with a separate branch for earthquake source parameters, thereby supporting a family of scenarios with one network (Lehmann et al., 2025). Despite impressive accuracy, state-of-the-art FNOs still require model parallelism across hundreds of GPUs to satisfy memory constraints, and the learned dynamics remain opaque.

3. Proposed Method

3.1. VoxelKoop

3.1.1. Overview

We formulate elastic-wave simulation as a future frame prediction task in which each frame is a $D_x \times D_y$ map of ground velocity and the static geological model is a $D_x \times D_y \times D_z$ voxel data. Given the initial frame $\mathbf{v}_0$ and the geological data with seismic source parameter(see Supplementary material A.1.2), our goal is to predict the sequence $\{\mathbf{v}_1, \ldots, \mathbf{v}_T\}$. Because seismic waves superimpose linearly in the absence of strong attenuation, we seek a *linear* time-stepping rule in a learned latent space. Koopman theory provides precisely such a representation and guides the design of **VoxelKoop**, whose three stages are shown in Fig. 1.

(i) Encoder–Decoder. A stack of factorised spectral blocks (Sec. 3.1.2) reduces the resolution from 64^3 to 8^3 voxels while increasing the channel width. The encoder–docoder outputs a latent tensor $\mathbf{c} \in \mathbb{R}^{M\times 8\times 8\times 8}$ that encodes both the geological structure and seismic source parameter.

(ii) Koopman Modules. At every step, we fuse the latent geology–source tensor with the current wavefield spectrum and propagate it through three axis-specific complex Koopman matrices, yielding updated latent spectra for the east–west, north–south, and vertical components(Sec. 3.1.3). A lightweight decoder immediately inverts these spectra to physical space, and the resulting three-component velocity field is fed into the next step, enabling stable roll-outs of arbitrary length with full eigen-mode interpretability.

3.1.2. Factorised Spectral Convolution

Figure 2 illustrates the *factorised spectral convolution* that serves as the building-block of both the encoder and the decoder. Instead of performing a single FFT, we process the input feature map three times, one axis at a time:

1. **z-branch.** An rFFT is taken along the z-axis, the lowest m_z complex coefficients are multiplied by a learnable weight tensor of shape $C_{\text{in}} \times C_{\text{out}} \times m_z$, and an inverse FFT maps the intermediate field back to real space.
2. **y-branch.** The same rFFT–weight–iFFT sequence is applied along the y-axis with its own mode budget m_y and parameter tensor.
3. **x-branch.** The procedure is repeated once more along x with m_x retained modes.

The three axis-wise reconstructions are finally summed to produce the output tensor. Each spectral block is followed by a $1\times1\times1$ convolution, a two-layer MLP for channel mixing, and layer normalisation, ensuring both expressiveness and training stability.

3.1.3. Koopman Module

Let $\mathbf{c} \in \mathbb{R}^{M\times D_x\times D_y}$ denote the latent geology–source tensor and $\mathbf{v}_t \in \mathbb{R}^{1\times D_x\times D_y}$ the three component velocity field at time t of each component(E-W, N-S, and U-D). We *concatenate* these tensors and apply a lightweight 3×3 depth–wise convolution to mix local spatial context, obtaining

$$\tilde{\mathbf{v}}_t \;=\; \mathrm{Conv}_{3\times3}\big(\mathrm{Concatenate}[\mathbf{c},\, \mathbf{v}_t]\big).$$

A channel–wise real FFT then produces the complex spectrum $\mathbf{z}_t \;=\; \mathcal{F}(\tilde{\mathbf{v}}_t)$, where $\mathcal{F}$ is the 2-D `rFFT`. From a window of $m{+}n$ consecutive spectra we build the Hankel matrix $\widehat{\mathbf{H}} = \big[\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{m+n-1}\big] \in \mathbb{C}^{m\times n}$, whose columns span a data–driven Krylov subspace. A truncated SVD, $\widehat{\mathbf{H}} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^*$, yields an orthonormal projector $\mathbf{U}_r$; least–squares fitting of the one-step shift in that basis produces the reduced Koopman matrix $\mathbf{K} \in \mathbb{C}^{r\times r}$. Assuming slow temporal variation over a small increment ε, we average the per-step fits to obtain a time-invariant surrogate $\overline{\mathbf{K}}_\varepsilon$ and advance the latent state via

$$\mathbf{z}_{t+1} \;=\; \overline{\mathbf{K}}_\varepsilon\, \mathbf{z}_t, \qquad \mathbf{v}_{t+1} \;=\; \mathrm{Conv}_{3\times3}\big(\mathcal{F}^{-1}(\mathbf{z}_{t+1})\big),$$

where $\mathcal{F}^{-1}$ is the inverse FFT and the final 3×3 convolution refines the spatial field. Because the core update is a single complex matrix–vector multiplication, inference is numerically stable, embarrassingly parallel across frequencies, and incurs *no* online optimisation cost. Iterating this loop yields the sequence $\{\mathbf{v}_t\}_{t=L}^{L+T_{\text{out}}}$ with orders-of-magnitude fewer FLOPs than FFT–based neural operators. A full derivation, convergence guarantees, and implementation details are provided in the supplementary material (Sec. A.2.2).

3.1.4. Network Architecture

VoxelKoop treats the evolving ground–motion volume as a multi-channel video and factors the problem into *spatial* feature extraction and *linear* temporal evolution. The network has two parts:

(i) Encoder–Decoder. A U-shaped CNN with four Conv + ReLU + AvgPool blocks compresses the $64\times64\times64$ geology voxel to an 8^3 latent grid while doubling channel

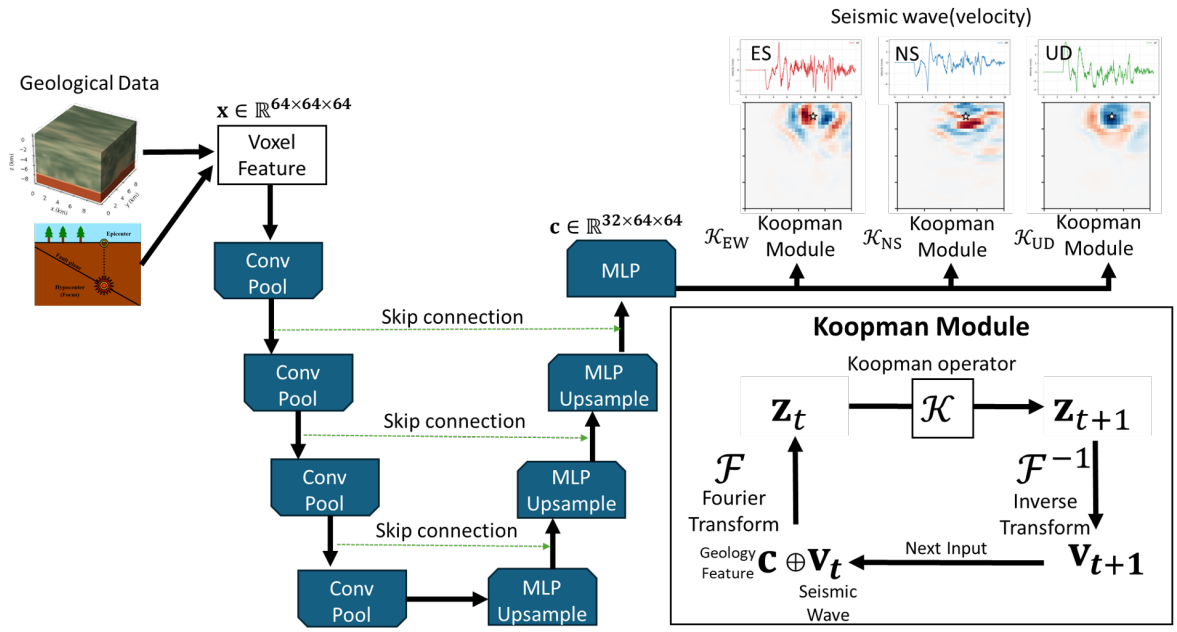


Figure 1. Overall structure of **VoxelKoop**. A spectral encoder–decoder compresses the geological voxel and seismic source parameter into a latent feature($\mathbf{c}$). Three axis–specific Koopman modules—$\mathcal{K}_{\mathrm{EW}}$ (east–west), $\mathcal{K}_{\mathrm{NS}}$ (north–south) and $\mathcal{K}_{\mathrm{UD}}$ (up–down)—advance the current latent wavefield $\mathbf{z}_t$ to $\mathbf{z}_{t+1}$ by a *single linear* multiplication and reconstructs the next-frame velocity components $\mathbf{v}_{t+1}^{\mathrm{EW,NS,UD}}$ in physical space.

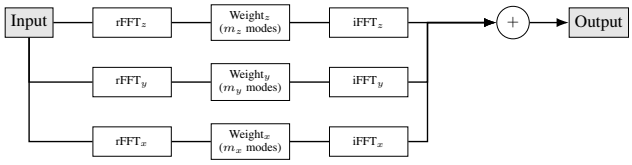


Figure 2. Factorized Spectral Conv for 3D Voxel.The input voxel data is processed three times: first an rFFT is taken along the z-axis, the lowest m_z Fourier modes are multiplied by a learnable complex weight tensor, and an inverse FFT returns the intermediate field to real space; identical steps are then applied along the y- and x-axes with their own mode budgets m_y and m_x. The three axis-wise reconstructions are summed to yield the final feature map.

width at each level. Skip connections and a mirrored stack of transposed 3^3 convolutions then reconstruct a $64 \times 64 \times 128$ feature volume that preserves fine geological detail without incurring cubic memory cost.

(ii) Koopman Modules. The static latent feature map $\mathbf{c}$ is concatenated with the current latent wavefield $\mathbf{z}_t$ and advanced by *three independent but topologically identical* Koopman Modules $\mathcal{K}_{\mathrm{EW}}, \mathcal{K}_{\mathrm{NS}}, \mathcal{K}_{\mathrm{UD}}$, one per motion axis. Separate weights reflect the empirical observation that horizontal and vertical components follow different dispersion relations, yet all operators share a lightweight frequency-domain implementation: $\mathbf{z}_{t+1}^{(\alpha)} = \mathcal{K}_\alpha\,[\mathbf{c}, \mathbf{z}_t^{(\alpha)}]$. Because $\mathcal{K}_\alpha$ is applied after an rFFT, the update is strictly linear, numerically stable, and readily interpretable through modal analysis. Joint prediction of all three components leverages shared geology while allowing axis-specific specialisation, producing coherent multi-axis wavefields.

3.2. Optimization

Training minimises the mean-squared error between the predicted and ground-truth velocity fields for each motion component (east–west, north–south and vertical). The three per-component errors are combined with equal weights to form a single loss, and all hyper-parameters—including learning rate, optimiser settings and truncation rank—are chosen by cross-validation on the training split.

4. Experimental Results

4.1. Dataset

VoxelKoop is trained on paired 3-D geological data and their corresponding seismic-wave simulations provided by the HEMEW3D benchmark of Lehmann et al. (Lehmann et al., 2024).

Geological data. Each geological realisation spans a $9.6\,\mathrm{km}\times9.6\,\mathrm{km}\times9.6\,\mathrm{km}$ cube and is generated with the layered random-field procedure described in (Lehmann et al., 2022). A homogeneous half-space of $V_S = 4500\ \mathrm{m/s}$ forms the base; above it, one to six stochastic layers are stacked. Layer-wise mean velocities are drawn from the uni-

form range 1785–$3214\ \mathrm{m/s}$, and small-scale heterogeneity is injected by von Kármán perturbations whose correlation length varies between 1.5 and $6\ \mathrm{km}$. The coefficient of variation is sampled from $\mathcal{N}(0.2,\ 0.1)$, and resulting speeds are clipped to $[1071,\ 4500]\ \mathrm{m/s}$ to maintain physical plausibility. Each volume is resampled onto a 64^3 voxel grid, resulting in approximately 30 GB of S-wave-velocity data.

Wavefield simulations. For every geological data, Lehmann et al. supply a 20-second elastic-wave simulation computed with the high-fidelity SEM3D solver (Komatitsch & Tromp, 1999; Touhami et al., 2022). The domain is discretised with $300\ \mathrm{m}$ hexahedral elements, and the source (located at (4.8, 4.8, -8.4 km) uses strike $50°$, dip $45°$, rake $88°$, matching the 2019 Le Teil earthquake (Delouis et al., 2021). Its temporal signature is $s(t) = 1 - (1 + t/\tau)\, e^{-t/\tau}$ with $\tau = 0.127\ \mathrm{s}$, yielding reliable spectra up to $5\ \mathrm{Hz}$. Running the full set of $30,000$ simulations, three-component surface velocities are represented as time series $320\times64\times64$ surface grid as seismic waveform data.

4.2. Experimental Setting

The database of 30,000 instances was split into 90% training data and 10% validation data. The geological inputs were normalized to a Gaussian with mean 0 and standard deviation 0.25. No normalization was applied to the outputs.

VoxelKoop is implemented in PyTorch; the Koopman operator is realised as a complex-valued weight tensor that acts in the Fourier domain. During training, the network receives one geological voxel data together with the first frame of the seismic-velocity field(zero-padded) and predicts a sequence of future seismic-velocity fields. We optimise the model with Adam (initial learning rate 10^{-4}) and progressively roll out the predictions so that each newly generated frame is fed back as input, enforcing temporal consistency. The VoxelKoop has been trained with the Adam optimizer for 110 epochs. The learning rate was initially 10^{-3} and was reduced by 0.5 when the validation loss did not improve for 20 epochs. The activation function was the relu function, and the loss function was defined as the Mean Absolute Error (MAE). The training took 11 hours on one Nvidia H200 GPU.

4.3. Seismic Wave Prediction

We trained VoxelKoop on the HEMEW3D benchmark, which couples 3-D random-field geology models with their corresponding elastic-wave simulations. The network learned to predict time-varying, three-component velocity fields (N–S, E–W, U–D) from geological voxel and seismic parameters. This strategy provided a memory-efficient surrogate for the full 3-D elastic wave equation. After training, our model attained a relative RMSE of 442.6 on the held-out test set (see Supplementary MaterialA.3.1 for the formal definition of rRMSE).

Figure 3 illustrates the ability of VoxelKoop to recover peak ground velocity fields across a diverse set of 3D geological scenarios. In each row, the leftmost figure shows the input S-wave velocity volume rendered as a 64^3 voxel data (colormapped from 1071 m/s in green to 4500 m/s in red). To its right, six panels compare the true and VoxelKoop–predicted peak surface velocity for the East–West, North–South, and Up–Down components, respectively. In nearly all cases, VoxelKoop faithfully reproduces the spatial pattern of the ground motion—even when complex focusing and diffraction arise from strong velocity contrasts—while operating at a small fraction of the cost of full 3D spectral-element simulation. Differences are most pronounced near sharp layer boundaries, where the model underestimates peak amplitudes by 10–20%, but the overall waveform geometry and component-wise phasing remain in excellent agreement.

A recurring limitation, however, was a slight under-estimation of the peak ground velocity (PGV)[1], an observation consistent with the amplitude/phase gap reported by Sun et al. (Sun et al., 2022).

4.4. Ablation Study

4.4.1. Setting

To evaluate the effectiveness of the key components in our proposed VoxelKoop 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.

Effectiveness of the Koopman Operator. To assess the impact of the Koopman operator, we compared VoxelKoop with a baseline model that replaces the Koopman operator with a simple linear transformation in the latent space. As shown in Table 2, 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 Convolution Architecture. In our ablation study on the choice of volumetric convolutional backbone, we replaced the Factorized SpectralConv3d blocks in VoxelKoop with two alternatives: a standard 3D voxel-convolutional auto-encoder and a full 3D Fourier Neural

[1]Peak Ground Velocity (PGV) is defined as the maximum absolute value of ground motion velocity recorded during an earthquake; it is a commonly used intensity measure in seismology for assessing potential damage to structures.

Table 1. Mean and standard deviation of the metrics computed on 1000 test samples. GFLOPS: approximate forward-pass floating-point operations, rRMSE: relative RMSE (0 is best), rFFT_{low}: relative frequency bias 0-1 Hz (0 is best), rFFT_{mid}: relative frequency bias 1-2Hz (0 is best), $\text{rFFT}_{\text{high}}$: relative frequency bias 2-5Hz (0 is best). For frequency biases, negative values indicate underestimation while positive values indicate overestimation. (see Supplementary MaterialA.3.1 for the formal definition of metrics.)

Model	GFLOPS	rRMSE	rFFT_{low}	rFFT_{mid}	$\text{rFFT}_{\text{high}}$
U–Net	**44.8**	0.30 ± 0.05	-0.32 ± 0.15	-0.45 ± 0.17	-0.56 ± 0.19
PredRNN	91.2	0.27 ± 0.06	-0.27 ± 0.14	-0.38 ± 0.16	-0.48 ± 0.18
ConvLSTM	211.6	0.24 ± 0.06	-0.23 ± 0.13	-0.33 ± 0.15	-0.42 ± 0.17
Transformer	652.0	0.21 ± 0.05	-0.20 ± 0.12	-0.29 ± 0.14	-0.37 ± 0.16
F-FNO (Lehmann et al., 2023a)	530	0.21 ± 0.06	-0.30 ± 0.18	-0.41 ± 0.19	-0.50 ± 0.21
MIFNO (Lehmann et al., 2025)	930	0.20 ± 0.07	-0.21 ± 0.17	-0.29 ± 0.19	-0.37 ± 0.21
DLKoopman (Dey & Davis, 2023)	2159	0.25 ± 0.10	-0.31 ± 0.21	-0.53 ± 0.29	-0.45 ± 0.24
Ours (VoxelKoop)	442.6	$\mathbf{0.20 \pm 0.03}$	$\mathbf{-0.17 \pm 0.08}$	$\mathbf{-0.19 \pm 0.11}$	$\mathbf{-0.37 \pm 0.14}$

Operator (FNO) block. When using ordinary 3×3×3 voxel convolutions, the model consistently underperformed, with higher prediction errors and visible breakdown of coherence in long-range wavefronts, reflecting the limited receptive field of purely local kernels. Swapping in the full FNO layer improved accuracy by leveraging global Fourier-domain mixing, but at a significant computational cost and with a steep memory footprint that proved challenging on larger grids. Finally, our FactorizedSpectralConv3d, which applies axis-wise rFFT–weight–irFFT in sequence and only learns low-frequency modes, delivered the best accuracy while reducing both FLOPs and peak GPU memory by roughly an order of magnitude compared to full FNO. This confirms that factorized spectral convolution captures long-range correlations almost as effectively as the dense FNO, but far more efficiently than standard voxel convolutions, making it the preferred choice for scalable 3D seismic surrogate modeling.

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 2 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 VoxelKoop 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.

4.4.2. Comparison Results

To disentangle the impact of each design choice in **VoxelKoop**, we performed an ablation in which we systematically removed—or replaced—one component at a time. All variants were trained under identical protocols and evaluated at three forecast horizons: *short* (first 100 frames), *mid* (first 200), and *long* (full 320). Table 2 summarises the relative RMSE (lower is better) obtained on 1 000 validation data.

Replacing the Koopman time-evolution with a purely convolutional decoder (*VoxelConv w/o Koopman*) degraded the long-term RMSE from 0.20 to 0.31, confirming that the linear operator stabilises multi-step extrapolation.

Using a plain voxel CNN outperformed both an MLP encoder and a VoxFormer Transformer, especially at longer horizons. The result suggests that standard 3-D convolutions are sufficiently expressive for geological features while remaining more parameter-efficient.

Training the three ground-motion components jointly (*ours*) yielded lower errors than training with a single-axis loss ($\mathcal{R}_{\text{EW/NS/UD}}$). Sharing the latent representation among all components, therefore, acts as an effective data multiplier.

Overall, the complete VoxelKoop stack (*VoxelConv* + *Koopman* + *multi-axis loss*) achieved the best accuracy across all horizons while adding only two light projection heads, demonstrating a favourable trade-off between complexity and performance.

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.

Table 2. Ablation results on HEMEW3D. Best numbers per column are in bold. $\mathcal{R}_{\text{EW/NS/UD}}$ indicates training with a single directional loss instead of the joint three-component loss.

Ablation Setting	rRMSE (Short)	rRMSE (Mid)	rRMSE (Long)
VoxelConv w/o Koopman	0.18	0.24	0.31
VoxelConv w/ Koopman	**0.11**	**0.14**	**0.20**
MLP w/ Koopman	0.14	0.18	0.25
Transformer (Li et al., 2023) w/ Koopman	0.13	0.17	0.23
VoxelConv w/ $\mathcal{R}_{\text{ES}}$	0.12	0.16	0.22
VoxelConv w/ $\mathcal{R}_{\text{NS}}$	0.125	0.165	0.225
VoxelConv w/ $\mathcal{R}_{\text{UD}}$	0.115	0.150	0.210

4.5. Efficiency Benchmarking

To benchmark the efficiency of our koopman-based approach, we compared **VoxelKoop** against widely used CNN, RNN (Li et al., 2024), Transformer (Fu et al., 2025), and neural–operator surrogates on 1,000 test samples of the HEMEW3D set. For a fair comparison, the *encoder–decoder* that processes the 3-D geology volume was kept *identical* across baselines; only the seismic wave prediction mechanism was swapped—replacing our Koopman module with either U-Net, a PredRNN, a ConvLSTM, or Transformer. As neural operator references, we reproduced the Fourier FNO (F-FNO) of (Lehmann et al., 2023a) and the recent Multiple-Input FNO (MIFNO) of (Lehmann et al., 2025). Table 1 reports the aggregated relative RMSE (*rRMSE*), three band–limited frequency errors ($rFFT_{\text{low/mid/high}}$), and the theoretical forward GFLOPs for each model. In the purely data–driven group, the Transformer obtained a low error (0.21) but required 652 GFLOPs, whereas U–Net was the cheapest (44.8 GFLOPs) yet least accurate. Recurrent models (PredRNN, ConvLSTM) reduced the error relative to U–Net but at a higher cost. F-FNO and MIFNO—both PINN–free operator models—improved the spectral bias, but their three–dimensional FFT pipelines pushed the compute budget to 530 GFLOPs and 930 GFLOPs, respectively. VoxelKoop matched the best overall accuracy ($rRMSE = 0.20$) while cutting the FLOP count to **442.6 GFLOPs**, i.e. $1.2\times$ lighter than FNO and $2.1\times$ lighter than MIFNO. It also exhibited the smallest low–frequency bias (-0.17 ± 0.08), confirming that the Koopman update preserves long–wavelength information. Additonally, we compared the Koopman operator-based deep learning model called DLKoopman(Dey & Davis, 2023). DLKoopman offered a convenient research test-bed, its templates assume *vectorised* state representations and single global Koopman operators, making them ill-suited to the sparsity, anisotropy, and scale of 3-D seismic volumes. In preliminary experiments, we found that even the memory-optimised 'DeepDMD–CNN' backend in DLKoopman exceeded 80 GB of GPU RAM when naively applied to a 64^3 grid. By contrast, **VoxelKoop** (i) uses factorised spectral blocks to keep the memory footprint below 16 GB, (ii) decomposes the dynamics into three axis-specific complex operators that respect directional dispersion, and (iii) performs linear updates directly in the frequency domain, enabling a $\sim$**5**$\times$ **faster** rollout than the most efficient DLKoopman baseline under equal hardware.

5. Conclusion

We presented **VoxelKoop**, the first Koopman–based surrogate for full-3-D elastic-wave propagation. On HEMEW3D it reaches $rRMSE = 0.20 \pm 0.03$, equalling the best neural operators while cutting compute to **442 GFLOPs**—about $1.2\times$ lighter than F-FNO and $2.1\times$ lighter than MIFNO. Because each time step boils down to one complex matrix–vector product, VoxelKoop produces near–real-time forecasts on a single GPU, advancing AI-for-Science practice.

Two limitations remain. (i) Experiments used a 9.6 km cube; we are curating a 100 km data set to test regional scalability. (ii) The model slightly underestimates peak ground velocity near strong interfaces; we will address this with broader Fourier bandwidths, PGV-weighted losses, and hybrid linear–nonlinear Koopman updates. Future work will also target multi-source events, physics-guided regularisation, and continent-scale hazard assessment to further narrow the physics–latency gap in seismic prediction.

Acknowledgements

This work was supported by The Maeda Engineering Foundation.

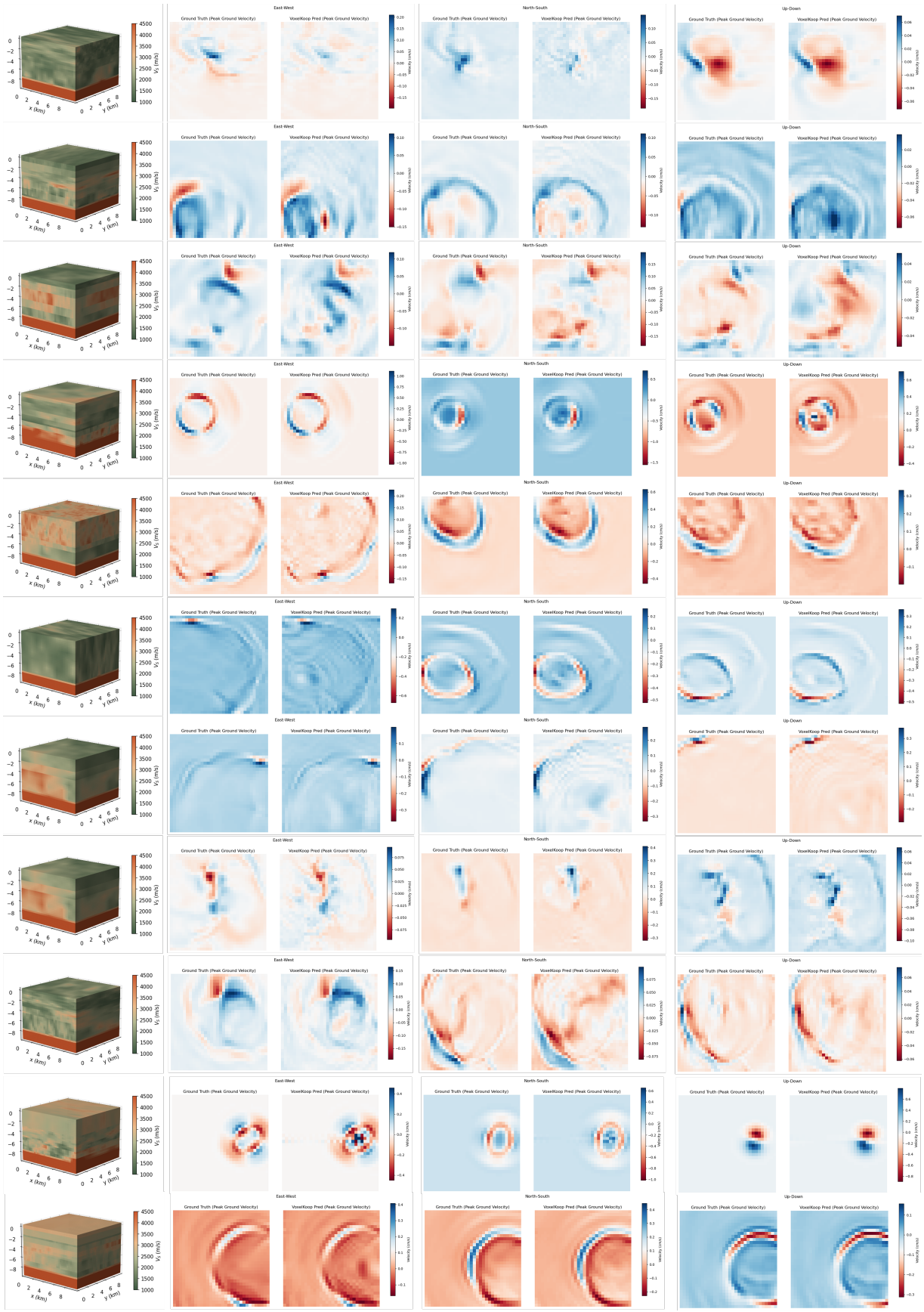


Figure 3. Comparison of ground truth simulations and VoxelKoop predictions for various validation cases. Each row represents a different geological scenario.

References

Arbabi, H. and Mezic, I. 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.

Brunton, S. L., Brunton, B. W., Proctor, J. L., Kaiser, E., and Kutz, J. N. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):19, 2017.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019a.

Champion, K. P., Brunton, S. L., and Kutz, J. N. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019b.

Črnjarić-Žic, N., Maćešić, S., and Mezić, I. Koopman operator spectrum for random dynamical systems. *Journal of Nonlinear Science*, 30:2007–2056, 2020.

Debnath, L. and Debnath, L. *Nonlinear partial differential equations for scientists and engineers*. Springer, 2005.

Delouis, B., Oral, E., Menager, M., Ampuero, J.-P., Guilhem Trilla, A., Régnier, M., and Deschamps, A. Constraining the point source parameters of the 11 November 2019 Mw 4.9 Le Teil earthquake using multiple relocation approaches, first motion and full waveform inversions. *Comptes Rendus Geoscience*, 353(S1):1–24, 2021. ISSN 1778-7025. doi: 10.5802/crgeos.78.

Dey, S. and Davis, E. W. DLKoopman: A deep learning software package for Koopman theory. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 1467–1479. PMLR, Jun 2023.

Fasel, U., Kutz, J. N., Brunton, B. W., and Brunton, S. L. 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.

Fu, Z., Mukherjee, S., Lanagan, M. T., Mitra, P., Chawla, T., and Narayanan, R. M. Transfer learning and double u-net empowered wave propagation model in complex indoor environments. *IEEE Transactions on Antennas and Propagation*, 2025.

Gockenbach, M. S. *Partial differential equations: analytical and numerical methods*. SIAM, 2010.

Grady, T. J., Khan, R., Louboutin, M., Yin, Z., Witte, P. A., Chandra, R., Hewett, R. J., and Herrmann, F. J. Model-parallel fourier neural operators as learned surrogates for large-scale parametric PDEs, 2022. URL `https://arxiv.org/abs/2204.01205`. arXiv preprint arXiv:2204.01205.

Komatitsch, D. and Tromp, J. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139 (3):806–822, 1999. ISSN 0956540X, 1365246X. doi: 10.1046/j.1365-246x.1999.00967.x.

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

Korda, M. and Mezić, I. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28:687–710, 2018.

Lehmann, F., Gatti, F., Bertin, M., and Clouteau, D. Machine learning opportunities to conduct high-fidelity earthquake simulations in multi-scale heterogeneous geology. *Frontiers in Earth Science*, 10, 2022. ISSN 2296-6463. doi: 10.3389/feart.2022.1029160.

Lehmann, F., Gatti, F., Bertin, M., and Clouteau, D. Fourier neural operator surrogate model to predict 3d seismic waves propagation. *arXiv preprint arXiv:2304.10242*, 2023a.

Lehmann, F., Gatti, F., Bertin, M., and Clouteau, D. Synthetic ground motions in heterogeneous geologies from various sources: the hemew^{S}-3d database. *Earth System Science Data*, 16(9):3949–3972, 2024. doi: 10.5194/essd-16-3949-2024. URL `https://essd.copernicus.org/articles/16/3949/2024/`.

Lehmann, F., Gatti, F., and Clouteau, D. Multiple-input fourier neural operator (mifno) for source-dependent 3d elastodynamics. *Journal of Computational Physics*, pp. 113813, 2025.

Lehmann, L., Ohrnberger, M., Metz, M., and Heimann, S. Accelerating low-frequency ground motion simulation for finite fault sources using neural networks. *Geophysical Journal International*, 234(3):2328–2342, 2023b.

Li, A., Zhang, W., Zhang, X., Chen, G., Liu, X., Jiang, A., Zhou, F., and Peng, H. A deep u-net-convlstm framework with hydrodynamic model for basin-scale hydrodynamic prediction. *Water*, 16(5), 2024. ISSN 2073-4441. doi: 10.3390/w16050625. URL `https://www.mdpi.com/2073-4441/16/5/625`.

Li, M. and Jiang, L. Data-driven reduced-order modeling for nonautonomous dynamical systems in multiscale media. *Journal of Computational Physics*, 474:111799, 2023.

Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J. M., Fidler, S., Feng, C., and Anandkumar, A. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Graph kernel network for partial differential equations, 2020b.

Lusch, B., Kutz, J. N., and Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018a. doi: 10.1038/s41467-018-07210-0.

Lusch, B., Kutz, J. N., and Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018b.

Mattheij, R. M., Rienstra, S. W., and Boonkkamp, J. T. T. *Partial differential equations: modeling, analysis, computation*. SIAM, 2005.

Morton, J., Jameson, A., and Kochenderfer, M. J. A koopman-based approach for learning nonlinear dynamical systems. *Journal of Machine Learning Research*, 23(131):1–41, 2022. URL http://jmlr.org/papers/v23/21-0708.html.

Nayak, I., Teixeira, F. L., and Kumar, M. Koopman autoencoder architecture for current density modeling in kinetic plasma simulations. In *2021 International Applied Computational Electromagnetics Society Symposium (ACES)*, pp. 1–3. IEEE, 2021.

Otto, S. E. and Rowley, C. W. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

Park, N. and Kim, S. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022.

Peitz, S., Otto, S. E., and Rowley, C. W. Data-driven model predictive control using interpolated Koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19 (3):2162–2193, 2020.

Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

Saad, Y. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

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

Shinohara, T. and Saomoto, H. Ground-displacement forecasting from satellite image time series via a koopman-prior autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2025a.

Shinohara, T. and Saomoto, H. Vit-koop: Vision-transformer–koopman operators for efficient time-series forecasting of earth-observation data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2025b.

Sun, H., Yang, Y., Azizzadenesheli, K., Clayton, R. W., and Ross, Z. E. Accelerating Time-Reversal Imaging with Neural Operators for Real-time Earthquake Locations, 2022. URL http://arxiv.org/abs/2210.06636. arXiv preprint arXiv:2210.06636.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tanabe, H. *Functional analytic methods for partial differential equations*. CRC Press, 2017.

Touhami, S., Gatti, F., Lopez-Caballero, F., Cottereau, R., de Abreu Corrêa, L., Aubry, L., and Clouteau, D. SEM3D: A 3D High-Fidelity Numerical Earthquake Simulator for Broadband (0–10 Hz) Seismic Response Prediction at a Regional Scale. *Geosciences*, 12(3):112, 2022. ISSN 2076-3263. doi: 10.3390/geosciences12030112.

Wenhui Peng, Zelong Yuan, Zhijie Li, and Wang, J. Linear attention coupled Fourier neural operator for simulation of three-dimensional turbulence, 2022. URL http://arxiv.org/abs/2210.04259. arXiv preprint arXiv:2210.04259.

Williams, M. O., Kevrekidis, I. G., and Rowley, C. W. A data–driven approximation of the Koopman operator: extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, Dec 2015.

Yeung, E., Kundu, S., and Hodas, N. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pp. 4832–4839, 2019. doi: 10.23919/ACC.2019.8815339.

A. Supplementaly Material

A.1. Problem Definition

A.1.1. Seismic Wave Propagation

Seismic wave propagation in a heterogeneous, isotropic medium can be characterized by the elastic wave equation:

$$\rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} = \nabla \lambda \left(\nabla \cdot \boldsymbol{u} \right) + \nabla \mu \left[\nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u} \right)^T \right] + \left(\lambda + 2\mu \right) \nabla \left(\nabla \cdot \boldsymbol{u} \right) - \mu \nabla \times \nabla \times \boldsymbol{u}, \tag{6}$$

where ρ is the density, $\boldsymbol{u}$ is the displacement vector, and λ, μ are Lamé parameters of the material. Alternatively, one can represent the medium in terms of compressional (V_P) and shear (V_S) wave velocities. In this study, we assume a constant ratio $\alpha = V_P/V_S = 1.7$, reducing the medium description to a single parameter V_S. Thus, equation (6) can be rewritten in terms of V_S as:

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = (\alpha^2 - 2) \nabla V_S^2 \big(\nabla \cdot \boldsymbol{u} \big) + \nabla V_S^2 \Big[\nabla \boldsymbol{u} + \big(\nabla \boldsymbol{u} \big)^T \Big] + (\alpha^2 - 1) V_S^2 \nabla \big(\nabla \cdot \boldsymbol{u} \big) - V_S^2 \nabla \times \nabla \times \boldsymbol{u}. \tag{7}$$

Consider a material domain $D_a \subset \mathbb{R}^3$, and let $\mathcal{A} = \mathcal{F}\big(D_a; \mathbb{R}^+\big)$ be the space of functions describing the spatial distribution of V_S, i.e., $a(\boldsymbol{x}) = V_S(\boldsymbol{x})^2$. In principle, the displacement $\boldsymbol{u}$ in such a domain depends on four variables $(x, y, z, t) \in D_a \times [0, T]$, which leads to a high-dimensional problem if one directly seeks a neural network approximation over space-time.

Moreover, in practice, measurements of $\boldsymbol{u}$ are usually only available at the top surface of the domain. Let us define

$$D_u = \big\{ (x, y, t) \,\big|\, (x, y) \in \partial D_a^{\text{top}},\ t \in [0, T] \big\},$$

where $\partial D_a^{\text{top}}$ is the upper boundary of D_a. We then focus on the velocity field $\boldsymbol{u}$ restricted to this surface region over time. Denote by $\mathcal{U} = \mathcal{C}\big(D_u; \mathbb{R}^3\big)$ the set of continuous functions representing the three velocity components (e.g., East-West, North-South, and vertical) measured at the surface.

Under appropriate initial conditions (e.g., seismic source specification) and absorbing boundaries, we can summarize the wave equation (7) symbolically as $\mathcal{L}(a, \boldsymbol{u}) = 0$. The overarching goal is to learn an operator.

$$\mathcal{G} : \mathcal{A} \longrightarrow \mathcal{U}, \quad a \mapsto \boldsymbol{u},$$

that predicts the surface velocity field $\boldsymbol{u}$ corresponding to a given geological distribution a. We approximate this operator via a neural network

$$\mathcal{G} \approx G_\theta,$$

where G_θ denotes a parametric model with trainable parameters θ. The remainder of this paper focuses on the design, training, and evaluation of G_θ as a surrogate for full numerical simulations of seismic wave propagation.

A.1.2. Seismic Source Parameters

The seismic source in our experiments is parameterized by six quantities: the fault location (x_s, y_s, z_s) and the fault orientation (strike, dip, rake). These are defined as follows:

- **Source coordinates** (x_s, y_s, z_s): the Cartesian position of the rupture centroid, measured in meters from the model origin. Here x_s and y_s denote the horizontal east–west and north–south offsets, respectively, while z_s is the depth below the surface (positive downward).
- **Strike** (°): the angle, measured clockwise from the north, of the fault trace on the horizontal plane. A strike of $0°$ indicates a north–south fault, $90°$ an east–west fault, and so on.
- **Dip** (°): the angle between the fault plane and the horizontal, measured in the vertical plane perpendicular to the strike. A dip of $0°$ corresponds to a horizontal fault, whereas $90°$ indicates a vertical fault plane.
- **Rake** (°): the angle of slip direction measured within the fault plane, relative to the horizontal line of maximum dip. Positive rake indicates reverse (thrust) motion, negative rake indicates normal motion, and $\pm 90°$ corresponds to pure dip–slip.

These six parameters uniquely specify both the position and kinematics of the seismic source used to generate the synthetic wavefields.

A.2. Koopman Operator

A.2.1. Time-Dependent Koopman Operator

Partial differential equations (PDEs) are fundamental in science and engineering but frequently lack closed-form solutions, necessitating computational approximations (Debnath & Debnath, 2005; Gockenbach, 2010). Let $\Phi = \Phi\left(D; \mathbb{R}^{d_\phi}\right)$ be a Banach space of inputs and $\Gamma = \Gamma\left(D; \mathbb{R}^{d_\gamma}\right)$ 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:

$$\partial_t \gamma\left(x_t\right) = \left(\mathcal{L}_\phi \gamma\right)\left(x_t\right) + \eta\left(x_t\right), \quad x_t \in D \times T, \tag{8}$$

$$\gamma\left(x_t\right) = \gamma_B, \quad x_t \in \partial D \times T, \tag{9}$$

$$\gamma\left(x_0\right) = \gamma_I, \quad x_0 \in D \times \{0\}. \tag{10}$$

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 $\mathbf{g} : \Gamma \to \mathcal{G}$, 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:

$$\mathbf{g}(\gamma_{t+\varepsilon}) = \mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t). \tag{11}$$

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

$$\partial_t \mathbf{g}(\gamma_t) = \lim_{\varepsilon \to 0} \frac{\mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) - \mathbf{g}(\gamma_t)}{\varepsilon}. \tag{12}$$

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

$$\partial_t \mathcal{M} + [\mathcal{M}, \mathcal{N}] = 0, \tag{13}$$

and identifying

$$\mathcal{N} = \lim_{\varepsilon \to 0} \frac{\mathcal{K}_t^{t+\varepsilon} \mathbf{g}(\gamma_t) - \mathbf{g}(\gamma_t)}{\varepsilon}, \tag{14}$$

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

A.2.2. 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 (Arbabi & Mezic, 2017), sHankel-DMD (Črnjarić-Žic et al., 2020), and HAVOK (Brunton et al., 2017)—we construct a Krylov sequence of observables with a temporal step size $\varepsilon \in [0, \infty)$:

$$\mathcal{R}_n = \left[\mathbf{g}\left(\gamma_0\right), \mathbf{g}\left(\gamma_\varepsilon\right), \mathbf{g}\left(\gamma_{2\varepsilon}\right), \ldots, \mathbf{g}\left(\gamma_{n\varepsilon}\right)\right], \tag{15}$$

$$\begin{aligned} \mathcal{R}_n = \Big[&\mathbf{g}\left(\gamma_0\right), \mathcal{K}_0^\varepsilon \mathbf{g}\left(\gamma_0\right), \mathcal{K}_\varepsilon^{2\varepsilon} \mathcal{K}_0^\varepsilon \mathbf{g}\left(\gamma_0\right), \ldots, \\ &\mathcal{K}_{(n-1)\varepsilon}^{n\varepsilon} \cdots \mathcal{K}_0^\varepsilon \mathbf{g}\left(\gamma_0\right)\Big]. \end{aligned} \tag{16}$$

This sequence is analogous to those used in Krylov subspace methods for eigenvalue computations (Saad, 2011). To further capture the temporal dynamics, we arrange these observables into a structured Hankel matrix:

$$\mathcal{H}_{m\times n} = \begin{bmatrix} \mathbf{g}\left(\gamma_0\right) & \mathbf{g}\left(\gamma_\varepsilon\right) & \cdots & \mathbf{g}\left(\gamma_{n\varepsilon}\right) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}\left(\gamma_{(m-1)\varepsilon}\right) & \mathbf{g}\left(\gamma_{m\varepsilon}\right) & \cdots & \mathbf{g}\left(\gamma_{(m+n-1)\varepsilon}\right) \end{bmatrix}. \tag{17}$$

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

$$\mathbb{K} = \operatorname{span}\left(\mathcal{R}_n\right) \simeq \operatorname{span}\left(\mathcal{H}_{(m,n)}\right). \tag{18}$$

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

$$\begin{aligned} &\langle \widehat{\mathcal{K}}_t^{t+\varepsilon}\mathbf{h}(\gamma_t), \mathbf{g}(\gamma_{i\varepsilon})\rangle = \langle \mathcal{K}_t^{t+\varepsilon}\mathbf{h}(\gamma_t), \mathbf{g}(\gamma_{i\varepsilon})\rangle \\ &\forall i = 0, \ldots, m, \end{aligned} \tag{19}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Under appropriate conditions, as $m \to \infty$, the projected operator $\widehat{\mathcal{K}}_t^{t+\varepsilon}$ converges to the true Koopman operator $\mathcal{K}_t^{t+\varepsilon}$ (Korda & Mezić, 2018):

$$\lim_{m\to\infty} \int_{\mathcal{G}\left(\mathbb{R}^{d_\gamma}\times T\right)} \|\widehat{\mathcal{K}}_t^{t+\varepsilon}\mathbf{h}(\gamma_t) - \mathcal{K}_t^{t+\varepsilon}\mathbf{h}(\gamma_t)\|_F \, \mathrm{d}\mu = 0, \quad \forall \mathbf{h}. \tag{20}$$

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

$$\mathcal{H}_{m\times n}(k+1) = \widehat{\mathcal{K}}_{k\varepsilon}^{(k+1)\varepsilon}\, \mathcal{H}_{m\times n}(k), \quad \forall k = 1, \ldots, n, \tag{21}$$

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 (Li & Jiang, 2023), we assume that for sufficiently small ε, the Koopman operator exhibits approximate temporal invariance. This assumption allows us to perform an offline temporal averaging:

$$\begin{aligned} \overline{\mathcal{K}}_\varepsilon &= \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbf{g}\left(\gamma_\tau\right)^{-1} \mathbf{g}\left(\gamma_{\tau+\varepsilon}\right) \, \mathrm{d}\tau \\ &\simeq \operatorname*{argmin}_{P\in\mathbb{R}^{d_\gamma+1}} \sum_{k=1}^{n} \|\mathcal{H}_{m\times n}(k+1) - P\,\mathcal{H}_{m\times n}(k)\|_F. \end{aligned} \tag{22}$$

For a fixed ε, this averaged operator $\overline{\mathcal{K}}_\varepsilon : \mathcal{G}\left(\mathbb{R}^{d_\gamma} \times T\right) \to \mathbb{K}$ 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.

A.2.3. 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 VoxelKoop with a koopman operator-based multi-step forecasting framework, which comprises six integrated components:

(1) **Latent Encoding:** The initial PDE input $\phi_t = \phi\left(D \times t\right)$ from Eqs. (8-10) undergoes transformation through an encoder network to produce $\mathbf{g}\left(\widehat{\gamma}_t\right)$ in space $\mathcal{G}\left(\mathbb{R}^{d_{\widehat{\gamma}_t}} \times T\right)$. This encoder implements the observation function $\mathbf{g}\left(\cdot\right)$ using a single nonlinear layer with $\tanh\left(\cdot\right)$ activation. Refer to Figure 1 for a visual representation.

(2) **Fourier Transformation:** A Fourier transform maps $\mathbf{g}\left(\widehat{\gamma}_t\right)$ to its frequency domain representation $\mathbf{g}_{\mathcal{F}}\left(\widehat{\gamma}_t\right) = \mathcal{F} \circ \mathbf{g}\left(\widehat{\gamma}_t\right)$. Following (Li et al., 2020a), we employ fast Fourier transform with frequency mode truncation at ω. This transformation serves two purposes:

First, it facilitates efficient computation of iterative updates:

$$\gamma\left(x_{t+\varepsilon}\right) = \int_{D \times t} \mathcal{J}\phi\left(xt, y_t\right) \eta\left(y_t\right) \mathrm{d}y_t, \quad \forall; x_t \in D \times t. \tag{23}$$

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 $\mathbf{g}_{\mathcal{F}}\left(\widehat{\gamma}_t\right)$ at timesteps $t \in \varepsilon\mathbb{N}^+$, we construct a Hankel matrix $\widehat{\mathcal{H}}_{m \times n}$ with embedding dimension m. The framework employs a $o \times o$ linear layer to learn $\overline{\mathcal{K}}_\varepsilon^{\,r} : \mathcal{G}\left(\mathbb{R}^{d_{\widehat{\gamma}}} \times T\right) \to \widehat{\mathbb{K}}$ following Eqs. (21-22). This enables future state prediction through $\mathbf{g}_{\mathcal{F}}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right) = \left[\overline{\mathcal{K}}_\varepsilon^{\,r} \widehat{\mathcal{H}}_{m \times n}\left(n\right)\right]^{\mathsf{T}}\left(m\right)$. The parameter r provides temporal resolution control, allowing finer-grained evolution modeling.

(4) **Inverse Spectral Transform:** Predicted states undergo inverse Fourier transformation: $\mathbf{g}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right) = \mathcal{F}^{-1} \circ \mathbf{g}_{\mathcal{F}}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right)$. 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 (Park & Kim, 2022), we implement network $\mathcal{C}$ to extract and predict low-level information $\mathbf{g}_{\mathcal{C}}\left(\widehat{\gamma}_t\right)$. The prediction follows $\left[\mathbf{g}_{\mathcal{C}}\left(\widehat{\gamma}_{(i+1)\varepsilon}\right), \ldots, \mathbf{g}_{\mathcal{C}}\left(\widehat{\gamma}_{(i+m+1)\varepsilon}\right)\right]^{\mathsf{T}} = \mathcal{C}\left[\mathbf{g}\left(\widehat{\gamma}_{i\varepsilon}\right), \ldots, \mathbf{g}\left(\widehat{\gamma}_{(i+m)\varepsilon}\right)\right]^{\mathsf{T}}$. For complex PDEs, advanced architectures incorporating inception modules (Szegedy et al., 2015) may be employed.

(6) **Ground Motion Reconstruction:** The framework combines predictions from both pathways through weighted averaging: $\mathbf{g}_{\mathcal{U}}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right) = (1-\lambda)\,\mathbf{g}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right) + \lambda \mathbf{g}_{\mathcal{C}}\left(\widehat{\gamma}_{(m+n)\varepsilon}\right)$, where $\lambda \in [0,1]$ balances spectral components. A nonlinear decoder with $\tanh\left(\cdot\right)$ activation implements $\mathbf{g}^{-1}\left(\cdot\right) \simeq \mathbf{g}_{\mathcal{U}}^{-1}\left(\cdot\right)$, producing the final solution state $\widehat{\gamma}_{(m+n)\varepsilon}$ in $\mathbb{R}^{d_{\widehat{\gamma}}}$.

Based on the framework described in the upper mechanisms, we propose an alternative iterative update approach that differs from Eq. (23). Given $t' > t \in \varepsilon\mathbb{N}$, our formulation is expressed as:

$$\begin{aligned} \widehat{\gamma}_{t'} = \Big[\mathbf{g}^{-1}\Big(& \underbrace{\mathcal{F}^{-1} \circ \overline{\mathcal{K}}_\varepsilon^{\,r} \circ \mathcal{F} \circ \mathbf{g}\left(\widehat{\gamma}_{[t-m\varepsilon,t]}\right)}_{\textbf{Processing in component 1-4}} \\ & + \underbrace{\mathcal{C} \circ \mathbf{g}\left(\widehat{\gamma}_{[t-m\varepsilon,t]}\right)}_{\textbf{Integration of component 1 and 5}} \Big)\Big]^{\mathsf{T}}\left(m\right). \end{aligned} \tag{24}$$

Here, $\widehat{\gamma}_{[t-m\varepsilon,t]}$ represents the vector $[\widehat{\gamma}_{t-m\varepsilon}, \ldots, \widehat{\gamma}_t]$, where $m \in \mathbb{N}$ denotes the delay-embedding dimension. Drawing inspiration from the architecture of Fourier neural operators (Li et al., 2020a), 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:

$$\begin{aligned} \mathcal{L} = & \alpha \|\widehat{\gamma}_{t'} - \gamma_{t'}\|_F \\ & + \beta \sum_{i=0}^{m} \|\mathbf{g}^{-1} \circ \mathbf{g}\left(\widehat{\gamma}_{t-im\varepsilon}\right) - \gamma_{t-im\varepsilon}\|_F, \end{aligned} \tag{25}$$

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 (Champion et al., 2019a;b; Fasel et al., 2022). While our framework shares conceptual elements with previous work on neural network parameterization of Koopman operators (Lusch et al., 2018b), 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.

A.2.4. Koopman Operator for Neural Network

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 $\mathbf{g} : \Gamma \to \mathcal{G}$ where the time-dependent Koopman operator $\mathcal{K}_t^{t+\varepsilon}$ propagates observables linearly:

$$\mathbf{g}(\gamma_{t+\varepsilon}) = \mathcal{K}_t^{t+\varepsilon}\mathbf{g}(\gamma_t) \tag{26}$$

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:

$$\mathbf{z}_{t+\varepsilon} \approx \mathcal{K}_t^{t+\varepsilon}\mathbf{z}_t \tag{27}$$

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

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

$$\begin{aligned}\mathcal{R}_n = \Big[&\mathbf{g}\left(\gamma_0\right), \mathcal{K}_0^{\varepsilon}\mathbf{g}\left(\gamma_0\right), \mathcal{K}_{\varepsilon}^{2\varepsilon}\mathcal{K}_0^{\varepsilon}\mathbf{g}\left(\gamma_0\right), \ldots, \\ &\mathcal{K}_{(n-1)\varepsilon}^{n\varepsilon}\cdots\mathcal{K}_0^{\varepsilon}\mathbf{g}\left(\gamma_0\right)\Big]\end{aligned} \tag{28}$$

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

$$\mathcal{H}_{m\times n} = \begin{bmatrix} \mathbf{g}\left(\gamma_0\right) & \mathbf{g}\left(\gamma_\varepsilon\right) & \cdots & \mathbf{g}\left(\gamma_{n\varepsilon}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}\left(\gamma_{(m-1)\varepsilon}\right) & \mathbf{g}\left(\gamma_{m\varepsilon}\right) & \cdots & \mathbf{g}\left(\gamma_{(m+n-1)\varepsilon}\right). \end{bmatrix} \tag{29}$$

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

$$\mathcal{H}_{m\times n}(k+1) = \widehat{\mathcal{K}}_{k\varepsilon}^{(k+1)\varepsilon}\,\mathcal{H}_{m\times n}(k), \forall k = 1, \ldots, n. \tag{30}$$

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

$$\overline{\mathcal{K}}_\varepsilon \simeq \underset{P\in\mathbb{R}^{d_\gamma+1}}{\operatorname{argmin}} \sum_{k=1}^{n} \|\mathcal{H}_{m\times n}(k+1) - P\,\mathcal{H}_{m\times n}(k)\|_F. \tag{31}$$

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.

A.3. Experimental Results

A.3.1. Metrics

In this paper, we used some metrics to evaluate our trained VoxelKoop. The relative Root Mean Square Error (rRMSE) is 20% on the test dataset. The rRMSE and relative Mean Absolute Error (rMAE) are computed between (normalized)

predicted and simulated velocity time series at each sensor, following

$$rRMSE(x,y) = \sqrt{\frac{1}{N_t}\sum_{k=1}^{Nt}\frac{(\hat{u}(x,y,t_k)-u(x,y,t_k))^2}{u(x,y,t_k)^2+\epsilon^2}} \tag{32}$$

$$rMAE(x,y) = \frac{1}{N_t}\sum_{k=1}^{Nt}\frac{|\hat{u}(x,y,t_k)-u(x,y,t_k)|}{|u(x,y,t_k)+\epsilon|} \tag{33}$$

where $\epsilon = 0.01$, x and y indicate the sensor coordinates, and t iterates from 0 s to 6.4 s. Errors are then averaged over all sensors and samples. Relative errors on the order of 20% are common in 3D PDE predictions, especially when complexity arises from the presence of heterogeneous parameters.

Due to the limited interpretability of the RMSE, it is complemented by the frequency biases, computed as the errors between the amplitudes of Fourier coefficients in simulated $u(x,y)$ and predicted $\hat{u}(x,y)$ velocity fields. Frequency biases are given for low frequency (0- 1Hz), medium frequency (1- 2Hz), and high frequency (2- 5Hz) components following

$$\text{rFFT}_{\text{low}} = \frac{\overline{\mathcal{F}\left(\hat{u}(x,y)\right)}_{low} - \overline{\mathcal{F}\left(u(x,y)\right)}_{low}}{\overline{\mathcal{F}\left(u(x,y)\right)}_{low}} \tag{34}$$

with

$$\overline{\mathcal{F}\left(u(x,y)\right)}_{low} = \frac{1}{N_f}\sum_{0\leq f<1}|\mathcal{F}(u(x,y))(f)| \tag{35}$$

In equation (35), N_f denotes the number of frequencies f between 0 and 1Hz and $\mathcal{F}$ is the Fourier transform. Definitions naturally extend to medium and high frequencies. It should be noted that positive frequency biases indicate an overestimation of the frequency content compared to the reference ($\overline{\mathcal{F}\left(\hat{u}(x,y)\right)} \geq \overline{\mathcal{F}\left(u(x,y)\right)}$) while negative values indicate underestimation. Noting that the mean of Fourier coefficients amplitudes $\overline{\mathcal{F}\left(u(x,y)\right)}$ is always positive, this implies that the frequency biases cannot be lower than -1.

A.3.2. Toy Model Experiments

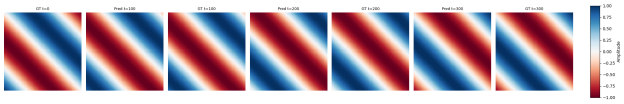
To validate our Koopman module on idealized waveforms, we constructed three synthetic datasets with known analytic solutions. For the synthetic dataset, we trained a Koopman Module on the first 100 saved frames, then rolled out predictions for the remaining frames. Figure 4 summarizes the eigenvalue spectra, leading spatial modes, and comparison between ground-truth (GT) and Koopman forecasts.

In the first toy model, a planar wavefront travels at $45°$ across a 64×64 grid. The DMD eigenvalue spectrum for this case (Figure 4(a)) shows that almost all eigenvalues lie very near the unit circle, indicating that the algorithm has learned a nearly lossless oscillatory operator. The leading two spatial modes correspond precisely to the forward and backward components of the plane wave, each with modulus $|\lambda| \approx 1$, confirming that amplitude is preserved. As illustrated in Figure 4(b), predictions at $t = 100, 200, 300$ remain phase-accurate and spatially aligned with the ground truth, demonstrating excellent long-range fidelity for periodic, non-attenuating dynamics.

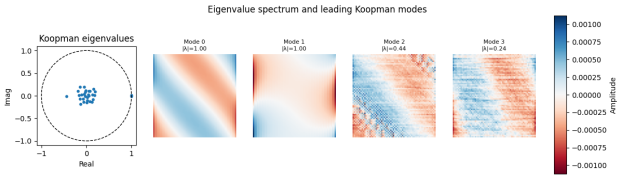
In the second experiment, we generated a circular wave emanating from the domain center with an exponential attenuation factor $\exp(-\alpha r)$. Here, the Koopman eigenvalues shrink inside the unit circle, with the magnitude deficit directly quantifying the per-step loss due to radial damping (Figure 4(c)). The dominant DMD modes form concentric ring patterns that capture the radial oscillation, whereas higher modes encode finer spatial details and residual noise. Despite the decay, the learned operator accurately predicts both the wavefront position and the amplitude drop through frame 300 (Figure 4(d)), illustrating that linear time-marching in the lifted Fourier basis can model attenuating processes.

The third scenario begins with a localized Gaussian pulse at the center, which disperses non-periodically over time. In this case, the eigenvalue cloud broadens significantly, reflecting the inherently transient and broadband nature of a single pulse (Figure 4(e)). The top few Koopman modes exhibit petal-shaped and quadrupolar structures, revealing how the method decomposes the initial pulse into spatial patterns that evolve with distinct decay rates. While early-time forecasts ($t \leq 100$) track the ground truth closely, later predictions diverge somewhat (Figure 4(f)), indicating that a purely linear surrogate of fixed rank may struggle to capture strongly non-harmonic dispersive behavior over very long horizons. Nonetheless, these experiments confirm that our approach provides an interpretable spectrum of dynamics—preserving amplitude for ideal

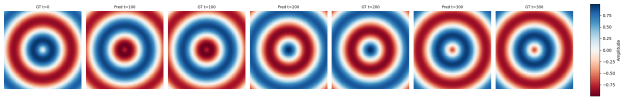
waves, encoding attenuation via eigenvalue magnitude, and revealing modal decomposition even for complex, non-stationary excitations.



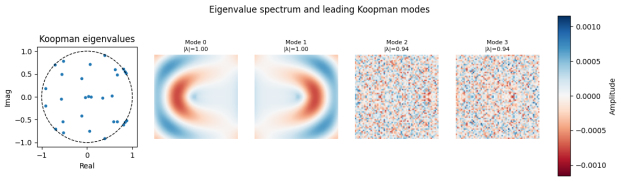
(a) Planar wave: ground truth vs. Koopman prediction.



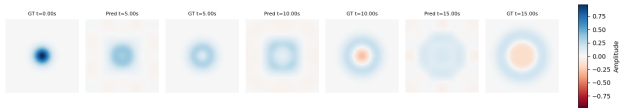
(b) Planar wave: Koopman eigenvalue spectrum and leading modes.



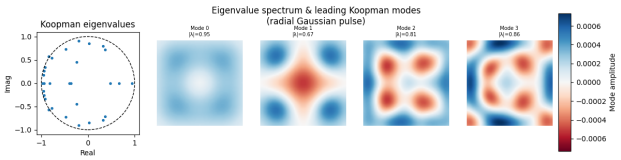
(c) Radial attenuating wave: ground truth vs. Koopman prediction.



(d) Radial attenuating wave: eigenvalue spectrum and dominant modes.



(e) Gaussian pulse: ground truth vs. Koopman prediction.



(f) Gaussian pulse: Koopman eigenvalue spectrum and principal modes.

Figure 4. Comparison of ground truth and Koopman-based predictions alongside eigenvalue spectra and leading spatial modes for three toy problems.

In all three cases, the Koopman-DMD approach offers a transparent modal decomposition that explains both stable propagation ($|\lambda| \approx 1$) and decay ($|\lambda| < 1$), while delivering accurate forecasts over hundreds of frames with a compact latent basis.

A.4. 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 (Debnath & Debnath, 2005). Despite significant progress (Tanabe, 2017), many important PDEs (e.g., the Navier–Stokes equations) still lack analytic solutions (Gockenbach, 2010), motivating the development of various numerical and data-driven methods for approximating their solutions (Mattheij et al., 2005).

Let $\Phi = \Phi\left(D; \mathbb{R}^{d_\phi}\right)$ be a Banach space of inputs and $\Gamma = \Gamma\left(D; \mathbb{R}^{d_\gamma}\right)$ 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 (Li et al., 2020b).

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 $\mathbf{g} : \Gamma \to \mathcal{G}(\mathbb{R}^{d_\gamma} \times T)$, 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

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

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, yielding a Koopman auto-encoder (KAE). 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.