Energy-Based Operator Learning in Function Space

Hong Chul Nam[*1†] **Jeonghwan Cheon**[*2] **Jae Mo Shin**[3] **Hyun Kwon**[2]
[*]Equally contributed, [†]Corresponding author,
[1]Center for Data Science, New York University, New York, NY, USA,
[2]Korea Military Academy, Seoul, Republic of Korea,
[3]Seoul National University, Seoul, Republic of Korea
hn2557@nyu.edu, {jeonghwan518, jaemoshin, hkwon.cs}@gmail.com

Abstract

We propose Energy-Based Operators (EBOs), an architecture-agnostic framework for learning conditional energy functionals over functions on continuous domains. An EBO defines a scalar energy over target functions given a context function and induces a probability model through a Gaussian reference measure. The resulting score field is obtained by taking the gradient of the parameterized energy, enabling iterative energy minimization (EM) in function space. Our model demonstrates strong performance in function generation tasks, such as super-resolution and forecasting, on smooth 1D function classes compared with prediction and denoising operators across various architectural backbones. Moreover, it achieves strong performance on PDEs, including Navier–Stokes, Darcy flow, and Burgers' equations. Notably, our model successfully detects anomalous functions by automatically assigning them high energy without any anomaly supervision. It enables seizure detection and volatility prediction by learning neural dynamics and market microstructure dynamics without anomaly labels during training. These results highlight the effectiveness of EBOs in both learning dynamical systems and detecting functional anomalies arising in scientific applications.

1 Introduction

Energy-based models (EBMs) [1–3] provide a flexible framework for modeling complex distributions by associating a scalar energy with each configuration, where lower energy corresponds to higher probability. Initially inspired by theoretical models of neural dynamics underlying associative memory in the brain, EBMs have demonstrated significant efficacy in capturing structures of various machine learning tasks, including image generation [4, 5], language modeling [6] and logical reasoning [7].

While EBMs have achieved success in finite-dimensional settings, extending them to *function spaces* presents unique challenges. Many scientific and engineering problems—from solving partial differential equations to modeling physiological signals—are naturally posed over continuous domains, where the objects of interest are functions. Naively discretizing these functions and applying standard EBMs ignores the underlying continuum structure, which introduces sensitivity to resolution changes [8].

Recent extensions to score-based diffusion operators [9–12], such as the denoising diffusion operator (DDO), parameterize the score directly, thereby supporting sampling but exposing no scalar energy and tying inference to a fixed noise schedule with a time-dependent score. We propose *Energy-Based Operators* (EBOs), which parameterize the energy functional itself, combining the flexibility of energy-based modeling with the discretization-invariance of neural operators (Figure 1). The scalar energy enables unsupervised anomaly detection on real-world signals, and the time-independent energy supports test-time scaling, where more optimization steps can be allocated to harder instances without retraining [6, 13].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

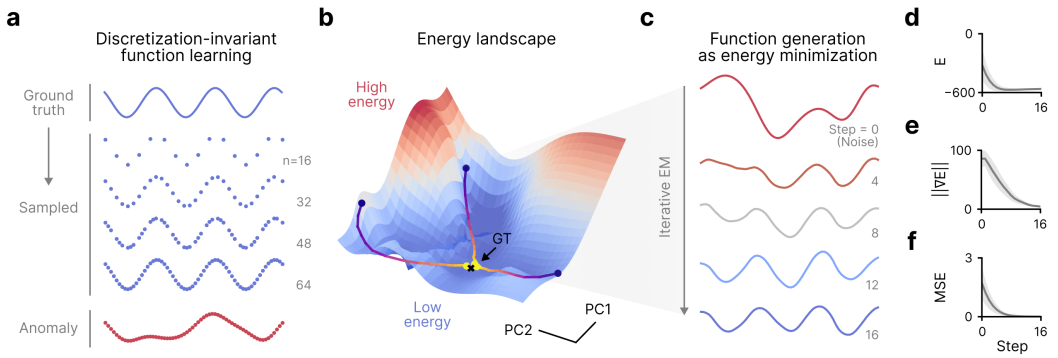


Figure 1: **Energy-Based Operator for discretization-invariant function learning. a,** A sinusoidal ground truth function and various sampled examples with different sampling ratios. An anomaly function is defined as a function that violates the dynamics of the ground truth function. **b,** Energy landscape of the EBO learned for the sinusoidal function, obtained by Principal Component Analysis. Red and blue colors indicate high energy and low energy, respectively. Three trajectories of energy minimization starting from different initial noises are displayed. **c,** Function generation as energy minimization. Sampled functions displayed from a representative EM trajectory in the energy landscape. **d-f,** Convergence of the iterative EM trajectory. **d,** Energy. **e,** Norm of the gradient of energy. **f,** Mean squared error between the sampled function and the given context.

Our key contributions are:

- We generalize energy-based models to function space neural operators, defining a scalar energy over functions and deriving the score as its gradient.
- We theoretically analyze EBO's discretization invariance and universal approximation properties.
- We demonstrate strong performance of EBO in function generation and functional anomaly detection over both diverse function classes and realistic datasets.

2 Energy-Based Operators

We fix two finite-dimensional domains $\Omega_X \subset \mathbb{R}^{n_x}$ and $\Omega_Y \subset \mathbb{R}^{n_y}$ equipped with Borel σ-algebras $\mathcal{B}(\Omega_X), \mathcal{B}(\Omega_Y)$ and reference measures λ_X and λ_Y[1]. We consider a *context function space* $\mathcal{H}_X \triangleq \mathcal{F}(\Omega_X, \mathbb{R}^d)$, and a *target function space* $\mathcal{H}_Y \triangleq \mathcal{F}(\Omega_Y, \mathbb{R}^m)$, where $\mathcal{H}_X$ and $\mathcal{H}_Y$ are assumed to be real separable Hilbert spaces with appropriate inner products[2]. Moreover, we assume point-wise evaluation is continuous on $\mathcal{H}_X$ and $\mathcal{H}_Y$ with $u(y) \triangleq \delta_y(u)$.

2.1 Conditional Energy Functionals

Our goal is to model the conditional distribution of a target function $u \in \mathcal{H}_Y$ given a context function $f \in \mathcal{H}_X$. Instead of predicting u directly, an EBO assigns to each context an energy functional that scores every candidate target, with low energy for favoured targets.

Definition 2.1 (Energy-Based Operator)**.** Let $C(\mathcal{H}_Y; \mathbb{R})$ denote the space of continuous functionals $\mathcal{H}_Y \to \mathbb{R}$. An **Energy-Based Operator (EBO)** is an operator

$$\Psi_\theta : \mathcal{H}_X \longrightarrow C(\mathcal{H}_Y; \mathbb{R}), \qquad f \longmapsto \Psi_\theta(f), \tag{1}$$

that assigns to each context function $f \in \mathcal{H}_X$ an **energy functional** $\Psi_\theta(f) : \mathcal{H}_Y \to \mathbb{R}$ on the target function space. We write $\Psi_\theta(f, u) \triangleq \Psi_\theta(f)(u)$ for the energy of a target function $u \in \mathcal{H}_Y$ given f.

[1]We take λ_X and λ_Y to be the uniform probability measures $\lambda_X(A) = |A|/|\Omega_X|$ and $\lambda_Y(A) = |A|/|\Omega_Y|$, where $|\cdot|$ is Lebesgue measure, so that integrals against them are spatial averages.

[2]Typical choices are Sobolev spaces H^s with $s > n_x/2$ for $\mathcal{H}_X$ and $s > n_y/2$ for $\mathcal{H}_Y$, on which point evaluation is continuous.

2.2 Reference Measure and Score

We define a target measure on the function space $\mathcal{H}_Y$ via a log Radon–Nikodym derivative with respect to a Gaussian reference. Let $\mu_0 = \mathcal{N}(0, C)$ be a Gaussian random field (GRF) on $\mathcal{H}_Y$ with covariance operator $C : \mathcal{H}_Y \to \mathcal{H}_Y$ (self-adjoint, positive and trace-class). Given $f \in \mathcal{H}_X$, we define a conditional measure on $\mathcal{H}_Y$ by

$$\frac{d\nu_\theta}{d\mu_0}(u \mid f) \;=\; \frac{1}{Z_\theta(f)} \exp\big(-\Psi_\theta(f, u)\big), \tag{2}$$

where $Z_\theta(f) = \int_{\mathcal{H}_Y} \exp\big(-\Psi_\theta(f, u)\big)\, d\mu_0(u)$ is a normalizing constant assumed finite.

Assumption 2.2 (Differentiability of the energy)**.** For each fixed $f \in \mathcal{H}_X$, the map $u \mapsto \Psi_\theta(f, u)$ is Fréchet differentiable on $\mathcal{H}_Y$. We write $\nabla_u \Psi_\theta(f, u) \in \mathcal{H}_Y$ for its gradient, the Riesz representative of the Fréchet derivative in the inner product of $\mathcal{H}_Y$.

We define the *score operator* by

$$S_\theta(f; u) \triangleq -C\, \nabla_u \Psi_\theta(f, u). \tag{3}$$

Applying C damps the high-frequency components of the gradient: $S_\theta(f; u)$ lies in the Cameron–Martin space $C^{1/2}(\mathcal{H}_Y)$ of μ_0 and is the steepest-descent direction of $\Psi_\theta(f, \cdot)$ in its inner product (Appendix B.4). On a grid, u is stored as a vector in $\mathbb{R}^n$ after flattening points and channels, automatic differentiation gives $\nabla_u \Psi_\theta \in \mathbb{R}^n$, and C becomes the GRF covariance matrix $C_n \in \mathbb{R}^{n \times n}$, so that $S_\theta(f; u) = -C_n \nabla_u \Psi_\theta(f, u)$.

2.3 Inference by Energy Minimization

Given a context f, EBO predicts a target function by iterative energy minimization (EM): it starts from a GRF sample and takes K steps along the score with step size $h > 0$,

$$u^{(0)} \sim \mu_0, \qquad u^{(k+1)} = u^{(k)} + h\, S_\theta\big(f; u^{(k)}\big), \quad k = 0, \dots, K-1. \tag{4}$$

We write $u_\theta^{(K)}$ for the final iterate. Check Algorithm 1 for a detailed pseudoalgorithm. When part of the target is observed, as in reconstruction, the observed values are held fixed, and only the remaining points are updated.

2.4 Training with Iterative Energy Minimization

We assume supervised training data $(f, u^*) \sim \mathcal{D}$, where $f \in \mathcal{H}_X$ and $u^* \in \mathcal{H}_Y$, and introduce a *task loss functional*

$$\mathcal{J} : \mathcal{H}_Y \times \mathcal{H}_Y \times \mathcal{H}_X \to \mathbb{R}_{\geq 0}, \quad (u, u^*; f) \mapsto \mathcal{J}(u, u^*; f). \tag{5}$$

This allows modeling discrepancies with application-specific functionals[3].

Definition 2.3 (Sampling-based training objective)**.** We define the training objective derived from the task loss functional $\mathcal{J}$ as

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{\substack{(f,u^*)\sim\mathcal{D} \\ u^{(0)}\sim\mu_0}} \Big[\mathcal{J}\big(u_\theta^{(K)}, u^*; f\big)\Big], \tag{6}$$

where $u^{(K)}$ is computed from the iterative EM for K steps.

This definition is similar to the loss defined in energy-based transformer [6], whereas we use deterministic GRF-preconditioned energy-minimization steps. Check Algorithm 2 for the training pipeline. If one backpropagates through all EM steps, gradients $\nabla_\theta \mathcal{L}(\theta)$ may involve Hessian-vector products of Ψ_θ. This can be computed by auto-differentiation, but may increase memory and runtime. In practice, we detach intermediate states to avoid backpropagation through time, so that only the last step is differentiated (Algorithm 2). Appendix B.4 shows that, under standard smoothness assumptions and sufficiently small step size, the deterministic GRF-preconditioned update monotonically decreases the discretized energy.

[3]For our paper, we assume L_2 functional as our default task loss functional.

Table 1: **Across architectures, EBO is the best method on smooth signals (Oscillation, Damping); Operator is best on the non-smooth Izhikevich model.** Test MSE $\pm$ std on B=128 held-out samples at resolution $N = 128$ with 60% context, in both super-resolution (SR) and forecasting (FC) settings. Bold marks the best method within each (backbone, dataset, mask) cell.

		Oscillation		Damping		Izhikevich	
Backbone	**Method**	SR	FC	SR	FC	SR	FC
Transformer	Op.	8.83e-04$_{\pm 5.98e-04}$	1.59e-03$_{\pm 1.58e-03}$	1.47e-03$_{\pm 1.26e-03}$	1.00e-03$_{\pm 1.15e-03}$	**1.99e-02$_{\pm 2.33e-02}$**	**3.85e-02$_{\pm 4.12e-02}$**
	EBO	**2.75e-04$_{\pm 1.87e-04}$**	**1.42e-03$_{\pm 2.34e-03}$**	**2.54e-04$_{\pm 1.80e-04}$**	**3.22e-04$_{\pm 3.89e-04}$**	3.21e-02$_{\pm 3.01e-02}$	4.81e-02$_{\pm 4.40e-02}$
	DDO	9.76e-03$_{\pm 2.42e-02}$	2.02e-02$_{\pm 7.72e-02}$	1.39e-01$_{\pm 7.88e-02}$	5.09e-03$_{\pm 7.89e-03}$	1.78e-01$_{\pm 1.35e-01}$	9.55e-02$_{\pm 1.05e-01}$
Factformer	Op.	1.43e-03$_{\pm 1.39e-03}$	1.90e-03$_{\pm 1.73e-03}$	1.65e-03$_{\pm 1.93e-03}$	7.11e-04$_{\pm 1.02e-03}$	**1.95e-02$_{\pm 2.33e-02}$**	**3.90e-02$_{\pm 4.09e-02}$**
	EBO	**2.71e-04$_{\pm 1.65e-04}$**	**1.03e-03$_{\pm 1.80e-03}$**	**4.16e-04$_{\pm 4.47e-04}$**	**3.73e-04$_{\pm 5.17e-04}$**	3.24e-02$_{\pm 3.27e-02}$	4.57e-02$_{\pm 4.26e-02}$
	DDO	1.92e-02$_{\pm 4.95e-02}$	1.08e-02$_{\pm 2.37e-02}$	1.34e-01$_{\pm 7.83e-02}$	9.11e-03$_{\pm 2.00e-02}$	2.28e-01$_{\pm 1.11e-01}$	1.03e-01$_{\pm 8.26e-02}$
Galerkin	Op.	1.93e-03$_{\pm 2.00e-03}$	2.26e-03$_{\pm 2.36e-03}$	3.04e-03$_{\pm 3.22e-03}$	8.56e-04$_{\pm 1.14e-03}$	**3.70e-02$_{\pm 4.31e-02}$**	**4.20e-02$_{\pm 4.30e-02}$**
	EBO	**4.66e-04$_{\pm 3.28e-04}$**	**1.65e-03$_{\pm 1.49e-03}$**	**4.63e-04$_{\pm 5.00e-04}$**	**3.23e-04$_{\pm 5.39e-04}$**	4.19e-02$_{\pm 4.17e-02}$	4.51e-02$_{\pm 4.25e-02}$
	DDO	1.38e-02$_{\pm 3.75e-02}$	1.25e-02$_{\pm 2.86e-02}$	1.42e-01$_{\pm 7.81e-02}$	1.54e-02$_{\pm 3.12e-02}$	1.84e-01$_{\pm 1.17e-01}$	1.11e-01$_{\pm 8.37e-02}$

2.5 Operator Architecture for the Energy Functional

We specify the architecture for $\Psi_\theta(f, u)$. Let $\{x_i\}_{i=1}^{M_X} \subset \Omega_X$ and $\{y_j\}_{j=1}^{M_Y} \subset \Omega_Y$ be sampling points with quadrature weights $\{w_i^X\}$ and $\{w_j^Y\}$. We represent $f_i \triangleq f(x_i) \in \mathbb{R}^d$ and $u_j \triangleq u(y_j) \in \mathbb{R}^m$ as a discretization of the context and target functions. EBO encodes the context function and the target function with a neural operator

$$\mathcal{G}_\theta : \mathcal{H}_X \times \mathcal{H}_Y \longrightarrow C(\overline{\Omega_Y}; \mathbb{R}^{d_h}),$$

such as transformer [14] or FNO [15], which maps (f, u) to a continuous latent function $\mathcal{G}_\theta(f, u)$ on the target domain with d_h channels. Continuity makes the point values $\mathcal{G}_\theta(f, u)(y_j)$ used below well-defined. In practice, $\mathcal{G}_\theta$ reads the pairs (x_i, f_i) and (y_j, u_j), optionally with a positional encoding of the coordinates such as SIREN [16], and returns its output at every y_j. A pointwise energy head $Q : \mathbb{R}^{d_h} \to \mathbb{R}$ maps the latent function to a per-point energy $Q\big(\mathcal{G}_\theta(f, u)(y)\big)$, and the energy functional averages it over the target domain, as in the averaging neural operator [17],

$$\Psi_\theta(f, u) = \int_{\Omega_Y} Q\big(\mathcal{G}_\theta(f, u)(y)\big)\, d\lambda_Y(y) \;=\; \frac{1}{|\Omega_Y|} \int_{\Omega_Y} Q\big(\mathcal{G}_\theta(f, u)(y)\big)\, dy.$$

In practice, we compute the quadrature sum $\Psi_\theta^{M_X, M_Y}(f, u) = \sum_{j=1}^{M_Y} w_j^Y\, Q\big(\mathcal{G}_\theta(f, u)(y_j)\big)$ where the integrals inside $\mathcal{G}_\theta$ are also replaced by quadrature sums over the sampled values f_i and u_j (Definition B.6). With uniform weights $w_j^Y = 1/M_Y$, this is the mean of the per-point energies.

3 Related Works

Neural operators. Neural operators [18] are architectures respecting discretization invariance and can efficiently learn mappings between infinite-dimensional function spaces. Transformer [14] architecture can be extended to proper neural operators with examples such as GNOT [19], UPT [20], and Transolvers [21]. On the other hand, there have been diverse attention mechanisms such as Galerkin [22] and FactFormer [23] for improving naive attention mechanisms in PDE modeling. Unlike other methods, EBO is agnostic to underlying operator architecture.

Energy-based models. Energy-based models have been explored in various forms [24]. Energy transformer [25] was proposed to generalize transformers into energy-based paradigms. Energy-based transformer [6] succeeded in scaling general energy-based models into language models and video models. However, since they are all trained with regular networks, they cannot adapt to irregular grids and do not enjoy discretization invariance in comparison with neural operators. There are several works focusing on function-space diffusion models [9–12], which are intrinsically energy-based operators. However, since they are not directly learning the energy functional itself, it is not possible to estimate the energy directly with those methods. Recent works have introduced EBM frameworks for operator learning [26, 27], and EBO generalizes this construction to a backbone-agnostic framework with discretization-invariance (Theorem 4.1) and universal-approximation (Theorem 4.2) guarantees.

4 Theoretical Analysis

In this section, we establish two fundamental theoretical properties of Energy-Based Operators: (i) *discretization invariance*, ensuring that our energy estimates converge to well-defined continuum limits despite discretization choices, and (ii) *universal approximation*, demonstrating that EBOs can approximate any jointly continuous energy to arbitrary precision.

4.1 Discretization Invariance

A central motivation for EBOs is robustness to changes in discretization. Unlike naive grid-based approaches that re-learn models for each resolution, an EBO trained on one resolution should generalize to others. We formalize this through the relationship between continuum and discretized energy functionals under the architectural assumptions detailed in Appendix B.

Theorem 4.1 (Discretization invariance of EBO)**.** *Let Ψ_θ be an EBO as in Definition B.5 with C^p kernels $\kappa_{YX}^{(t)}$ on $\overline{\Omega_Y} \times \overline{\Omega_X}$ and $\kappa_{YY}^{(t)}$ on $\overline{\Omega_Y} \times \overline{\Omega_Y}$, Lipschitz activations with constant L_σ, lifting maps P_X, P_Y that are Lipschitz in their first arguments, and a Lipschitz energy head Q with constant L_Q. Assume that Assumptions B.1 and B.2 hold on a compact set $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$. Let D_{M_X}, D_{M_Y} be discrete refinements of Ω_X, Ω_Y with quadrature rules of order $p \geq 1$. Then the discretized energy $\Psi_\theta^{M_X, M_Y}$ of Definition B.6 satisfies*

$$\sup_{(f,u)\in\mathcal{K}} \left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| = O\Big(M_X^{-p/n_x} + M_Y^{-p/n_y}\Big), \qquad M_X, M_Y \to \infty. \quad (7)$$

The proof decomposes the error into layer-wise quadrature error (via Lemma B.8) and error propagation through Lipschitz-stable layers (Lemma B.9). Lemma B.10 combines these bounds layer-by-layer, yielding $O(M_X^{-p/n_x} + M_Y^{-p/n_y})$ error in the latent functions from the cross and self kernels. Output pooling contributes a further $O(M_Y^{-p/n_y})$, and Lipschitz continuity of the energy head returns the scalar energy. Full details appear in Appendix B, including the extension to softmax attention through continuum attention [28] (Appendix B.3).

4.2 Universal Approximation

We next establish that EBOs are universal approximators of jointly continuous energies.

Theorem 4.2 (Universal approximation for EBO)**.** *Under Assumptions B.1 and B.24, let $\Psi^\dagger : \mathcal{H}_X \to C(\mathcal{H}_Y; \mathbb{R})$ be jointly continuous (Definition B.23). For any compact set $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$ and $\varepsilon > 0$, there exists an EBO Ψ_θ as in Definition B.5, with sufficiently large latent dimension d_h, such that*

$$\sup_{(f,u)\in\mathcal{K}} \left|\Psi_\theta(f,u) - \Psi^\dagger(f,u)\right| < \varepsilon. \quad (8)$$

The proof reduces the energy to finitely many localized averages of f and u. The cross and self kernels gather these averages into a constant latent function (Lemma B.25), and an energy head approximates their nonlinear combination using Assumption B.24. See Appendix B for details.

Corollary 4.3 (Weak convergence of induced measures)**.** *Let $\Psi^\dagger$ be as in Theorem 4.2, fix $f \in \mathcal{H}_X$, and suppose that $\Psi^\dagger$ induces a well-defined conditional probability measure $\nu^\dagger(\mathrm{d}u \mid f) \propto \exp\big(-\Psi^\dagger(f,u)\big)\,\mu_0(\mathrm{d}u)$, where μ_0 is a Gaussian reference measure on $\mathcal{H}_Y$. Then there are EBOs Ψ_{θ_n} with $\Psi_{\theta_n}(f,\cdot) \to \Psi^\dagger(f,\cdot)$ μ_0-almost everywhere, and if $\exp\big(-\Psi_{\theta_n}(f,\cdot)\big)$ is bounded by a μ_0-integrable function independent of n, then $\nu_{\theta_n}(\cdot \mid f)$ converges to $\nu^\dagger(\cdot \mid f)$ weakly.*

In the main experiments, we use deterministic GRF-preconditioned energy minimization as a scalable refinement procedure rather than exact sampling from this measure.

5 Experiments

In this section, we conduct extensive experiments to empirically validate the performance of EBO. Since EBO is architecture-agnostic, we separate the *backbone*, the neural operator that encodes the functions, from the *paradigm*, the way the target function is produced. As backbones, we

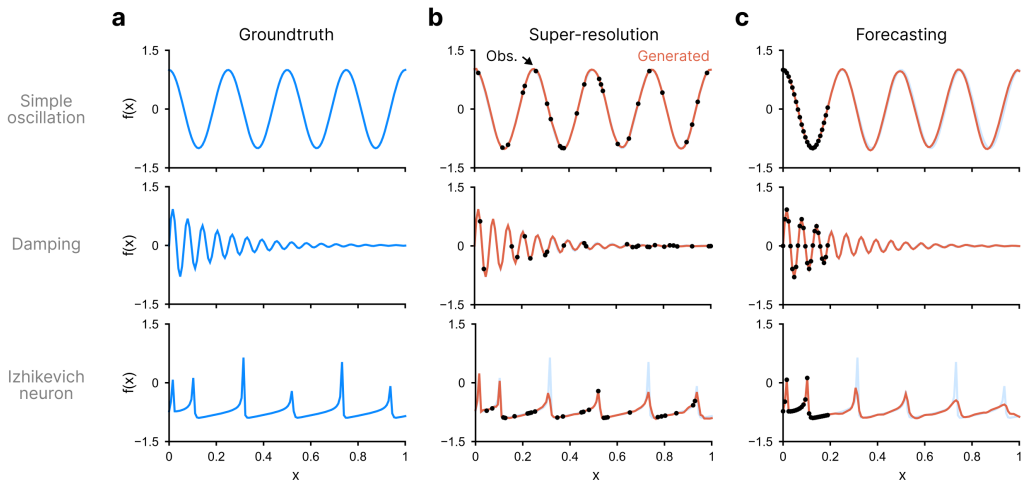


Figure 2: **Generation of various functions with limited context. a,** Ground-truth functions. Three function classes are used: simple oscillation, damping, and Izhikevich neuron model. **b,** Super-resolution experiment. Only sparse observations (20%) are given, and reconstruction to 128 samples is performed. Dots and orange lines indicate the given context and the generated functions by EBO, respectively. **c,** Forecasting experiment. The first 20% sample points are given, and the model reconstructs the full future values.

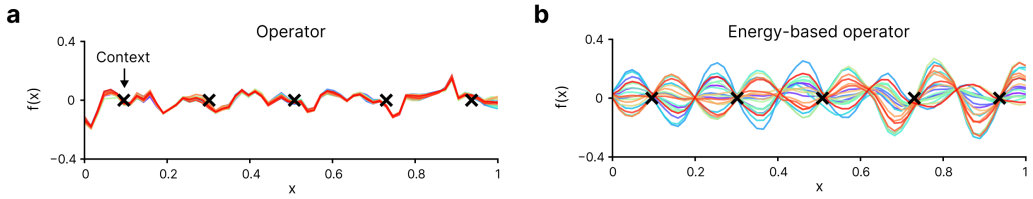


Figure 3: **Generating various plausible functions from minimal constraints.** Provided five zero-crossing as given contexts, **a,** Generated functions through direct prediction (Operator). **b,** Generated functions through EBO.

use a transformer neural operator [14] for the 1D function classes, EEG and financial data, with FactFormer [23] and Galerkin Transformer [22] as further backbones for 1D super-resolution and forecasting, and FNO [15] for the PDEs. Within each comparison, all paradigms share the same backbone. The *operator* (Op.) takes the backbone for embedding and a point-wise projection layer as a predictor, and predicts the target function in one forward pass. EBO adapts the same backbone minimally by intentionally adding an integral layer for scalar outputs, and produces the target by iterative energy minimization from a GRF sample. DDO [9] produces it by denoising a GRF sample.

5.1 Function-level Reconstruction and Generation

During training of EBO, we used various functions with different discretizations and sampling ratios over irregular grids (Figure 1a). Our model maps regular functions to low energy, while anomalous functions are assigned high energy values in the energy landscape (Figure 1b). Starting from initial GRF noise, we perform iterative EM to generate functions conditioned on a given context (Figure 1b,c). The refined candidate function's energy gradually decreases (Figure 1d). The norm of the energy gradient also converges toward zero, indicating that the iterative EM dynamics stabilize (Figure 1e). During this process, the gap between the generated function and the ground-truth function narrows (Figure 1f). Check Appendix C for details.

Super-resolution and forecasting We tested super-resolution and forecasting with 60% context ratio across three function classes (oscillation, damping, and Izhikevich; Figure 2a), using three backbones (Transformer [14], FactFormer [23], and Galerkin Transformer [22]) with detailed setups in Appendix C. According to Table 1, EBO is the best method on smooth signals (oscillation and damping) for every backbone tested, improving over one-shot prediction (Op.) by up to $\sim$6.5$\times$ in

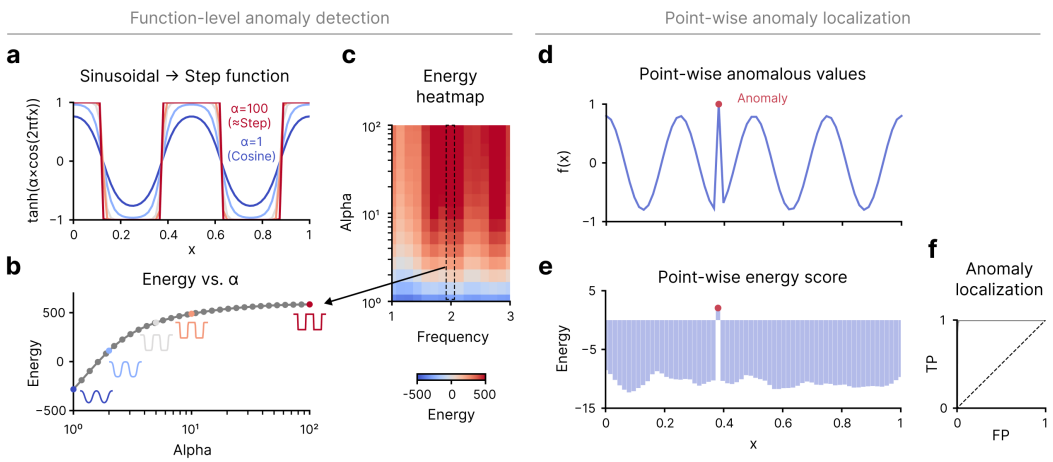


Figure 4: **Function-level anomaly detection and anomaly localization using energy. a-c,** Function-level anomaly detection. **a,** Synthesized function space transition from sinusoidal function (learned) to step function (anomaly). **b,** Energy vs. α. $\alpha = 1$ gives $\tanh(\cos(2\pi\omega x))$, a smooth waveform close to a sinusoid, but increasing α modulates the waveform into a radical transition, finally converging to a step function. **c,** Generalized energy heatmap with varying α and frequency of sinusoidal function. **d-f,** Point-wise energy score for anomaly localization. **d,** Sinusoidal function with point-wise anomaly. **e,** Point-wise energy score. **f,** Anomaly localization task performance.

MSE, and is never beaten by the diffusion baseline (DDO). On the non-smooth Izhikevich regime, one-shot prediction (Op.) wins across all backbones — consistent with the GP prior in EBO being best matched to smooth functions. Even with limited context, EBO successfully captures dynamics such as periodicity (oscillation), decay (damping), and neural spike timing (Izhikevich) in both super-resolution (Figure 2b) and forecasting (Figure 2c).

Function generation from sparse observations To investigate whether EBO can generate various plausible functions given minimal constraints, we provided 5 uniformly spaced zero-crossing context points and performed super-resolution to predict the full function. While direct predictions fail to capture the global periodicity of cosine functions (Figure 3a), EBO successfully generates diverse cosine-like functions (Figure 3b).

Function-level anomaly detection We use total energy as an OOD score. Test functions $f_\alpha(x) = \tanh(\alpha \cdot \cos(2\pi\omega x))$ transition from smooth oscillation ($\alpha \approx 1$) to square wave ($\alpha \gg 1$) (Figure 4a). At $\alpha = 1$, the function is smooth but not a sinusoid (its third harmonic is about 7% of the fundamental), and increasing α also raises its peak value $\tanh \alpha$ and its root mean square, from 0.58 at $\alpha = 1$ to nearly 1. The energy therefore responds to amplitude as well as to shape. Energy increases monotonically with α, enabling energy-score-based anomaly detection (Figure 4b). We built an energy heatmap, where the x-axis indicates diverse cosine functions with different frequencies and the y-axis indicates α. Energy increases with α at every frequency. At $\alpha \approx 1$ it is low at every frequency, as EBO is trained only on the cosine function class, while for larger α its level also depends on frequency: it is lowest near frequency 1 and reaches the top of the colour scale between frequencies 1.6 and 2.1 (Figure 4c). Table 3 compares EBO with the operator, scored by its reconstruction error: EBO has the higher AUROC on these step-like functions (0.981 against 0.969), on seizures (0.848 against 0.709) and on high future volatility (0.986 against 0.978). These results suggest that the energy score, trained without supervision, can capture the general energy landscape.

Point-wise anomaly detection To track not only whether a function is anomalous but also where the anomaly occurs, we use the per-point energy $Q(\mathcal{G}_\theta(f, u)(y_j))$ for anomaly localization. After injecting a discontinuity at a single point of a cosine function instance (Figure 4d), the anomalous location consistently exhibits the highest energy (Figure 4e). We tested various random anomalous locations and confirmed that simple energy thresholding is sufficient for anomaly localization (Figure 4f). These results empirically validate that EBO can perform both detection and localization.

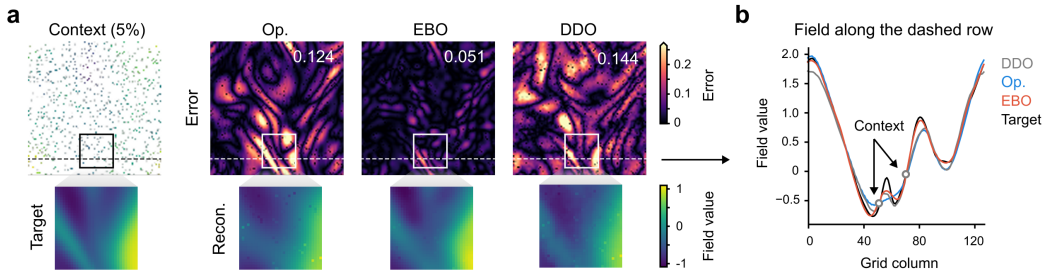


Figure 5: **Navier–Stokes sparse reconstruction at resolution 128 and 5% context. a,** Context and absolute errors with relative L^2 (Top) and Boxed window (Bottom). **b,** Field along the dashed row.

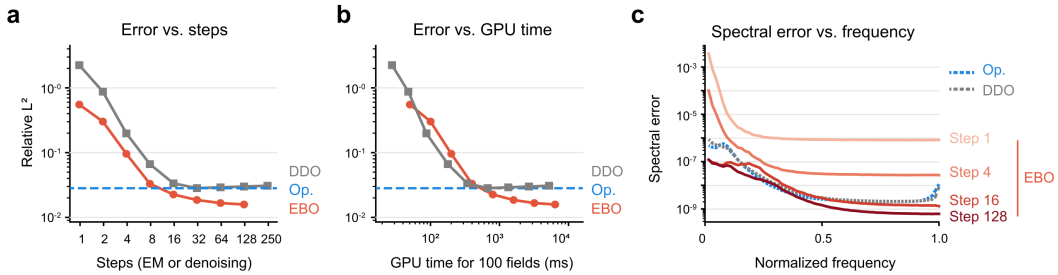


Figure 6: **EBO inference dynamics over iterative EM steps on Darcy flow. a,** Missing-region relative L^2 on 100 test fields against EM steps (EBO) or denoising steps (DDO); Op. (dashed) is one forward pass. **b,** The same against GPU time on one A100, with step counts on the curves (Op.: 18.3 ms). **c,** Spectral error after 1, 4, 16 and 128 EBO steps.

Sparse reconstruction of PDE solution fields EBO has the lowest missing-region relative L^2 in all 12 settings of Table 2, 25–64% below the operator, while DDO stays close to the operator. Over all 72 resolution and context settings, EBO is significantly better than the operator in 69 and never significantly worse (paired 95% intervals). On a Navier–Stokes field at 5% context, EBO's relative L^2 is 0.051, against 0.124 for the operator and 0.144 for DDO (Figure 5a). EBO recovers fine structure that both baselines blur (Figure 5a, boxed window) and follows the target most closely along the dashed row (Figure 5b). On Darcy flow at 20% context, EBO's error falls with the number of EM steps and passes the operator between 8 and 16 steps, while DDO levels off near the operator (Figure 6a). In GPU time, EBO passes the operator at about $43\times$ the operator's cost (Figure 6b), and after 128 steps its spectral error is below DDO's at every frequency (Figure 6c). See Appendix D for the detailed setups and results at every resolution.

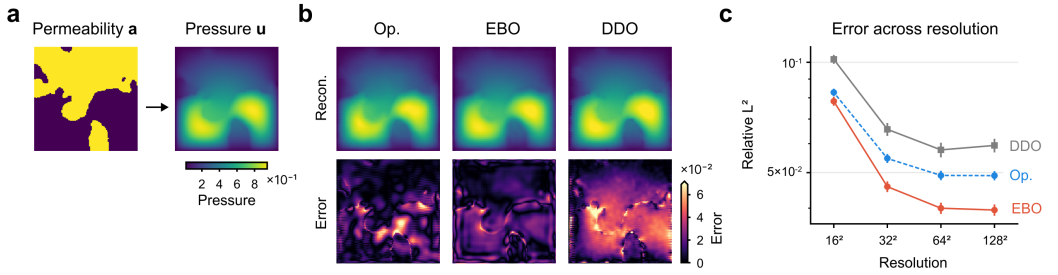


Figure 7: **Darcy flow from permeability a to pressure u. a,** A test pair at 128^2: the permeability a and the true pressure u. **b,** Predictions of the operator, EBO and DDO (top) and their absolute errors (bottom). **c,** Relative L^2 at each resolution over the 1,000 test pairs, with one model per method trained on all four resolutions.

Forward inference We also learn the map from Darcy permeability a to pressure u, where the input and target are different functions (Figure 7a). On a test pair at 128^2, EBO's relative L^2 is 0.035, against 0.040 for the operator and 0.069 for DDO (Figure 7b). With one FNO model per method trained across resolutions, EBO has the lowest error at every resolution, 19% below the operator and 33% below DDO at 128^2 (Figure 7c). See Appendix E for the detailed setup.

Table 2: **Sparse PDE reconstruction at resolution 128**. Each value reports the missing-region relative L^2 ($\times 10^2$; mean $\pm$ std). Appendix D.4 reports all resolutions and context ratios.

PDE	Context	Op. ↓	EBO ↓	DDO ↓
Darcy	20%	2.8±0.5	**1.6±0.3**	2.9±0.5
	40%	2.4±0.4	**1.4±0.2**	2.4±1.1
	60%	2.3±0.4	**1.4±0.2**	2.2±0.4
	80%	2.3±0.4	**1.7±0.3**	2.3±0.9
Burgers	20%	2.0±1.9	**0.7±0.6**	1.8±1.7
	40%	1.0±0.8	**0.4±0.2**	0.8±0.7
	60%	0.7±0.6	**0.3±0.2**	0.6±0.4
	80%	0.6±0.6	**0.3±0.2**	0.5±0.3
Navier–Stokes	20%	7.5±1.0	**3.5±0.4**	7.7±1.0
	40%	6.4±0.8	**2.8±0.3**	6.3±0.8
	60%	6.0±0.8	**2.7±0.3**	5.8±0.8
	80%	5.9±0.7	**3.1±0.3**	5.6±1.9

Table 3: **Energy-based unsupervised anomaly detection across tasks.** We report AUROC, AUPRC, and detection rates (Det.) at different false alarm rates (α).

Anomaly	Metric	Op. ↑	EBO ↑
Cosine → Step	AUROC	0.9685	**0.9809**
	AUPRC	0.9701	**0.9827**
	Det. (α=0.01)	0.8375	**0.9100**
	Det. (α=0.05)	0.8750	**0.9150**
High volatility	AUROC	0.9783	**0.9860**
	AUPRC	0.9670	**0.9710**
	Det. (α=0.01)	0.6503	**0.8703**
	Det. (α=0.05)	0.9410	**0.9610**
Seizure	AUROC	0.7088	**0.8480**
	AUPRC	0.8081	**0.8859**
	Det. (α=0.05)	0.4887	**0.7375**
	Det. (α=0.1)	0.7212	**0.7388**

5.2 Function-level Anomaly in Real-World Tasks

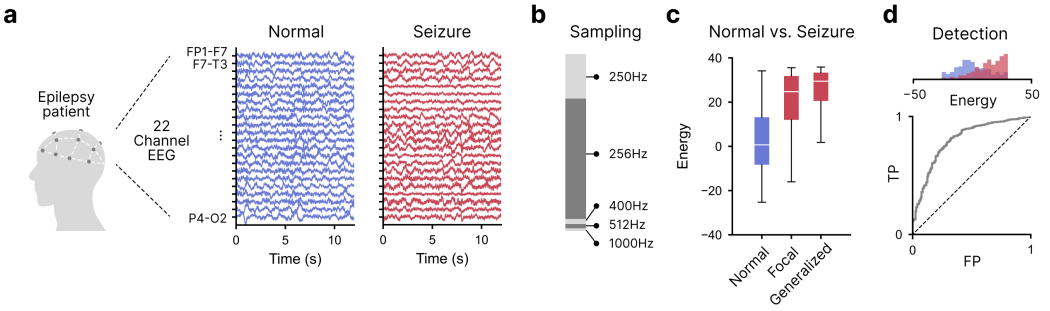


Figure 8: **Seizure as function level anomaly on electroencephalogram trained model. a,** Electroencephalogram (EEG) measured on epilepsy patients using the 10-20 system and TCP bipolar montage. Example shows a 22-channel EEG signal during normal and seizure states. **b,** Diversity of sampling frequency. The EEG corpus intrinsically contains 250 Hz, 256 Hz, 400 Hz, 512 Hz, and 1000 Hz due to medical measurement setup. **c,** Energy of normal and seizure states. Two types of seizure are used: focal seizure and generalized seizure. **d,** Seizure detection performance. Energy distribution (top) and receiver operating characteristic curve (bottom).

Epilepsy seizure detection. We evaluate EBO on the Temple University Hospital Seizure Corpus (TUSZ) [29, 30], a large-scale annotated EEG dataset for seizure detection. We use the 22-channel TCP bipolar montage and segment recordings into 12-second windows (Figure 8a; see Appendix F for detailed setups). Notably, the EEG datasets are recorded at various sampling frequencies due to measurement conditions (Figure 8b). Although conventional approaches interpolate or downsample the raw signal to match regular grids, we used raw signals to train EBO by leveraging discretization invariance. The model is trained to learn neural dynamics on background (normal) EEG segments and evaluated on its ability to distinguish seizure events via energy-based scoring (Figure 8c). Remarkably, the energy measured on seizures is notably higher than on normal EEG. In addition, energy scores for generalized seizure EEG are higher than for focal seizure, indicating that energy does not merely reflect the presence of an anomaly but provides meaningful information regarding the degree of anomaly. Overall, our model, trained to learn neural dynamics, is capable of distinguishing function-level anomalies such as seizures even without supervision (Figure 8d, AUROC $= 0.848$).

Financial volatility detection. We apply EBO to limit order book (LOB) data from high-frequency financial markets to learn market microstructure dynamics (Figure 9a; we used custom-collected LOB data from the Korea futures market from July 2, 2024 to Dec 12, 2024 for training and from Jan 17, 2025 to Mar 13, 2025 for testing. See Appendix G for detailed setups). LOB contains raw-level market events related to orders and executions, providing rich information to model dynamics

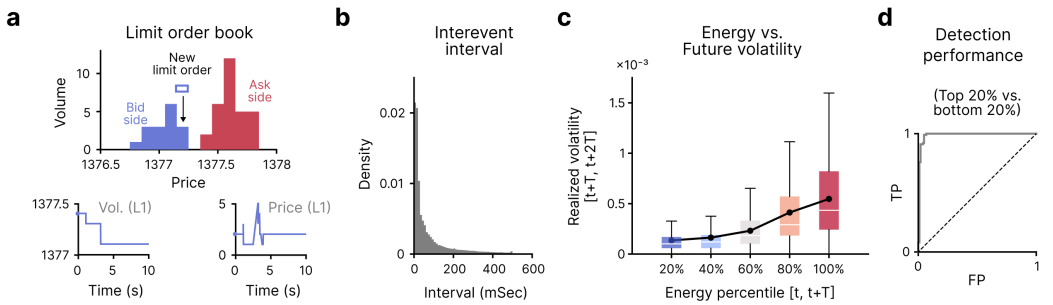


Figure 9: **Energy predicts future volatility. a,** Example snapshot of a limit order book (left). Red indicates ask orders and blue indicates bid orders, with cumulative volume at each price level. The volume and price at each level vary over time due to market events such as new limit orders, market orders, and cancellations (right). The data were collected from the Korea Exchange. **b,** Interevent intervals. **c,** Energy vs. future volatility. Current energy (computed from data in [t, t+T]) is binned by percentile, and future realized volatility (computed from data in [t+T, t+2T]) is measured for each bin. **d,** Separation of the top 20% of future realized volatility from the bottom 20% by energy.

across market participants [31]. Previous literature leverages LOB data to predict toxic market flow, specifically anomalous market behavior such as volatility regime shifts or market crashes [32, 33]. We evaluated whether the energy of EBO trained on LOB data indicates high future volatility. Notably, since market events arrive at irregular times, the limit order book time series inherently lies on an irregular temporal grid (Figure 9b). EBO's discretization invariance naturally encodes 10-second windows containing a variable number of market events.[4] After learning the energy landscape of market dynamics, we compute energy with respect to various LOB snapshots. Notably, we confirmed that energy computed from current LOB snapshots is positively correlated with future realized volatility (Spearman's $\rho = 0.563$). As current energy increases, the corresponding future volatility also increases (Figure 9c). Energy separates the top 20% of future realized volatility from the bottom 20% (Figure 9d; AUROC $= 0.986$). These labels depend only on future volatility, not on the volatility of the current window, so a window that is already volatile and stays volatile counts as high-volatility. The results therefore show that energy separates high from low future volatility levels.

6 Conclusion

We introduce Energy-Based Operators (EBO), a framework for learning conditional energy functionals over function spaces. By combining neural operators with energy-based models and iterative energy minimization, EBO reconstructs functions by refining random initial functions toward low energy. We prove discretization invariance and universal approximation of continuous energy functionals for EBOs built from integral-kernel layers. Empirically, EBO achieves strong results in function reconstruction and generation, and its energy enables effective detection in tasks such as EEG-based seizure identification and financial market volatility prediction.

Limitations. EBO is more expensive at inference than a one-shot operator, because every step evaluates the energy gradient. On Darcy flow it needs about $43\times$ the operator's GPU time to reach a lower error than the operator, and about $350\times$ at $K = 128$ (Appendix D.5), so our results do not establish an advantage at equal compute; Equilibrium Matching [5] may reduce this cost. EBO's gains are also limited to smooth function classes. On the Izhikevich model, whose spikes end in discontinuous resets, the operator has lower error than EBO with every backbone (Table 1), and EBO places the spikes at the right times but underestimates their amplitudes (Figure 2b,c). Future work includes objective-guided inverse sampling [34] and applications to physical and neural dynamics with known energy structure [35, 36].

7 Code availability

Python 3.12 (Python software foundation) with PyTorch 2.12 was used to perform the simulation. The code used in this work is available at `https://github.com/bizoffermark/ebo`.

[4] Event times are not uniformly distributed within a window, so here the mean of the per-point energies estimates their average with respect to the distribution of event times.

Acknowledgments and Disclosure of Funding

This study was supported by the AI Research and Development Center of the Korea Military Academy. Hong Chul Nam is supported by a PhD fellowship for the Center for Data Science at NYU.

References

[1] Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* **79**, 2554–2558 (1982).

[2] Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for boltzmann machines. *Cognitive Science* **9**, 147–169 (1985).

[3] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M. & Huang, F. J. A tutorial on energy-based learning. In *Predicting Structured Data* (MIT Press, 2006).

[4] Du, Y. & Mordatch, I. Implicit generation and modeling with energy based models. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019).

[5] Wang, R. & Du, Y. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300* (2025).

[6] Gladstone, A. *et al.* Energy-based transformers are scalable learners and thinkers. In *International Conference on Learning Representations* (2026).

[7] Oarga, A. & Du, Y. Generalizable reasoning through compositional energy minimization. In *Advances in Neural Information Processing Systems*, vol. 38 (2025).

[8] Berner, J. *et al.* Principled approaches for extending neural architectures to function spaces for operator learning. *arXiv preprint arXiv:2506.10973* (2025).

[9] Lim, J. H. *et al.* Score-based diffusion models in function space. *Journal of Machine Learning Research* **26**, 1–62 (2025).

[10] Wang, S., Dou, Z., Shan, S., Liu, T.-R. & Lu, L. Fundiff: Diffusion models over function spaces for physics-informed generative modeling. *arXiv preprint arXiv:2506.07902* (2025).

[11] Franzese, G. *et al.* Continuous-time functional diffusion processes. *Advances in Neural Information Processing Systems* **36**, 37370–37400 (2023).

[12] Zhang, B. & Wonka, P. Functional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4723–4732 (2024).

[13] Du, Y., Li, S., Tenenbaum, J. & Mordatch, I. Learning iterative reasoning through energy minimization. In *International Conference on Machine Learning*, 5570–5582 (PMLR, 2022).

[14] Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).

[15] Li, Z. *et al.* Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations* (2021).

[16] Sitzmann, V., Martel, J., Bergman, A., Lindell, D. & Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems* **33**, 7462–7473 (2020).

[17] Lanthaler, S., Li, Z. & Stuart, A. M. Nonlocality and nonlinearity implies universality in operator learning. *Constructive Approximation* (2025).

[18] Kovachki, N. *et al.* Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research* **24**, 1–97 (2023).

[19] Hao, Z. *et al.* Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, 12556–12569 (PMLR, 2023).

[20] Alkin, B. *et al.* Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems* **37**, 25152–25194 (2024).

[21] Wu, H., Luo, H., Wang, H., Wang, J. & Long, M. Transolver: A fast transformer solver for pdes on general geometries. In *International Conference on Machine Learning* (2024).

[22] Cao, S. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems* **34**, 24924–24940 (2021).

[23] Li, Z., Shu, D. & Barati Farimani, A. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems* **36**, 28010–28039 (2023).

[24] Song, Y. & Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288* (2021).

[25] Hoover, B. *et al.* Energy transformer. *Advances in Neural Information Processing Systems* **36**, 27532–27559 (2023).

[26] Zhang, Q., Krotov, D. & Karniadakis, G. E. Operator learning for reconstructing flow fields from sparse measurements: an energy transformer approach. *Journal of Computational Physics* 114148 (2025).

[27] Oh, T. I., Nam, H. C., Park, C. & Cho, H. Funcanode: A function level anomaly detection in device simulation. In *2024 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 01–04 (IEEE, 2024).

[28] Calvello, E., Kovachki, N. B., Levine, M. E. & Stuart, A. M. Continuum attention for neural operators. *arXiv preprint arXiv:2406.06486* (2024).

[29] Shah, V. *et al.* The temple university hospital seizure detection corpus. *Frontiers in Neuroinformatics* **12**, 83 (2018).

[30] Obeid, I. & Picone, J. The temple university hospital eeg data corpus. *Frontiers in neuroscience* **10**, 196 (2016).

[31] Gould, M. D. *et al.* Limit order books. *Quantitative Finance* **13**, 1709–1742 (2013).

[32] Easley, D., López de Prado, M. M. & O'Hara, M. The microstructure of the" flash crash": Flow toxicity, liquidity crashes, and the probability of informed trading. *Journal of Portfolio Management* **37**, 118–128 (2011).

[33] Easley, D., López de Prado, M. M. & O'Hara, M. Flow toxicity and liquidity in a high-frequency world. *The Review of Financial Studies* **25**, 1457–1493 (2012).

[34] Wu, T. *et al.* Compositional generative inverse design. In *The Twelfth International Conference on Learning Representations* (2024). URL `https://openreview.net/forum?id=wmX0CqFSd7`.

[35] Friston, K. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience* **11**, 127–138 (2010).

[36] Khona, M. & Fiete, I. R. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience* **23**, 744–766 (2022).

[37] Rahimi, A. & Recht, B. Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y. & Roweis, S. (eds.) *Advances in Neural Information Processing Systems*, vol. 20 (Curran Associates, Inc., 2007).

[38] Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366 (1989).

[39] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* **2**, 303–314 (1989).

[40] Leshno, M., Lin, V. Y., Pinkus, A. & Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks* **6**, 861–867 (1993).

[41] Izhikevich, E. M. Simple model of spiking neurons. *IEEE Transactions on Neural Networks* **14**, 1569–1572 (2003).

A Random Fourier Features for Gaussian Random Field Sampling

While FFT can be applied to efficiently sample Gaussian random fields with regular grids, it is not applicable to dense and irregular fields. Sampling from a Gaussian Random Field (GRF) with covariance kernel K at n arbitrary locations traditionally requires $\mathcal{O}(n^3)$ computation for Cholesky decomposition. We adopt Random Fourier Features (RFF) [37] to reduce this to $\mathcal{O}(nD)$, where D is the number of features.

For a shift-invariant kernel $K(x, y) = \sigma^2 k(x - y)$ with $k(0) = 1$, Bochner's theorem guarantees a spectral probability density $p(\omega)$ such that:

$$k(x-y) = \int p(\omega) e^{i\omega^\top (x-y)} d\omega = \mathbb{E}_{\omega \sim p}\left[2\cos(\omega^\top x + b)\cos(\omega^\top y + b)\right] \tag{9}$$

where $b \sim \text{Uniform}(0, 2\pi)$. For the RBF kernel $k(x-y) = \exp\left(-\frac{\|x-y\|^2}{2\ell^2}\right)$ and the spectral density is $p(\omega) = \mathcal{N}(0, \ell^{-2} I)$.

We approximate the kernel via Monte Carlo sampling:

$$K(x, y) \approx \phi(x)^\top \phi(y), \quad \text{where} \quad \phi(x) = \sigma\sqrt{\frac{2}{D}}\cos(Wx + b) \tag{10}$$

with $W \in \mathbb{R}^{D \times n_y}$, where n_y is the dimension of the coordinates, having rows $\omega_j \sim \mathcal{N}(0, \ell^{-2} I)$ and $b \in \mathbb{R}^D$ with entries $b_j \sim \text{Uniform}(0, 2\pi)$. As a result, we apply the kernel trick to sample from approximate Gaussian random fields.

To sample $f \sim \mathcal{N}(0, C)$ at coordinates $X = \{x_1, \ldots, x_n\}$, we compute:

$$f(X) = \Phi w, \quad w \sim \mathcal{N}(0, I_D) \tag{11}$$

where $\Phi \in \mathbb{R}^{n \times D}$ is the feature matrix with rows $\phi(x_i)^\top$.

The approximation error satisfies $\sup_{x,y} |K(x, y) - \hat{K}(x, y)| = \mathcal{O}\big(\sqrt{\log D / D}\big)$ with high probability, with constants depending on σ^2, the length scale ℓ, the coordinate dimension and the domain diameter [37]. For the RBF kernel, each entry of $\hat{K}$ is an average of D independent terms, with standard deviation between $\sigma^2/\sqrt{2D}$ (at $x = y$) and $\sigma^2/\sqrt{D}$ (for distant points); at $D = 512$ this is $0.031\sigma^2$ to $0.044\sigma^2$. The largest error over many pairs of points is several times larger: about $0.1\sigma^2$ over 128 points on $[0, 1]$ with $\ell = 0.1$.

This approach offers three key benefits: (1) $\mathcal{O}(nD)$ complexity enables sampling at thousands of points, (2) no matrix factorization avoids numerical instability from ill-conditioned covariance matrices, and (3) arbitrary coordinates are handled naturally without grid assumptions.

B Theoretical Analysis

In this section, we establish two fundamental theoretical properties of Energy-Based Operators: (i) discretization invariance, ensuring that our energy estimates converge to well-defined continuum limits, and (ii) universal approximation, demonstrating that EBOs can approximate any jointly continuous energy to arbitrary precision.

B.1 Preliminaries and Notation

We first establish the necessary notation and assumptions, building on the notation introduced in Section 2. Recall that $\Omega_X \subset \mathbb{R}^{n_x}$ and $\Omega_Y \subset \mathbb{R}^{n_y}$ are bounded domains with the uniform probability measures $\lambda_X(A) = |A|/|\Omega_X|$ and $\lambda_Y(A) = |A|/|\Omega_Y|$, where $|\cdot|$ is Lebesgue measure (hence $\lambda_X(\Omega_X) = \lambda_Y(\Omega_Y) = 1$, and integrals against them are spatial averages). Theorems B.11 and B.26 hold for any finite reference measures, with $\lambda_X(\Omega_X)$ and $\lambda_Y(\Omega_Y)$ entering the constants. The attention result of Appendix B.3 needs the normalized measures, because there λ_X and λ_Y are the laws of the sample points and the limits of their empirical measures. The spaces $\mathcal{H}_X$, $\mathcal{H}_Y$ are separable Hilbert spaces of functions on these domains.

Additional notation for this section. We introduce the following notation for the architecture specification:

- $d_h \in \mathbb{N}$: latent dimension, the number of channels of the latent functions $v^{(t)}$.
- $T \in \mathbb{N}$: number of kernel integral layers.
- $P_X : \mathbb{R}^d \times \mathbb{R}^{n_x} \to \mathbb{R}^{d_h}$, $P_Y : \mathbb{R}^m \times \mathbb{R}^{n_y} \to \mathbb{R}^{d_h}$: lifting maps.
- $\kappa_{YX}^{(t)} : \Omega_Y \times \Omega_X \to \mathbb{R}^{d_h \times d_h}$: cross kernel (context to target) at layer t.
- $\kappa_{YY}^{(t)} : \Omega_Y \times \Omega_Y \to \mathbb{R}^{d_h \times d_h}$: self kernel (target to target) at layer t.
- $Q : \mathbb{R}^{d_h} \to \mathbb{R}$: energy head (pointwise map from the latent function to the per-point energy).
- $\mathcal{K}$: compact subsets of function spaces (to avoid confusion with minimization steps K).

Assumption B.1 (Regularity of function spaces)**.** We assume:

1. $\mathcal{H}_X \hookrightarrow C(\overline{\Omega_X}; \mathbb{R}^d)$ and $\mathcal{H}_Y \hookrightarrow C(\overline{\Omega_Y}; \mathbb{R}^m)$ with continuous embeddings.
2. There exist constants $C_X, C_Y > 0$ such that $\|f\|_{C(\overline{\Omega_X})} \le C_X \|f\|_{\mathcal{H}_X}$ and $\|u\|_{C(\overline{\Omega_Y})} \le C_Y \|u\|_{\mathcal{H}_Y}$.

This assumption is satisfied, for instance, when $\mathcal{H}_X = H^s(\Omega_X; \mathbb{R}^d)$ with $s > n_x/2$ by the Sobolev embedding theorem.

Assumption B.2 (C^p regularity on $\mathcal{K}$)**.** Fix an integer $p \ge 1$ and a compact set $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$. Assume that for all $(f, u) \in \mathcal{K}$:

1. The lifted input satisfies
$$h_f \in C^p(\overline{\Omega_X}; \mathbb{R}^{d_h}) \quad \text{with} \quad \|h_f\|_{C^p(\overline{\Omega_X})} \le C_{\mathcal{K}}.$$
2. Every latent function integrated against the self kernel satisfies
$$v^{(t)} \in C^p(\overline{\Omega_Y}; \mathbb{R}^{d_h}), \qquad \max_{0 \le t < T} \|v^{(t)}\|_{C^p(\overline{\Omega_Y})} \le C_{\mathcal{K}}.$$
3. The per-point energy $Q \circ v^{(T)}$ satisfies
$$Q \circ v^{(T)} \in C^p(\overline{\Omega_Y}) \quad \text{with} \quad \|Q \circ v^{(T)}\|_{C^p(\overline{\Omega_Y})} \le C_{\mathcal{K}}.$$

Since $v^{(0)} = h_u = P_Y(u(\cdot), \cdot)$, item 2 is a smoothness condition on the target functions in $\mathcal{K}$: with an affine lifting that is injective in u, it holds only if $u \in C^p(\overline{\Omega_Y}; \mathbb{R}^m)$, which is stronger than $u \in H^s$ with $s > n_y/2$. Sample paths of a Matérn-ν field with $\nu \in \{0.5, 1.5, 2.5\}$ are C^p only for $p < \nu$, so

the rate of Theorem B.11 does not cover energies evaluated at exact Matérn samples with $\nu < p$, such as $\nu = 0.5$ (Table A4). The random Fourier feature samples of Appendix A are finite sums of cosines and therefore smooth, but their C^p norms, and with them the constant in the rate, grow with the sampled frequencies.

Definition B.3 (Discrete refinement)**.** A **discrete refinement** of a bounded domain $\Omega \subset \mathbb{R}^n$ is a sequence of finite sets $(D_L)_{L=1}^{\infty}$ with $|D_L| = L$ such that for any $\varepsilon > 0$, there exists $L(\varepsilon) \in \mathbb{N}$ with

$$\Omega \subseteq \bigcup_{x \in D_L} B(x, \varepsilon) \quad \text{for all } L \geq L(\varepsilon), \tag{12}$$

where $B(x, \varepsilon) = \{y \in \mathbb{R}^n : \|y - x\| < \varepsilon\}$.

Definition B.4 (Quadrature rule)**.** A **quadrature rule** on Ω associated with discretization $D_L = \{x_1, \ldots, x_L\}$ is a set of weights $\{w_\ell\}_{\ell=1}^{L} \subset \mathbb{R}_+$ satisfying $\sum_{\ell=1}^{L} w_\ell = \lambda(\Omega)$. We say the quadrature rule has **order** p if for all $g \in C^p(\overline{\Omega})$:

$$\left| \int_\Omega g(x)\, d\lambda(x) - \sum_{\ell=1}^{L} w_\ell g(x_\ell) \right| \leq C_p \|g\|_{C^p} L^{-p/n}. \tag{13}$$

B.2 Discretization Invariance

We now formalize and prove discretization invariance for EBOs.

Definition B.5 (EBO Architecture - Continuum Form)**.** An EBO $\Psi_\theta : \mathcal{H}_X \to C(\mathcal{H}_Y; \mathbb{R})$ (Definition 2.1) has the following structure, which specifies the energy $\Psi_\theta(f, u) = \Psi_\theta(f)(u)$:

1. **Lifting:** Continuous pointwise maps $P_X : \mathbb{R}^d \times \mathbb{R}^{n_x} \to \mathbb{R}^{d_h}$ and $P_Y : \mathbb{R}^m \times \mathbb{R}^{n_y} \to \mathbb{R}^{d_h}$ yield lifted representations:

$$h_f(x) = P_X(f(x), x), \quad h_u(y) = P_Y(u(y), y). \tag{14}$$

2. **Kernel Integral Layers**: For $t = 1, \ldots, T$,

$$\begin{aligned} v^{(t)}(y) = \sigma\Big(& W^{(t)} v^{(t-1)}(y) + b^{(t)}(y) \\ & + \int_{\Omega_X} \kappa_{YX}^{(t)}(y, x) h_f(x)\, d\lambda_X(x) \\ & + \int_{\Omega_Y} \kappa_{YY}^{(t)}(y, z) v^{(t-1)}(z)\, d\lambda_Y(z) \Big), \end{aligned} \tag{15}$$

   where $v^{(0)} = h_u$, σ is Lipschitz, and

$$\kappa_{YX}^{(t)} \in C^p(\overline{\Omega_Y} \times \overline{\Omega_X}; \mathbb{R}^{d_h \times d_h}), \qquad \kappa_{YY}^{(t)} \in C^p(\overline{\Omega_Y} \times \overline{\Omega_Y}; \mathbb{R}^{d_h \times d_h}).$$

   The biases $b^{(t)}$ are continuous on $\overline{\Omega_Y}$.

3. **Energy Head:** A continuous map $Q : \mathbb{R}^{d_h} \to \mathbb{R}$ applied pointwise gives the per-point energy $Q(v^{(T)}(y))$, which is integrated:

$$\Psi_\theta(f, u) = \int_{\Omega_Y} Q(v^{(T)}(y))\, d\lambda_Y(y). \tag{16}$$

The layers above are one instance of the neural operator $\mathcal{G}_\theta : \mathcal{H}_X \times \mathcal{H}_Y \to C(\overline{\Omega_Y}; \mathbb{R}^{d_h})$ of Section 2.5, with latent function $\mathcal{G}_\theta(f, u) = v^{(T)}$. Each $v^{(t)}$ is continuous on $\overline{\Omega_Y}$ because the lifting maps, kernels, biases and activation are continuous and u is continuous by Assumption B.1. For the same reasons each layer depends continuously on u in the sup norm, and $\mathcal{H}_Y$ embeds continuously in $C(\overline{\Omega_Y}; \mathbb{R}^m)$, so $u \mapsto \Psi_\theta(f, u)$ is continuous on $\mathcal{H}_Y$ and $\Psi_\theta(f) \in C(\mathcal{H}_Y; \mathbb{R})$ as Definition 2.1 requires.

Interpretation of the two kernels. The kernel $\kappa_{YX}^{(t)}$ transfers information from the fixed input representation h_f to the output representation, while $\kappa_{YY}^{(t)}$ allows every output location to interact with every other output location. Thus the self kernel is bidirectional (there is no causal mask), while h_f is held fixed across layers. Bidirectionality does not require $\kappa_{YY}^{(t)}(y,z) = \kappa_{YY}^{(t)}(z,y)^\top$. These are integral-operator kernels on the spatial product domains $\Omega_Y \times \Omega_X$ and $\Omega_Y \times \Omega_Y$; they act on latent functions and are distinct from the covariance kernel of the Gaussian reference measure. Both kernels depend only on positions and are fixed for a given θ, as in the kernel integral operators of neural operators [18]; the spectral convolution of FNO is one example (Remark B.13). Softmax attention, whose kernel also depends on the latent features, is treated in Appendix B.3.

Definition B.6 (Discretized EBO)**.** Given discretizations $D_{M_X} = \{x_i\}_{i=1}^{M_X} \subset \Omega_X$ and $D_{M_Y} = \{y_j\}_{j=1}^{M_Y} \subset \Omega_Y$ with quadrature weights $\{w_i^X\}$ and $\{w_j^Y\}$, the **discretized EBO** is:

1. **Discretized Kernel Integral Layers**: Set $v_j^{(0)} = h_u(y_j)$ and

$$\begin{aligned} v_j^{(t)} = \sigma\Big(& W^{(t)} v_j^{(t-1)} + b^{(t)}(y_j) \\ & + \sum_{i=1}^{M_X} w_i^X \kappa_{YX}^{(t)}(y_j, x_i) h_f(x_i) \\ & + \sum_{\ell=1}^{M_Y} w_\ell^Y \kappa_{YY}^{(t)}(y_j, y_\ell) v_\ell^{(t-1)} \Big). \end{aligned} \tag{17}$$

2. **Discretized Energy:**

$$\Psi_\theta^{M_X, M_Y}(f, u) = \sum_{j=1}^{M_Y} w_j^Y Q(v_j^{(T)}). \tag{18}$$

Definition B.7 (Discretization Invariance for EBOs)**.** Let $\Theta \subseteq \mathbb{R}^{p_\theta}$ be a parameter space. A parametric family of EBOs $\{\Psi_\theta\}_{\theta \in \Theta}$ is **discretization-invariant** if, for any discrete refinements (D_{M_X}) of Ω_X and (D_{M_Y}) of Ω_Y with associated quadrature rules of order $p \geq 1$, the following holds:

For any $\theta \in \Theta$ and any compact set $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$:

$$\lim_{M_X, M_Y \to \infty} \sup_{(f,u) \in \mathcal{K}} \left| \Psi_\theta^{M_X, M_Y}(f, u) - \Psi_\theta(f, u) \right| = 0. \tag{19}$$

We now prove discretization invariance through a sequence of lemmas.

Lemma B.8 (Quadrature Convergence for Integral Kernels)**.** *Let $\kappa \in C^p(\overline{\Omega_Y} \times \overline{\Omega_X}; \mathbb{R}^{d_h \times d_h})$ and $h \in C^p(\overline{\Omega_X}; \mathbb{R}^{d_h})$. For a quadrature rule of order p on Ω_X with M_X points $\{x_i\}$ and weights $\{w_i^X\}$, one has:*

$$\begin{aligned} \sup_{y \in \Omega_Y} \Big\| \int_{\Omega_X} \kappa(y, x) h(x) \, d\lambda_X(x) - \sum_{i=1}^{M_X} w_i^X \, \kappa(y, x_i) h(x_i) \Big\| \\ \leq \; C \, \|\kappa\|_{C^p(\overline{\Omega_Y} \times \overline{\Omega_X})} \, \|h\|_{C^p(\overline{\Omega_X})} \, M_X^{-p/n_x}, \end{aligned}$$

where C depends on p, the latent dimension d_h and the domain Ω_X.

Proof. Fix $y \in \Omega_Y$ and define $g_y : \Omega_X \to \mathbb{R}^{d_h}$ by $g_y(x) = \kappa(y, x) h(x)$.

Step 1: We show g_y inherits regularity from κ and h. Since κ is continuous on the compact set $\overline{\Omega_Y} \times \overline{\Omega_X}$, it is uniformly continuous. For any α with $|\alpha| \leq p$:

$$\|D_x^\alpha g_y\|_\infty \leq \sum_{\beta \leq \alpha} C_{\alpha,\beta} \|D_x^\beta \kappa(y, \cdot)\|_\infty \|D_x^{\alpha-\beta} h\|_\infty. \tag{20}$$

Thus $\|g_y\|_{C^p(\Omega_X)} \leq C_1 \|\kappa\|_{C^p} \|h\|_{C^p}$ uniformly in y.

Step 2: Apply the quadrature error bound componentwise. For each component $k \in \{1, \dots, d_h\}$:

$$\left| \int_{\Omega_X} [g_y(x)]_k \, d\lambda_X(x) - \sum_{i=1}^{M_X} w_i^X [g_y(x_i)]_k \right| \leq C_p \|[g_y]_k\|_{C^p} M_X^{-p/n_x}. \tag{21}$$

Step 3: Combine components.

$$\left\| \int_{\Omega_X} \kappa(y,x) h(x) \, d\lambda_X(x) - \sum_{i=1}^{M_X} w_i^X \kappa(y,x_i) h(x_i) \right\| \tag{22}$$

$$\leq \sqrt{d_h} \max_k \left| \int_{\Omega_X} [g_y(x)]_k \, d\lambda_X - \sum_{i=1}^{M_X} w_i^X [g_y(x_i)]_k \right| \tag{23}$$

$$\leq \sqrt{d_h} C_p \|g_y\|_{C^p} M_X^{-p/n_x} \tag{24}$$

$$\leq C \|\kappa\|_{C^p} \|h\|_{C^p} M_X^{-p/n_x}, \tag{25}$$

where $C = \sqrt{d_h} C_p C_1$.

Step 4: The bound is uniform in y since the constants depend only on $\|\kappa\|_{C^p(\Omega_Y \times \Omega_X)}$. □

The same quadrature estimate applies to the self kernel after replacing Ω_X, h, M_X, n_x by $\Omega_Y, v^{(t-1)}, M_Y, n_y$.

Lemma B.9 (Stability of Iterated Layers)**.** *Let $\mathcal{L}[v,h]$ be the layer in* (15)*, with layer indices omitted, and let σ be L_σ-Lipschitz. If $\|v - \tilde{v}\|_\infty \leq \delta_v$ and $\|h - \tilde{h}\|_\infty \leq \delta_h$, then*

$$\begin{aligned} \|\mathcal{L}[v,h] - \mathcal{L}[\tilde{v},\tilde{h}]\|_\infty \leq L_\sigma \Big[&\big(\|W\|_{\mathrm{op}} + \lambda_Y(\Omega_Y) \|\kappa_{YY}\|_\infty \big) \delta_v \\ &+ \lambda_X(\Omega_X) \|\kappa_{YX}\|_\infty \delta_h \Big], \end{aligned} \tag{26}$$

where kernel sup norms use the matrix operator norm.

Proof. Subtract the two layer preactivations. The local term is bounded by $\|W\|_{\mathrm{op}} \delta_v$, the input integral by $\lambda_X(\Omega_X) \|\kappa_{YX}\|_\infty \delta_h$, and the output integral by $\lambda_Y(\Omega_Y) \|\kappa_{YY}\|_\infty \delta_v$. Apply the triangle inequality and the Lipschitz bound for σ, then take the supremum over y. The same estimate holds for the quadrature layers, since the weights are nonnegative and sum to the corresponding domain measures. □

Lemma B.10 (Layer-wise Discretization Error)**.** *Let $v^{(t)}$ be the continuum representation and define the quadrature extension by*

$$\begin{aligned} \tilde{v}^{(t)}(y) = \sigma\Big(&W^{(t)} \tilde{v}^{(t-1)}(y) + b^{(t)}(y) \\ &+ \sum_{i=1}^{M_X} w_i^X \kappa_{YX}^{(t)}(y, x_i) h_f(x_i) \\ &+ \sum_{\ell=1}^{M_Y} w_\ell^Y \kappa_{YY}^{(t)}(y, y_\ell) \tilde{v}^{(t-1)}(y_\ell) \Big), \end{aligned} \tag{27}$$

with $\tilde{v}^{(0)} = v^{(0)} = h_u$. Then $\tilde{v}^{(t)}(y_j) = v_j^{(t)}$ at every output quadrature point. Under Assumption B.2 and order-p quadrature on both domains,

$$\|v^{(t)} - \tilde{v}^{(t)}\|_\infty \leq C_t \big(M_X^{-p/n_x} + M_Y^{-p/n_y} \big), \tag{28}$$

where C_t is uniform over $\mathcal{K}$ and independent of M_X, M_Y.

Proof. Write $e_t = \|v^{(t)} - \tilde{v}^{(t)}\|_\infty$ and set $e_0 = 0$. Introduce $\hat{v}^{(t)}$ by using the same quadrature formula as $\tilde{v}^{(t)}$, but with the exact previous representation $v^{(t-1)}$ in both the local term and the output sum.

Step 1: Quadrature errors. Apply Lemma B.8 separately to the two integrals. Assumption B.2 bounds h_f and every $v^{(t-1)}$ in C^p uniformly on $\mathcal{K}$, so

$$\|v^{(t)} - \hat{v}^{(t)}\|_\infty \le A_t^X M_X^{-p/n_x} + A_t^Y M_Y^{-p/n_y}. \tag{29}$$

The constants A_t^X, A_t^Y include L_σ, the respective kernel C^p norms, and the uniform bounds on the source representations.

Step 2: Propagation of representation error. The input sums agree. Positivity and normalization of the output quadrature weights give

$$\|\hat{v}^{(t)} - \tilde{v}^{(t)}\|_\infty \le B_t e_{t-1}, \qquad B_t = L_\sigma\big(\|W^{(t)}\|_{\mathrm{op}} + \lambda_Y(\Omega_Y)\|\kappa_{YY}^{(t)}\|_\infty\big). \tag{30}$$

Step 3: Induction. Combining the bounds yields

$$e_t \le A_t^X M_X^{-p/n_x} + A_t^Y M_Y^{-p/n_y} + B_t e_{t-1}. \tag{31}$$

Thus $C_0 = 0$ and $C_t = \max\{A_t^X, A_t^Y\} + B_t C_{t-1}$ suffice. These constants remain finite for fixed depth T. □

Theorem B.11 (Discretization Invariance of EBO)**.** *Let Ψ_θ be an EBO as in Definition B.5 with:*

1. *Kernels $\kappa_{YX}^{(t)}$ and $\kappa_{YY}^{(t)}$ of class C^p on their respective product domains, as in* (15)*.*
2. *Lipschitz activations σ with constant L_σ.*
3. *Lipschitz energy head Q with constant L_Q.*
4. *Lifting maps P_X, P_Y that are Lipschitz in their first arguments.*

Assume further that Assumptions B.1 and B.2 hold on the compact set $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$. Let (D_{M_X}), (D_{M_Y}) be discrete refinements with quadrature rules of order $p \ge 1$.

Then on this compact set $\mathcal{K}$,

$$\sup_{(f,u)\in\mathcal{K}} \left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| = \mathcal{O}\left(M_X^{-p/n_x} + M_Y^{-p/n_y}\right) \quad \text{as } M_X, M_Y \to \infty. \tag{32}$$

Proof. Fix $(f,u) \in \mathcal{K}$. The proof proceeds in four steps.

Step 1: Uniform bounds on compact sets.

Since $\mathcal{K}$ is compact in $\mathcal{H}_X \times \mathcal{H}_Y$ and the embeddings into continuous functions are continuous (Assumption B.1), we have:

$$\sup_{(f,u)\in\mathcal{K}} \|f\|_{C(\overline{\Omega_X})} \le C_{\mathcal{K}}^X, \qquad \sup_{(f,u)\in\mathcal{K}} \|u\|_{C(\overline{\Omega_Y})} \le C_{\mathcal{K}}^Y. \tag{33}$$

For lifting maps continuous in both arguments and Lipschitz in their first arguments,

$$\begin{aligned} \|h_f\|_\infty &\le L_{P_X} C_{\mathcal{K}}^X + \sup_{x\in\overline{\Omega_X}} \|P_X(0,x)\|, \\ \|h_u\|_\infty &\le L_{P_Y} C_{\mathcal{K}}^Y + \sup_{y\in\overline{\Omega_Y}} \|P_Y(0,y)\|. \end{aligned} \tag{34}$$

The spatial C^p bounds needed for quadrature are supplied separately by Assumption B.2.

Step 2: Discretization error in the kernel integral layers. Use the quadrature extension $\tilde{v}^{(T)}$ from Lemma B.10. At every output quadrature point,

$$\|v^{(T)}(y_j) - v_j^{(T)}\| \le C_T\big(M_X^{-p/n_x} + M_Y^{-p/n_y}\big). \tag{35}$$

Here C_T depends on both kernel families, the layer parameters, and the uniform regularity bounds on $\mathcal{K}$.

Step 3: Discretization error in energy pooling.

The continuum energy is:

$$\Psi_\theta(f,u) = \int_{\Omega_Y} Q(v^{(T)}(y))\, d\lambda_Y(y). \tag{36}$$

The discretized energy (using exact $v^{(T)}$) would be:

$$\hat{\Psi}_\theta(f,u) = \sum_{j=1}^{M_Y} w_j^Y Q(v^{(T)}(y_j)). \tag{37}$$

By Assumption B.2, the function

$$y \mapsto Q(v^{(T)}(y))$$

belongs to $C^p(\overline{\Omega_Y})$ with a uniform C^p bound over $(f,u) \in \mathcal{K}$. Since the quadrature rule on Ω_Y has order p, the quadrature error satisfies

$$\left|\Psi_\theta(f,u) - \hat{\Psi}_\theta(f,u)\right| \le C_Q M_Y^{-p/n_y}, \tag{38}$$

where C_Q depends on $\|Q \circ v^{(T)}\|_{C^p}$ but is bounded uniformly over $(f,u) \in \mathcal{K}$.

Step 4: Combine errors.

The actual discretized energy is:

$$\Psi_\theta^{M_X,M_Y}(f,u) = \sum_{j=1}^{M_Y} w_j^Y Q(\tilde{v}^{(T)}(y_j)). \tag{39}$$

We decompose:

$$\left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| \tag{40}$$

$$\le \underbrace{\left|\sum_{j=1}^{M_Y} w_j^Y Q(\tilde{v}^{(T)}(y_j)) - \sum_{j=1}^{M_Y} w_j^Y Q(v^{(T)}(y_j))\right|}_{(I)} \tag{41}$$

$$+ \underbrace{\left|\sum_{j=1}^{M_Y} w_j^Y Q(v^{(T)}(y_j)) - \int_{\Omega_Y} Q(v^{(T)}(y))\, d\lambda_Y(y)\right|}_{(II)}. \tag{42}$$

Bound on (I): By Lipschitz property of Q and Step 2:

$$(I) \le \sum_{j=1}^{M_Y} w_j^Y \left|Q(\tilde{v}^{(T)}(y_j)) - Q(v^{(T)}(y_j))\right| \tag{43}$$

$$\le L_Q \sum_{j=1}^{M_Y} w_j^Y \|\tilde{v}^{(T)}(y_j) - v^{(T)}(y_j)\| \tag{44}$$

$$\le L_Q \lambda_Y(\Omega_Y) C_T \big(M_X^{-p/n_x} + M_Y^{-p/n_y}\big). \tag{45}$$

Bound on (II): By Step 3:

$$(II) \le C_Q M_Y^{-p/n_y}. \tag{46}$$

Final bound:

$$\begin{aligned}\left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| \le{}& L_Q \lambda_Y(\Omega_Y) C_T \big(M_X^{-p/n_x} + M_Y^{-p/n_y}\big) \\ &+ C_Q M_Y^{-p/n_y}.\end{aligned} \tag{47}$$

Since all constants are uniform over the compact set $\mathcal{K}$, taking the supremum:

$$\sup_{(f,u)\in\mathcal{K}} \left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| = \mathcal{O}\left(M_X^{-p/n_x} + M_Y^{-p/n_y}\right) \quad \text{as } M_X, M_Y \to \infty. \tag{48}$$

This completes the proof. □

Remark B.12 (Convergence rates). On quasi-uniform grids with $M_X = M_Y = M$ and first-order quadrature ($p = 1$), the bound becomes

$$O\Big(M^{-1/\max(n_x,n_y)}\Big).$$

Higher-order quadrature rules increase p and yield faster convergence. This shows that incorporating quadrature weights into the kernel integrals and pooling makes EBOs inherently compatible with multi-resolution and irregular discretizations, a practical advantage for scientific datasets.

Remark B.13 (FNO). The spectral convolution of FNO [15] keeps the Fourier modes $|k| \le k_{\max}$ of a function on the torus and multiplies them by learned matrices R_k. It equals $\int \kappa(y-z)\, v(z)\, dz$ with the trigonometric polynomial $\kappa(x) = \sum_{|k|\le k_{\max}} R_k e^{2\pi i k\cdot x}$, a C^∞ self kernel that depends only on positions. On the uniform grid $\{i/n\}$, the FFT computes this integral with equal weights, which is the trapezoidal rule and has order p for periodic C^p integrands. Our FNO reads the context as input channels at each grid point. Since context and target share the grid, this is a lifting $P_Y(u(y), f(y), y)$, which is pointwise and adds no quadrature error.

B.3 Softmax Attention

The kernels of Definition B.5 depend only on positions. In softmax attention, the kernel also depends on the latent features. For this case, we use the continuum attention operators [28] and take their approximation results as given.

Definition B.14 (Continuum softmax attention [28])**.** Let $W_Q, W_K \in \mathbb{R}^{d_k\times d_h}$ and $W_V \in \mathbb{R}^{d_h\times d_h}$. For $v \in C(\overline{\Omega_Y};\mathbb{R}^{d_h})$ and $h \in C(\overline{\Omega_X};\mathbb{R}^{d_h})$, self-attention and cross-attention are

$$\mathcal{A}_{YY}(v)(y) = \frac{\int_{\Omega_Y} \exp\big(\langle W_Q v(y), W_K v(z)\rangle\big)\, W_V v(z)\, dz}{\int_{\Omega_Y} \exp\big(\langle W_Q v(y), W_K v(z)\rangle\big)\, dz},$$

$$\mathcal{A}_{YX}(v,h)(y) = \frac{\int_{\Omega_X} \exp\big(\langle W_Q v(y), W_K h(x)\rangle\big)\, W_V h(x)\, dx}{\int_{\Omega_X} \exp\big(\langle W_Q v(y), W_K h(x)\rangle\big)\, dx},$$

as in Definitions 4–5 and 10–11 of [28]. A factor $1/\sqrt{d_k}$ is absorbed into W_Q, and multiple heads are sums of such terms after output projections. An *attention EBO* is the EBO of Definition B.5 with the two kernel integrals in (15) replaced by $\mathcal{A}^{(t)}_{YX}(v^{(t-1)}, h_f)(y)$ and $\mathcal{A}^{(t)}_{YY}(v^{(t-1)})(y)$, with separate parameters for each layer and each operator. Its discretization at points $\{x_i\}_{i=1}^{M_X}$ and $\{y_j\}_{j=1}^{M_Y}$ replaces every integral by the sum over the points, which is standard softmax attention, and averages the per-point energies: $\Psi_\theta^{M_X,M_Y}(f,u) = \frac{1}{M_Y}\sum_{j=1}^{M_Y} Q(v_j^{(T)})$.

Lemma B.15 (Stability of softmax attention)**.** *Let $q, \tilde q, a_1, \dots, a_N, \tilde a_1, \dots, \tilde a_N \in \mathbb{R}^{d_h}$ have norms at most R, and let $\delta = \max\{\|q-\tilde q\|, \max_\ell \|a_\ell - \tilde a_\ell\|\}$. Let $s_\ell = \langle W_Q q, W_K a_\ell\rangle$, $p_\ell = e^{s_\ell}/\sum_m e^{s_m}$, and define $\tilde s_\ell, \tilde p_\ell$ in the same way. Then*

$$\Big\|\sum_\ell p_\ell W_V a_\ell - \sum_\ell \tilde p_\ell W_V \tilde a_\ell\Big\| \le \|W_V\|\big(1 + 4R^2\|W_Q\|\|W_K\|\big)\,\delta.$$

The same bound holds with the sums replaced by integrals against a probability measure.

Proof. Split the difference into $\sum_\ell (p_\ell - \tilde p_\ell) W_V a_\ell$ and $\sum_\ell \tilde p_\ell W_V (a_\ell - \tilde a_\ell)$. The second term is at most $\|W_V\|\delta$ because $\tilde p$ is a probability vector. The Jacobian of the softmax is $\mathrm{diag}(p) - pp^\top$, and $\|(\mathrm{diag}(p) - pp^\top)\Delta\|_1 \le 2\|\Delta\|_\infty$, so integrating along the segment from $\tilde s$ to s gives $\|p - \tilde p\|_1 \le 2\max_\ell |s_\ell - \tilde s_\ell|$. Since $|s_\ell - \tilde s_\ell| \le \|W_Q\|\|W_K\|\big(\|q-\tilde q\|\|a_\ell\| + \|\tilde q\|\|a_\ell - \tilde a_\ell\|\big) \le 2R\|W_Q\|\|W_K\|\delta$, the first term is at most $4R^2\|W_Q\|\|W_K\|\|W_V\|\delta$. The argument for integrals is identical. □

Proposition B.16 (Consistency of attention EBOs)**.** *Let Ψ_θ be an attention EBO with bounded open domains Ω_X, Ω_Y, normalized Lebesgue measures λ_X, λ_Y, continuous lifting maps, a Lipschitz activation σ with constant L_σ and a Lipschitz energy head Q with constant L_Q, and let Assumption B.1 hold. Let $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$ be compact, and let $\{x_i\}$ and $\{y_j\}$ be i.i.d. uniform samples of $\overline{\Omega_X}$ and $\overline{\Omega_Y}$. Then*

$$\lim_{M_X, M_Y \to \infty} \sup_{(f,u)\in\mathcal{K}} \mathbb{E}\left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| = 0.$$

Proof. Compactness. By Assumption B.1, $f \mapsto h_f$ and $u \mapsto h_u$ are continuous into $C(\overline{\Omega_X})$ and $C(\overline{\Omega_Y})$, because continuous lifting maps are uniformly continuous on bounded sets. By the integral form of Lemma B.15, both attention operators are Lipschitz on bounded subsets of C, so every layer is continuous. Hence the sets $\{(v^{(t-1)}, h_f) : (f,u) \in \mathcal{K}\} \subset C(\overline{\Omega_Y}) \times C(\overline{\Omega_X})$ and $\{v^{(t-1)} : (f,u) \in \mathcal{K}\} \subset C(\overline{\Omega_Y})$ are compact, and all continuum latent values are bounded.

One layer. Write $\mathcal{A}^M$ for an attention operator with its integrals replaced by sums over the samples and applied to the exact functions $v^{(t-1)}$ and h_f. Theorems 6 and 12 of [28], applied to these compact sets, give

$$\varepsilon_t := \sup_{(f,u)\in\mathcal{K}} \mathbb{E}\Big(\big\|\mathcal{A}_{YX}^{(t),M_X}(v^{(t-1)}, h_f) - \mathcal{A}_{YX}^{(t)}(v^{(t-1)}, h_f)\big\|_C + \big\|\mathcal{A}_{YY}^{(t),M_Y}(v^{(t-1)}) - \mathcal{A}_{YY}^{(t)}(v^{(t-1)})\big\|_C \Big) \longrightarrow 0.$$

Propagation. Let $e_t = \max_j \|v_j^{(t)} - v^{(t)}(y_j)\|$, so $e_0 = 0$. Each discrete attention output is a convex combination of the vectors $W_V v_\ell^{(t-1)}$ or $W_V h_f(x_i)$, so the discrete latent values are bounded by a constant R that does not depend on the samples; we take R to bound the continuum values as well. At y_j, the discrete layer differs from the continuum layer by two terms: the discrete attention at the discrete features minus the discrete attention at the exact features, which Lemma B.15 bounds by a constant times e_{t-1}, and the error of the discrete attention at the exact features, which is at most the C-norm error in ε_t. Hence $e_t \le L_\sigma\big(\|W^{(t)}\| + L_t\big)e_{t-1} + L_\sigma\,\eta_t$, where L_t collects the constants of Lemma B.15 for the two operators and η_t is the sum of the two C-norm errors. Taking expectations and the supremum over $\mathcal{K}$, induction gives $\sup_{\mathcal{K}} \mathbb{E}\, e_T \to 0$.

Pooling. Finally,

$$\left|\Psi_\theta^{M_X,M_Y}(f,u) - \Psi_\theta(f,u)\right| \le L_Q\, e_T + \left|\frac{1}{M_Y}\sum_{j=1}^{M_Y} Q\big(v^{(T)}(y_j)\big) - \int_{\Omega_Y} Q\big(v^{(T)}(y)\big)\, d\lambda_Y(y)\right|,$$

and the expectation of the last term is at most $\sup_{\mathcal{K}} \|Q \circ v^{(T)}\|_\infty / \sqrt{M_Y}$, uniformly over $\mathcal{K}$. □

B.4 Stability of GRF-Preconditioned Energy Minimization

We analyze the deterministic inference procedure used by EBO after the GRF initialization. For a fixed context function $f \in \mathcal{H}_X$, write

$$\Psi(u) \equiv \Psi_\theta(f,u).$$

Let $\mathcal{H}_Y$ be a Hilbert space equipped with inner product $\langle\cdot,\cdot\rangle$. Let $C : \mathcal{H}_Y \to \mathcal{H}_Y$ denote the covariance operator of the Gaussian random field reference. We assume that C is self-adjoint, positive semidefinite, and bounded; for a trace-class covariance operator this boundedness holds automatically. In the discretized implementation, C is represented by the GRF covariance matrix C_n.

The deterministic GRF-preconditioned energy-minimization flow is

$$\frac{du_t}{dt} = -C\nabla\Psi(u_t), \tag{49}$$

where $\nabla\Psi$ denotes the Riesz representative of the Fréchet derivative of Ψ with respect to the ambient Hilbert-space inner product.

Proposition B.17 (Energy descent under GRF-preconditioned flow)**.** *Assume $\Psi : \mathcal{H}_Y \to \mathbb{R}$ is Fréchet differentiable and $C : \mathcal{H}_Y \to \mathcal{H}_Y$ is self-adjoint positive semidefinite. Then along the flow* (49)*, the energy is non-increasing:*

$$\frac{d}{dt}\Psi(u_t) = -\left\langle \nabla\Psi(u_t), C\nabla\Psi(u_t)\right\rangle = -\left\|C^{1/2}\nabla\Psi(u_t)\right\|^2 \le 0. \tag{50}$$

If C is injective on the subspace containing $\nabla\Psi(u_t)$, then every stationary point of the preconditioned flow is also a stationary point of the energy functional.

Proof. By the chain rule for Fréchet differentiable functionals,

$$\frac{d}{dt}\Psi(u_t) = \left\langle \nabla\Psi(u_t), \frac{du_t}{dt}\right\rangle.$$

Substituting (49) gives

$$\frac{d}{dt}\Psi(u_t) = -\left\langle \nabla\Psi(u_t), C\nabla\Psi(u_t)\right\rangle.$$

Since C is self-adjoint and positive semidefinite, it admits a positive semidefinite square root $C^{1/2}$, and therefore

$$\left\langle \nabla\Psi(u_t), C\nabla\Psi(u_t)\right\rangle = \left\|C^{1/2}\nabla\Psi(u_t)\right\|^2 \ge 0.$$

Hence $\frac{d}{dt}\Psi(u_t) \le 0$. If u_t is stationary for the preconditioned flow, then $C\nabla\Psi(u_t) = 0$. If C is injective on the subspace containing $\nabla\Psi(u_t)$, this implies $\nabla\Psi(u_t) = 0$. □

We next state the corresponding finite-dimensional descent guarantee for the discretized update used in Algorithm 1. After flattening all spatial and channel dimensions, write the update as

$$u^{(k+1)} = u^{(k)} - hC_n\nabla\Psi(u^{(k)}), \tag{51}$$

where Ψ now denotes the discretized scalar energy.

Proposition B.18 (Discrete energy descent)**.** *Assume $\Psi : \mathbb{R}^n \to \mathbb{R}$ has L-Lipschitz gradient with respect to the Euclidean norm, and let $C_n \in \mathbb{R}^{n\times n}$ be symmetric positive semidefinite. If $C_n \neq 0$ and*

$$0 < h < \frac{2}{L\|C_n\|_{\mathrm{op}}}, \tag{52}$$

then the update (51) *satisfies*

$$\Psi(u^{(k+1)}) \le \Psi(u^{(k)}) - h\left(1 - \frac{hL\|C_n\|_{\mathrm{op}}}{2}\right)\left\|C_n^{1/2}\nabla\Psi(u^{(k)})\right\|^2. \tag{53}$$

In particular, $\Psi(u^{(k+1)}) \le \Psi(u^{(k)})$ for any step size satisfying (52)*. If $C_n = 0$, the update is the identity map and the energy is unchanged.*

Proof. By L-smoothness of Ψ,

$$\Psi(u^{(k+1)}) \le \Psi(u^{(k)}) + \left\langle \nabla\Psi(u^{(k)}), u^{(k+1)} - u^{(k)}\right\rangle + \frac{L}{2}\left\|u^{(k+1)} - u^{(k)}\right\|^2.$$

Using $u^{(k+1)} - u^{(k)} = -hC_n\nabla\Psi(u^{(k)})$ gives

$$\Psi(u^{(k+1)}) \le \Psi(u^{(k)}) - h\left\langle \nabla\Psi(u^{(k)}), C_n\nabla\Psi(u^{(k)})\right\rangle + \frac{Lh^2}{2}\left\|C_n\nabla\Psi(u^{(k)})\right\|^2.$$

Since C_n is symmetric positive semidefinite,

$$\left\langle \nabla\Psi(u^{(k)}), C_n\nabla\Psi(u^{(k)})\right\rangle = \left\|C_n^{1/2}\nabla\Psi(u^{(k)})\right\|^2.$$

Moreover, for a symmetric positive semidefinite matrix, $C_n^2 \preceq \|C_n\|_{\mathrm{op}}C_n$, so

$$\left\|C_n\nabla\Psi(u^{(k)})\right\|^2 = \left\langle \nabla\Psi(u^{(k)}), C_n^2\nabla\Psi(u^{(k)})\right\rangle \le \|C_n\|_{\mathrm{op}}\left\langle \nabla\Psi(u^{(k)}), C_n\nabla\Psi(u^{(k)})\right\rangle.$$

Combining the above inequalities yields

$$\Psi(u^{(k+1)}) \le \Psi(u^{(k)}) - h\left(1 - \frac{hL\|C_n\|_{\mathrm{op}}}{2}\right)\left\|C_n^{1/2}\nabla\Psi(u^{(k)})\right\|^2.$$

The coefficient is positive whenever $0 < h < 2/(L\|C_n\|_{\mathrm{op}})$. □

Corollary B.19 (Vanishing preconditioned gradient along stable EM)**.** *Under the assumptions of Proposition B.18, suppose in addition that Ψ is bounded below and that h satisfies* (52)*. Then*

$$\sum_{k=0}^{\infty} \left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\|^2 < \infty, \tag{54}$$

and hence

$$\left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\| \longrightarrow 0. \tag{55}$$

If C_n is positive definite with smallest eigenvalue $\lambda_{\min}(C_n) > 0$, then $\|\nabla \Psi(u^{(k)})\| \to 0$ as well.

Proof. Let

$$\alpha = h \left(1 - \frac{hL \|C_n\|_{\mathrm{op}}}{2} \right) > 0.$$

By Proposition B.18,

$$\Psi(u^{(k+1)}) \leq \Psi(u^{(k)}) - \alpha \left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\|^2.$$

Summing from $k = 0$ to $K - 1$ gives

$$\alpha \sum_{k=0}^{K-1} \left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\|^2 \leq \Psi(u^{(0)}) - \Psi(u^{(K)}).$$

Since Ψ is bounded below, the right-hand side is bounded uniformly in K. Taking $K \to \infty$ proves summability. A nonnegative summable sequence must converge to zero, yielding

$$\left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\| \to 0.$$

If C_n is positive definite, then

$$\left\| C_n^{1/2} \nabla \Psi(u^{(k)}) \right\|^2 \geq \lambda_{\min}(C_n) \left\| \nabla \Psi(u^{(k)}) \right\|^2,$$

so $\|\nabla \Psi(u^{(k)})\| \to 0$. □

Remark B.20 (Interpretation of the GRF preconditioner). The covariance preconditioner does not merely rescale the gradient numerically. It defines the geometry of the refinement dynamics. The direction $-C\nabla\Psi$ is the covariance-preconditioned descent direction induced by the Gaussian reference. Formally, on a finite-dimensional discretization, or on a subspace where C^{-1} is well-defined, this direction is the steepest descent direction under the covariance-weighted inner product

$$\langle v, w \rangle_{C^{-1}} = \langle C^{-1} v, w \rangle.$$

Equivalently, high-frequency directions corresponding to small eigenvalues of C are damped by the update. This prevents the refinement dynamics from reducing to grid-wise Euclidean descent and keeps the inference procedure compatible with the function-space geometry induced by the GRF reference.

Remark B.21 (Relation to stochastic Langevin sampling). The deterministic update analyzed above should be interpreted as GRF-preconditioned energy minimization, not exact sampling from the Gaussian-reference Gibbs measure

$$\nu_\theta(du \mid f) \propto \exp\{-\Psi_\theta(f, u)\}\, \mu_0(du).$$

To obtain a Langevin-style sampler for this measure, one must include both the Gaussian-reference drift and matched GRF noise, yielding the continuous-time preconditioned Langevin dynamics

$$du_t = (-C \nabla_u \Psi_\theta(f, u_t) - u_t)\, dt + \sqrt{2}\, dW_t^C,$$

where W_t^C is a C-Wiener process. The corresponding Euler–Maruyama update is

$$u^{(k+1)} = u^{(k)} + h \left(S_\theta(f; u^{(k)}) - u^{(k)} \right) + \sqrt{2h}\, \xi_k, \qquad \xi_k \sim \mathcal{N}(0, C_n).$$

Thus, adding noise alone to the deterministic EM update is not sufficient to recover Langevin sampling for the Gaussian-reference Gibbs measure; the reference drift $-u^{(k)}$ is also required. In the main experiments, we use the deterministic update in Algorithm 1 for stable and scalable energy refinement from a GRF initialization.

Remark B.22 (Scope of the descent guarantee). Propositions B.17 and B.18 prove monotonic descent of the learned scalar energy under the deterministic GRF-preconditioned update. They do not assert global convergence to a unique minimizer, which would require additional assumptions such as convexity or a suitable gradient-dominance condition. For nonconvex neural energies, the guarantee is stability of the refinement dynamics and convergence of the preconditioned gradient norm under the conditions of Corollary B.19.

B.5 Universal Approximation

We now establish that the class of EBOs is expressive enough to approximate any jointly continuous energy.

Definition B.23 (Jointly Continuous Energy)**.** An operator $\Psi^\dagger : \mathcal{H}_X \to C(\mathcal{H}_Y;\mathbb{R})$ is **jointly continuous** if $(f,u) \mapsto \Psi^\dagger(f,u) \triangleq \Psi^\dagger(f)(u)$ is continuous with respect to the product topology on $\mathcal{H}_X \times \mathcal{H}_Y$.

Assumption B.24 (Neural Network Approximation)**.** We assume that:

1. The lifting maps P_X, P_Y can be chosen from a class of neural networks that is dense in continuous functions on compact sets.
2. The kernels $\kappa_{YX}^{(t)}$ and $\kappa_{YY}^{(t)}$ admit C^p neural parameterizations dense in the continuous matrix-valued functions on $\overline{\Omega_Y} \times \overline{\Omega_X}$ and $\overline{\Omega_Y} \times \overline{\Omega_Y}$, respectively.
3. The energy head Q can be chosen from a class dense in $C(\mathbb{R}^{d_h};\mathbb{R})$ on compact sets.
4. The architecture class allows the componentwise SiLU activation $\sigma(x) = x/(1+e^{-x})$ and affine lifting maps with zero padding, as used in the construction below.

The density requirements follow from classical neural network universal approximation [38, 39]; smooth non-polynomial activations such as tanh provide C^p kernel parameterizations, while the latent aggregation layer below uses SiLU, the activation of our experiments. Energy heads built from SiLU networks are dense in $C(\mathbb{R}^{d_h};\mathbb{R})$ on compact sets because SiLU is continuous and not a polynomial [40].

Lemma B.25 (Aggregation with Input and Output Kernels)**.** *Let $\eta_i \in C^p(\overline{\Omega_X})$ for $i = 1,\dots,N_X$ and $\xi_j \in C^p(\overline{\Omega_Y})$ for $j = 1,\dots,N_Y$. Define the moment vector*

$$\begin{aligned} s(f,u) &= (a_1(f),\dots,a_{N_X}(f), b_1(u),\dots,b_{N_Y}(u)) \in \mathbb{R}^D, \\ a_i(f) &= \int_{\Omega_X} \eta_i(x) f(x)\, d\lambda_X(x), \qquad b_j(u) = \int_{\Omega_Y} \xi_j(z) u(z)\, d\lambda_Y(z), \end{aligned} \tag{56}$$

where $D = N_X d + N_Y m$. With affine lifting maps and continuous kernels, one layer of (15) *of width $d_h \geq \max\{2D, d, m\}$ can produce*

$$v^{(1)}(y) = \big(\sigma(s(f,u)), \sigma(-s(f,u)), 0\big), \qquad y \in \overline{\Omega_Y}. \tag{57}$$

Here σ is SiLU, applied componentwise, and the final zero block pads the width. A fixed linear map $\Pi : \mathbb{R}^{d_h} \to \mathbb{R}^D$ recovers $s(f,u) = \Pi v^{(1)}(y)$.

Proof. Lift f and u into the first d and m channels of h_f and h_u, respectively, and set all remaining channels to zero. Choose kernels independent of the receiving location y so that

$$\begin{aligned} \kappa_{YX}^{(1)}(y,x) h_f(x) &= (a(x), -a(x), 0), \qquad & a(x) &= (\eta_1(x)f(x),\dots,\eta_{N_X}(x)f(x),0), \\ \kappa_{YY}^{(1)}(y,z) h_u(z) &= (b(z), -b(z), 0), \qquad & b(z) &= (0, \xi_1(z)u(z),\dots,\xi_{N_Y}(z)u(z)). \end{aligned}$$

The two D-dimensional blocks $a(x), b(z)$ occupy complementary feature channels. These identities specify matrix-valued kernels using the scalar weights η_i, ξ_j; the kernels do not depend on f or u. Set $W^{(1)} = 0$, $b^{(1)} = 0$, and $\sigma = \mathrm{SiLU}$. The preactivation is $(s, -s, 0)$, and since $\sigma(0) = 0$, SiLU gives the stated constant latent function. For every $x \in \mathbb{R}$,

$$\sigma(x) - \sigma(-x) = \frac{x}{1+e^{-x}} + \frac{x}{1+e^{x}} = x,$$

so subtracting the second D channels from the first D channels recovers s and defines Π. □

Theorem B.26 (Universal Approximation for EBO)**.** *Under Assumptions B.1 and B.24, let* $\Psi^\dagger : \mathcal{H}_X \to C(\mathcal{H}_Y;\mathbb{R})$ *be jointly continuous (Definition B.23). For every compact* $\mathcal{K} \subset \mathcal{H}_X \times \mathcal{H}_Y$ *and* $\varepsilon > 0$*, there is an EBO* Ψ_θ *as in Definition B.5, with sufficiently large latent width and neural parameterizations, such that*

$$\sup_{(f,u)\in\mathcal{K}} |\Psi_\theta(f,u) - \Psi^\dagger(f,u)| < \varepsilon. \tag{58}$$

Proof. **Step 1: Compact-set reduction.** The continuous embeddings in Assumption B.1 send $\mathcal{K}$ injectively to a compact set $\tilde{\mathcal{K}} \subset C(\overline{\Omega_X};\mathbb{R}^d) \times C(\overline{\Omega_Y};\mathbb{R}^m)$. A continuous bijection from a compact space to a Hausdorff space is a homeomorphism onto its image. Consequently, $\Psi^\dagger$ induces a continuous function on $\tilde{\mathcal{K}}$; no extension from the entire Hilbert space to the sup-norm space is needed.

Coordinate evaluations of f and u separate points of $\tilde{\mathcal{K}}$. The algebra of polynomials in finitely many such evaluations contains the constants, so the Stone–Weierstrass theorem gives points $x_1,\dots,x_{N_X} \in \overline{\Omega_X}$, $y_1,\dots,y_{N_Y} \in \overline{\Omega_Y}$ and a polynomial $\rho : \mathbb{R}^D \to \mathbb{R}$, with $D = N_X d + N_Y m$, such that

$$\sup_{(f,u)\in\tilde{\mathcal{K}}} |\Psi^\dagger(f,u) - \rho(f(x_1),\dots,f(x_{N_X}),u(y_1),\dots,u(y_{N_Y}))| < \varepsilon/3. \tag{59}$$

The two projections of $\tilde{\mathcal{K}}$ are uniformly bounded and equicontinuous. Choose smooth, nonnegative, normalized weights η_i, ξ_j supported near the respective evaluation points. Their localized averages $a_i(f), b_j(u)$ approximate those evaluations uniformly. All evaluations and averages lie in a fixed compact box, so uniform continuity of ρ on that box gives weights for which

$$\sup_{(f,u)\in\tilde{\mathcal{K}}} |\Psi^\dagger(f,u) - \rho(s(f,u))| < 2\varepsilon/3. \tag{60}$$

Step 2: Realize both sets of moments in the latent function. Lemma B.25 constructs an ideal latent function $v^\dagger(y) = (\sigma(s), \sigma(-s), 0)$ using κ_{YX} for the input moments and κ_{YY} for the output moments. Define the continuous ideal energy head

$$Q^\dagger(\zeta) = \frac{\rho(\Pi\zeta)}{\lambda_Y(\Omega_Y)}, \qquad \zeta \in \mathbb{R}^{d_h}.$$

Then $\int_{\Omega_Y} Q^\dagger(v^\dagger(y))\, d\lambda_Y(y) = \rho(s(f,u))$.

Step 3: Neural parameterization and error control. The ideal latent functions have a common bounded range on $\tilde{\mathcal{K}}$. Choose a compact box containing that range with a positive margin. Assumption B.24 gives a neural energy head Q_θ satisfying

$$\sup_\zeta |Q_\theta(\zeta) - Q^\dagger(\zeta)| < \frac{\varepsilon}{6\lambda_Y(\Omega_Y)}$$

on this box. By uniform continuity of $Q^\dagger$, choose $\delta > 0$ small enough that latent perturbations below δ stay in the box and change $Q^\dagger$ by less than $\varepsilon/(6\lambda_Y(\Omega_Y))$.

Approximate each of the two ideal kernels uniformly by its neural parameterization. The affine lifts can be represented exactly. Boundedness of h_f, h_u on $\tilde{\mathcal{K}}$ and the Lipschitz activation give

$$\begin{aligned}\|v_\theta^{(1)} - v^\dagger\|_\infty \le L_\sigma\Big(&\lambda_X(\Omega_X)\|\kappa_{YX,\theta} - \kappa^\dagger_{YX}\|_\infty \|h_f\|_\infty \\ &+ \lambda_Y(\Omega_Y)\|\kappa_{YY,\theta} - \kappa^\dagger_{YY}\|_\infty \|h_u\|_\infty\Big).\end{aligned} \tag{61}$$

The kernel approximation errors can therefore be chosen to make this bound less than δ uniformly. Integration then yields

$$\sup_{(f,u)\in\tilde{\mathcal{K}}} |\Psi_\theta(f,u) - \rho(s(f,u))| < \varepsilon/3.$$

Combining with (60) proves the claim. □

Remark B.27 (Role of the self kernel). The YY kernel gathers information from the entire candidate target function before the nonlinear scalar readout. With $\kappa_{YY} = 0$, each latent value $v^{(T)}(y)$ depends only on f and $u(y)$, so the per-point energy at y sees u only through $u(y)$. The two-kernel construction can instead express nonlinear interactions between global output features, such as $\left(\int_{\Omega_Y} u(y)\, d\lambda_Y(y)\right)^2$ for scalar u.

Remark B.28 (Smoothness of the approximating EBO). Take the energy head Q_θ in Step 3 to be a SiLU network, which Assumption B.24 allows. Every map in the construction is then smooth in u: the lifting is affine, the kernel integrals are linear in u, SiLU is C^∞, and Q_θ is C^∞. Since $\mathcal{H}_Y$ embeds continuously in $C(\overline{\Omega_Y}; \mathbb{R}^m)$, the energy $u \mapsto \Psi_\theta(f, u)$ is Fréchet differentiable on $\mathcal{H}_Y$. The EBO of Theorem B.26 therefore satisfies Assumption 2.2, and its score $-C\nabla_u \Psi_\theta(f, u)$ is well defined.

Corollary B.29 (Convergence of induced measures)**.** *Assume the conditions of Theorem B.26, fix $f \in \mathcal{H}_X$, and let μ_0 be a Gaussian measure on $\mathcal{H}_Y$. Suppose that $Z^\dagger(f) = \int_{\mathcal{H}_Y} \exp(-\Psi^\dagger(f, u))\, d\mu_0(u) < \infty$, so that*

$$\frac{d\nu^\dagger}{d\mu_0}(u \mid f) = \frac{\exp(-\Psi^\dagger(f, u))}{Z^\dagger(f)} \tag{62}$$

defines a probability measure. Then:

1. *There are EBOs Ψ_{θ_n}, $n \in \mathbb{N}$, such that $\Psi_{\theta_n}(f, u) \to \Psi^\dagger(f, u)$ for μ_0-almost every u.*
2. *If, in addition, there is a μ_0-integrable $G : \mathcal{H}_Y \to [0, \infty)$ with $\exp(-\Psi_{\theta_n}(f, u)) \le G(u)$ for all n and μ_0-almost every u, then $\nu_{\theta_n}(\cdot \mid f)$ converges to $\nu^\dagger(\cdot \mid f)$ in total variation, and hence weakly.*

Proof. **Step 1: Almost-everywhere convergence.** $\mathcal{H}_Y$ is a separable Hilbert space, hence a Polish space, so the Borel probability measure μ_0 is tight: there are compact sets $L_n \subset \mathcal{H}_Y$ with $\mu_0(L_n) \ge 1 - 1/n$. Replacing L_n by $L_1 \cup \cdots \cup L_n$, we may take them increasing. Each $\{f\} \times L_n$ is compact in $\mathcal{H}_X \times \mathcal{H}_Y$, so Theorem B.26 gives θ_n with

$$\sup_{u \in L_n} \left|\Psi_{\theta_n}(f, u) - \Psi^\dagger(f, u)\right| < \frac{1}{n}. \tag{63}$$

If $u \in L_m$, then $u \in L_n$ for every $n \ge m$, so $|\Psi_{\theta_n}(f, u) - \Psi^\dagger(f, u)| < 1/n$ for all $n \ge m$ and the energies converge at u. Since $\mu_0\big(\bigcup_m L_m\big) = 1$, the convergence holds μ_0-almost everywhere.

Step 2: Normalizing constants. Write $p_n = \exp(-\Psi_{\theta_n}(f, \cdot))$ and $p^\dagger = \exp(-\Psi^\dagger(f, \cdot))$. By Step 1, $p_n \to p^\dagger$ μ_0-almost everywhere, and $p_n \le G$. Dominated convergence gives

$$Z_{\theta_n}(f) = \int_{\mathcal{H}_Y} p_n\, d\mu_0 \;\to\; \int_{\mathcal{H}_Y} p^\dagger\, d\mu_0 = Z^\dagger(f). \tag{64}$$

Moreover $Z^\dagger(f) > 0$ because $p^\dagger > 0$ and μ_0 is a probability measure, so $Z_{\theta_n}(f) \ge Z^\dagger(f)/2$ for all large n.

Step 3: Total variation. The densities $q_n = p_n / Z_{\theta_n}(f)$ of $\nu_{\theta_n}(\cdot \mid f)$ and $q^\dagger = p^\dagger / Z^\dagger(f)$ of $\nu^\dagger(\cdot \mid f)$ with respect to μ_0 satisfy $q_n \to q^\dagger$ μ_0-almost everywhere and, for all large n, $|q_n - q^\dagger| \le 2G/Z^\dagger(f) + q^\dagger$, which is μ_0-integrable. Dominated convergence gives $\int_{\mathcal{H}_Y} |q_n - q^\dagger|\, d\mu_0 \to 0$. For every bounded measurable $g : \mathcal{H}_Y \to \mathbb{R}$,

$$\left| \int_{\mathcal{H}_Y} g\, d\nu_{\theta_n}(\cdot \mid f) - \int_{\mathcal{H}_Y} g\, d\nu^\dagger(\cdot \mid f) \right| \le \|g\|_\infty \int_{\mathcal{H}_Y} \left|q_n - q^\dagger\right| d\mu_0 \;\to\; 0. \tag{65}$$

This is convergence in total variation. Taking g bounded and continuous gives weak convergence. □

C Synthetic function experiments

In this section, we provide a detailed summary of 1D function classes defined for EBO as well as training details.

C.1 Function Class

We evaluate EBO on three function classes as defined in the following paragraphs.

Cosine Functions. We begin with a synthetic benchmark to validate our approach under controlled conditions and perform ablation studies. We generate functions of the form $f(x) = A\cos(2\pi\omega x+\phi)$, where amplitude $A \sim \mathcal{U}(0.5, 1.5)$, frequency $\omega \sim \mathcal{U}(4, 6)$, and phase $\phi \sim \mathcal{U}(0, 2\pi)$. Observations are sampled at irregular locations $\{x_i\}_{i=1}^N$ drawn uniformly from $[0, 1]$ and sorted. We initialize the initial function with Gaussian random field samples from a GP prior with a Matérn ν=1.5 kernel (Table A4).

Damping. We consider $y(x) = e^{-\alpha x}\sin(2\pi f x)$ with decay rate $\alpha \sim \mathcal{U}(3, 6)$ and frequency $f \sim \mathcal{U}(8, 16)$. The exponential envelope breaks translation invariance, coupling local oscillatory structure to global amplitude decay. Such non-stationary waveforms arise naturally in RLC circuits, seismology, and underdamped mechanical systems, where energy dissipation manifests as exponential attenuation of periodic motion.

Izhikevich neuron. We sample membrane potential trajectories from the Izhikevich spiking neuron model [41], a two-dimensional dynamical system that balances biological plausibility with computational efficiency:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \quad \frac{du}{dt} = a(bv - u), \tag{66}$$

with reset $v \leftarrow c$, $u \leftarrow u + d$ when $v \geq 30$ mV. Here v is the membrane potential, u is a recovery variable modeling slow ionic currents, and I is the injected current. The four parameters (a, b, c, d) determine the firing pattern: we sample regular spiking (RS), intrinsically bursting (IB), and fast spiking (FS) neuron types with $I \sim \mathcal{U}(5, 15)$. Unlike the smooth function classes above, the resulting spike trains exhibit discontinuous resets, testing the model's capacity to learn distributions over non-smooth trajectories prevalent in computational neuroscience and brain-computer interfaces.

C.2 Training Modes

We compare three training paradigms: direct prediction (Operator), energy-based minimization (EBO) and denoising diffusion (DDO). All three use the same backbone, coordinate encoding and context conditioning, and differ only in how the target function is produced (Table A2). The operator predicts it in one forward pass from a zero-valued input. EBO starts from a GRF sample and takes K energy-minimization steps. DDO is trained to remove GRF noise added to the target at a random noise level, which the network receives through a timestep embedding, and generates a target by running the reverse diffusion from a GRF sample. Table A1 summarizes the training configuration for different backbones.

C.3 Training Configuration

Table A2 summarizes the training configuration for EBO on the various functions datasets, and Table A3 summarizes the configuration of the shared optimizer and scheduler.

Multi-Resolution Training. Models are trained on variable resolution grids, with N drawn from the resolution list of Table A3, with irregular sampling locations. This approach improves generalization across different discretization scales and prevents the model from overfitting to specific grid structures.

Context Masking. We sample the context ratio uniformly $r \sim \mathcal{U}[0.1, 0.9]$ and employ mixed masking modes (random and block) to expose the model to diverse conditioning scenarios during training.

Table A1: **Backbone-specific hyperparameters and parameter counts.** All architectures use the layer and head counts below with the same coord encoding / token-concat conditioning.

Parameter	Transformer	FactFormer	Galerkin
Hidden dim	128	128	128
# Layers	4	4	4
# Heads	4	4	4
Attention type	Vanilla MHA	Factorized	Galerkin (linear)
Special params	—	geotype=structured_1D	—
Backbone params	476,544	370,049	615,169
Pred. head params	16,641	16,641	16,641
Embedding/coord	66,561	66,561	66,561
Total params	**559,746**	**453,251**	**698,371**

Table A2: **Training mode hyperparameters (Operator/ EBO / DDO).**

Parameter	Operator	EBO	DDO
Prediction method	One-shot MLP head	iterative EM	Reverse diffusion
Prediction head	2-layer MLP	—	—
Pred. head LR multiplier	$10\times$	—	—
Input initialization	Zero-valued	GRF samples from GP prior	GRF noise
Number of steps (training)	—	$K \sim \mathcal{U}[8, 64]$	—
Step size h	—	0.01 (fixed, non-learnable)	—
Time conditioning	none	none	timestep embedding

C.4 Gaussian Process Kernel Selection

We employ function-specific GP kernels for function initialization, selected empirically across several training setups as shown in Table A4.

The Matérn kernel smoothness parameter ν controls differentiability: $\nu = 0.5$ yields rough samples suitable for high-frequency components, $\nu = 1.5$ produces once-differentiable samples, and $\nu = 2.5$ generates smooth, twice-differentiable samples.

C.5 Pseudo-algorithms for Inference and Training

Algorithm 1 predicts a target function by iterative energy minimization (Section 2.3), and Algorithm 2 trains the EBO through that prediction (Section 2.4). The mask M equals 1 at observed target values u_{obs} and 0 otherwise. Here C_n is the GRF covariance matrix on the grid. In the PDE experiments, the initial sample and the preconditioner use different covariance matrices (Appendix D).

Algorithm 1 Inference

Require: EBO Ψ_θ, context f, steps K, step size h, covariance C_n, observed values u_{obs}, mask M

Sample $u^{(0)} \sim \mathcal{N}(0, C_n)$
$u^{(0)} \leftarrow M \odot u_{\text{obs}} + (1 - M) \odot u^{(0)}$
for $k = 0$ **to** $K - 1$ **do**
  $S^{(k)} \leftarrow -C_n \nabla_u \Psi_\theta\big(f, u^{(k)}\big)$
  $u^{(k+1)} \leftarrow u^{(k)} + h\,(1 - M) \odot S^{(k)}$
end for
return $u^{(K)}$ and its energy $\Psi_\theta\big(f, u^{(K)}\big)$

Algorithm 2 Training step

Require: EBO Ψ_θ, data $\mathcal{D}$, task loss $\mathcal{J}$, steps K, step size h, covariance C_n

Sample $(f, u^*) \sim \mathcal{D}$
$u^{(K)} \leftarrow$ Algorithm 1 on f, differentiating only its last step
$\widehat{\mathcal{L}}(\theta) \leftarrow \mathcal{J}\big(u^{(K)}, u^*; f\big)$
Update θ using $\nabla_\theta \widehat{\mathcal{L}}(\theta)$

Table A3: **Shared optimizer and training schedule (all architectures, all training modes).**

Parameter	Value
Optimizer	AdamW
Learning rate	5×10^{-4}
Warmup steps	5,000
LR schedule	Cosine decay (min_lr_scale = 10)
Batch size	64
Resolution list	$\{16, 32, 64, 128, 256\}$ (multi-resolution)
Train iterations	50,000
Mixed precision	bfloat16 (AMP)
Coordinate encoding	SIREN ($\omega_0 = 30$, hidden_dim = 256, layers = 1)
Conditioning mode	Token concat
Seed	42

Table A4: **GP kernel configurations used per function type.**

Function Type	Kernel	ν	Length Scale
Oscillation	Matérn	1.5	0.1
Damping	Matérn	2.5	0.1
Izhikevich	Matérn	0.5	0.1

D PDE Sparse Reconstruction Experiments

This section describes the sparse reconstruction experiments on Darcy flow, Navier–Stokes and Burgers. For each PDE and method, one FNO model is trained across several resolutions and evaluated at every resolution.

D.1 Task, Data and Metric

Each model receives the values of a solution field at a random subset of grid points and reconstructs the full field. The observed mask is denoted by M, with $M_i = 1$ at observed points and $M_i = 0$ elsewhere. Training draws one context ratio per sample from $\mathrm{Unif}[0.05, 0.95]$, and evaluation uses context ratios of 5%, 10%, 20%, 40%, 60% and 80%, with the same masks for all methods. Table A5 lists the resolutions and splits.

Table A5: **Sparse PDE reconstruction datasets.** Burgers at 512 points is used only for evaluation.

Task	Resolutions	Output field	Train / val / test
Darcy flow	16^2, 32^2, 64^2, 128^2	scalar solution field	4500 / 500 / 1000
Navier–Stokes	32^2, 64^2, 128^2	scalar vorticity field	9500 / 500 / 2000
Burgers	32, 64, 128, 256, 512	$u(x, t{=}1)$	4000 / 500 / 1000

Darcy flow and Navier–Stokes use the 128×128 datasets released with the neuraloperator library: the solution field of the Darcy files and the vorticity of the forced Navier–Stokes files. All grids are $x_i = i/n$, so each coarser resolution is an exact strided subsample of the stored grid and all resolutions share one train, validation and test split. The Darcy files contain a few single-pixel negative values on the domain boundary, which are kept as published.

Burgers follows the setup of neuraloperator's 1D Burgers data: $u_t + (u^2/2)_x = \nu u_{xx}$ on the periodic domain $[0, 1]$ with $\nu = 0.01$, and the target is $u(x, t{=}1)$. Initial conditions are Gaussian random fields from the generator of the FNO code, which samples $\mathcal{N}(0, 1250(-\Delta + 25I)^{-2})$. The equation is solved pseudo-spectrally (integrating-factor RK4, 2/3 dealiasing, $\Delta t = 10^{-4}$) on 1024 points. Against the exact Cole–Hopf solution, the solver agrees to 3×10^{-13} relative L^2, and the stored fields agree to 2.7×10^{-8}, the precision of float32 storage.

Fields are standardized with the training-set mean and standard deviation. The metric is the missing-region relative L^2 error,

$$\mathrm{rel}L^2_{\mathrm{miss}} = \frac{\|(1-M)(\hat{u}-u)\|_2}{\|(1-M)u\|_2 + \varepsilon}, \tag{67}$$

computed per test field in physical units and averaged over the test set. It leaves out the observed points, which are known at test time. Tables give the mean $\pm$ the standard deviation across test fields; the standard error of the mean is smaller by a factor $\sqrt{n}$, with $n = 1000$ or 2000 test fields, which is 2–3% of the standard deviation. Displayed reconstructions insert the observed values, $\hat{u}_{\mathrm{ctx}} = Mu + (1-M)\hat{u}$.

D.2 Models and Training

All three methods share one FNO backbone: a pointwise lift, two spectral layers, RMSNorm and a pointwise projection, with coordinates as input channels (sizes in Table A6). The spectral layers keep a fixed number of Fourier modes, so the same weights apply at every resolution, and each training step draws one training resolution uniformly. With the same weights, our Fourier layer computes the same function as neuraloperator's `SpectralConv` to within 2×10^{-7}. The FFTs are periodic on every dataset; Darcy flow is not padded, so all three methods treat its boundary as periodic.

Operator. input $[Mu, M, \text{coordinates}]$ and output $\hat{u}$, trained on $\mathcal{L}_{\mathrm{op}} = \mathcal{L}_{\mathrm{miss}} + \lambda_{\mathrm{ctx}}\mathcal{L}_{\mathrm{ctx}}$ with

$$\mathcal{L}_{\mathrm{miss}} = \frac{\sum_i (1-M_i)(\hat{u}_i - u_i)^2}{\sum_i (1-M_i) + \varepsilon}, \qquad \mathcal{L}_{\mathrm{ctx}} = \frac{\sum_i M_i(\hat{u}_i - u_i)^2}{\sum_i M_i + \varepsilon}. \tag{68}$$

The context term makes the operator reproduce the observed values, which discourages discontinuities when they are inserted after prediction.

EBO. input $[u, Mu, M, \text{coordinates}]$. The network outputs a per-point energy, averaged over points into the scalar energy Ψ. Reconstruction starts from a GRF sample with the context clamped and updates the unobserved points by $u \leftarrow u - \alpha m\,(1-M) \odot P\nabla_u \Psi$, where α is a learned step size, m is an inference step multiplier, and $P = nD^{-1/2}KD^{-1/2}$ is the GRF covariance applied per axis and normalized so that a step has the same meaning at every resolution. This is the EM update (4) with step size $h = \alpha m$ and covariance matrix $C_n = P$, applied to the unobserved points only. Training draws the number of minimization steps uniformly from $\{1, \ldots, 48\}$ and differentiates only the last step. A fixed-point term (weight 1) asks that one step started at the true field does not move it, and a quarter of the training iterations start from the true field plus GRF noise ($\sigma = 0.1$). On Burgers, the preconditioner is restricted to the missing points: the gradient at context points is set to zero before P is applied, so the minimization stops only where the gradient with respect to the missing values vanishes.

DDO. input $[z_t, Mu, M, \text{coordinates}]$ and a sinusoidal embedding of the noise level t. The network is a denoising diffusion operator, a score-based diffusion model in function space [9], and predicts the noise ε in the noisy field $z_t = \alpha_t u + \sigma_t \varepsilon$, where ε is a GRF sample with an RBF kernel and (α_t, σ_t) follow the truncated variance-preserving cosine schedule (log-SNR from 10 to -10). Reconstruction starts from $z_1 \sim \mathrm{GRF}$, applies the ancestral denoising step for the chosen number of steps, and returns the posterior mean at the last step. Training draws t and minimizes the squared error of the predicted noise, averaged over grid points. Context is inserted only at the end, as for the other methods.

All models are trained for 50,000 steps with AdamW, a 1000-step linear warm-up and cosine decay to a tenth of the learning rate. For each PDE and method, we trained a few configurations and kept the one, and its checkpoint, with the lowest validation error: missing-region relative L^2 averaged over the training resolutions and 20–80% context. The operator's context weight λ_{ctx} was swept over $\{0.1, 1, 10\}$ on Burgers and fixed to 1 on Darcy flow and Navier–Stokes, and DDO's noise length scale over $\{0.05, 0.02, 0.01\}$, plus 0.005 on Burgers. EBO's sweep covered the training trajectory length, the fixed-point term and the preconditioner length scale, with 13 runs on Burgers and 2 on each of Darcy flow and Navier–Stokes. Table A6 lists the selected values and all other settings.

Table A6: **FNO hyperparameters across the three PDEs.**

Hyperparameter	Darcy flow	Navier–Stokes	Burgers
Fourier modes	6×6	6×6	16
Hidden channels	64	64	64
Spectral layers	2	2	2
Activation	SiLU	SiLU	SiLU
Output normalization	RMSNorm	RMSNorm	RMSNorm
Field channels	1	1	1
Optimizer	AdamW	AdamW	AdamW
Learning rate	10^{-3}	10^{-3}	10^{-3}
Weight decay	10^{-6}	10^{-6}	10^{-6}
Gradient clipping	1.0	1.0	1.0
Batch size	16	16	32
Training steps (Op., EBO, DDO)	50,000	50,000	50,000
Op. context-loss weight λ_{ctx}	1	1	10
EBO training minimization steps	1–48	1–48	1–48
EBO step size α	learned; initialized at 1.0, $10\times$ learning rate		
EBO preconditioner length scale	0.03	0.03	0.005
EBO preconditioner restricted to missing points	no	no	yes
EBO fixed-point term weight	1	1	1
EBO starts near the true field (share, σ)	0.25, 0.1	0.25, 0.1	0.25, 0.1
EBO initialization GRF	RBF kernel, length scale 0.1, $\sigma = 1$		
DDO noise length scale	0.02	0.01	0.02

Table A7: **Inference settings for the PDE experiments**, chosen on validation.

Task	EBO steps	EBO step multiplier	EBO schedule	DDO denoising steps
Darcy flow	96	0.5	constant	32
Burgers	128	1.0	constant	64
Navier–Stokes	96	0.25	constant	32

D.3 Inference

EBO's minimization is deterministic (no Langevin noise) and starts from one GRF sample. Each method uses one inference setting per dataset at every resolution and context ratio (Table A7), chosen by the geometric-mean validation error over the training resolutions and the six context ratios. The EBO candidates are step counts $\{16, 32, 64, 96, 128\}$, step multipliers $\{0.1, 0.25, 0.5, 0.75, 1.0, 2.0\}$ and a constant or decaying schedule (multiplier $m(1 + k/8)^{-1/2}$ at step k), 60 settings in total; the DDO candidates are $\{16, 32, 64, 128, 250\}$ denoising steps. Burgers at 512 points is not used for this choice, so it stays zero-shot for both the weights and the inference setting.

D.4 Results Across Resolutions

Tables A8, A9 and A10 report every evaluation resolution. At resolution 128, EBO has the lowest error at every context ratio and lowers the operator's error by 25–51% on Darcy flow, 46–56% on Navier–Stokes and 49–64% on Burgers. The smallest Darcy flow gain is at 80% context, where the missing region is small. On Navier–Stokes the gain is smaller at 32^2.

Over all 72 resolution and context cells, EBO has the lowest mean error in 68. We compare methods by a paired 95% confidence interval on the mean difference over test fields. EBO is significantly better than the operator in 69 cells (23 of 24 on Darcy flow, 17 of 18 on Navier–Stokes and 29 of 30 on Burgers) and never significantly worse. The other three cells are ties: Navier–Stokes at 32^2 and 5% context (0.4388 against 0.4367; EBO is lower on 53% of the fields), Darcy flow at 64^2 and 80% context (0.0216 against 0.0215), and Burgers at 512 points and 60% context. EBO is significantly better than DDO in 69 cells (23 of 24, all 18 and 28 of 30). In the other three, DDO has the lowest error of the three methods, each significantly: Darcy flow at 64^2 and 80% context (0.0212), and Burgers at 512 points with 60% and 80% context (0.0044 against 0.0053, and 0.0045 against 0.0048).

Table A8: **Darcy flow: sparse reconstruction at every evaluation resolution.** Missing-region relative L^2 error, mean $\pm$ standard deviation across test fields. Bold marks the lowest unrounded mean per resolution and context.

Resolution	Context	Op. ↓	EBO ↓	DDO ↓
16^2	5%	$0.275_{\pm 0.071}$	$\mathbf{0.261}_{\pm \mathbf{0.094}}$	$0.407_{\pm 0.102}$
	10%	$0.190_{\pm 0.057}$	$\mathbf{0.156}_{\pm \mathbf{0.069}}$	$0.280_{\pm 0.088}$
	20%	$0.119_{\pm 0.033}$	$\mathbf{0.083}_{\pm \mathbf{0.031}}$	$0.160_{\pm 0.056}$
	40%	$0.072_{\pm 0.017}$	$\mathbf{0.047}_{\pm \mathbf{0.015}}$	$0.080_{\pm 0.026}$
	60%	$0.053_{\pm 0.013}$	$\mathbf{0.036}_{\pm \mathbf{0.011}}$	$0.055_{\pm 0.016}$
	80%	$0.042_{\pm 0.013}$	$\mathbf{0.029}_{\pm \mathbf{0.011}}$	$0.043_{\pm 0.016}$
32^2	5%	$0.133_{\pm 0.037}$	$\mathbf{0.089}_{\pm \mathbf{0.034}}$	$0.195_{\pm 0.062}$
	10%	$0.080_{\pm 0.019}$	$\mathbf{0.051}_{\pm \mathbf{0.015}}$	$0.096_{\pm 0.031}$
	20%	$0.050_{\pm 0.010}$	$\mathbf{0.034}_{\pm \mathbf{0.008}}$	$0.049_{\pm 0.014}$
	40%	$0.034_{\pm 0.007}$	$\mathbf{0.026}_{\pm \mathbf{0.006}}$	$0.031_{\pm 0.007}$
	60%	$0.028_{\pm 0.006}$	$\mathbf{0.023}_{\pm \mathbf{0.005}}$	$0.026_{\pm 0.006}$
	80%	$0.025_{\pm 0.006}$	$\mathbf{0.020}_{\pm \mathbf{0.005}}$	$0.024_{\pm 0.006}$
64^2	5%	$0.067_{\pm 0.014}$	$\mathbf{0.037}_{\pm \mathbf{0.009}}$	$0.095_{\pm 0.026}$
	10%	$0.044_{\pm 0.008}$	$\mathbf{0.026}_{\pm \mathbf{0.005}}$	$0.049_{\pm 0.011}$
	20%	$0.032_{\pm 0.006}$	$\mathbf{0.020}_{\pm \mathbf{0.004}}$	$0.031_{\pm 0.006}$
	40%	$0.025_{\pm 0.005}$	$\mathbf{0.017}_{\pm \mathbf{0.004}}$	$0.024_{\pm 0.005}$
	60%	$0.022_{\pm 0.004}$	$\mathbf{0.018}_{\pm \mathbf{0.003}}$	$0.022_{\pm 0.004}$
	80%	$0.022_{\pm 0.004}$	$0.022_{\pm 0.005}$	$\mathbf{0.021}_{\pm \mathbf{0.006}}$
128^2	5%	$0.051_{\pm 0.009}$	$\mathbf{0.025}_{\pm \mathbf{0.004}}$	$0.071_{\pm 0.015}$
	10%	$0.036_{\pm 0.006}$	$\mathbf{0.019}_{\pm \mathbf{0.003}}$	$0.042_{\pm 0.008}$
	20%	$0.028_{\pm 0.005}$	$\mathbf{0.016}_{\pm \mathbf{0.003}}$	$0.029_{\pm 0.005}$
	40%	$0.024_{\pm 0.004}$	$\mathbf{0.014}_{\pm \mathbf{0.002}}$	$0.024_{\pm 0.011}$
	60%	$0.023_{\pm 0.004}$	$\mathbf{0.014}_{\pm \mathbf{0.002}}$	$0.022_{\pm 0.004}$
	80%	$0.023_{\pm 0.004}$	$\mathbf{0.017}_{\pm \mathbf{0.003}}$	$0.023_{\pm 0.009}$

DDO is above the operator at 5% and 10% context in every cell, and below it in 10 of 24 Darcy flow, 10 of 18 Navier–Stokes and 17 of 30 Burgers cells, all from 20% context up.

At 512 points, which Burgers training never sees, EBO keeps its advantage at sparse context (0.014 against 0.047 for the operator at 5%). With dense context, a minority of fields with large operator errors carries EBO's advantage in the mean: EBO is lower than the operator on only 50%, 33% and 37% of the fields at 40%, 60% and 80% context.

A Burgers reference without learning. At $t = 1$ with $\nu = 0.01$, almost every Burgers field is a single shock on a ramp of slope close to $1/t$: 5230 of the 5500 fields have exactly one shock and the others have none. Fields differ mainly in the position and height of the shock, and 699 of the 1000 test fields have a training field within 5% relative L^2. Predicting each test field by the circularly shifted training field that best fits its context points is therefore a strong reference at low context. This lookup beats the operator in 16 of 30 cells, including every cell up to 10% context. It also beats EBO at 5% context at every resolution, at 10% context up to 128 points, and at 20% context at 32 points: at 128 points and 5% context, the error is 0.031 for the best match and 0.025 for the mean of the eight best matches, against 0.051 for EBO and 0.101 for the operator. In the other cells, EBO is well below the lookup (at 128 points and 20% context, 0.007 against 0.013 for the mean of the eight best matches). At low context, the Burgers task mostly consists of locating one shock.

Appendix H shows reconstructions of single test fields at every context ratio.

D.5 Number of Steps and Inference Cost

Table A11 and Figure A6 give the error and GPU time against the number of steps K at resolution 128 and 20% context, and Figure 6a,b plots the Darcy flow part. EBO's error falls with K on all three datasets until it levels off. EBO passes the operator between 8 and 16 steps on Darcy flow (0.0226 against 0.0282 at $K = 16$, about $43\times$ the operator's GPU time; 0.0158 at $K = 128$), between 16 and 32 steps on Navier–Stokes (0.0424 against 0.0746 at $K = 32$, about $86\times$; 0.0343 at $K = 128$),

Table A9: **Navier–Stokes: sparse reconstruction at every evaluation resolution.** Missing-region relative L^2 error, mean $\pm$ standard deviation across test fields. Bold marks the lowest unrounded mean per resolution and context.

Resolution	Context	Op. ↓	EBO ↓	DDO ↓
32^2	5%	**0.437**$\pm$**0.084**	0.439$\pm$0.105	0.640$\pm$0.108
	10%	0.280$\pm$0.055	**0.230**$\pm$**0.055**	0.384$\pm$0.080
	20%	0.172$\pm$0.029	**0.124**$\pm$**0.023**	0.187$\pm$0.037
	40%	0.108$\pm$0.016	**0.085**$\pm$**0.012**	0.102$\pm$0.016
	60%	0.085$\pm$0.012	**0.071**$\pm$**0.010**	0.081$\pm$0.012
	80%	0.075$\pm$0.011	**0.060**$\pm$**0.009**	0.071$\pm$0.011
64^2	5%	0.204$\pm$0.035	**0.125**$\pm$**0.025**	0.273$\pm$0.047
	10%	0.132$\pm$0.020	**0.078**$\pm$**0.012**	0.139$\pm$0.021
	20%	0.091$\pm$0.012	**0.054**$\pm$**0.007**	0.088$\pm$0.011
	40%	0.070$\pm$0.009	**0.043**$\pm$**0.005**	0.067$\pm$0.008
	60%	0.063$\pm$0.008	**0.043**$\pm$**0.006**	0.060$\pm$0.007
	80%	0.060$\pm$0.008	**0.048**$\pm$**0.007**	0.057$\pm$0.023
128^2	5%	0.132$\pm$0.018	**0.062**$\pm$**0.007**	0.176$\pm$0.021
	10%	0.096$\pm$0.013	**0.046**$\pm$**0.005**	0.105$\pm$0.012
	20%	0.075$\pm$0.010	**0.035**$\pm$**0.004**	0.077$\pm$0.010
	40%	0.064$\pm$0.008	**0.028**$\pm$**0.003**	0.063$\pm$0.008
	60%	0.060$\pm$0.008	**0.027**$\pm$**0.003**	0.058$\pm$0.008
	80%	0.059$\pm$0.007	**0.031**$\pm$**0.003**	0.056$\pm$0.019

Table A10: **Burgers: sparse reconstruction at every evaluation resolution.** Missing-region relative L^2 error, mean $\pm$ standard deviation across test fields. Bold marks the lowest unrounded mean per resolution and context.

Resolution	Context	Op. ↓	EBO ↓	DDO ↓
32	5%	0.260$\pm$0.332	**0.211**$\pm$**0.407**	0.405$\pm$0.443
	10%	0.147$\pm$0.196	**0.108**$\pm$**0.204**	0.228$\pm$0.305
	20%	0.070$\pm$0.103	**0.038**$\pm$**0.079**	0.080$\pm$0.143
	40%	0.028$\pm$0.027	**0.011**$\pm$**0.009**	0.021$\pm$0.024
	60%	0.017$\pm$0.018	**0.007**$\pm$**0.005**	0.012$\pm$0.013
	80%	0.010$\pm$0.011	**0.005**$\pm$**0.004**	0.008$\pm$0.009
64	5%	0.178$\pm$0.212	**0.119**$\pm$**0.240**	0.332$\pm$0.355
	10%	0.083$\pm$0.117	**0.049**$\pm$**0.099**	0.119$\pm$0.165
	20%	0.034$\pm$0.044	**0.013**$\pm$**0.019**	0.031$\pm$0.044
	40%	0.015$\pm$0.013	**0.006**$\pm$**0.004**	0.011$\pm$0.011
	60%	0.010$\pm$0.008	**0.004**$\pm$**0.002**	0.007$\pm$0.006
	80%	0.008$\pm$0.007	**0.003**$\pm$**0.002**	0.006$\pm$0.004
128	5%	0.101$\pm$0.131	**0.051**$\pm$**0.098**	0.192$\pm$0.212
	10%	0.043$\pm$0.053	**0.018**$\pm$**0.026**	0.055$\pm$0.063
	20%	0.020$\pm$0.019	**0.007**$\pm$**0.006**	0.018$\pm$0.017
	40%	0.010$\pm$0.008	**0.004**$\pm$**0.002**	0.008$\pm$0.007
	60%	0.007$\pm$0.006	**0.003**$\pm$**0.002**	0.006$\pm$0.004
	80%	0.006$\pm$0.006	**0.003**$\pm$**0.002**	0.005$\pm$0.003
256	5%	0.064$\pm$0.085	**0.025**$\pm$**0.043**	0.111$\pm$0.130
	10%	0.028$\pm$0.029	**0.010**$\pm$**0.010**	0.035$\pm$0.031
	20%	0.014$\pm$0.012	**0.006**$\pm$**0.004**	0.015$\pm$0.013
	40%	0.008$\pm$0.006	**0.004**$\pm$**0.002**	0.007$\pm$0.005
	60%	0.006$\pm$0.005	**0.003**$\pm$**0.002**	0.005$\pm$0.003
	80%	0.006$\pm$0.005	**0.003**$\pm$**0.001**	0.005$\pm$0.003
512 (unseen)	5%	0.047$\pm$0.044	**0.014**$\pm$**0.015**	0.082$\pm$0.085
	10%	0.023$\pm$0.021	**0.008**$\pm$**0.006**	0.032$\pm$0.027
	20%	0.011$\pm$0.009	**0.006**$\pm$**0.003**	0.013$\pm$0.010
	40%	0.007$\pm$0.006	**0.005**$\pm$**0.002**	0.006$\pm$0.004
	60%	0.005$\pm$0.004	0.005$\pm$0.002	**0.004**$\pm$**0.003**
	80%	0.005$\pm$0.005	0.005$\pm$0.002	**0.005**$\pm$**0.003**

Table A11: **Inference cost and reconstruction accuracy.** Resolution 128, 20% context, 100 test fields; means over ten trials on one NVIDIA A100 (TF32). K is the number of EBO refinement steps or DDO denoising steps, with $K = 0$ for the direct operator. Error is missing-region relative L^2; time is for all 100 fields.

		Darcy flow		Navier–Stokes		Burgers	
Type	K	**Error**	**Time (ms)**	**Error**	**Time (ms)**	**Error**	**Time (ms)**
Op.	0	2.823×10^{-2}	18.30	7.457×10^{-2}	18.40	1.932×10^{-2}	1.194
EBO	1	5.527×10^{-1}	50.01	1.101×10^{0}	50.05	6.657×10^{-1}	2.730
EBO	2	3.020×10^{-1}	99.59	8.351×10^{-1}	99.61	2.252×10^{-1}	5.204
EBO	4	9.592×10^{-2}	198.8	4.809×10^{-1}	198.7	1.642×10^{-2}	10.25
EBO	8	3.306×10^{-2}	397.1	1.979×10^{-1}	397.0	6.778×10^{-3}	20.31
EBO	16	2.256×10^{-2}	794.0	8.257×10^{-2}	793.5	6.734×10^{-3}	40.30
EBO	32	1.858×10^{-2}	1588	4.242×10^{-2}	1586	6.735×10^{-3}	80.34
EBO	64	1.665×10^{-2}	3174	3.540×10^{-2}	3172	6.735×10^{-3}	160.6
EBO	128	1.582×10^{-2}	6348	3.431×10^{-2}	6344	6.735×10^{-3}	320.8
DDO	1	2.243×10^{0}	27.52	2.766×10^{0}	26.75	4.297×10^{0}	2.004
DDO	2	8.743×10^{-1}	47.38	1.381×10^{0}	46.83	1.642×10^{0}	3.719
DDO	4	1.984×10^{-1}	86.72	2.516×10^{-1}	85.72	5.698×10^{-2}	7.326
DDO	8	6.645×10^{-2}	177.4	9.243×10^{-2}	192.9	2.741×10^{-2}	26.98
DDO	16	3.355×10^{-2}	338.6	7.774×10^{-2}	343.6	2.172×10^{-2}	29.02
DDO	32	2.830×10^{-2}	669.7	7.690×10^{-2}	680.8	1.907×10^{-2}	57.98
DDO	64	2.911×10^{-2}	1332	7.843×10^{-2}	1353	1.866×10^{-2}	115.8
DDO	128	2.996×10^{-2}	2655	8.014×10^{-2}	2699	1.745×10^{-2}	230.6
DDO	250	3.083×10^{-2}	5188	8.143×10^{-2}	5265	1.786×10^{-2}	451.0

and between 2 and 4 steps on Burgers (about $9\times$), where it settles at 0.0068 from $K = 8$ against 0.0193 for the operator. On Darcy flow and Navier–Stokes, DDO stays above the operator at every number of denoising steps, coming within 0.3% of it on Darcy flow at 32 steps; on Burgers it is below the operator from 32 steps and reaches 0.0175 at 128 steps. The operator is therefore much cheaper, while EBO trades extra computation for lower error without retraining.

Tables A12–A14 give EBO's error after K steps at every resolution and context ratio. With sparse context at coarse resolutions (Darcy flow at 16^2 and Navier–Stokes at 32^2, with 5% and 10% context), the error is still falling at 128 steps. On Burgers with 40% context or more, it levels off after 8 steps at every resolution up to 128 points. On Navier–Stokes with dense context, steps beyond about 64 raise the error slightly (64^2 and 80% context: 0.045 at 64 steps, 0.052 at 128). Figure A7 shows EBO's reconstructions after 1 to 128 steps on each PDE, and Figure 6c shows the Darcy flow spectral error after 1, 4, 16 and 128 steps, which falls with refinement at all frequencies but one.

Table A12: **Darcy flow: EBO error after K energy-minimization steps.** Missing-region relative L^2, mean over the test fields, along the trajectories of the reported results (step count in Table A7); Op.: the operator's error. Bold: lowest error in each row at the precision shown.

Resolution	Context	Op.	K=1	K=2	K=4	K=8	K=16	K=32	K=64	K=96	K=128
16^2	5%	0.275	0.630	0.490	0.406	0.360	0.325	0.296	0.271	0.261	**0.256**
	10%	0.190	0.593	0.418	0.310	0.256	0.218	0.186	0.163	0.156	**0.154**
	20%	0.119	0.565	0.359	0.227	0.166	0.126	0.099	0.086	**0.083**	**0.083**
	40%	0.072	0.563	0.329	0.171	0.101	0.066	0.050	**0.047**	**0.047**	**0.047**
	60%	0.053	0.547	0.309	0.146	0.075	0.045	0.037	**0.036**	**0.036**	**0.036**
	80%	0.042	0.535	0.299	0.131	0.060	0.034	0.030	**0.029**	**0.029**	**0.029**
32^2	5%	0.133	0.594	0.412	0.275	0.196	0.144	0.110	0.093	0.089	**0.088**
	10%	0.080	0.565	0.347	0.190	0.119	0.078	0.057	**0.051**	**0.051**	**0.051**
	20%	0.050	0.564	0.322	0.137	0.067	0.041	**0.033**	**0.033**	0.034	0.034
	40%	0.034	0.573	0.318	0.114	0.043	0.027	**0.024**	0.025	0.026	0.026
	60%	0.028	0.572	0.321	0.113	0.041	0.025	**0.021**	0.022	0.023	0.023
	80%	0.025	0.565	0.317	0.115	0.043	0.024	**0.019**	**0.019**	0.020	0.020
64^2	5%	0.067	0.578	0.362	0.181	0.092	0.053	0.040	**0.037**	**0.037**	**0.037**
	10%	0.044	0.543	0.303	0.118	0.052	0.033	0.027	**0.026**	**0.026**	**0.026**
	20%	0.032	0.553	0.300	0.101	0.038	0.025	0.021	**0.020**	**0.020**	**0.020**
	40%	0.025	0.571	0.312	0.101	0.034	0.022	0.018	**0.017**	**0.017**	**0.017**
	60%	0.022	0.580	0.328	0.109	0.036	0.022	0.018	**0.017**	0.018	0.018
	80%	0.022	0.566	0.322	0.117	0.044	0.025	**0.020**	**0.020**	0.022	0.023
128^2	5%	0.051	0.564	0.337	0.138	0.053	0.032	0.027	**0.025**	**0.025**	**0.025**
	10%	0.036	0.532	0.287	0.097	0.038	0.026	0.022	0.020	**0.019**	**0.019**
	20%	0.028	0.558	0.300	0.096	0.034	0.023	0.019	0.017	**0.016**	**0.016**
	40%	0.024	0.577	0.314	0.099	0.033	0.021	0.017	0.015	**0.014**	**0.014**
	60%	0.023	0.578	0.326	0.106	0.034	0.021	0.016	0.014	0.014	**0.013**
	80%	0.023	0.570	0.325	0.116	0.040	0.024	0.019	**0.017**	**0.017**	**0.017**

Table A13: **Navier–Stokes: EBO error after K energy-minimization steps.** Missing-region relative L^2, mean over the test fields, along the trajectories of the reported results (step count in Table A7); Op.: the operator's error. Bold: lowest error in each row at the precision shown.

Resolution	Context	Op.	K=1	K=2	K=4	K=8	K=16	K=32	K=64	K=96	K=128
32^2	5%	0.437	1.172	1.004	0.832	0.715	0.631	0.554	0.478	0.439	**0.416**
	10%	0.280	1.095	0.875	0.637	0.480	0.389	0.315	0.254	0.230	**0.219**
	20%	0.172	1.111	0.867	0.558	0.331	0.227	0.166	0.133	0.124	**0.122**
	40%	0.108	1.139	0.888	0.544	0.280	0.157	0.100	**0.084**	0.085	0.086
	60%	0.085	1.137	0.883	0.539	0.280	0.150	0.085	**0.070**	0.071	0.073
	80%	0.075	1.106	0.846	0.516	0.280	0.154	0.083	0.061	**0.060**	0.061
64^2	5%	0.204	1.118	0.887	0.600	0.368	0.239	0.168	0.132	0.125	**0.123**
	10%	0.132	1.046	0.779	0.459	0.235	0.138	0.095	0.080	**0.078**	**0.078**
	20%	0.091	1.104	0.843	0.491	0.218	0.105	0.063	0.055	**0.054**	0.055
	40%	0.070	1.140	0.883	0.519	0.233	0.099	0.049	**0.043**	**0.043**	0.044
	60%	0.063	1.140	0.883	0.525	0.247	0.106	0.048	**0.041**	0.043	0.045
	80%	0.060	1.122	0.870	0.529	0.260	0.122	0.058	**0.045**	0.048	0.052
128^2	5%	0.132	1.108	0.855	0.519	0.238	0.117	0.076	0.064	**0.062**	**0.062**
	10%	0.096	1.036	0.756	0.414	0.174	0.084	0.054	0.047	**0.046**	**0.046**
	20%	0.075	1.105	0.841	0.482	0.199	0.083	0.043	0.036	**0.035**	**0.035**
	40%	0.064	1.139	0.882	0.516	0.225	0.088	0.036	0.029	**0.028**	**0.028**
	60%	0.060	1.137	0.880	0.520	0.240	0.096	0.036	**0.027**	**0.027**	**0.027**
	80%	0.059	1.129	0.880	0.535	0.255	0.114	0.046	**0.031**	**0.031**	0.032

Table A14: **Burgers: EBO error after K energy-minimization steps.** Missing-region relative L^2, mean over the test fields, along the trajectories of the reported results (step count in Table A7); Op.: the operator's error. Bold: lowest error in each row at the precision shown.

Resolution	Context	Op.	K=1	K=2	K=4	K=8	K=16	K=32	K=64	K=128
32	5%	0.260	0.843	0.551	0.394	0.310	0.246	0.215	**0.207**	0.211
	10%	0.147	0.797	0.433	0.239	0.164	0.125	0.111	**0.108**	**0.108**
	20%	0.070	0.710	0.295	0.087	0.047	**0.038**	**0.038**	**0.038**	**0.038**
	40%	0.028	0.693	0.234	0.020	**0.011**	**0.011**	**0.011**	**0.011**	**0.011**
	60%	0.017	0.660	0.208	0.012	**0.007**	**0.007**	**0.007**	**0.007**	**0.007**
	80%	0.010	0.659	0.192	0.008	**0.005**	**0.005**	**0.005**	**0.005**	**0.005**
64	5%	0.178	0.846	0.495	0.295	0.206	0.149	0.124	0.120	**0.119**
	10%	0.083	0.753	0.329	0.110	0.061	0.050	**0.049**	**0.049**	**0.049**
	20%	0.034	0.696	0.250	0.031	0.014	0.014	**0.013**	**0.013**	**0.013**
	40%	0.015	0.716	0.258	0.021	**0.006**	**0.006**	**0.006**	**0.006**	**0.006**
	60%	0.010	0.699	0.250	0.023	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**
	80%	0.008	0.707	0.257	0.021	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
128	5%	0.101	0.777	0.379	0.146	0.077	0.055	**0.051**	**0.051**	**0.051**
	10%	0.043	0.711	0.264	0.040	0.019	**0.018**	**0.018**	**0.018**	**0.018**
	20%	0.020	0.703	0.234	0.016	**0.007**	**0.007**	**0.007**	**0.007**	**0.007**
	40%	0.010	0.687	0.228	0.012	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**
	60%	0.007	0.684	0.226	0.011	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
	80%	0.006	0.677	0.225	0.010	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
256	5%	0.064	0.713	0.287	0.064	0.029	**0.025**	**0.025**	**0.025**	**0.025**
	10%	0.028	0.683	0.232	0.027	0.014	0.012	0.011	**0.010**	**0.010**
	20%	0.014	0.686	0.230	0.028	0.014	0.009	0.007	**0.006**	**0.006**
	40%	0.008	0.692	0.239	0.034	0.015	0.010	0.006	0.005	**0.004**
	60%	0.006	0.707	0.253	0.038	0.015	0.009	0.005	0.004	**0.003**
	80%	0.006	0.701	0.260	0.039	0.011	0.006	0.004	**0.003**	**0.003**
512 (unseen)	5%	0.047	0.745	0.283	0.037	0.015	**0.014**	**0.014**	**0.014**	**0.014**
	10%	0.023	0.704	0.239	0.021	0.010	0.009	0.009	**0.008**	**0.008**
	20%	0.011	0.689	0.232	0.025	0.011	0.008	0.007	**0.006**	**0.006**
	40%	0.007	0.689	0.237	0.040	0.018	0.011	0.008	0.006	**0.005**
	60%	**0.005**	0.688	0.265	0.063	0.026	0.014	0.010	0.007	**0.005**
	80%	**0.005**	0.703	0.306	0.096	0.033	0.015	0.010	0.007	**0.005**

E Darcy Flow: From Permeability to Pressure

This section learns the map from the permeability a of a Darcy flow problem to the pressure u it produces, the standard Darcy benchmark of the FNO papers, so the input and output are different functions.

E.1 Data, Models and Protocol

Data. The neuraloperator Darcy files hold 5,000 training and 1,000 test pairs (a, u) on a 128×128 grid, and 500 training pairs are held out for validation. The pressure fields and the resolutions (16^2 to 128^2, strided subsamples) are those of Appendix D; neuraloperator's own coarser files equal these subsamples. Both fields are standardized with their training statistics. The metric is the full-field relative L^2 error $\|\hat{u} - u\|/\|u\|$ in physical units, averaged over the test pairs.

Models and training. All methods use the Darcy FNO backbone and optimizer of Table A6, with one model per method trained on all four resolutions for 50,000 iterations. The operator maps $[a, \text{coordinates}]$ to $\hat{u}$ with an MSE loss. EBO reads $[u, a, \text{coordinates}]$ and returns a per-point energy whose mean is $\Psi(a, u)$; a prediction starts from a GRF sample and takes K EM steps. DDO, set up as in Appendix D, denoises $[z_t, a, \text{coordinates}]$. EBO's training settings (preconditioner length scale 0.03) and DDO's noise length scale (0.02) are those selected for Darcy flow in Appendix D; none was tuned on this task. Each run keeps the checkpoint with the lowest error on the 100 validation pairs.

Protocol. EBO's inference setting is chosen from the 60 candidates of Appendix D and DDO's from 9 step counts between 1 and 250, each by the geometric mean of the validation error over the four resolutions; the test set is used once. EBO uses 128 steps with multiplier 0.25 and DDO 250 denoising steps at every resolution, with one prediction per test pair.

E.2 Results

EBO has the lowest test error at every resolution (Table A15): 5% below the operator at 16×16 and 16–19% below at the finer grids, and 23–33% below DDO everywhere. All 8 paired comparisons favour EBO significantly, and DDO is significantly worse than the operator at every resolution. Most of the gain does not depend on the inference search: with its training-time setting (32 steps, multiplier 1), EBO is still significantly better than both baselines at every resolution, by 5–14% against the operator and 23–28% against DDO.

Figure A8 shows the error against the number of steps. EBO's error falls below both baselines' between 16 and 32 EM steps, while DDO's error falls steeply up to 16 denoising steps and little after that; even at its best step count on the test set, a choice made in its favour, DDO stays significantly above EBO. In the sample predictions (Figure A9), the errors concentrate along the jumps of a and the edges of the domain; the operator's error spreads over the domain as ripples, EBO's is lower in the interior, and DDO's is larger around some of the jumps.

Cost and limitations. At 128×128, inference on the 1,000 test pairs takes 72.6 s for EBO's 128 EM steps (18.2 s at its training-time setting), 57.8 s for DDO's 250 denoising steps and 0.2 s for the operator. Each method has one training run, so the intervals cover the spread over test pairs and not over training seeds. DDO's validation error was still falling slowly at the end of training.

Table A15: **Darcy flow, permeability a to pressure u.** Test relative L^2 error at each resolution, mean $\pm$ standard deviation over the 1,000 test pairs. Bold marks the lowest error per row.

Resolution	Op.	EBO	DDO
16×16	$0.0827_{\pm 0.0319}$	$\mathbf{0.0783_{\pm 0.0354}}$	$0.1019_{\pm 0.0464}$
32×32	$0.0547_{\pm 0.0248}$	$\mathbf{0.0458_{\pm 0.0248}}$	$0.0656_{\pm 0.0428}$
64×64	$0.0491_{\pm 0.0236}$	$\mathbf{0.0400_{\pm 0.0228}}$	$0.0577_{\pm 0.0426}$
128×128	$0.0491_{\pm 0.0232}$	$\mathbf{0.0395_{\pm 0.0227}}$	$0.0593_{\pm 0.0409}$
Geometric mean	0.0575	**0.0488**	0.0692

F TUSZ Dataset and Preprocessing

The Temple University Hospital Seizure Corpus (TUSZ) v2.0.3 [29] is the largest publicly available annotated EEG dataset for seizure detection, containing clinical EEG recordings from over 600 patients with expert annotations for seizure onset/offset times and type classifications. Automated seizure detection is clinically significant for continuous EEG monitoring in intensive care units, where timely identification of seizure activity can guide treatment decisions and improve patient outcomes.

We adopt a one-class learning paradigm: the model is trained exclusively on background (non-seizure) EEG segments and learns to model the dynamics of normal brain activity. At inference time, seizure segments—which the model has never seen during training—produce higher energy scores because they deviate from the learned normal distribution. This formulation naturally frames seizure detection as anomaly detection. We use the official train/dev/eval splits provided with the corpus.

F.1 Seizure Types

The TUSZ corpus employs a hierarchical annotation taxonomy that distinguishes between focal (partial) and generalized seizure types, as well as specific clinical manifestations. Table A16 summarizes 9 seizure type annotations used in the dataset.

Table A16: **TUSZ seizure type annotations.**

Label	Type	Description
bckg	Background	Normal, non-seizure activity
fnsz	Focal Non-Specific	Focal seizure, unclassified subtype
gnsz	Generalized Non-Specific	Generalized seizure, unclassified subtype
spsz	Simple Partial	Focal seizure without impaired awareness
cpsz	Complex Partial	Focal seizure with impaired awareness
absz	Absence	Brief generalized seizure with staring
tnsz	Tonic	Sustained muscle stiffening
tcsz	Tonic-Clonic	Stiffening followed by rhythmic jerking
mysz	Myoclonic	Brief, shock-like muscle jerks

F.2 Electrode Montage

We use the standard TCP (Temporal Central Parasagittal) bipolar montage with 22 differential channels. Bipolar montages compute the voltage difference between adjacent electrode pairs, which offers several advantages for clinical EEG analysis: they reduce common-mode artifacts, emphasize local electrical activity by attenuating distant sources, and provide better spatial localization of abnormal discharges. The TCP montage organizes electrodes into anatomically meaningful chains—temporal chains capture activity from the temporal lobes (important for temporal lobe epilepsy), parasagittal chains cover the sensorimotor cortex, and central chains bridge the hemispheres. Files missing any required electrodes are excluded from preprocessing.

Table A17: **22-channel TCP bipolar montage organization.**

Chain	Channels	Count
Left Temporal	FP1-F7, F7-T3, T3-T5, T5-O1	4
Right Temporal	FP2-F8, F8-T4, T4-T6, T6-O2	4
Central	A1-T3, T3-C3, C3-CZ, CZ-C4, C4-T4, T4-A2	6
Left Parasagittal	FP1-F3, F3-C3, C3-P3, P3-O1	4
Right Parasagittal	FP2-F4, F4-C4, C4-P4, P4-O2	4
Total		**22**

F.3 Preprocessing Pipeline

Our preprocessing pipeline consists of frequency filtering, normalization, and segmentation stages designed to prepare clinical EEG data for training the model.

Frequency Filtering. Raw EEG signals contain artifacts and noise outside the frequency range of interest for learning neural dynamics. We apply a two-stage filtering process:

- **Bandpass filter (0.5–70 Hz)**: A 4th-order IIR Butterworth filter removes DC drift and slow baseline wander (below 0.5 Hz) as well as high-frequency noise and muscle artifacts (above 70 Hz). The 0.5 Hz lower cutoff preserves slow EEG rhythms relevant to seizure activity while rejecting electrode drift.
- **Notch filter (60 Hz)**: A 4th-order IIR Butterworth notch filter attenuates power line interference, which is ubiquitous in clinical recordings and can obscure EEG features.

Normalization and Segmentation. We apply per-file z-score normalization *before* segmentation, computing channel-wise statistics over the entire recording: $\tilde{x}_c(t) = (x_c(t) - \mu_c)/(\sigma_c + \varepsilon)$. This approach preserves the relative amplitude relationships between segments from the same recording, which is important because seizure activity often manifests as amplitude changes relative to baseline.

The preprocessed signal is then segmented using non-overlapping windows of $T = 12$s duration with stride $S = 12$s. The 12-second window length provides sufficient temporal context to capture seizure morphology while keeping segment sizes manageable. Segments are labeled based on annotation overlap: a segment receives a seizure label if the maximum overlap with any seizure annotation exceeds the threshold, and background otherwise. Segments with ambiguous overlap are excluded.

F.4 Training Configuration

Table A18 summarizes the training configuration for EBO on the TUSZ dataset.

Table A18: **EBO training configuration for EEG training.**

Parameter	**Value**
Input channels	22
Hidden size / Depth / Heads	256 / 4 / 4
Learning rate	1×10^{-4}
Batch size	4
Position encoding	SIREN
EM steps	1–32 (randomized during training)
EM step size	0.01 (learnable, $100\times$ LR multiplier)
GRF kernel	Matérn ($\nu = 0.5$, $\ell = 0.1$, $\sigma = 1.0$)

G LOB Dataset and Preprocessing

Limit Order Book (LOB) data captures the full depth of buy and sell orders in financial markets, providing a comprehensive view of market microstructure at the tick level. LOB data is particularly valuable for understanding market dynamics because it reveals not just executed trades but also the latent supply and demand at various price levels.

We preprocess high-frequency LOB snapshots into fixed-format segments suitable for energy-based anomaly detection, with the realized volatility (RV) of the next window as the evaluation label. Unlike the TUSZ dataset where we employ one-class learning (training only on normal data), for LOB we use an *unsupervised* paradigm: the model trains on all market segments without label information, learning to model the joint distribution of LOB features. At inference time, we examine whether the model's energy correlates with future realized volatility—high energy indicating that the current market state deviates from typical patterns and may precede volatile periods.

G.1 Data Collection

We collected high-frequency limit order book data from the Korea Exchange (KRX) futures market, specifically focusing on KOSDAQ150 futures front-month contracts. The KOSDAQ150 futures contract is a derivative instrument based on the KOSDAQ150 index, which tracks 150 representative stocks listed on the KOSDAQ market. The data was gathered using a custom-developed collection program designed to capture the complete market microstructure at tick-level granularity.

The dataset spans two temporally separated periods: a training set from July 2, 2024 to December 12, 2024 (approximately 5.5 months), and a test set from January 17, 2025 to March 13, 2025 (approximately 2 months). This temporal separation ensures out-of-sample evaluation and prevents information leakage from future market conditions into the training process.

We focused exclusively on the regular trading session (8:45–15:35 KST), excluding the opening and closing auction periods. The auction mechanisms employ different price discovery processes (call auction vs. continuous matching) that would introduce heterogeneity in the order book dynamics, making them unsuitable for our analysis of continuous market behavior.

Our collection program recorded complete limit order book snapshots at every market event, capturing the full depth of the order book whenever the market state changed. The recorded events include new limit order submissions, order cancellations, market order arrivals, and trade executions. This event-driven approach preserves the complete information flow of the market, capturing both the evolution of latent liquidity (through limit order submissions and cancellations) and realized demand (through market orders and executions). For each event, we recorded the top 5 price levels on both the bid and ask sides of the order book, along with the corresponding order quantities at each level.

G.2 Feature Channels

We extract 21 feature channels from raw LOB data, capturing price levels, order volumes, and returns. The features are designed to be scale-invariant and comparable across different assets and time periods.

Price Features (Channels 0–9). Price features encode the distance of each order book level from the mid-price, normalized by the tick size $\tau = 10$ and log-transformed:

$$\text{Bid Price}^{(i)} = \ln\left(\frac{|P_{\text{mid}} - P_{\text{bid}}^{(i)}|}{\tau} + 1\right) \tag{69}$$

$$\text{Ask Price}^{(i)} = \ln\left(\frac{|P_{\text{ask}}^{(i)} - P_{\text{mid}}|}{\tau} + 1\right) \tag{70}$$

where $P_{\text{mid}} = (P_{\text{bid}}^{(1)} + P_{\text{ask}}^{(1)})/2$ is the mid-price. The log transformation compresses the dynamic range of price distances, making features from different market conditions (tight vs. wide spreads) more comparable. The tick size normalization ($\tau = 10$) converts absolute price differences to tick units.

Table A19: **LOB feature channels (21 total)**. Tick size $\tau = 10$.

Channels	Name	Formula
0–4	Bid Price (1–5)	$\ln(\lvert P_{\text{mid}} - P_{\text{bid}}^{(i)} \rvert/\tau + 1)$
5–9	Ask Price (1–5)	$\ln(\lvert P_{\text{ask}}^{(i)} - P_{\text{mid}} \rvert/\tau + 1)$
10–14	Bid Amount (1–5)	$\ln(A_{\text{bid}}^{(i)} + 1)$
15–19	Ask Amount (1–5)	$\ln(A_{\text{ask}}^{(i)} + 1)$
20	Log Return	$\ln(P_{\text{mid}}^{(t)}/P_{\text{mid}}^{(t-1)})$

Volume Features (Channels 10–19). Volume features capture the liquidity available at each price level using log-transformed order amounts:

$$\text{Amount}^{(i)} = \ln(A^{(i)} + 1) \tag{71}$$

The log transformation is essential because order sizes span multiple orders of magnitude (from single-lot retail orders to large institutional blocks), and raw volumes would dominate the feature space.

Return Feature (Channel 20). The log return captures price momentum at the tick level:

$$r_t = \ln\left(\frac{P_{\text{mid}}^{(t)}}{P_{\text{mid}}^{(t-1)}}\right), \quad P_{\text{mid}} = (P_{\text{bid}}^{(1)} + P_{\text{ask}}^{(1)})/2 \tag{72}$$

Log returns are preferred over simple returns because they are additive over time and approximately symmetric for small price changes, providing a stationary transformation of the non-stationary price process.

G.3 Target: Realized Volatility

For each segment, we compute the Realized Volatility (RV) of the *next* 10-second time window as the target label:

$$\text{RV} = \sqrt{\sum_{k=1}^{K} r_k^2} \tag{73}$$

where $r_k = \ln(P_{\text{mid}}^{(k)}/P_{\text{mid}}^{(k-1)})$ are the log returns in the future window, and K is the number of events. High RV indicates volatile periods, while low RV indicates calm markets.

It is important to note that our energy-based model does not directly predict RV as a regression target. Instead, the model learns the distribution of LOB feature sequences in an unsupervised manner, and we evaluate whether the learned energy function correlates with future volatility. The hypothesis is that market states preceding high-volatility periods may exhibit distinctive patterns (e.g., order book imbalances, unusual spread dynamics) that the energy model assigns higher energy to, as these states deviate from typical market conditions.

G.4 Preprocessing Pipeline

The LOB preprocessing pipeline converts raw tick data into fixed-format segments suitable for neural network training.

Time parsing and filtering. Raw timestamps are stored in HHMMSSCC format (hours, minutes, seconds, centiseconds). We parse these to seconds from midnight and filter to keep only events within the core trading session (8:45–15:35), excluding the opening and closing auction periods where price discovery mechanisms differ from continuous trading.

Sliding window segmentation. We use overlapping windows with $T = 10$s duration and $S = 1$s stride. The shorter stride creates more training samples and provides finer temporal resolution for detecting regime transitions. Windows with fewer than 10 events are discarded to ensure sufficient information content.

Variable sequence lengths. Each window contains a variable number of events depending on market activity—quiet periods may have only 10–20 events per 10 seconds, while active periods can have hundreds. This natural variation is preserved in our preprocessing, and the model handles variable-length inputs through subsampling to a fixed length at training time.

G.5 Training Configuration

Table A20 summarizes the training configuration for EBO on the LOB dataset.

Table A20: **EBO training configuration for LOB volatility detection.**

Parameter	Value
Input channels	21 (all LOB features)
Hidden size / Depth / Heads	256 / 4 / 4
Learning rate	1×10^{-5}
Batch size	16
Epochs	3
Position encoding	SIREN
EM steps	1–32 (randomized during training)
EM step size	0.01 (learnable, $10\times$ LR multiplier)
GRF kernel	Matérn ($\nu = 1.5$, $\ell = 0.1$, $\sigma = 1.0$)

G.6 Evaluation Metrics

For market volatility prediction, we evaluate the model's ability to identify market states that precede high-volatility periods using three metrics:

Energy–Volatility Correlation. We compute Spearman correlation coefficients between energy and future realized volatility. Spearman correlation is robust to outliers and captures monotonic relationships. A positive correlation indicates that the model assigns higher energy to market states preceding volatile periods.

Quintile Analysis. We partition test segments into 5 groups based on energy percentiles (0–20%, 20–40%, ..., 80–100%) and examine the distribution of future RV within each quintile using box plots.

High-Volatility Detection. Test segments in the top 20% of future RV are labeled high-volatility and those in the bottom 20% low-volatility, and the middle 60% is left out. EBO's AUROC and detection rates use its energy as the score that separates the two groups (Figure 9d and Table 3). Leaving out the middle quintiles makes the two groups easier to separate than a split of all segments at a single threshold. Because a segment counts as high-volatility whatever its current volatility, these metrics measure discrimination of future volatility levels, not of transitions from calm to volatile markets. Volatility persists, so the realized volatility of the current window, $\mathrm{RV}_{t:t+T}$, is a natural baseline score, and we do not compare energy against it. Consecutive windows overlap by 9 of their 10 seconds, so test segments are strongly dependent; we report point estimates only, and intervals for these metrics would need to account for this dependence.

H Additional Results

This section collects the figures for the PDE experiments of Appendices D and E.

Sparse reconstruction at every context ratio. Figures A1 and A2 show single test fields at 5% context for Darcy flow and Burgers, as Figure 5 does for Navier–Stokes. In both, EBO's error is the lowest (relative L^2 0.027 against 0.051 for the operator and 0.082 for DDO on Darcy flow, 0.019 against 0.034 and 0.142 on Burgers), and on Burgers it follows the shock more closely than either baseline. Figures A3, A4 and A5 show one test field per PDE at every context ratio.

Number of energy-minimization steps. Figure A6 plots the error against the number of steps from Table A11, and Figure A7 shows EBO's reconstructions after 1 to 128 steps on each PDE (Appendix D.5).

Darcy flow from permeability to pressure. Figure A8 shows the test error against the number of steps, and Figure A9 shows two test pairs at 128×128 (Appendix E).

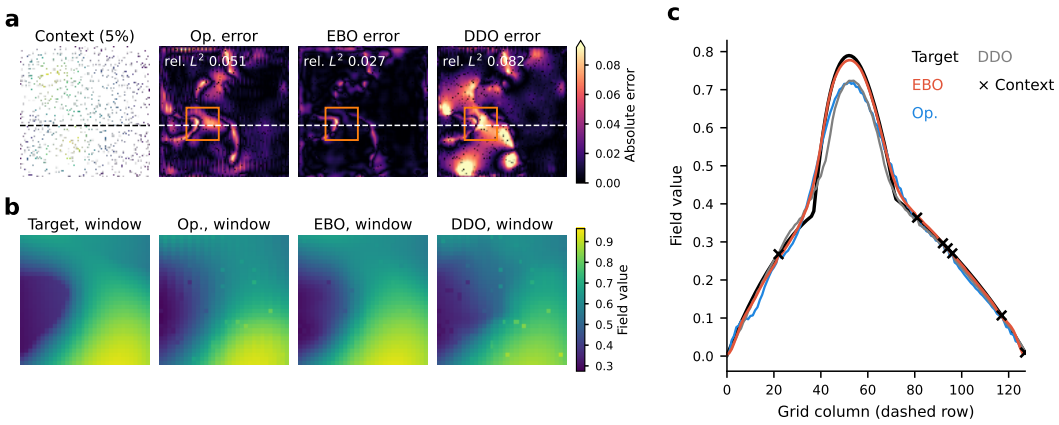


Figure A1: **Darcy flow sparse reconstruction at resolution 128 and 5% context.** Along the dashed row (**c**), EBO follows the peak near column 52 (0.78 against 0.79; Op. and DDO 0.72).

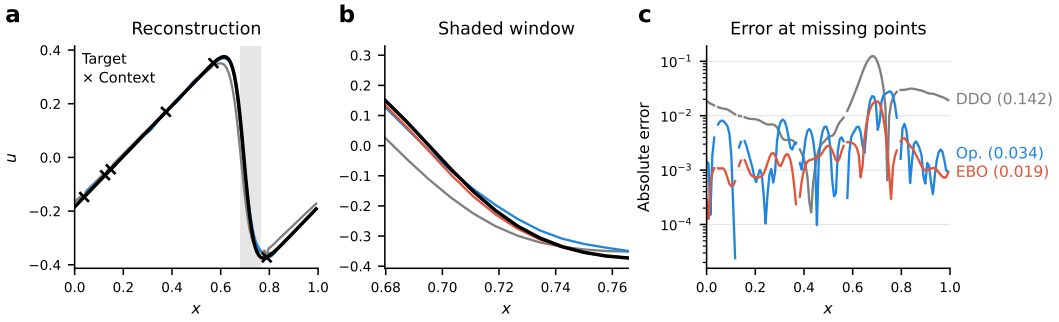


Figure A2: **Burgers sparse reconstruction at resolution 128 and 5% context. a,** Target and reconstructions with the six context points; the shaded window around the shock is enlarged in **b**. **c,** Absolute error at the missing points, with each method's missing-region relative L^2 in brackets.

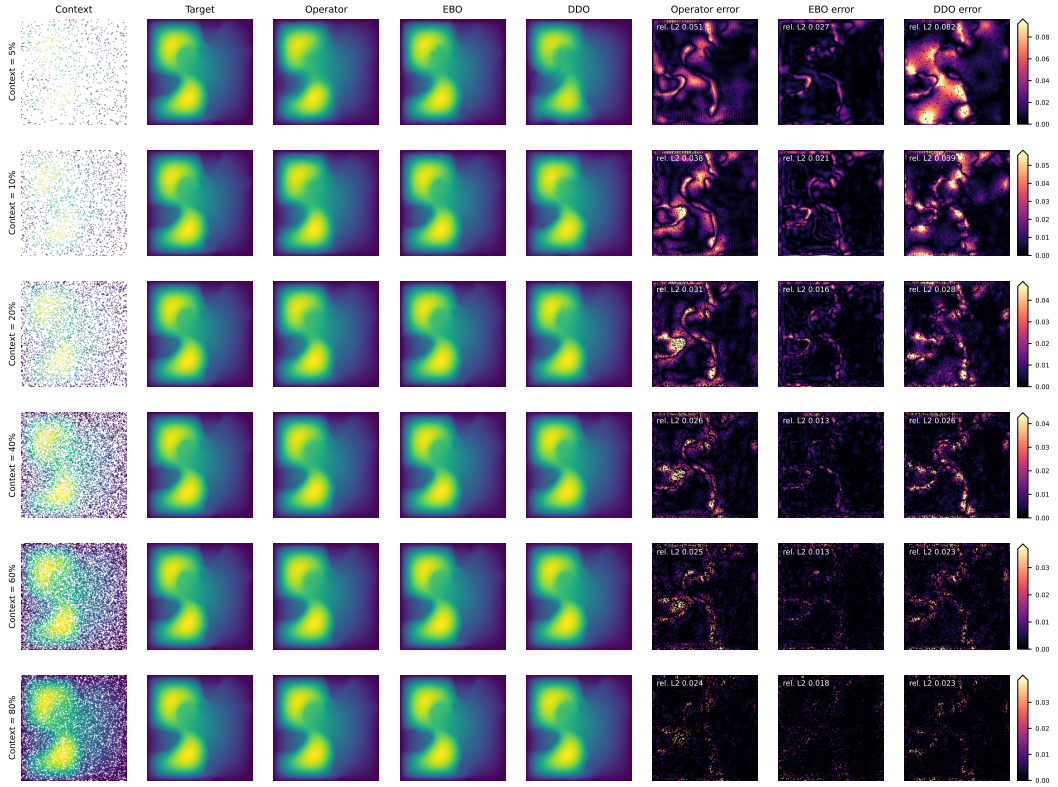


Figure A3: **Sparse Darcy-flow reconstruction over context ratios at resolution 128.** Each row is one context ratio from 5% to 80%. Columns, from left: context, target, the operator, EBO and DDO reconstructions, and each method's missing-region error map with its relative L^2 error. Error maps in a row share one colour scale. EBO has the lowest error on this test field at every context ratio.

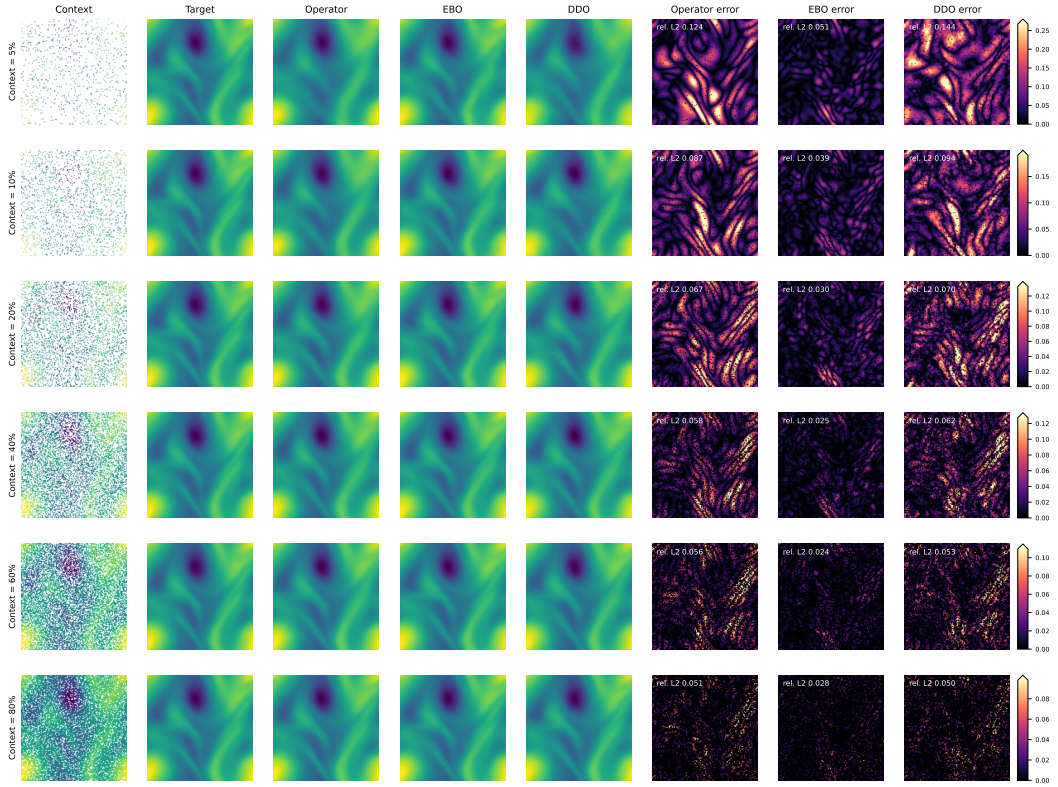


Figure A4: **Sparse Navier–Stokes reconstruction over context ratios at resolution 128**, in the layout of Figure A3. EBO has the lowest error on this test field at every context ratio and recovers coherent flow patterns from sparse observations.

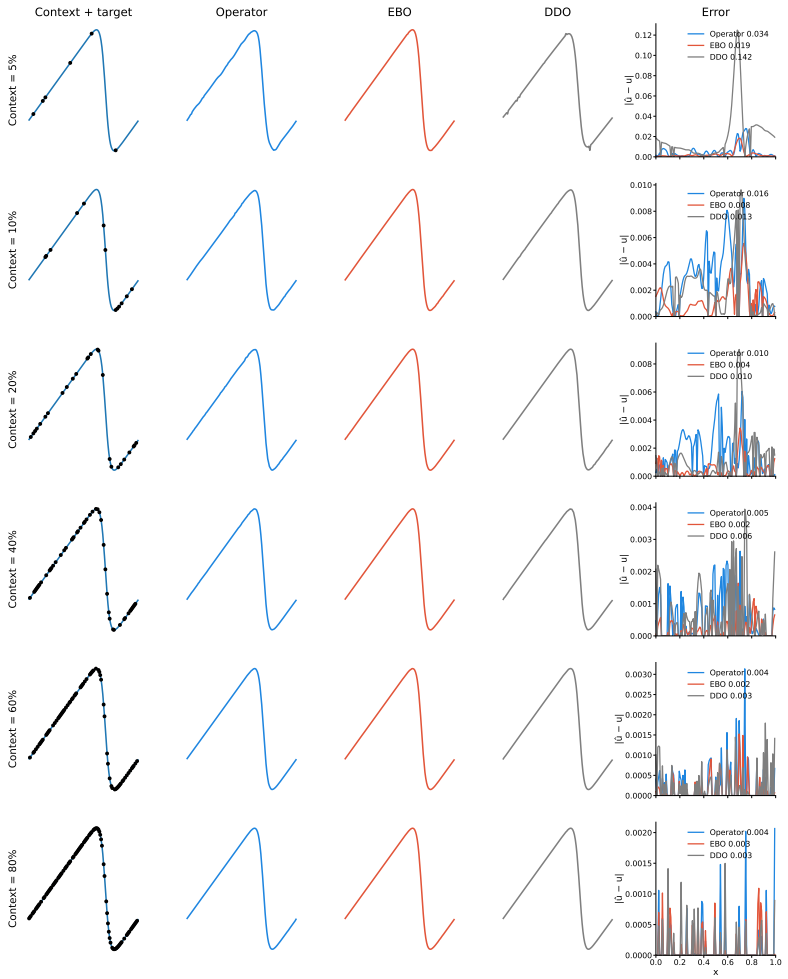


Figure A5: **Sparse Burgers reconstruction over context ratios at resolution 128.** Each row is one context ratio from 5% to 80%. Columns, from left: context points on the target, the operator, EBO and DDO reconstructions, and the absolute error $|\hat{u} - u|$ at each point, with each method's relative L^2. EBO's error is lowest at every context ratio, tied with DDO at 80% at the precision shown.

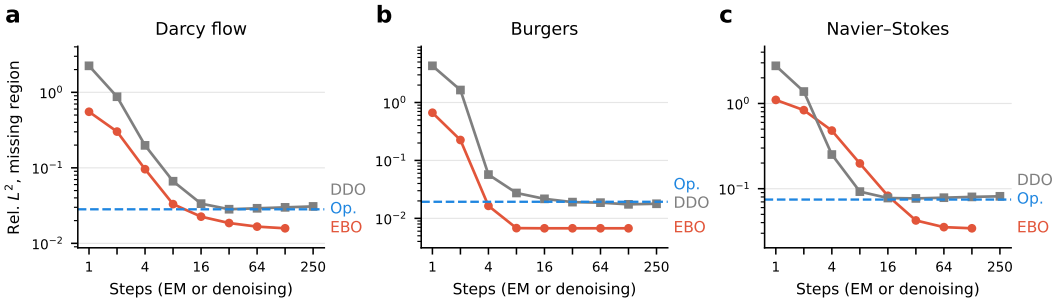


Figure A6: **Error against the number of steps on all three PDEs**, from the measurements of Table A11. Missing-region relative L^2 of EBO after K energy-minimization steps and of DDO with K denoising steps; the operator (dashed) is one forward pass.

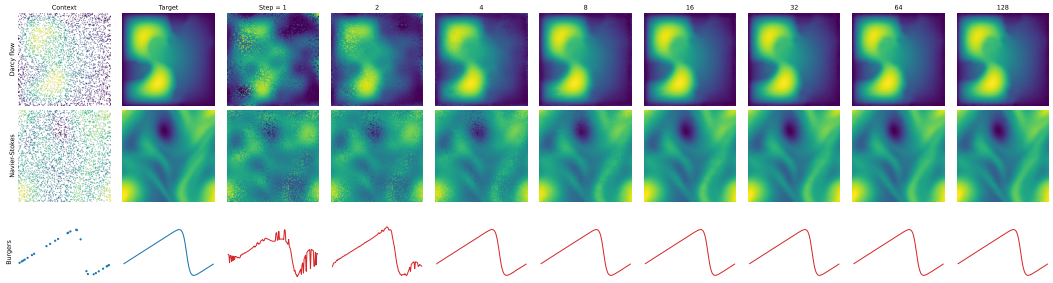


Figure A7: **EBO energy-minimization reconstruction trajectory.** EBO reconstructions at resolution 128 and 20% context for Darcy flow, Navier–Stokes and Burgers. Columns, from left: context, target, and EBO after 1, 2, 4, 8, 16, 32, 64 and 128 steps. As refinement proceeds, EBO removes high-frequency artifacts and moves the candidate function toward the target while keeping the context.

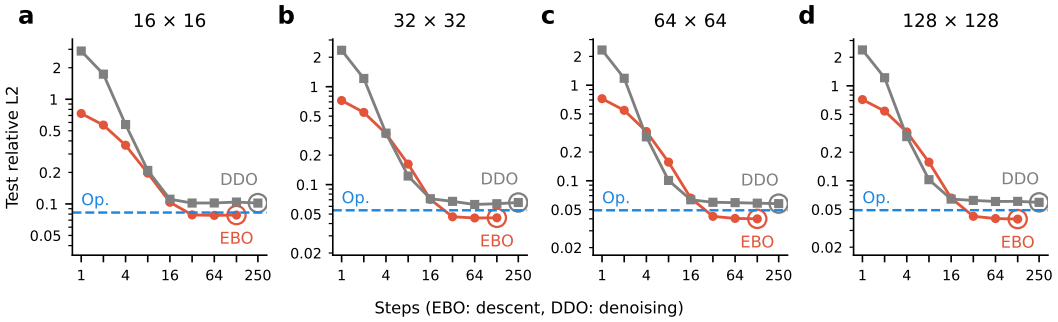


Figure A8: **Darcy flow, permeability to pressure: test error against the number of steps** at (**a**) 16×16, (**b**) 32×32, (**c**) 64×64 and (**d**) 128×128. Red: EBO after k EM steps. Grey: DDO with k denoising steps, one sampling pass per k. Rings mark the selected settings; the dashed line is the operator. Each method counts steps of its own kind, so equal positions do not mean equal cost.

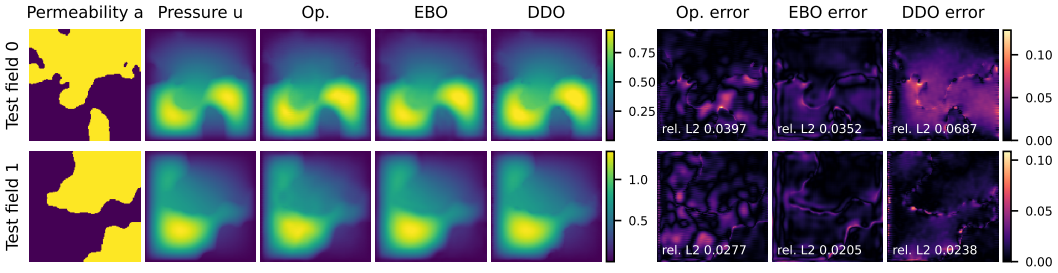


Figure A9: **Darcy flow, permeability to pressure: two test pairs at** 128×128**.** Columns: the permeability a, the true pressure u, the operator, EBO and DDO predictions on the colour scale of u, and each prediction's absolute error with its relative L^2 error, on one error scale per row.

I Broader Impacts

EBO assigns an energy to a whole function without labels, which can help flag unusual signals in scientific and clinical data and reconstruct physical fields from sparse measurements. A high energy only means that a signal differs from the training data, so the seizure-detection results (AUROC 0.848 on TUSZ) do not support clinical use without validation on independent cohorts and review by clinicians. Energy scores that track future volatility could be used for trading or risk management, where errors, or market regimes absent from the training period, can cause financial losses; our results cover KOSDAQ150 front-month futures from July 2024 to March 2025 and do not establish reliability on other markets or periods. Iterative energy minimization also uses more computation at inference than a one-shot operator (Appendix D.5), which adds to energy use when deployed at scale.

J Licenses of Existing Assets

The neuraloperator library is released under the MIT License, and its Darcy flow files are distributed on Zenodo (doi:10.5281/zenodo.12784353) under the Creative Commons Attribution 4.0 International license. The Navier–Stokes files are distributed for the same library. The code of the Fourier neural operator [15], whose Gaussian random field generator produces the Burgers initial conditions, is released under the MIT License. Access to the TUH EEG Seizure Corpus [29] requires signing its data-use agreement with the Neural Engineering Data Consortium.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly state the paper's main claims, including its contributions, assumptions, and limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The "Limitations" paragraph at the end of the Conclusion (Section 6) discusses the inference cost of EBO and the function classes on which it does not improve over the operator.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: Our paper includes the full set of assumptions and a complete proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We describe detailed methods and instructions for reproducing the experimental results in the corresponding sections of the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes].

Justification: We provide open access to the data and code via a GitHub repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: We provide experimental details, such as data splits and hyperparameters, in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: We report error bars for all results shown in the figures, as well as detailed numerical statistical values.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We provide information on the compute resources used for the experiments in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We reviewed the NeurIPS Code of Ethics, and our research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: We discuss both the potential positive and negative societal impacts of the work in Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: Since our paper is basic academic research, it does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We cite the original papers that provide the publicly available assets, and Appendix J gives their licenses or terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: We provide detailed documentation, such as README files, alongside the assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.