FROM DISTRIBUTIONS TO STOCHASTIC PROCESSES: NEURAL APPROXIMATION OF MEASURE-VALUED MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning mappings between probability distributions arises naturally in settings where both inputs and outputs are represented by populations of samples rather than individual observations. This paper develops an approximation-theoretic framework for such distribution-to-distribution learning problems and extends it to mappings between stochastic processes. For continuous operators acting on W_2 -compact families of finite-dimensional input laws, we establish uniform neural approximation in the 2-Wasserstein metric. Our construction relies on finite law statistics as input representations, a simplex-valued neural map, and a shared atomic output support that guarantees the output is always a valid probability measure. We then extend this approximation principle to probability laws on separable Hilbert spaces via fixed finite-rank orthogonal projections. Together, these results establish the representational feasibility of learning transformations whose inputs and outputs are probability laws rather than deterministic vectors or functions.

To assess practical relevance, we construct numerical problems whose target transformations are naturally defined at the level of distributions or probability laws: predicting the first-passage-time distribution of an Ornstein–Uhlenbeck process and the nonlinear response-path laws of a Duffing oscillator. Since the theoretical framework is deliberately model-agnostic and covers a substantially broader class of operators than can be efficiently represented by any single practical architecture, our experiments instantiate the framework using task-adapted neural models rather than reproducing the theoretical construction verbatim. Both models outperform a fixed-feature MLP baseline and distribution-space kernel regression. These proof-of-concept demonstrations complement the approximation theory by supporting the learnability and practical relevance of distribution-to-distribution transformations for random systems.

1 INTRODUCTION

Motivation and problem setup. Many random systems are naturally described at the population level. In uncertainty propagation, mean-field dynamics, ensemble forecasting, stochastic PDEs, and Bayesian inverse problems, the desired prediction is an output law rather than the response of one prescribed input realization (Lord et al., 2014; Walsh, 1986; Sznitman, 1991; Carmona & Delarue, 2018; Stuart, 2010; Evensen, 2003). When the observations are paths or fields, the same question concerns laws on function spaces. In both cases, probability distributions are the mathematical objects of interest, while finite ensembles are observations of those objects.

We formulate this task as learning a *distributional operator*

$$\mathcal{G} : \mathcal{K} \subset \mathcal{P}_2(\mathcal{X}) \longrightarrow \mathcal{P}_2(\mathcal{Y}), \qquad \mu \longmapsto \mathcal{G}(\mu),$$

where $\mathcal{P}_2$ denotes probability measures with finite second moment. Training observations are pairs of empirical laws $\{(\widehat{\mu}_i, \widehat{\nu}_i)\}_{i=1}^n$. The input and output ensembles within a pair may be sampled independently, so there need not be a meaningful map between individual particles. The target is instead the law-level relation $\nu_i = \mathcal{G}(\mu_i)$. We measure approximation in the 2-Wasserstein metric W_2. The finite-dimensional setting takes $\mathcal{X} = \mathbb{R}^{d_x}$ and $\mathcal{Y} = \mathbb{R}^{d_y}$; the process setting takes $\mathcal{X}$ and $\mathcal{Y}$ to be separable Hilbert spaces of paths or fields.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Related work and the theoretical gap. Classical universal approximation theory treats continuous maps between finite-dimensional vectors (Cybenko, 1989). Neural-operator theory extends the encoding–mapping–reconstruction principle to deterministic functions, with DeepONet and Fourier neural operators providing prominent examples (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). Our input and output, however, are probability laws. A deterministic operator between functions does not establish that an input law can be encoded by finitely many observable statistics, transformed continuously, and decoded into a valid output law with uniform error control.

Distribution regression and invariant set networks address complementary parts of the problem. Distribution regression maps an empirical law to a scalar or vector, while distribution-to-distribution regression permits a distributional response (Szabó et al., 2016; Oliva et al., 2013). DeepSets and Set Transformers provide practical permutation-invariant encoders for unordered samples (Zaheer et al., 2017; Lee et al., 2019), but an architecture for finite sets alone does not give a uniform approximation theorem on a space of laws. More specifically, Oliva et al. (2013) assume densities on compact Euclidean domains and derive an expected L_2 density-risk bound for a kernel-smoothed orthogonal-series estimator. Their setting does not yield uniform W_2 approximation or guarantee that a truncated output series is a probability density. Pham & Warin (2023) approximate maps of the form $(x, \mu) \mapsto V(x, \mu)$, whose outputs are vectors or functions rather than probability laws.

Measure-valued approximation is closer to our objective. The probabilistic transformer maps finite-dimensional features to mixtures of fixed measures and proves compact-uniform approximation for $\mathcal{P}_1(\mathbb{R}^D)$-valued maps in W_1 (Kratsios, 2021); the general framework of Kratsios et al. (2023) covers maps between broad classes of metric spaces, including probability-valued targets. Transformer dynamics can also interpolate finitely many prescribed input–target measure pairs (Geshkovski et al., 2024). These results establish important general measure-valued approximation principles. What remains unresolved for our setting is their combination with a law-valued input accessed through an empirical ensemble: a finite encoder that captures W_2 rather than only weak behavior, a decoder that always returns a probability measure, a uniform bound for replacing feature integrals by sample averages, and a lifting from finite-dimensional laws to laws on path spaces. The gap is particularly delicate because W_2 convergence requires both weak convergence and control of second moments, and path-space reduction adds input- and output-projection errors.

Our theory. For a W_2-compact family $\mathcal{K} \subset \mathcal{P}_2(\mathbb{R}^{d_x})$ and a W_2-continuous operator $\mathcal{G} : \mathcal{K} \to \mathcal{P}_2(\mathbb{R}^{d_y})$, Theorem 2.1 constructs a finite law encoder from bounded test-function expectations and a truncated second-moment feature. A neural map with a softmax output assigns weights to a common finite set of output atoms. The bounded tests control weak behavior, the moment feature supplies the additional information required by W_2, and the simplex-valued decoder guarantees nonnegative weights with unit mass. Compactness and continuity then give uniform W_2 approximation of the target operator. For empirical inputs, Proposition B.5 replaces population feature integrals by independent sample averages and separates the population approximation error from a uniform high-probability query-sampling error.

Theorem 2.2 extends the result to laws on separable Hilbert spaces. Fixed finite-rank orthogonal maps encode paths into coordinate vectors, the finite-dimensional distributional operator acts on the induced coordinate laws, and synthesis maps reconstruct output paths. Uniform projection on W_2-compact law families decomposes the total error into input projection, finite-dimensional neural approximation, and output reconstruction. The argument remains entirely at the level of probability laws and therefore does not require eventwise pairing of input and output paths. Together, the two theorems provide the missing W_2-specific link between empirical law encoding, valid probability-valued decoding, and process-law approximation. They establish representational feasibility rather than statistical training rates or a mandatory practical architecture.

Contributions. Our contributions are:

- We formulate distribution-to-distribution learning from finite, unpaired input and output ensembles as approximation of a continuous operator between probability-law spaces.
- We prove uniform neural approximation in W_2 on compact families of finite-dimensional input laws using finitely many law statistics, neural simplex weights, and a valid common-support output law. We also give a uniform query-sample bound for empirical feature averages.

- We lift the approximation result to process and random-field laws on separable Hilbert spaces through fixed finite-rank projections, with explicit control of input projection, finite-dimensional approximation, and output reconstruction.
- We demonstrate the framework on a censored Ornstein–Uhlenbeck first-passage law and nonlinear Duffing response-path laws. Task-specific categorical and generative decoders assess practical learnability against fixed-feature and distribution-space kernel baselines without identifying the existence construction with a unique architecture.

2 Universal Approximation Theory for Operators between Probability Laws

We establish two universal approximation results: one for operators between finite-dimensional probability laws and one for laws on separable Hilbert spaces. Both use encoding, neural mapping, and valid probability-law reconstruction. Supporting constructions, empirical-feature bounds, and projection lemmas are summarized below; their formal statements and all proofs appear in Appendix B.

2.1 Problem setup and assumptions

We begin with the standard probability-space notation used throughout. If $(\mathcal{X}, d_{\mathcal{X}})$ is a Polish space, $\mathcal{P}(\mathcal{X})$ denotes its Borel probability measures and, for $1 \le p < \infty$,

$$\mathcal{P}_p(\mathcal{X}) = \left\{ \mu \in \mathcal{P}(\mathcal{X}) : \int_{\mathcal{X}} d_{\mathcal{X}}(x, x_0)^p \, d\mu(x) < \infty \right\}, \tag{1}$$

where the definition does not depend on the fixed point $x_0 \in \mathcal{X}$. For $\mu, \nu \in \mathcal{P}_p(\mathcal{X})$, let $\Pi(\mu, \nu)$ be their set of couplings and define

$$W_p^{\mathcal{X}}(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \left(\int_{\mathcal{X}\times\mathcal{X}} d_{\mathcal{X}}(x, y)^p \, d\pi(x, y) \right)^{1/p}. \tag{2}$$

When the underlying space is clear, we simply write W_p. For a measurable map $F : \mathcal{X} \to \mathcal{Y}$, $F_{\#}\mu$ denotes the pushforward law, and for i.i.d. samples $X_1, \dots, X_{N_{\text{in}}} \sim \mu$ we write

$$\widehat{\mu}_{N_{\text{in}}} = \frac{1}{N_{\text{in}}} \sum_{b=1}^{N_{\text{in}}} \delta_{X_b} \tag{3}$$

for the empirical measure.

Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. A distributional operator is a map

$$\mathcal{G} : \mathcal{K} \subset \mathcal{P}_2(\mathcal{X}) \longrightarrow \mathcal{P}_2(\mathcal{Y}).$$

Its uniform approximation error on $\mathcal{K}$ is measured by $\sup_{\mu\in\mathcal{K}} W_2^{\mathcal{Y}}(\mathcal{N}(\mu), \mathcal{G}(\mu))$.

The finite-dimensional theory takes $\mathcal{X} = \mathbb{R}^{d_x}$ and $\mathcal{Y} = \mathbb{R}^{d_y}$ and uses the following assumptions throughout:

(A1) $\mathcal{K} \subset \mathcal{P}_2(\mathbb{R}^{d_x})$ is W_2-compact. Consequently, the family is tight and its second moments are uniformly integrable:

$$\lim_{R\to\infty} \sup_{\mu\in\mathcal{K}} \int_{\{\|x\|_2 > R\}} \|x\|_2^2 \, d\mu(x) = 0. \tag{4}$$

(A2) $\mathcal{G} : \mathcal{K} \to \mathcal{P}_2(\mathbb{R}^{d_y})$ is W_2-continuous. Hence $\mathcal{G}$ is uniformly continuous on $\mathcal{K}$ and $\mathcal{G}(\mathcal{K})$ is W_2-compact.

(A3) The chosen continuous, nonpolynomial activation gives uniform neural-network approximation on compact Euclidean sets, as in the classical universal approximation theorem. Bounded clipping, or an equivalent bounded output activation, is available for the feature networks.

Two kinds of input statistics are needed. Choose bounded weak-convergence tests $\psi_1, \dots, \psi_M \in C_b(\mathbb{R}^{d_x})$ and retain the second moment as a separate coordinate:

$$E_{\mathrm{wk}}(\mu) = \left(\int \psi_1 \, d\mu, \dots, \int \psi_M \, d\mu \right),$$
$$m_2(\mu) = \int \|x\|_2^2 \, d\mu(x),$$
$$E(\mu) = (E_{\mathrm{wk}}(\mu), m_2(\mu)) \in \mathbb{R}^{M+1}. \tag{5}$$

The bounded coordinates determine weak behavior, while convergence of the second moments upgrades weak convergence to W_2 convergence. Compactness then permits a finite choice of tests at any prescribed accuracy.

The construction has three layers. First, E represents the input law to the accuracy required by the uniform continuity of $\mathcal{G}$. Second, a continuous simplex-valued map assigns weights to a common finite output support. Third, ordinary neural networks approximate the feature functions and simplex map. The Tietze–Dugundji extension theorem supplies the simplex-valued extension needed in the second layer, while tightness and (4) control neural approximation of the feature integrals.

2.2 Universal approximation of finite-dimensional laws

The first theorem combines finite law representation with neural realization. The encoder integrates bounded neural features, including a truncated approximation of the second moment, and a softmax network predicts weights on a common finite output support.

Theorem 2.1 (Universal approximation of distributional operators)**.** *Assume* (A1)–(A3). *Let* $\Delta_N = \{w \in [0,1]^N : \sum_{j=1}^N w_j = 1\}$. *For every* $\varepsilon > 0$*, there exist* $M, N \in \mathbb{N}$*, common output atoms* $a_1, \dots, a_N \in \mathbb{R}^{d_y}$*, bounded scalar feature networks* $\widehat{\psi}_i$*, a bounded neural approximation* $\widehat{q}_R$ *of* $q(x) = \|x\|_2^2$ *obtained by truncation at some radius* R*, and a softmax network* $\widehat{p} : \mathbb{R}^{M+1} \to \Delta_N$. *Define*

$$\widehat{E}_{\mathrm{wk},R}(\mu) = \left(\int \widehat{\psi}_1 \, d\mu, \dots, \int \widehat{\psi}_M \, d\mu \right),$$
$$\widehat{E}_R(\mu) = \left(\widehat{E}_{\mathrm{wk},R}(\mu), \int \widehat{q}_R \, d\mu \right), \tag{6}$$

and

$$\widehat{\mathcal{N}}_\theta(\mu) = \sum_{j=1}^N \widehat{p}_j(\widehat{E}_R(\mu)) \delta_{a_j}. \tag{7}$$

Then the networks can be chosen so that

$$\sup_{\mu \in \mathcal{K}} W_2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

Construction and proof outline. Compactness permits finitely many bounded tests and a second-moment coordinate to distinguish input laws to any required W_2 tolerance (Lemma B.1). Uniform continuity of $\mathcal{G}$ transfers this tolerance to the outputs. Local finite-support output approximations are collected on common atoms, and a partition of unity gives continuous simplex weights. The formal continuous construction is Theorem B.2 in Appendix B.1. Bounded tests are then approximated by clipped networks on large balls; tightness controls the complement, and uniform integrability controls truncation of the quadratic feature (Lemma B.3). Regularizing the weights, approximating their logits, and applying softmax preserves nonnegativity and unit mass. Common-support stability converts weight error to W_2 error (Lemma B.4). The full proof of Theorem 2.1 is in Appendix B.2.

The fixed atoms are an existence device shared by all input laws. Practical models may use richer structured decoders, but their approximation properties require separate justification.

Access through empirical ensembles. For an already realized neural operator, the population feature integrals can be replaced by averages of independent query samples. Hoeffding concentration

and the output network's modulus of continuity give a high-probability bound uniform over the queried law in $\mathcal{K}$. This separates population approximation error from empirical feature-estimation error. With a Lipschitz weight network, the latter contributes an output W_2 bound of order $N_{\text{in}}^{-1/4}$ for fixed confidence and realized architecture. The formal bound and its constants are in Proposition B.5, Appendix B.3. This is a query-sample guarantee, not a training, optimization, or network-width rate. Boundedness includes the truncated quadratic feature; the same concentration conclusion need not hold for an untruncated second moment.

2.3 Universal approximation of Hilbert-space laws

For process and random-field laws, finite-dimensionalization acts on each path and therefore produces a coordinate law, rather than a single feature vector representing the whole input distribution. Let $\mathcal{X}$ and $\mathcal{Y}$ be separable Hilbert spaces. Fix increasing finite-dimensional subspaces $X_n \subset \mathcal{X}$ and $Y_m \subset \mathcal{Y}$ with dense unions, with $d_n = \dim X_n$ and $q_m = \dim Y_m$. Orthonormal coordinates define analysis maps $\mathcal{E}_{\mathcal{X}}^n : \mathcal{X} \to \mathbb{R}^{d_n}$ and $\mathcal{E}_{\mathcal{Y}}^m : \mathcal{Y} \to \mathbb{R}^{q_m}$ and synthesis maps $\mathcal{D}_{\mathcal{X}}^n : \mathbb{R}^{d_n} \to \mathcal{X}$ and $\mathcal{D}_{\mathcal{Y}}^m : \mathbb{R}^{q_m} \to \mathcal{Y}$. Their compositions

$$P_{\mathcal{X}}^n = \mathcal{D}_{\mathcal{X}}^n \circ \mathcal{E}_{\mathcal{X}}^n, \qquad P_{\mathcal{Y}}^m = \mathcal{D}_{\mathcal{Y}}^m \circ \mathcal{E}_{\mathcal{Y}}^m$$

are orthogonal projections. Their pushforwards approximate every W_2-compact law family uniformly; the formal reconstruction and stability results are Lemmas B.7 and B.6 in Appendix B.4.

Theorem 2.2 (Universal approximation on separable Hilbert-space laws)**.** *Assume the activation condition* (A3)*. Let $\mathcal{X}$ and $\mathcal{Y}$ be separable Hilbert spaces equipped with the orthogonal coordinate systems above. Let $\mathcal{K} \subset \mathcal{P}_2(\mathcal{X})$ be $W_2^{\mathcal{X}}$-compact, and let*

$$\mathcal{G} : \mathcal{U} \longrightarrow \mathcal{P}_2(\mathcal{Y})$$

be W_2-continuous on an open neighborhood $\mathcal{U}$ of $\mathcal{K}$. For every $\varepsilon > 0$, there exist $n, m \in \mathbb{N}$ and a finite-dimensional neural distributional operator $\widehat{\mathcal{N}}_\theta : \mathcal{P}_2(\mathbb{R}^{d_n}) \to \mathcal{P}_2(\mathbb{R}^{q_m})$ of the form in Theorem 2.1 such that

$$\mathcal{N}_{\theta,n,m} = (\mathcal{D}_{\mathcal{Y}}^m)_{\#} \circ \widehat{\mathcal{N}}_\theta \circ (\mathcal{E}_{\mathcal{X}}^n)_{\#} \tag{8}$$

satisfies

$$\sup_{\mu \in \mathcal{K}} W_2^{\mathcal{Y}}(\mathcal{N}_{\theta,n,m}(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

Proof outline and interpretation. Uniform projection accuracy first makes $(P_{\mathcal{X}}^n)_{\#}\mu$ close to μ over $\mathcal{K}$. The open neighborhood ensures that projected input laws remain in the domain of $\mathcal{G}$, and continuity controls the induced output perturbation. Uniform output projection reduces the target to a finite-dimensional coordinate law. Theorem 2.1 approximates the resulting law operator, and synthesis is an isometry. The triangle inequality thus separates input projection, finite-dimensional neural approximation, and output reconstruction errors. The complete proof is in Appendix B.4; further path-law construction details are in Appendix B.5.

The result requires no eventwise coupling of input and output paths. Its orthogonal coordinate systems are fixed independently of the input law; a distribution-specific PCA or Karhunen–Loève basis, or a learned encoder, requires additional stability and uniform-reconstruction analysis.

3 Experiments

Theoretical results are typically established through general constructions to ensure broad applicability, but such constructions are often not directly implementable in practice. We therefore design concrete and computationally feasible distributional neural operators for controlled finite-dimensional Gaussian distributions, first-passage times of Ornstein–Uhlenbeck (OU) processes, and nonlinear Duffing oscillators. Rather than mechanically reproducing the theoretical constructions, we follow their underlying principles and tailor each model to the structure of its problem. The controlled Gaussian experiment acts on finite-dimensional distributions, whereas the OU and Duffing tasks use empirical laws of stochastic-process paths. Owing to space limitations, the main text focuses on the OU and Duffing tasks; complete results for the controlled Gaussian task are provided in Appendix E, and the corresponding theoretical extensions appear in Appendix C.

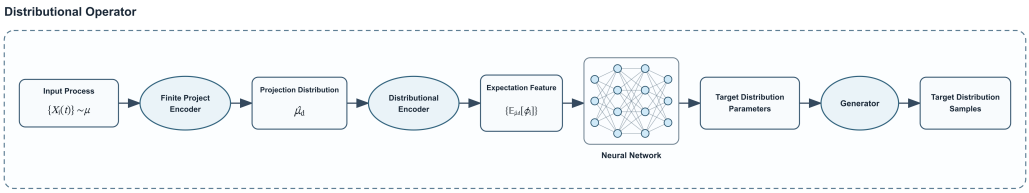

Figure 1: A general example of the model pipeline. The model used for a specific task may differ in certain details, but broadly follows this design rationale. For task-specific model structures, see Figures 5 and 6 in the appendix.

To the best of our knowledge, these tasks have no established benchmark or off-the-shelf baseline. We therefore compare against task-appropriate fixed-feature and kernel-regression alternatives. Implementation details for data generation and the comparison methods are provided in Appendix D.

Unless otherwise specified, input and output ensembles are sampled independently within each law pair, so the models cannot exploit trajectory-level correspondence. Every experiment uses 1,000 training law pairs and 200 test law pairs, which exhaust the 1,200-law dataset. The test set is used only to evaluate final model performance. Neural experiments are repeated with five random seeds. The default neural training configuration is AdamW for 1,000 epochs with initial learning rate 10^{-3} and weight decay 10^{-2}. All computations are performed on one NVIDIA GeForce RTX 4090 GPU with 24 GB of memory. Source code is available at `https://anonymous.4open.science/r/distributional_operator-3E6C`.

3.1 Experiment I: OU First-Passage Distribution

3.1.1 Problem

The OU first-passage problem has a standard interpretation as a leaky integrate-and-fire (LIF) neuron. In this model, the membrane potential evolves until it first reaches a firing threshold, at which point a spike is recorded and the potential is reset (Burkitt, 2006). Under the classical diffusion approximation with Gaussian white-noise input, the subthreshold membrane potential is an OU process. Consequently, the elapsed time from a reset to the next threshold crossing is one inter-spike interval (ISI), whose law is the OU first-passage-time law (Ricciardi & Sato, 1988; Alili et al., 2005). Specifically, the membrane potential follows

$$dV_t = \left(-\frac{V_t}{\gamma} + m\right) dt + \sqrt{q}\, dW_t, \qquad V(0) = v_0, \tag{9}$$

with threshold V_{th} and first-passage time

$$T = \inf\{t \geq 0 : V(0) = v_0, V_t \geq V_{\mathrm{th}}\}. \tag{10}$$

The model does not observe (m, q) directly. Instead, the input law is that of the drifted Brownian process

$$dI_t = m\, dt + \sqrt{q}\, dW_t, \qquad I(0) = 0, \tag{11}$$

with the same (m, q). The target distributional operator is

$$\mathcal{G} : \mathcal{L}(I) \longmapsto \mathcal{L}(T),$$

where the first-passage law is represented by 48 finite-time categories and one censoring category. The dataset contains 1,200 input–output law pairs. Each input law is represented by N_{in} paths on 256 observation times, and each target law by N_{out} independently simulated reset trials. The ensembles share the same law parameters but have no samplewise pairing. The parameter family, simulation scheme, censoring rule, binning, and split are detailed in Appendix D.1.1.

3.1.2 Model Instantiation

Following the process pipeline, we use a training-fitted PCA encoder $\mathcal{E}_I$ to preprocess the paths, selecting the truncation dimension d_X so that the cumulative explained-variance ratio exceeds 0.99. For the ith input ensemble, the Distributional Operator computes

$$\begin{aligned} \alpha_{ij} &= \mathcal{E}_I(I_{ij}) \in \mathbb{R}^{d_X}, \\ f_{ij} &= \phi_{\text{path}}(\alpha_{ij}), \\ h_i &= \frac{1}{N_{\text{in}}} \sum_{j=1}^{N_{\text{in}}} \phi_{\text{set}}(f_{ij}), \\ \ell_i &= \rho_\theta(h_i), \qquad \widehat{p}_i = \text{softmax}(\ell_i) \in \Delta_{49}. \end{aligned}$$

The shared path and set encoders followed by mean aggregation make the model invariant to the ordering of the input paths. The first 48 entries of $\widehat{p}_i$ parameterize the finite first-passage intervals and the last entry represents censoring. Training minimizes the count-weighted categorical negative log-likelihood. PCA fitting, layer widths, parameter counts, optimization, checkpointing, and complete baseline configurations are reported in Appendix D.1.2.

3.1.3 Evaluation Metrics and Results

Negative log-likelihood is the categorical training objective and measures predictive fit. Hellinger distance and KL divergence compare the full 49-category distributions from complementary symmetric and directed perspectives. Finite-bin W_2 measures displacement of probability mass over passage time after removing and renormalizing the censoring category, while tail error measures the absolute error in the probability of no crossing. Lower is better for every metric; formal definitions are in Appendix D.1.3.

Model	NLL	Hellinger	KL divergence	$W_{2,\text{fin}}$	Tail error
Distributional Operator	3.11176 ± 0.00085	0.18055 ± 0.00048	0.10636 ± 0.00085	0.16634 ± 0.00286	0.00373 ± 0.00018
Fixed-Feature MLP	3.11680 ± 0.00011	0.18592 ± 0.00013	0.11140 ± 0.00011	0.20957 ± 0.00026	0.00399 ± 0.00003
Kernel regression	3.15241	0.21616	0.14701	0.34567	0.00639

Table 1: Test results for input process laws mapped to OU first-passage-time laws. Entries are the mean $\pm$ standard deviation over five runs with random initializations. Kernel regression is deterministic and therefore has zero standard deviation. Lower is better for every metric.

For a qualitative theoretical reference, we evaluate the OU first-passage-time density using its Bessel-bridge representation (Alili et al., 2005). This reference density is independent of the empirical target histograms used for training and lets us compare the shapes predicted by the learned and kernel-based models against a continuous first-passage-time calculation.

The Distributional Operator gives the lowest mean NLL, Hellinger distance, KL divergence, finite-bin W_2, and tail error. Kernel regression is worse on every metric, indicating that local averaging between training laws does not capture the map as accurately. The density comparisons in Figure 2 show that the learned operator recovers the location, peak, and tail of the theoretical first-passage-time density across representative test laws.

3.2 Experiment II: Duffing Uncertainty Propagation

3.2.1 Problem

We study a distribution-to-distribution map induced by the fixed hardening Duffing oscillator

$$\ddot{Y}(t) + 0.2\dot{Y}(t) + Y(t) + Y(t)^3 = X(t), \qquad Y(0) = \dot{Y}(0) = 0,$$

on $t \in [0, 20]$. Given a probability law μ of stochastic forcing paths, the target is the response law under the Duffing solution operator $\mathcal{S}$, namely

$$\mathcal{G} : \mu \longmapsto \mathcal{S}_{\#}\mu.$$

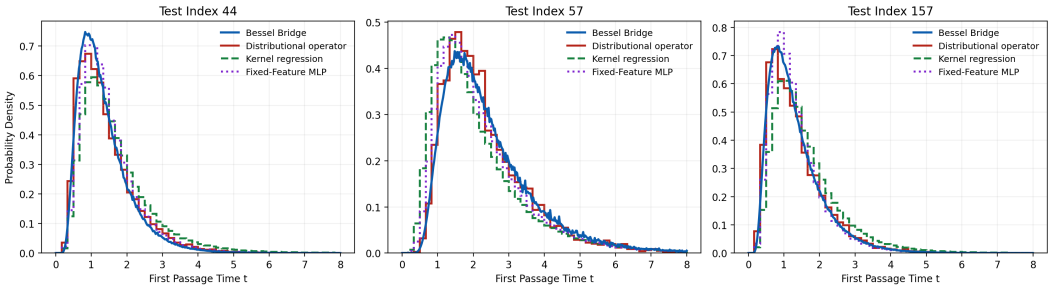

Figure 2: OU first-passage-time probability densities for three representative test laws. The continuous blue curve is the theoretical reference calculated from the Bessel-bridge representation. The step curves show the densities predicted by the Distributional Operator, kernel regression, and the Fixed-Feature MLP on the 48 finite time bins. Agreement with the reference curve compares how well the models recover the location, height, and tail of the first-passage-time density.

This is a distributional uncertainty-propagation problem: the model must characterize an ensemble of nonlinear responses rather than predict the response to one prescribed forcing trajectory. Such laws arise in nonlinear vibration and energy-harvesting models under stochastic or broadband excitation (Daqaq, 2010; Jia, 2020). The dataset contains 1,200 input–output law pairs. Each law is represented by N_{in} forcing paths and N_{out} independently sampled response paths on 256 observation times. The ensembles share the same forcing-law parameters but have no samplewise pairing. The forcing family, numerical solver, and measure-level construction are detailed in Appendix D.2.

3.2.2 Model Instantiation

Following the process pipeline, we use training-fitted PCA encoders $\mathcal{E}_X$ and $\mathcal{E}_Y$ to preprocess the forcing and response paths. Their truncation dimensions d_X and d_Y are selected so that each cumulative explained-variance ratio exceeds 0.99. For the ith input ensemble,

$$\begin{aligned}
\alpha_{ij} &= \mathcal{E}_X(X_{ij}) \in \mathbb{R}^{d_X}, \\
c_i &= \rho_\theta\left(\frac{1}{N_{\text{in}}}\sum_{j=1}^{N_{\text{in}}}\phi_\theta(\alpha_{ij})\right), \\
z_{ir} &\sim \mathcal{N}(0, I_{d_Y}), \qquad \beta_{ir} = F_\theta(z_{ir}; c_i), \\
\widehat{Y}_{ir} &= \mathcal{D}_Y(\beta_{ir}), \qquad r = 1, \ldots, N_{\text{out}},
\end{aligned}$$

where F_θ is a conditional RealNVP and $\mathcal{D}_Y$ is the inverse output-PCA reconstruction map (Dinh et al., 2017). Thus the empirical forcing-process law is projected, encoded by expectation features, combined with independent latent draws, and decoded into samples from the predicted response-process law.

The conditional generator uses six affine coupling transformations. For a masked partition $h = (h_{\mathcal{I}}, h_{\mathcal{J}})$, each transformation has the form

$$\begin{aligned}
h'_{\mathcal{I}} &= h_{\mathcal{I}}, \\
h'_{\mathcal{J}} &= h_{\mathcal{J}} \odot \exp(\widetilde{s}_\ell(h_{\mathcal{I}}, c_i)) + t_\ell(h_{\mathcal{I}}, c_i), \qquad \widetilde{s}_\ell = 2\tanh(s_\ell/2).
\end{aligned}$$

Training minimizes the debiased Sinkhorn divergence between N_{out} generated and N_{out} target response paths. Layer widths, parameter counts, PCA fitting, coupling masks, optimization schedules, and complete baseline configurations are reported in Appendices D.2.2 and D.2.3.

3.2.3 Evaluation Metrics and Results

Sinkhorn divergence is the training objective and measures regularized transport discrepancy between response ensembles. RBF MMD compares kernel mean embeddings and is sensitive to broad distributional differences. Sliced W_2 averages one-dimensional transport discrepancies over random

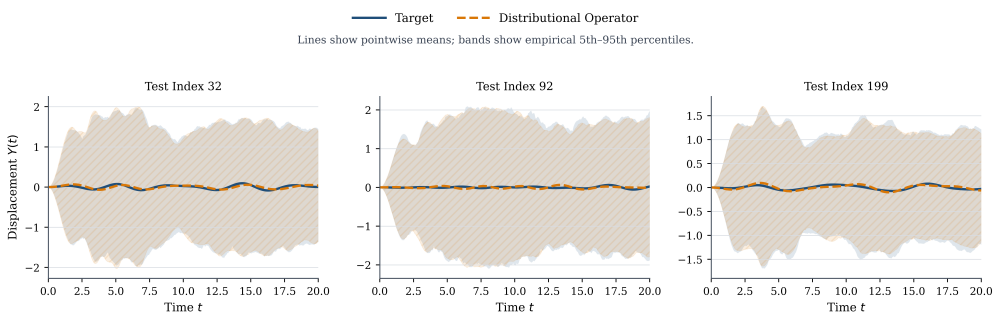

Figure 3: Conditional distribution prediction for the Duffing oscillator. Target and predicted response distributions are shown for three randomly chosen test forcing measures. Solid blue lines and shaded regions denote the reference mean and 5th–95th percentile envelope, while dashed orange lines and hatched regions show the corresponding model predictions. Agreement in both statistics demonstrates the model's accuracy in approximating the nonlinear pushforward from forcing to response distributions.

path-space projections, while energy distance compares within- and between-ensemble Euclidean distances. Formal definitions and evaluation settings are in Appendix D.2.4. We compared the performance of distributional operator (proposed model) with that of two baseline models on the test data in Table 2.Lower values indicate closer agreement for all four metrics. To mitigate the effects of randomness, the evaluation results were computed using five different random seeds and are reported as the mean ± standard deviation of the returns, except for the kernel regression model, which is deterministic and therefore reported as a single value. To intuitively illustrate the predictive performance, we present 3the results of the models on three randomly selected test datasets.

Model	Sinkhorn	MMD	SW_2	Energy
Distributional Operator	8.2050 ± 0.2786	0.0760 ± 0.0010	0.1639 ± 0.0036	0.3407 ± 0.0104
Fixed-Feature MLP	9.9426 ± 0.6795	0.0777 ± 0.0017	0.1718 ± 0.0036	0.3890 ± 0.0173
Kernel regression	18.6291	0.1255	0.2498	0.8650

Table 2: Test performance on the Duffing dataset. Neural-model entries are the mean $\pm$ sample standard deviation on the 200-law test set. Kernel regression is a single fit with a fixed bandwidth. Sinkhorn denotes the Sinkhorn divergence, MMD the maximum mean discrepancy, SW_2 the sliced 2-Wasserstein distance, and Energy the energy distance; lower values indicate better agreement.

4 Conclusion

We developed a constructive approximation theory for continuous distribution-to-distribution maps in the 2-Wasserstein metric. On W_2-compact input families, finitely many law statistics, a neural simplex-valued map, and a shared atomic support suffice for uniform approximation by valid output measures. We also quantified the error incurred when those statistics are estimated from a finite unordered ensemble. By combining finite-rank analysis and synthesis maps with the finite-dimensional construction, the same principle extends to probability laws on separable Hilbert spaces and therefore to distributional operators acting on stochastic processes.

The OU first-passage and Duffing uncertainty-propagation experiments show that this law-level formulation is computationally meaningful and learnable from unpaired ensembles. The task-adapted categorical and conditional-flow models outperform the fixed-feature and kernel-regression comparators on the reported metrics. These architectures instantiate the framework's central principles but do not reproduce the proof's finite-support construction, and the reported comparisons are not evidence of architectural optimality: the baseline set is necessarily limited, stronger designs may exist, and the remaining predictive errors leave substantial room for improvement. Future work should tighten the connection between constructive theory and practical architectures, weaken the compactness and neighborhood assumptions, obtain sharper statistical and optimization rates, and evaluate the framework on broader and more demanding stochastic systems.

REFERENCES

Larbi Alili, Pierre Patie, and Jesper Lund Pedersen. Representations of the first hitting time density of an ornstein–uhlenbeck process. *Stochastic Models*, 21(4):967–980, 2005. doi: 10.1080/15326340500294702. URL https://doi.org/10.1080/15326340500294702.

Anthony N. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95(1):1–19, 2006. doi: 10.1007/s00422-006-0068-6. URL https://doi.org/10.1007/s00422-006-0068-6.

René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications*. Springer, 2018.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Mohammed F. Daqaq. Response of uni-modal duffing-type harvesters to random forced excitations. *Journal of Sound and Vibration*, 329(18):3621–3631, 2010. doi: 10.1016/j.jsv.2010.04.002.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

Borjan Geshkovski, Philippe Rigollet, and Domènec Ruiz-Balet. Measure-to-measure interpolation using transformers. *arXiv preprint arXiv:2411.04551*, 2024.

Yu Jia. Review of nonlinear vibration energy harvesting: Duffing, bistability, parametric, stochastic and others. *Journal of Intelligent Material Systems and Structures*, 31(7):921–944, 2020. doi: 10.1177/1045389X20905989.

Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Anastasis Kratsios. Universal regular conditional distributions. *arXiv preprint arXiv:2105.07743*, 2021.

Anastasis Kratsios, Chong Liu, Matti Lassas, Maarten V. de Hoop, and Ivan Dokmanić. An approximation theory for metric space-valued functions with a view towards deep learning. *arXiv preprint arXiv:2304.12231*, 2023.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753, 2019.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Gabriel J. Lord, Catherine E. Powell, and Tony Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge University Press, 2014.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Junier Oliva, Barnabás Póczos, and Jeff Schneider. Distribution to distribution regression. In *International Conference on Machine Learning*, pp. 1049–1057, 2013.

Huyên Pham and Xavier Warin. Mean-field neural networks: Learning mappings on wasserstein space. *Neural Networks*, 168:380–393, 2023.

Luigi M. Ricciardi and Shunsuke Sato. First-passage-time density and moments of the ornstein–uhlenbeck process. *Journal of Applied Probability*, 25(1):43–57, 1988. doi: 10.1017/S0021900200040626. URL https://doi.org/10.1017/S0021900200040626.

Andrew M. Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Zoltán Szabó, Bharath K. Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40, 2016.

Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'Eté de Probabilités de Saint-Flour XIX–1989*, pp. 165–251. Springer, 1991.

John B. Walsh. An introduction to stochastic partial differential equations. In *Ecole d'Eté de Probabilités de Saint-Flour XIV–1984*, pp. 265–439. Springer, 1986.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

AI USE DISCLOSURE

Generative AI tools were used to polish the writing and improve readability; to search for and retrieve relevant literature and technical information; to prepare initial drafts of portions of the manuscript; and to assist in generating the synthetic datasets used in the experiments. The authors independently checked the retrieved sources and reviewed, revised, and validated all AI-assisted text, citations, data-generation procedures, and resulting datasets. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

A TABLE OF NOTATIONS

Notation	Meaning
$\mathbb{R}$, $\mathbb{N}$	Real numbers and positive integers.
$\mathcal{X}$, $\mathcal{Y}$	Input and output state spaces; in the process setting these are separable Hilbert spaces.
$\mathcal{P}_2(\mathcal{X})$	Borel probability measures on $\mathcal{X}$ with finite second moment.
W_2, $W_2^{\mathcal{X}}$	Quadratic Wasserstein distance, with the superscript indicating the underlying state space when needed.
$\mathcal{K}$	A W_2-compact family of admissible input laws.
μ, ν	Generic input and output probability laws.
$\mathcal{G}$	Target distributional operator, mapping an input law to an output law.
$\mathcal{N}$, $\widehat{\mathcal{N}}_\theta$	Continuous finite-support approximation and its neural realization.
δ_x	Dirac probability measure concentrated at x.
$F_\#$, $F_\#\mu$	Pushforward of a measure through a measurable map F.
$E(\mu)$, $\widehat{E}(\mu)$	Exact finite-dimensional law encoder and its neural approximation.
ψ_i, $\widehat{\psi}_i$	Bounded test functions and their neural approximations.
$m_2(\mu)$	Second moment $\int \|x\|_2^2 \, d\mu(x)$.
a_j, p_j	Shared output-support points and their simplex-valued weights.
Δ_N	Probability simplex $\{q \in [0,1]^N : \sum_{j=1}^N q_j = 1\}$.
M, N	Number of input features and output support points in the theoretical construction.
N_{in}	Number of samples representing an input distribution in a theoretical query or experiment.
N_{out}	Number of samples representing an output distribution in an experiment.
N_{batch}	Number of law-level examples in an optimization mini-batch.
$\mathcal{E}_{\mathcal{X}}^n$, $\mathcal{D}_{\mathcal{X}}^n$	Finite-rank encoder and decoder for the Hilbert-space state $\mathcal{X}$.
$P_{\mathcal{X}}^n$	Orthogonal projection $\mathcal{D}_{\mathcal{X}}^n \circ \mathcal{E}_{\mathcal{X}}^n$.
$\mathbb{E}$, Pr	Expectation and probability.
supp, Lip, diam	Support, Lipschitz constant, and diameter.
ε, δ	Approximation tolerance and confidence level.

B Definitions and Theory Details

We collect the definitions, supporting results, and proofs for the main-text universal approximation theorems.

B.1 Continuous finite-support construction

Lemma B.1 (Finite separation in W_2). *Let $\mathcal{K} \subset \mathcal{P}_2(\mathbb{R}^d)$ be W_2-compact and let $\eta > 0$. Then there are $M \in \mathbb{N}$, functions $\psi_1, \dots, \psi_M \in C_b(\mathbb{R}^d)$, and $r > 0$ such that, for all $\mu, \nu \in \mathcal{K}$,*

$$\begin{aligned} &\|E_{\mathrm{wk}}(\mu) - E_{\mathrm{wk}}(\nu)\|_2 < r, \\ &|m_2(\mu) - m_2(\nu)| < r \quad \Longrightarrow \quad W_2(\mu, \nu) < \eta. \end{aligned}$$

Proof. Choose a countable convergence-determining family in $C_b(\mathbb{R}^d)$. If no finite subfamily worked, compactness would give sequences μ_k, ν_k with all first k test integrals and their second moments approaching one another, but with $W_2(\mu_k, \nu_k) \geq \eta$. Passing to W_2-convergent subsequences gives limits with identical bounded-test integrals and hence the same weak limit. Convergence of the second moments then characterizes W_2 convergence, a contradiction. □

Theorem B.2 (Continuous finite-support approximation). *Let $\mathcal{K} \subset \mathcal{P}_2(\mathbb{R}^{d_x})$ be W_2-compact and let $\mathcal{G} : \mathcal{K} \to \mathcal{P}_2(\mathbb{R}^{d_y})$ be W_2-continuous. For every $\varepsilon > 0$, there exist $M, N \in \mathbb{N}$, functions $\psi_1, \dots, \psi_M \in C_b(\mathbb{R}^{d_x})$, fixed points $a_1, \dots, a_N \in \mathbb{R}^{d_y}$, and a continuous map $p = (p_1, \dots, p_N) : \mathbb{R}^{M+1} \to \Delta_N$, where $\Delta_N = \{q \in [0,1]^N : \sum_{j=1}^N q_j = 1\}$, such that*

$$\mathcal{N}_0(\mu) = \sum_{j=1}^{N} p_j(E(\mu))\delta_{a_j} \tag{12}$$

satisfies

$$\sup_{\mu \in \mathcal{K}} W_2(\mathcal{N}_0(\mu), \mathcal{G}(\mu)) < \varepsilon,$$

with E defined in (5)*.*

Proof of Theorem B.2. By uniform continuity of $\mathcal{G}$ and Lemma B.1, choose a finite feature map E and radius r so that nearby features produce outputs within $\varepsilon/4$. Cover the compact set $E(\mathcal{K})$ by finitely many balls centered at $E(\mu_\ell)$. A continuous partition of unity α_ℓ subordinate to this cover may be obtained from normalized tent functions.

Finitely supported laws are dense in $\mathcal{P}_2(\mathbb{R}^{d_y})$, so choose $\widehat{\nu}_\ell$ with $W_2(\widehat{\nu}_\ell, \mathcal{G}(\mu_\ell)) < \varepsilon/4$. Collect the finitely many atoms of all $\widehat{\nu}_\ell$ into a common list $a_1, \dots, a_N$ and write their weights as $\beta_{\ell j}$. Define on $E(\mathcal{K})$

$$q_j(z) = \sum_{\ell} \alpha_\ell(z)\beta_{\ell j}, \qquad \mathcal{N}_0(\mu) = \sum_{j} q_j(E(\mu))\delta_{a_j}.$$

Mixing couplings for the active ℓ gives

$$W_2^2(\mathcal{G}(\mu), \mathcal{N}_0(\mu)) \leq \sum_{\ell} \alpha_\ell(E(\mu)) W_2^2(\mathcal{G}(\mu), \widehat{\nu}_\ell) < \varepsilon^2.$$

Finally extend the continuous map $q : E(\mathcal{K}) \to \Delta_N$ to all of $\mathbb{R}^{M+1}$ using the Dugundji extension theorem (the simplex is convex). □

B.2 Neural realization and proof of Theorem 2.1

Lemma B.3 (Uniform neural approximation of the features). *Let $\mathfrak{K} \subset \mathcal{P}_2(\mathbb{R}^d)$ be tight and satisfy* (4)*. For finitely many $\psi_i \in C_b(\mathbb{R}^d)$ and every $\rho > 0$, bounded neural networks $\widehat{\psi}_i$ and a bounded neural network $\widehat{q}_R$ exist such that*

$$\sup_{\mu \in \mathfrak{K}} \left| \int (\widehat{\psi}_i - \psi_i) d\mu \right| < \rho, \qquad \sup_{\mu \in \mathfrak{K}} \left| \int \widehat{q}_R d\mu - m_2(\mu) \right| < \rho.$$

Proof. For a bounded test, choose a ball carrying uniformly high probability, approximate the test uniformly there, and clip the network to a fixed bound; tightness controls the complement. For $q(x) = \|x\|_2^2$, choose R so that the tail integral in (4) is small, multiply q by a continuous cutoff supported in the ball of radius $2R$, and uniformly approximate and clip that compactly supported function. Its integral differs from m_2 only by the approximation error and the uniformly small tail. □

Lemma B.4 (Common-support stability). *For $a_1, \dots, a_N \in \mathbb{R}^d$, $w, v \in \Delta_N$, and $D = \max_{j,k} \|a_j - a_k\|_2$,*

$$W_2^2\left(\sum_j w_j \delta_{a_j}, \sum_j v_j \delta_{a_j}\right) \le \frac{D^2}{2}\|w - v\|_1.$$

Proof. Match the mass common to w_j and v_j at zero cost and transport the remaining mass, whose total is $\|w - v\|_1/2$, across distances at most D. □

The same neural realization argument applies to a tight input family with uniformly integrable second moments whenever the continuous finite-support construction is uniform on that family.

Proof of Theorem 2.1. Apply Theorem B.2 with error $\varepsilon/2$. Lemma B.3 uniformly approximates its feature vector. The resulting feature vectors lie in a compact neighborhood of $E(\mathcal{K})$. Uniform continuity of p makes the induced weight error small. To realize p by softmax, use the positive regularization

$$p_j^\tau = \frac{p_j + \tau}{1 + N\tau}.$$

Approximate the continuous logits $\log p_j^\tau$ uniformly on that compact neighborhood, and apply softmax. Lemma B.4 converts the final weight error to W_2 error. □

B.3 Uniform empirical-feature bound

Proposition B.5 (Uniform empirical-feature error bound). *Fix any neural realization from Theorem 2.1, and write $f_i = \widehat{\psi}_i$ for $i \le M$ and $f_{M+1} = \widehat{q}_R$. Define the realized feature ranges*

$$\ell_i = \inf_{x \in \mathbb{R}^{d_x}} f_i(x), \qquad u_i = \sup_{x \in \mathbb{R}^{d_x}} f_i(x), \qquad s_i = u_i - \ell_i,$$

the feature box $\mathcal{Z} = \prod_{i=1}^{M+1}[\ell_i, u_i]$, and the modulus of continuity of the output network on this box by

$$\omega_{\widehat{p},\mathcal{Z}}(t) = \sup_{\substack{z, z' \in \mathcal{Z} \\ \|z - z'\|_\infty \le t}} \|\widehat{p}(z) - \widehat{p}(z')\|_1.$$

Let

$$\eta_\theta = \sup_{\mu \in \mathcal{K}} W_2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)), \qquad D = \max_{j,k} \|a_j - a_k\|_2.$$

If $X_1, \dots, X_{N_{\mathrm{in}}} \overset{\text{i.i.d.}}{\sim} \mu$ and the feature integrals in (6) *are replaced by empirical averages, then, for every $t > 0$,*

$$\sup_{\mu \in \mathcal{K}} \Pr\left(W_2(\widehat{\mathcal{N}}_{\theta,N_{\mathrm{in}}}(\mu), \mathcal{G}(\mu)) > \eta_\theta + D\sqrt{\frac{\omega_{\widehat{p},\mathcal{Z}}(t)}{2}}\right) \le 2 \sum_{\substack{1 \le i \le M+1 \\ s_i > 0}} \exp\left(-\frac{2N_{\mathrm{in}}t^2}{s_i^2}\right). \tag{13}$$

In particular, with $s_{\max} = \max_i s_i$ and

$$t_{N_{\mathrm{in}},\delta} = s_{\max}\sqrt{\frac{\log(2(M+1)/\delta)}{2N_{\mathrm{in}}}},$$

the following bound holds with probability at least $1 - \delta$, uniformly in $\mu \in \mathcal{K}$:

$$\sup_{\mu \in \mathcal{K}} \Pr\left(W_2(\widehat{\mathcal{N}}_{\theta,N_{\mathrm{in}}}(\mu), \mathcal{G}(\mu)) > \eta_\theta + D\sqrt{\frac{\omega_{\widehat{p},\mathcal{Z}}(t_{N_{\mathrm{in}},\delta})}{2}}\right) \le \delta. \tag{14}$$

If $\widehat{p}$ is L_p-Lipschitz from ℓ_∞ to ℓ_1 on $\mathcal{Z}$, the second term in (14) *is at most*

$$D\sqrt{\frac{L_p s_{\max}}{2}}\left(\frac{\log(2(M+1)/\delta)}{2N_{\text{in}}}\right)^{1/4}.$$

Consequently, for every $\varepsilon > 0$ and $\delta \in (0,1)$ the networks and N_{in} can be chosen so that

$$\sup_{\mu\in\mathcal{K}} \Pr\Big(W_2(\widehat{\mathcal{N}}_{\theta,N_{\text{in}}}(\mu), \mathcal{G}(\mu)) > \varepsilon\Big) < \delta.$$

Proof of Proposition B.5. For $\mu \in \mathcal{K}$, set

$$z(\mu) = \left(\int f_1\, d\mu, \ldots, \int f_{M+1}\, d\mu\right), \qquad z_{N_{\text{in}}}(\mu) = \frac{1}{N_{\text{in}}}\sum_{b=1}^{N_{\text{in}}} (f_1(X_b), \ldots, f_{M+1}(X_b)).$$

Both vectors belong to $\mathcal{Z}$. For every coordinate with $s_i > 0$, Hoeffding's inequality gives

$$\Pr(|z_{N_{\text{in}},i}(\mu) - z_i(\mu)| > t) \le 2\exp\left(-\frac{2N_{\text{in}}t^2}{s_i^2}\right).$$

Coordinates with $s_i = 0$ have zero empirical error. Hence a union bound, whose right-hand side is independent of μ, yields

$$\sup_{\mu\in\mathcal{K}} \Pr(\|z_{N_{\text{in}}}(\mu) - z(\mu)\|_\infty > t) \le 2\sum_{\substack{1\le i\le M+1\\ s_i>0}} \exp\left(-\frac{2N_{\text{in}}t^2}{s_i^2}\right). \tag{15}$$

On the complementary event, the definition of the modulus gives

$$\|\widehat{p}(z_{N_{\text{in}}}(\mu)) - \widehat{p}(z(\mu))\|_1 \le \omega_{\widehat{p},\mathcal{Z}}(t).$$

Applying Lemma B.4 to the empirical- and population-feature outputs therefore gives

$$W_2(\widehat{\mathcal{N}}_{\theta,N_{\text{in}}}(\mu), \widehat{\mathcal{N}}_\theta(\mu)) \le D\sqrt{\frac{\omega_{\widehat{p},\mathcal{Z}}(t)}{2}}.$$

The triangle inequality and the definition of η_θ, combined with (15), prove (13).

For $t = t_{N_{\text{in}},\delta}$, each nonzero-range summand on the right of (13) is at most $\delta/(M+1)$ after including its leading factor 2, so their sum is at most δ. This proves (14). If $\widehat{p}$ is L_p-Lipschitz, then $\omega_{\widehat{p},\mathcal{Z}}(t) \le L_p t$, which gives the displayed Lipschitz specialization.

Finally, apply (14) with confidence level $\delta/2$. Theorem 2.1 permits choosing $\eta_\theta < \varepsilon/2$. Since $\mathcal{Z}$ is compact and $\widehat{p}$ is continuous, $\omega_{\widehat{p},\mathcal{Z}}(t) \to 0$ as $t \downarrow 0$. Taking N_{in} sufficiently large makes the second term in (14) smaller than $\varepsilon/2$, proving the last assertion. □

B.4 Hilbert-space reconstruction and proof of Theorem 2.2

Lemma B.6 (Lipschitz pushforward and reconstruction)**.** *If $F : \mathcal{X} \to \mathcal{Y}$ is Lipschitz, then*

$$W_2^{\mathcal{Y}}(F_\#\mu, F_\#\nu) \le \text{Lip}(F) W_2^{\mathcal{X}}(\mu,\nu).$$

Moreover, for a measurable $P : \mathcal{X} \to \mathcal{X}$,

$$W_2^{\mathcal{X}}(P_\#\mu, \mu) \le \left(\int \|Px - x\|_{\mathcal{X}}^2 d\mu(x)\right)^{1/2}.$$

Proof. Push a coupling of μ, ν through (F, F) for the first inequality. For the second, push μ through $x \mapsto (Px, x)$. □

Lemma B.7 (Uniform Wasserstein autoencoding in a separable Hilbert space)**.** *Let $\mathcal{X}$ be a separable Hilbert space and let $X_1 \subset X_2 \subset \cdots \subset \mathcal{X}$ be finite-dimensional subspaces whose union is dense. Write $d_n = \dim X_n$, choose an orthonormal basis $e_1^n, \ldots, e_{d_n}^n$ of X_n, and define*

$$\mathcal{E}_{\mathcal{X}}^n x = \big(\langle x, e_1^n\rangle_{\mathcal{X}}, \ldots, \langle x, e_{d_n}^n\rangle_{\mathcal{X}}\big),$$
$$\mathcal{D}_{\mathcal{X}}^n z = \sum_{j=1}^{d_n} z_j e_j^n, \qquad P_{\mathcal{X}}^n = \mathcal{D}_{\mathcal{X}}^n \circ \mathcal{E}_{\mathcal{X}}^n.$$

Then $\mathcal{E}_{\mathcal{X}}^n$ is a contraction, $\mathcal{D}_{\mathcal{X}}^n$ is an isometry, and $P_{\mathcal{X}}^n$ is the orthogonal projection onto X_n. Moreover, every $W_2^{\mathcal{X}}$-compact set $\mathcal{K} \subset \mathcal{P}_2(\mathcal{X})$ satisfies

$$\lim_{n\to\infty} \sup_{\mu\in\mathcal{K}} W_2^{\mathcal{X}}\big((P_{\mathcal{X}}^n)_{\#}\mu, \mu\big) = 0. \tag{16}$$

Proof. The norm claims and the identity of $P_{\mathcal{X}}^n$ follow from orthonormality. For $\mu \in \mathcal{P}_2(\mathcal{X})$, set

$$r_n(\mu) = \left(\int_{\mathcal{X}} \|(I - P_{\mathcal{X}}^n)x\|_{\mathcal{X}}^2 \, d\mu(x)\right)^{1/2}.$$

The coupling $x \mapsto (P_{\mathcal{X}}^n x, x)$ and Lemma B.6 give $W_2^{\mathcal{X}}((P_{\mathcal{X}}^n)_{\#}\mu, \mu) \le r_n(\mu)$. Because $P_{\mathcal{X}}^n x \to x$ and $\|(I - P_{\mathcal{X}}^n)x\|_{\mathcal{X}} \le \|x\|_{\mathcal{X}}$, dominated convergence gives $r_n(\mu) \to 0$ for each μ.

The convergence is uniform on $\mathcal{K}$. Indeed, for any coupling π of μ and ν, the reverse triangle inequality in $L^2(\pi; \mathcal{X})$ and the contractivity of $I - P_{\mathcal{X}}^n$ yield

$$|r_n(\mu) - r_n(\nu)| \le \left(\int \|(I - P_{\mathcal{X}}^n)(x - y)\|_{\mathcal{X}}^2 \, d\pi(x, y)\right)^{1/2} \le \left(\int \|x - y\|_{\mathcal{X}}^2 \, d\pi(x, y)\right)^{1/2}.$$

Taking the infimum over π shows that every r_n is 1-Lipschitz on $\mathcal{P}_2(\mathcal{X})$. Pointwise convergence together with this common Lipschitz bound and compactness of $\mathcal{K}$ implies $\sup_{\mu\in\mathcal{K}} r_n(\mu) \to 0$, proving (16). □

Proof of Theorem 2.2. Write $P_{\mathcal{X}}^n = \mathcal{D}_{\mathcal{X}}^n \circ \mathcal{E}_{\mathcal{X}}^n$ and define $P_{\mathcal{Y}}^m$ similarly. Lemma B.7 gives

$$\sup_{\mu\in\mathcal{K}} W_2^{\mathcal{X}}\big((P_{\mathcal{X}}^n)_{\#}\mu, \mu\big) \longrightarrow 0. \tag{17}$$

Since $\mathcal{K}$ is compact and $\mathcal{U}$ is an open neighborhood of $\mathcal{K}$, (17) implies that $(P_{\mathcal{X}}^n)_{\#}\mu \in \mathcal{U}$ for every $\mu \in \mathcal{K}$ and all sufficiently large n. Continuity of $\mathcal{G}$ then implies

$$\sup_{\mu\in\mathcal{K}} W_2^{\mathcal{Y}}\big(\mathcal{G}((P_{\mathcal{X}}^n)_{\#}\mu), \mathcal{G}(\mu)\big) \longrightarrow 0. \tag{18}$$

To see the uniformity in (18), otherwise choose $n_k \to \infty$ and $\mu_k \in \mathcal{K}$ violating it. After passing to a subsequence, $\mu_k \to \mu \in \mathcal{K}$ in $W_2^{\mathcal{X}}$; then (17) also gives $(P_{\mathcal{X}}^{n_k})_{\#}\mu_k \to \mu$, contradicting continuity of $\mathcal{G}$ at μ. Fix n so that the left-hand side of (18) is less than $\varepsilon/3$.

The family

$$\mathfrak{L}_n = \{\mathcal{G}((P_{\mathcal{X}}^n)_{\#}\mu) : \mu \in \mathcal{K}\} \subset \mathcal{P}_2(\mathcal{Y})$$

is compact. Applying Lemma B.7 in $\mathcal{Y}$, choose m such that

$$\sup_{\nu\in\mathfrak{L}_n} W_2^{\mathcal{Y}}((P_{\mathcal{Y}}^m)_{\#}\nu, \nu) < \varepsilon/3. \tag{19}$$

The encoded input family $\mathcal{K}_n = (\mathcal{E}_{\mathcal{X}}^n)_{\#}\mathcal{K}$ is compact. On $\mathcal{K}_n$ define

$$\mathcal{S}_{n,m}(\lambda) = (\mathcal{E}_{\mathcal{Y}}^m)_{\#}\mathcal{G}((\mathcal{D}_{\mathcal{X}}^n)_{\#}\lambda).$$

Both pushforwards are Lipschitz by Lemma B.6, so $\mathcal{S}_{n,m}$ is a continuous operator between finite-dimensional law spaces. Theorem 2.1 therefore provides $\widehat{\mathcal{N}}_\theta$ such that

$$\sup_{\lambda\in\mathcal{K}_n} W_2\big(\widehat{\mathcal{N}}_\theta(\lambda), \mathcal{S}_{n,m}(\lambda)\big) < \varepsilon/3. \tag{20}$$

For $\mu \in \mathcal{K}$, put $\lambda = (\mathcal{E}_{\mathcal{X}}^n)_{\#}\mu$ and $\nu = \mathcal{G}((P_{\mathcal{X}}^n)_{\#}\mu)$. Since $\mathcal{D}_{\mathcal{Y}}^m$ is an isometry, Lemma B.6, (20), (19), and (18) give

$$\begin{aligned} W_2^{\mathcal{Y}}(\mathcal{N}_{\theta,n,m}(\mu), \mathcal{G}(\mu)) &\le W_2^{\mathcal{Y}}\big((\mathcal{D}_{\mathcal{Y}}^m)_{\#}\widehat{\mathcal{N}}_\theta(\lambda), (P_{\mathcal{Y}}^m)_{\#}\nu\big) \\ &\quad + W_2^{\mathcal{Y}}((P_{\mathcal{Y}}^m)_{\#}\nu, \nu) + W_2^{\mathcal{Y}}(\nu, \mathcal{G}(\mu)) < \varepsilon. \end{aligned}$$

Taking the supremum over $\mu \in \mathcal{K}$ proves the claim. □

B.5 Additional path-law construction details

B.5.1 Distributional Approximation in Wasserstein Space

From a distributional perspective, a random-field operator does not act directly on a representation of random variables defined on a fixed probability space; instead, it acts on probability measures over path spaces. Following the notation introduced earlier, let $\mathcal{P}_2(\mathcal{X})$ denote the space of probability measures on $\mathcal{X}$ with finite second moments, and let $W_2^{\mathcal{X}}$ denote the Wasserstein distance induced by the norm on $\mathcal{X}$. For a measurable map $F : \mathcal{X} \to \mathcal{Y}$, write $F_{\#}\mu$ for the pushforward measure.

In the distributional formulation, a random-field operator is expressed as a map between probability measures on path spaces. Let $\mathcal{X}$ and $\mathcal{Y}$ be the path spaces of the input and output random fields, respectively. We consider a continuous operator of the form

$$\mathcal{G} : \mathcal{U} \subset \mathcal{P}_2(\mathcal{X}) \longrightarrow \mathcal{P}_2(\mathcal{Y}), \tag{21}$$

where continuity is understood with respect to $W_2^{\mathcal{X}}$ and $W_2^{\mathcal{Y}}$ on the domain and codomain, respectively. Given a $W_2^{\mathcal{X}}$-compact set $\mathcal{K} \subset \mathcal{U}$, our goal is to construct a random-field distributional operator approximator $\mathcal{N}_{\theta,n,m}$ such that, for every $\varepsilon > 0$,

$$\sup_{\mu \in \mathcal{K}} W_2^{\mathcal{Y}}(\mathcal{N}_{\theta,n,m}(\mu), \mathcal{G}(\mu)) < \varepsilon. \tag{22}$$

The domain is denoted by $\mathcal{U}$ because the subsequent proof must compare not only $\mathcal{G}(\mu)$ with the output of the approximating operator, but also $\mathcal{G}(\mu)$ with the output associated with the input distribution reconstructed by the input autoencoder. The target operator must therefore be defined and continuous on a set containing both the original family of input distributions and its family of reconstructed distributions.

To apply the finite-dimensional distributional operator approximation result to random-field distributions on path spaces, we use an encoding–finite-dimensional distributional operator–decoding architecture. Let

$$\mathcal{E}_{\mathcal{X}}^n : \mathcal{X} \to \mathbb{R}^n, \qquad \mathcal{D}_{\mathcal{X}}^n : \mathbb{R}^n \to \mathcal{X}, \tag{23}$$

$$\mathcal{E}_{\mathcal{Y}}^m : \mathcal{Y} \to \mathbb{R}^m, \qquad \mathcal{D}_{\mathcal{Y}}^m : \mathbb{R}^m \to \mathcal{Y} \tag{24}$$

be the encoders and decoders for the input and output path spaces, respectively. Define the corresponding path-space reconstruction operators by

$$P_{\mathcal{X}}^n = \mathcal{D}_{\mathcal{X}}^n \circ \mathcal{E}_{\mathcal{X}}^n, \qquad P_{\mathcal{Y}}^m = \mathcal{D}_{\mathcal{Y}}^m \circ \mathcal{E}_{\mathcal{Y}}^m. \tag{25}$$

If X is an $\mathcal{X}$-valued random field with $\mu = \mathcal{L}(X)$, then the encoded random variable is $\mathcal{E}_{\mathcal{X}}^n(X) \in \mathbb{R}^n$, whose law is

$$(\mathcal{E}_{\mathcal{X}}^n)_{\#}\mu = \mathcal{L}(\mathcal{E}_{\mathcal{X}}^n(X)) \in \mathcal{P}_2(\mathbb{R}^n). \tag{26}$$

Thus, finite-dimensional encoding does not turn the random-field distribution μ into a single deterministic vector. Rather, it pushes the path-space random variable forward to a finite-dimensional random vector while preserving its probability distribution.

For fixed n and m, the finite-dimensional distributional operator approximation theory developed above provides an abstract finite-dimensional neural operator module

$$\widehat{\mathcal{N}}_\theta : \mathcal{P}_2(\mathbb{R}^n) \longrightarrow \mathcal{P}_2(\mathbb{R}^m). \tag{27}$$

The random-field distributional operator approximator induced by this module is defined by

$$\mathcal{N}_{\theta,n,m} = (\mathcal{D}_{\mathcal{Y}}^m)_{\#} \circ \widehat{\mathcal{N}}_\theta \circ (\mathcal{E}_{\mathcal{X}}^n)_{\#}. \tag{28}$$

This architecture can be represented by the following commutative diagram:

$$\begin{array}{ccc} \mathcal{P}_2(\mathcal{X}) & \xrightarrow{\mathcal{G}} & \mathcal{P}_2(\mathcal{Y}) \\ {\scriptstyle (\mathcal{E}_{\mathcal{X}}^n)_{\#}}\big\downarrow & & \big\uparrow{\scriptstyle (\mathcal{D}_{\mathcal{Y}}^m)_{\#}} \\ \mathcal{P}_2(\mathbb{R}^n) & \xrightarrow{\widehat{\mathcal{N}}_\theta} & \mathcal{P}_2(\mathbb{R}^m). \end{array} \tag{29}$$

In diagram (29), the upper arrow is the target random-field distributional operator $\mathcal{G}$; the left arrow pushes probability measures on the input path space forward to a finite-dimensional distribution space; the lower arrow is the neural operator module between finite-dimensional distribution spaces; and the right arrow decodes the finite-dimensional output distribution back to the output path space. The central question can therefore be stated as follows: under what conditions does the approximating path formed by the left, lower, and right arrows approximate the target operator $\mathcal{G}$ uniformly in $W_2^{\mathcal{Y}}$?

The proof strategy can be understood by examining the sources of error in the commutative diagram. First, on the input side, we must control the information loss caused by finite-dimensional encoding. In other words, after the original input distribution μ has been encoded and decoded, the resulting reconstructed distribution should be close to the original distribution in $W_2^{\mathcal{X}}$. If this error can be controlled uniformly over the family of input distributions under consideration, and if the target operator $\mathcal{G}$ is continuous with respect to Wasserstein distance, then the finite-dimensional reduction on the input side cannot produce an uncontrollable error in the output distribution.

Second, once the input has been encoded, the problem becomes one of operator approximation between finite-dimensional distribution spaces. This part is not developed again here; it is supplied by the finite-dimensional distributional operator approximation theory established above and is denoted uniformly by $\widehat{\mathcal{N}}_\theta$. Its role is to approximate the finite-dimensional distributional map induced by the original operator $\mathcal{G}$ between the input and output encoding spaces, namely, the lower arrow in the commutative diagram.

Finally, on the output side, we must control the error introduced when the finite-dimensional output distribution is decoded back to path space. Even if the finite-dimensional distributional module gives a good approximation in $\mathcal{P}_2(\mathbb{R}^m)$, the output decoder must still recover the target output distribution stably in $W_2^{\mathcal{Y}}$. The output path space must therefore possess a corresponding uniform autoencoding property in the Wasserstein sense.

In summary, the approximation error of a random-field distributional operator can be divided into three components: the input-side autoencoding error, the finite-dimensional distributional operator approximation error, and the output-side autoencoding error. The main theorem that follows is organized precisely around this decomposition. It first controls the reconstruction errors on the input and output sides through finite-dimensional autoencoding structures on the path spaces, then invokes the finite-dimensional distributional operator approximation result established above to control the error along the lower arrow of the commutative diagram, and finally obtains uniform Wasserstein approximation of $\mathcal{G}$ by $\mathcal{N}_{\theta,n,m}$.

It follows that the key to the subsequent theory is not to reconstruct the finite-dimensional distributional operator, but to prove that the three errors above can be controlled simultaneously in the Wasserstein sense. This formulation does not require the input and output random fields to be paired pointwise on the same event space. It is therefore better suited to learning problems in which random-field distributions are accessed only through sample sets or empirical distributions. The following theory will make this error decomposition based on diagram (29) rigorous.

C Supplementary Theory Proof For Experiments

C.1 Structured Output Representations and Neural Realizations

General finite-support output measures are expressive, but their support size can grow rapidly with the covering number of the input set and the complexity of local output laws. Practical problems often provide additional output structure. We consider common compact support, finite-dimensional parametric families, generative pushforwards, and conditional densities.

Lemma C.1 (Finite-dimensional factorization)**.** *Let $\mathcal{K} \subset \mathcal{P}_2(\mathbb{R}^{d_x})$ be W_2-compact, let $\mathcal{Y}$ be a Banach space, and let $F : \mathcal{K} \to \mathcal{Y}$ be continuous. For every $\varepsilon > 0$, there are a finite-dimensional encoder $E : \mathcal{K} \to \mathbb{R}^M$ and a continuous map $H : \mathbb{R}^M \to \mathcal{Y}$ such that*

$$\sup_{\mu \in \mathcal{K}} \|H(E(\mu)) - F(\mu)\|_{\mathcal{Y}} < \varepsilon.$$

If $F(\mathcal{K})$ lies in a convex set $\mathcal{Y}_0$, then $H(E(\mathcal{K}))$ may be required to lie in $\mathcal{Y}_0$.

Proof. Use Lemma B.1 and a finite cover of $E(\mathcal{K})$ with a partition of unity α_ℓ. For centers μ_ℓ, define

$$H(z) = \sum_{\ell=1}^{L} \alpha_\ell(z) F(\mu_\ell).$$

Uniform continuity controls the error. This is a convex combination, so it preserves $\mathcal{Y}_0$. Extend H from the compact feature set by the Tietze–Dugundji theorem. □

C.1.1 Common Support on a Compact Set

Proposition C.2 (Shared support points)**.** *Suppose the assumptions of Theorem B.2 hold and there is a compact $D \subset \mathbb{R}^{d_y}$ such that $\operatorname{supp} \mathcal{G}(\mu) \subset D$ for every $\mu \in \mathcal{K}$. For every $\varepsilon > 0$, there are common points $a_1, \dots, a_N \in D$, an encoder E, and continuous simplex-valued weights p_j such that*

$$\mathcal{N}(\mu) = \sum_{j=1}^{N} p_j(E(\mu)) \delta_{a_j}, \qquad \sup_{\mu \in \mathcal{K}} W_2(\mathcal{N}(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

The support size N may be bounded by the covering number of D at a scale proportional to ε.

Proof. Choose an r-net $\{a_j\}_{j=1}^N$ of D and a continuous partition of unity $\{\lambda_j\}$ subordinate to it. For any law ν on D, set

$$Q(\nu) = \sum_{j=1}^{N} \left(\int_D \lambda_j(y) d\nu(y) \right) \delta_{a_j}.$$

The coupling $\lambda_j(y)d\nu(y)$ gives $W_2(\nu, Q(\nu)) \le r$. The map $q(\mu) = (\int \lambda_1 d\mathcal{G}(\mu), \dots, \int \lambda_N d\mathcal{G}(\mu))$ is continuous. Apply Lemma C.1 to factor q through E and choose the weight error so that the common-support coupling bound is below $\varepsilon - r$. □

C.1.2 Finite-Dimensional Parametric Output Families

Let $\mathfrak{F} = \{\nu_a : a \in \Theta\} \subset \mathcal{P}_2(\mathbb{R}^{d_y})$ be a parametric family. Suppose $\mathcal{G}(\mu) = \nu_{A(\mu)}$ for a continuous parameter map $A : \mathcal{K} \to \Theta$.

Proposition C.3 (Parametric output family)**.** *Assume $a \mapsto \nu_a$ is W_2-continuous on a compact convex set $K_\Theta \subset \Theta$ containing $A(\mathcal{K})$. For every $\varepsilon > 0$, there are E and continuous $H : E(\mathcal{K}) \to K_\Theta$ such that*

$$\sup_{\mu \in \mathcal{K}} W_2(\nu_{H(E(\mu))}, \mathcal{G}(\mu)) < \varepsilon.$$

Proof. Uniform continuity of $a \mapsto \nu_a$ converts a sufficiently small parameter error into a W_2 error below ε. Apply Lemma C.1 to A while preserving the convex parameter set. □

Continuity of A is an additional assumption: continuity of $\mathcal{G}$ does not by itself imply a continuous parameter selection. Nonidentifiability, label switching in mixture models, or a discontinuous inverse parameterization can prevent such a selection. Legal parameters must also be enforced. For Gaussian outputs, a Cholesky factor or a positive transform of eigenvalues guarantees a positive semidefinite covariance.

Corollary C.4 (Neural realization of a parametric output family)**.** *Under Proposition C.3, suppose there is $r > 0$ such that*

$$C_r = \{a \in \mathbb{R}^p : \operatorname{dist}(a, A(\mathcal{K})) \le r\} \subset K_\Theta.$$

For every $\varepsilon > 0$, there are input-feature networks and a finite-dimensional network $\widehat{H}_\theta : \mathbb{R}^M \to \mathbb{R}^p$ such that

$$\widehat{\mathcal{N}}_\theta(\mu) = \nu_{\widehat{H}_\theta(\widehat{E}(\mu))}, \qquad \sup_{\mu \in \mathcal{K}} W_2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

C.1.3 Generative Pushforward Representation

Let $\rho \in \mathcal{P}_2(\mathbb{R}^{d_z})$ be a fixed latent law. Suppose there is a measurable map $G : \mathcal{K} \times \mathbb{R}^{d_z} \to \mathbb{R}^{d_y}$ with $\mathcal{G}(\mu) = G(\mu, \cdot)_{\#}\rho$.

Proposition C.5 (Generative pushforward)**.** *If $\mu \mapsto G(\mu, \cdot)$ is continuous from $(\mathcal{K}, W_2)$ to $L^2(\rho; \mathbb{R}^{d_y})$, then for every $\varepsilon > 0$ there are an encoder E and continuous $H : E(\mathcal{K}) \to L^2(\rho; \mathbb{R}^{d_y})$ such that*

$$\mathcal{N}_H(\mu) = H(E(\mu))_{\#}\rho, \qquad \sup_{\mu \in \mathcal{K}} W_2(\mathcal{N}_H(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

Proof. Apply Lemma C.1 in $L^2(\rho; \mathbb{R}^{d_y})$ with error ε. Coupling the two outputs by the same $Z \sim \rho$ gives

$$W_2^2(\mathcal{N}_H(\mu), \mathcal{G}(\mu)) \le \|H(E(\mu)) - G(\mu, \cdot)\|^2_{L^2(\rho)}.$$

□

The existence of a generator for each individual law does not imply a continuous choice of generators across a family. The pushforward map is many to one, and mode permutations or phase choices can make a generator selection discontinuous even when the laws vary continuously.

Polynomial chaos is one example. If $Z \sim \rho$ and $Y = f(Z) = \sum_{i=0}^{\infty} a_i \psi_i(Z)$ in $L^2(\rho)$, then the truncation $f_M(Z) = \sum_{i=0}^{M} a_i \psi_i(Z)$ satisfies

$$W_2^2(f_{\#}\rho, f_{M\#}\rho) \le \|f - f_M\|^2_{L^2(\rho)}.$$

For standard Gaussian ρ, the basis may be the Hermite polynomials. If the coefficients depend continuously on the input law, a network can predict them from finite distributional features.

Corollary C.6 (Neural-operator realization of a generator)**.** *Under Proposition C.5, for every $\varepsilon > 0$ there are input-feature networks and a deterministic neural operator $\widehat{H}_\theta : \mathbb{R}^M \to L^2(\rho; \mathbb{R}^{d_y})$ such that*

$$\widehat{\mathcal{N}}_\theta(\mu) = \widehat{H}_\theta(\widehat{E}(\mu))_{\#}\rho, \qquad \sup_{\mu \in \mathcal{K}} W_2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

C.1.4 Conditional Density Representation

Proposition C.7 (Existence of a conditional-density representation)**.** *Let $D \subset \mathbb{R}^{d_y}$ be compact with positive Lebesgue measure. Suppose every $\mathcal{G}(\mu)$ is supported on D and has density p_μ, and the map $P : \mathcal{K} \to L^1(D)$, $P(\mu) = p_\mu$, is continuous. For every $\varepsilon > 0$, there are E and a continuous map $H : E(\mathcal{K}) \to L^1(D)$ such that every $H(z)$ is a probability density and*

$$\mathcal{N}_H(\mu)(dy) = H(E(\mu))(y)dy, \qquad \sup_{\mu \in \mathcal{K}} W_2(\mathcal{N}_H(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

Proof. The density set

$$\mathcal{A}(D) = \left\{ p \in L^1(D) : p \ge 0 \text{ a.e.}, \int_D p(y)dy = 1 \right\}$$

is closed and convex. Apply Lemma C.1 to P in $L^1(D)$ while preserving $\mathcal{A}(D)$. For laws supported on D,

$$W_2^2(\mathcal{N}_H(\mu), \mathcal{G}(\mu)) \le \frac{\operatorname{diam}(D)^2}{2} \|H(E(\mu)) - p_\mu\|_{L^1(D)}.$$

□

A normalized exponential preserves positivity and unit mass:

$$\mathfrak{S}(h)(y) = \frac{\exp(h(y))}{\int_D \exp(h(\zeta))d\zeta}.$$

Alternatively, for a measurable partition $\{Q_j\}_{j=1}^J$ and simplex weights α_j, the bin density

$$p_\alpha(y) = \sum_{j=1}^{J} \frac{\alpha_j}{|Q_j|} \mathbf{1}_{Q_j}(y)$$

is automatically valid and converges in L^1 as the partition is refined.

Lemma C.8 (Stability of the normalized exponential)**.** *For compact D of positive Lebesgue measure and $h_1, h_2 \in C(D)$,*

$$\|\mathfrak{S}(h_1) - \mathfrak{S}(h_2)\|_{L^1(D)} \le 2\|h_1 - h_2\|_{C(D)}.$$

Proof. Set $r = h_1 - h_2$, $h_t = h_2 + tr$, and $p_t = \mathfrak{S}(h_t)$. Then

$$\partial_t p_t(y) = p_t(y) \left(r(y) - \int_D r(\zeta) p_t(\zeta) d\zeta \right),$$

so $\|\partial_t p_t\|_{L^1(D)} \le 2\|r\|_{C(D)}$. Integrating over $t \in [0,1]$ proves the result. □

Corollary C.9 (Neural realization of normalized-exponential densities)**.** *Under Proposition C.7, suppose a continuous energy map $A : \mathcal{K} \to C(D)$ satisfies*

$$p_\mu(y) = \frac{\exp(A(\mu)(y))}{\int_D \exp(A(\mu)(\zeta)) d\zeta}.$$

For every $\varepsilon > 0$, there are feature networks and a neural operator $\widehat{H}_\theta : \mathbb{R}^M \to C(D)$ such that

$$\widehat{p}_\theta(y \mid \mu) = \frac{\exp(\widehat{H}_\theta(\widehat{E}(\mu))(y))}{\int_D \exp(\widehat{H}_\theta(\widehat{E}(\mu))(\zeta)) d\zeta},$$
$$\widehat{\mathcal{N}}_\theta(\mu)(dy) = \widehat{p}_\theta(y \mid \mu) dy,$$

and

$$\sup_{\mu \in \mathcal{K}} W_2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)) < \varepsilon.$$

Proof. Approximate A after finite-dimensional factorization and input-feature approximation, obtaining

$$\sup_{\mu \in \mathcal{K}} \|\widehat{H}_\theta(\widehat{E}(\mu)) - A(\mu)\|_{C(D)} < \gamma.$$

Lemma C.8 gives an L^1 density error below 2γ, and therefore

$$W_2^2(\widehat{\mathcal{N}}_\theta(\mu), \mathcal{G}(\mu)) \le \operatorname{diam}(D)^2 \gamma.$$

Choose $\gamma < \varepsilon^2 / \operatorname{diam}(D)^2$. □

If p_μ is strictly positive and $\mu \mapsto \log p_\mu$ is continuous into $C(D)$, one may take $A(\mu) = \log p_\mu$. Densities with zeros require a positive regularization or bin-density approximation first.

D Data Generation and Experiments Protocol

D.1 OU First-Passage Time

D.1.1 Data Generation

The 1,200 laws are divided equally among three drift regimes:

subthreshold :	$m \sim \mathrm{Unif}(0.75, 0.95)$,	400 laws,
balanced :	$m \sim \mathrm{Unif}(0.95, 1.05)$,	400 laws,
suprathreshold :	$m \sim \mathrm{Unif}(1.05, 1.25)$,	400 laws.

The master seed is zero. Parameter sampling uses seed 101, and q is sampled independently with

$$\log q \sim \mathrm{Unif}(\log 0.1, \log 0.35).$$

Each law is assigned a unique identifier from 0 through 1199. For the target, 200 independent reset trials follow

$$dV_t = (-V_t + m)dt + \sqrt{q}\, dW_t,$$
$$V_0 = 0,$$

using random seed 303, step size 0.002, threshold 1, and terminal time 8. A crossing within a step is linearly interpolated; trials without a crossing by time 8 are censored. Uncensored times are assigned to 48 equal-width bins with edges $t_b = 8b/48$, and censored trials form category 49. Dividing the 49 category counts by 200 gives the target mass.

The input process file is generated from the same parameter table. For every law, 200 independent paths satisfy

$$dX_t = m\, dt + \sqrt{q}\, dW_t, \qquad X_0 = 0,$$

and are evaluated at $t_k = 8k/255$, $k = 0, \ldots, 255$. Independent Gaussian increments with variance $8/255$ give the exact transition law on this grid, and cumulative summation preserves the temporal correlation within each path. A deterministic stream generated by `SeedSequence([0, law_id])` makes every law reproducible independently of generation order. The resulting array has shape $1200 \times 200 \times 256$ and is stored in single precision. The input and target files are joined by their unique law identifiers rather than by row position, with matched parameters and regime labels verified before use.

A deterministic regime-stratified split with seed zero assigns 1,000 complete laws to training and 200 to testing; all paths from a law remain in the same split. There is no validation split, and the test laws are excluded from PCA fitting, optimization, scheduling, early stopping, and checkpoint selection.

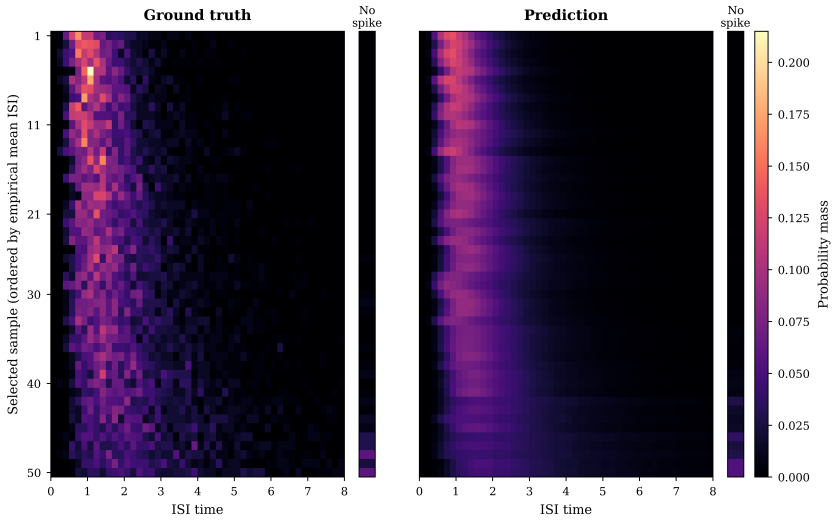

Figure 4: Predicted and empirical interspike-interval distributions. Ground-truth probability masses (left) and those predicted by the Process Distributional Operator (right) are shown for 50 test examples selected uniformly at random and ordered by increasing empirical mean interspike interval (ISI). Each heatmap row represents one input law, and the columns span the 48 finite ISI bins from 0 to 8. The narrow column beside each heatmap shows the probability of no threshold crossing before the end of the observation window. All panels use the same color scale.

D.1.2 Model and Baseline Implementations

For the Distributional Operator, one global PCA is fitted to the 200,000 training paths after flattening the law and path axes. The 256 time coordinates are centered using their training means but are not standardized coordinatewise. The mean and covariance are accumulated in double precision, and a symmetric eigendecomposition orders the components by decreasing eigenvalue. The smallest dimension whose cumulative explained-variance ratio is strictly greater than 0.99 is 14, with achieved

ratio 0.99015011. The fitted mean and components are then applied unchanged to both splits. The PCA state, threshold, split seed, and training-law identifiers are stored with the checkpoint so evaluation never refits the transform.

The resulting model input has shape $N_{\text{batch}} \times N_{\text{in}} \times 14$. Each compressed path passes through a shared $14 \to 32 \to 32$ SiLU path MLP. A DeepSets element network then maps $32 \to 32 \to 32$ with a SiLU hidden activation and no dropout, and averages the resulting 32-dimensional embeddings over the input-sample axis. Finally, a $32 \to 32 \to 49$ outer network with a GELU hidden activation returns the logits. The complete sequence of tensor shapes is

$$\begin{aligned}(N_{\text{batch}}, N_{\text{in}}, 256) &\xrightarrow{\text{PCA}} (N_{\text{batch}}, N_{\text{in}}, 14)\\ &\xrightarrow{\phi_{\text{path}}} (N_{\text{batch}}, N_{\text{in}}, 32)\\ &\xrightarrow{\text{DeepSets mean}} (N_{\text{batch}}, 32) \xrightarrow{\rho} (N_{\text{batch}}, 49).\end{aligned}$$

The neural model has 5,265 trainable parameters. Softmax returns the predicted bin masses and cumulative summation returns the predicted CDF. It is trained for 1,000 epochs with AdamW, learning rate 10^{-3}, weight decay 10^{-2}, mini-batches of 64 laws, no learning-rate scheduler, and no early stopping; the saved final checkpoint is evaluated once on the untouched test split.

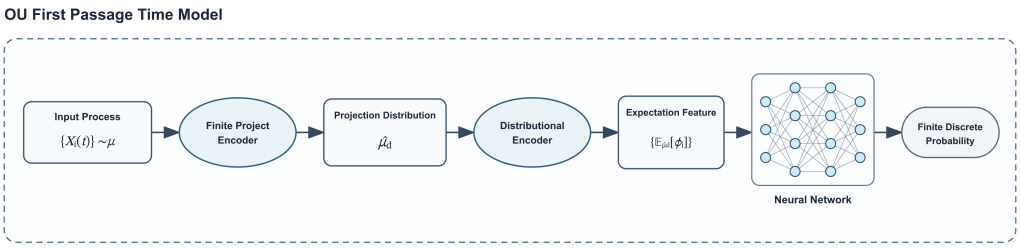

Figure 5: Distributional Operator used for the OU first-passage task. An empirical law of PCA-compressed input paths is encoded by shared path and DeepSets networks, mapped to categorical output-law parameters, and sampled when realizations of the predicted first-passage distribution are required.

The kernel baseline converts the input particles into a normalized 32-category histogram over the global training-particle range. For query X and training law X_i, let $d(X, X_i)$ be the exact one-dimensional quadratic Wasserstein distance between the piecewise-uniform histograms. It predicts

$$\widehat{p}(X) = \sum_i w_i(X) p_i, \qquad w_i(X) = \frac{\exp[-d(X, X_i)^2/(2h^2)]}{\sum_j \exp[-d(X, X_j)^2/(2h^2)]}, \qquad h = 0.15.$$

Self-matches are excluded for training-law evaluation, and distances are computed in blocks of 256 training laws.

The fixed-feature MLP replaces learned particle aggregation by the sample mean, population variance, and eight empirical sine and cosine moments at frequencies drawn once from $\mathcal{N}(0, 1)$. The resulting 18 features pass through a linear $18 \to 32$ map, GELU, and a $32 \to 49$ logit map. It is trained for 1,000 epochs with mini-batches of 64 laws using the count-weighted categorical NLL objective.

D.1.3 Evaluation Metrics

Five runs use mini-batches of 64 under the default training configuration.

For normalized target histogram p_i and prediction $\widehat{p}_i$, we report count-normalized NLL, Hellinger distance, empirical-to-prediction KL divergence, tail error, and finite-support W_2:

$$\mathrm{NLL}_i = -\sum_{k=1}^{49} p_{ik} \log \widehat{p}_{ik},$$

$$H(p_i, \widehat{p}_i) = \left[\frac{1}{2}\sum_{k=1}^{49}(\sqrt{p_{ik}} - \sqrt{\widehat{p}_{ik}})^2\right]^{1/2},$$

$$D_{\mathrm{KL}}(p_i \| \widehat{p}_i) = \sum_{k:p_{ik}>0} p_{ik} \log \frac{p_{ik}}{\widehat{p}_{ik}},$$

$$E_{\mathrm{tail}}(p_i, \widehat{p}_i) = |p_{i,49} - \widehat{p}_{i,49}|.$$

For finite-support W_2, category 49 is removed, the first 48 masses are renormalized, and the exact one-dimensional quadratic Wasserstein distance is computed between the associated piecewise-uniform laws. Metrics are averaged over the 200 test laws and then summarized across the five runs.

D.2 Nonlinear Duffing Oscillator

D.2.1 Data Generation

The Duffing coefficients are fixed throughout the dataset at linear stiffness $a = 1$, cubic stiffness $b = 1$, and damping $c = 0.2$. A forcing-law parameter is

$$\theta = (A, f, \sigma_\eta, \ell_\eta) \in [0.5, 2.0] \times [0.10, 0.30] \times [0.05, 0.40] \times [0.10, 1.00].$$

For each realization, independently sample $\Phi \sim \mathrm{Unif}[0, 2\pi]$ and a zero-mean Gaussian process η with squared-exponential covariance

$$K_\theta(s,t) = \sigma_\eta^2 \exp\left(-\frac{(s-t)^2}{2\ell_\eta^2}\right).$$

The resulting stochastic forcing path is

$$X(t) = A\sin(2\pi f t + \Phi) + \eta(t).$$

Thus $(A, f, \sigma_\eta, \ell_\eta)$ identifies a probability law, whereas the phase and Gaussian-process path are realization-level random variables and are resampled for every particle.

We use a Latin-hypercube design to choose 1,200 parameter vectors over the four-dimensional box above, providing space-filling coverage near potentially sharp response transitions. The split is performed at the measure level: 1,000 settings are used for training and 200 for testing. No realizations from one forcing law appear in multiple splits. An optional extrapolation benchmark may instead reserve boundary bands in A or f, but it is kept separate from the primary interpolation split.

Both forcing and displacement paths are stored at $N_t = 256$ equally spaced observation times

$$t_k = \frac{20k}{255}, \qquad k = 0, \ldots, 255.$$

The ODE is not solved directly on this coarse grid. Each forcing realization is generated on a fine grid with step size $\Delta t_{\mathrm{solve}} = 0.005$ and interpolated inside the first-order system

$$\dot{y}_1 = y_2, \qquad \dot{y}_2 = X(t) - 0.2y_2 - y_1 - y_1^3.$$

We integrate from zero initial conditions with adaptive Dormand–Prince RK45, using relative tolerance 10^{-8} and absolute tolerance 10^{-10}, and then sample the solution on the observation grid. Before full generation, a convergence study on representative parameter settings compares $\Delta t \in \{0.02, 0.01, 0.005, 0.0025\}$ and requires relative trajectory error below 10^{-4} against the finest reference.

For each parameter vector θ_i, let μ_i be the corresponding forcing-process law. Draw two independent ensembles,

$$X_{i,j} \overset{\mathrm{iid}}{\sim} \mu_i, \qquad X'_{i,j} \overset{\mathrm{iid}}{\sim} \mu_i, \qquad j = 1, \ldots, 128,$$

and propagate only the second ensemble through the fixed Duffing solution operator, $Y_{i,j} = \mathcal{S}(X'_{i,j})$. The empirical input and target laws are therefore

$$\widehat{\mu}_X^{(i)} = \frac{1}{128} \sum_{j=1}^{128} \delta_{X_{i,j}},$$

$$\widehat{\mu}_Y^{(i)} = \frac{1}{128} \sum_{j=1}^{128} \delta_{Y_{i,j}}.$$

This deliberately removes trajectory-level correspondence while retaining the shared measure parameter. The stored arrays X and Y each have shape $1,200 \times 128 \times 256$ in `measure,sample,time` order, and a $1,200 \times 4$ metadata array stores the parameter vectors for diagnostics and split construction, not as model input.

For the neural configurations, a fixed random permutation assigns the first 1,000 law indices to training and the remaining 200 to testing. The training loader is shuffled; the test loader is not shuffled. Mini-batches contain eight laws, use no worker processes, and contain all 128 particles from each selected law.

D.2.2 Distributional Operator Implementation

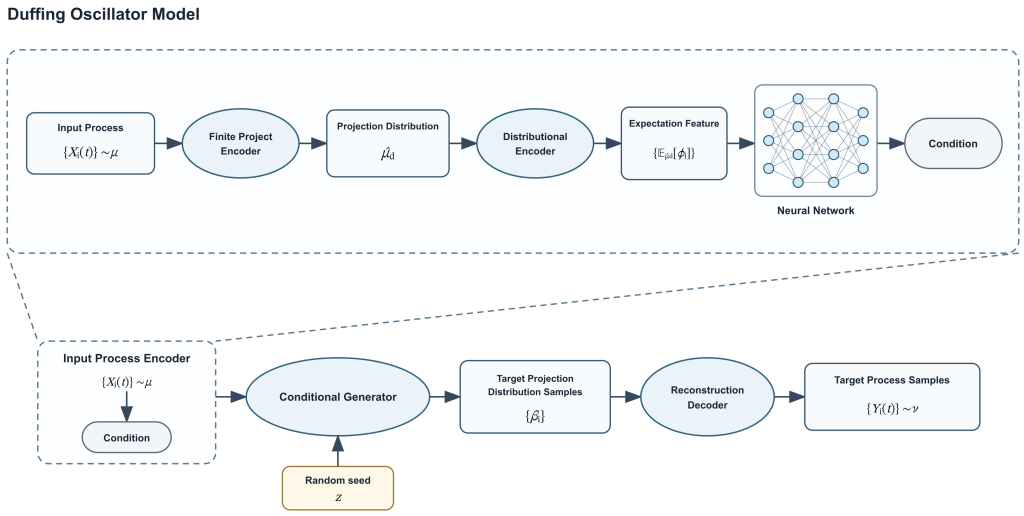

Figure 6: Process-valued Distributional Operator used for the Duffing task. Top: the input-process ensemble is projected to finite-dimensional coordinate laws and encoded into a condition through expectation features. Bottom: the condition and independent latent draws drive a conditional generator, whose projected output samples are reconstructed as response-process paths.

Separate PCA transforms are fitted to the pooled training forcing and response trajectories, never to test data. The 256-dimensional forcing and response grids are truncated to 24 and 13 principal coordinates, respectively. PCA fitting accumulates the training mean and scatter matrix in 64-law blocks in double precision and uses a symmetric eigendecomposition. Each retained coordinate is then standardized using its population mean and standard deviation over all trajectories in the training laws; test data reuse these stored transformations. Near-constant coordinates use unit scale. Inputs to the learned encoder are the normalized 24-dimensional forcing coefficients. Generated normalized 13-dimensional response coefficients are unstandardized and reconstructed on the original 256-point grid before the training loss and reported metrics are evaluated.

The DeepSets particle map and outer map each have two width-256 SiLU hidden layers. The particle embedding and law context both have dimension 128, and the aggregation is the sample mean.

The conditional RealNVP has six affine coupling layers with alternating even–odd masks. Each scale–translation conditioner takes the masked 13-dimensional coefficient vector together with the 128-dimensional context, uses two width-256 SiLU hidden layers, and has a zero-initialized output layer. Its log-scale is bounded as $\widetilde{s} = 2\tanh(s/2)$. The base law is $\mathcal{N}(0, I_{13})$, temperature is one, and 128 particles are generated for each input law.

Training minimizes the mean debiased GeomLoss Sinkhorn divergence with $p = 2$, blur 0.01, scaling 0.5, automatic backend selection, and no auxiliary mean loss. AdamW uses learning rate 10^{-3} and weight decay 10^{-3}; the global gradient norm is clipped to one. Training runs for at most 1,000 epochs.

D.2.3 Baseline Implementations

The Fixed-Feature MLP operates on the raw 256-point paths without PCA. A two-hidden-layer width-64 SiLU network encodes each forcing path into 64 features. A DeepSets module with two width-64 SiLU layers maps these features to 128-dimensional sample embeddings, concatenates their population mean and standard deviation (with variance stabilization 10^{-6}), and applies two more width-64 SiLU layers to produce a 64-dimensional law context. Given this context and 32-dimensional standard Gaussian noise at temperature one, a three-hidden-layer width-64 SiLU MLP generates a 32-dimensional output code; a two-hidden-layer width-64 SiLU decoder maps each code directly to a 256-point response path. It generates 128 paths per law and trains for 200 epochs with batches of eight laws. Its debiased Sinkhorn loss uses $p = 2$, blur 0.05, scaling 0.5, and automatic backend selection. AdamW uses learning rate 10^{-3} and weight decay 10^{-2}, with global gradient norm clipped to one.

The Nadaraya–Watson baseline computes Sinkhorn divergences d_k between a query forcing law and each of the 1,000 training laws, then uses $w_k \propto \exp[-d_k/(2h^2)]$. Its 64 predicted response particles are the elementwise weighted averages

$$\widehat{Y}_j = \sum_{k=1}^{1000} w_k Y_{k,j}, \qquad j = 1, \dots, 64,$$

of the stored target-particle arrays, rather than samples from a discrete mixture over training laws. The common train–test split and a fixed subset of 64 particles are used for every law. Input-law distances are debiased GeomLoss Sinkhorn divergences with $p = 2$, blur 0.05, scaling 0.5, the tensorized backend, four reference laws per distance chunk, and two query laws per batch. The bandwidth is fixed at $h = 0.005$, and the reported baseline is a single deterministic fit.

D.2.4 Evaluation Metrics

For a target empirical law $\nu = n^{-1}\sum_{i=1}^{n}\delta_{y_i}$ and a predicted empirical law $\widehat{\nu} = m^{-1}\sum_{j=1}^{m}\delta_{\widehat{y}_j}$, let $\Pi(\widehat{\nu}, \nu)$ denote their couplings. With ground cost c and regularization $\varepsilon > 0$, define

$$\mathrm{OT}_\varepsilon(\widehat{\nu}, \nu) = \inf_{\pi \in \Pi(\widehat{\nu}, \nu)} \left\{ \int c(x, y) d\pi(x, y) + \varepsilon\, \mathrm{KL}(\pi \| \widehat{\nu} \otimes \nu) \right\},$$
$$S_\varepsilon(\widehat{\nu}, \nu) = \mathrm{OT}_\varepsilon(\widehat{\nu}, \nu) - \tfrac{1}{2}\,\mathrm{OT}_\varepsilon(\widehat{\nu}, \widehat{\nu}) - \tfrac{1}{2}\,\mathrm{OT}_\varepsilon(\nu, \nu).$$

For a positive-definite kernel k, the biased empirical MMD and empirical energy distance are

$$\widehat{\mathrm{MMD}}_k = \left[\frac{1}{m^2}\sum_{j,j'} k(\widehat{y}_j, \widehat{y}_{j'}) + \frac{1}{n^2}\sum_{i,i'} k(y_i, y_{i'}) - \frac{2}{mn}\sum_{j,i} k(\widehat{y}_j, y_i) \right]^{1/2},$$
$$\widehat{\mathrm{ED}} = \frac{2}{mn}\sum_{j,i} \|\widehat{y}_j - y_i\|_2 - \frac{1}{m^2}\sum_{j,j'} \|\widehat{y}_j - \widehat{y}_{j'}\|_2 - \frac{1}{n^2}\sum_{i,i'} \|y_i - y_{i'}\|_2.$$

For $P_\theta(z) = \langle \theta, z \rangle$ and uniform measure σ on the unit sphere, sliced 2-Wasserstein distance is

$$\mathrm{SW}_2^2(\widehat{\nu}, \nu) = \int W_2^2((P_\theta)_\#\widehat{\nu}, (P_\theta)_\#\nu)\, d\sigma(\theta).$$

The model-specific training losses and schedules are given in Appendices D.2.2 and D.2.3. Final evaluation generates 128 predicted paths for each target law. We report the model's configured debiased Sinkhorn divergence, biased RBF MMD, sliced W_2, and empirical energy distance on the reconstructed 256-point paths. For MMD, a separate median-heuristic bandwidth is computed for each law from all prediction–target squared distances and the nonzero within-sample squared distances; the reported value is MMD, not MMD squared. Sliced W_2 uses 128 fixed unit-normalized Gaussian projections. Metrics are averaged over laws. The conditional-flow values in Table 2 are test-set mean $\pm$ sample standard deviation; kernel regression is evaluated once.

E Supplementary Experiments

The controlled Gaussian study below is a purely supplementary experiment. Its complete setup, results, and implementation details are reported here.

E.1 Controlled Gaussian Distributional Mapping

E.1.1 Problem Setup

This controlled task learns a map from four-dimensional Gaussian-mixture laws to four-dimensional Gaussian laws from unpaired ensembles of 200 input and 200 output particles. The target mean and covariance are nonlinear functions of analytic input-law moments and Fourier features, while the Distributional Operator sees only the empirical input law and predicts a full-covariance Gaussian. Complete data-generation, baseline, architecture, optimization, and metric definitions are given in Appendix E.1.4.

E.1.2 Model Instantiation

The Distributional Operator applies a shared particle encoder $\phi_\theta : \mathbb{R}^4 \to \mathbb{R}^{64}$ to each input sample and averages the resulting features. The pooled representation is processed by an outer network ρ_θ and a Gaussian output head:

$$
\begin{aligned}
h^{(b)} &= \frac{1}{200}\sum_{j=1}^{200} \phi_\theta(x_j^{(b)}),\\
(\widehat{\mu}^{(b)}, \widehat{u}^{(b)}) &= \rho_\theta(h^{(b)}),\\
\widehat{L}^{(b)} &= \mathcal{R}_{\mathrm{ltr}}(\widehat{u}^{(b)}), \qquad \widehat{\Sigma}^{(b)} = \widehat{L}^{(b)}\widehat{L}^{(b)\top}.
\end{aligned}
$$

The particle encoder has two width-64 GELU hidden layers, and the outer network has three width-128 GELU hidden layers. The Gaussian head outputs four mean coordinates and ten entries of a lower-triangular Cholesky factor. Softplus diagonal entries with an offset of 10^{-5} guarantee a positive-definite covariance. Thus the predicted law is $\mathcal{N}(\widehat{\mu}^{(b)}, \widehat{\Sigma}^{(b)})$ and is invariant to the ordering of the input particles.

E.1.3 Evaluation

Training minimizes empirical output-sample NLL. We additionally report the closed-form Gaussian W_2 distance, target-to-prediction KL divergence, and Hellinger distance, averaged over the 200 test laws. Full metric definitions and optimization details are in Appendix E.1.4.

We compare against the Fixed-Feature MLP and distribution-space kernel regression; see Appendix E.1.4 for their implementations.

Figure 7 compares the predicted and true mean components directly, while Figure 8 summarizes covariance error across all test distributions.

E.1.4 Additional Details

Data Generation For each of the 1,200 laws, the number of mixture components C_k is sampled uniformly from $\{1, 2, 3\}$. Independent unit-rate exponential variables are normalized to give mixture

Model	NLL	W_2	KL divergence	Hellinger distance
Distributional Operator	$\mathbf{4.016462 \pm 0.002379}$	$\mathbf{0.162049 \pm 0.004262}$	$\mathbf{0.048246 \pm 0.001804}$	$\mathbf{0.099313 \pm 0.002195}$
Fixed-Feature MLP	4.033581 ± 0.002687	0.187644 ± 0.003589	0.066297 ± 0.002362	0.112535 ± 0.002348
Kernel regression	4.776576 ± 0.000000	0.741837 ± 0.000000	0.806815 ± 0.000000	0.396992 ± 0.000000

Table 3: Test results for the controlled Gaussian mapping. Entries are the mean $\pm$ standard deviation over five runs. Kernel regression is deterministic and therefore has zero standard deviation. Lower is better for every metric.

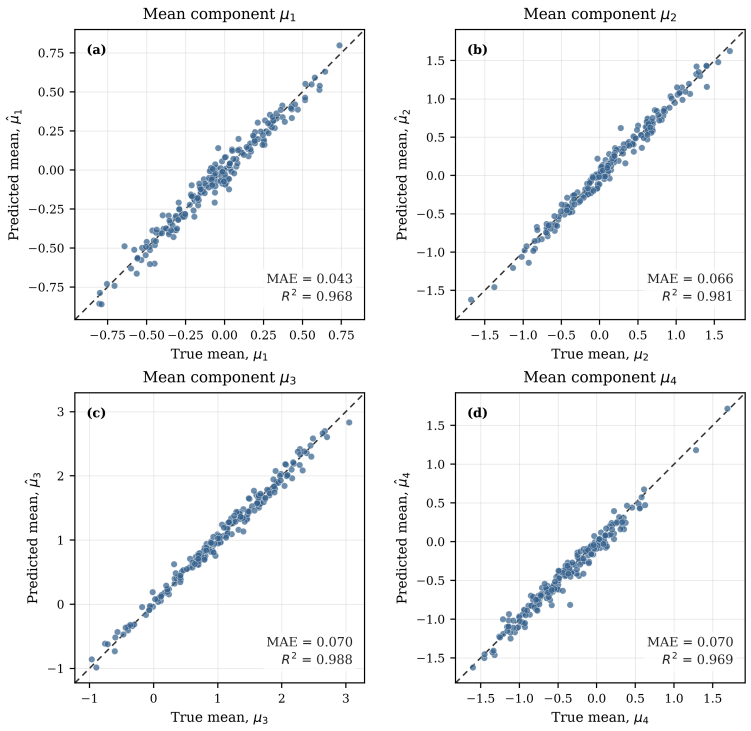

Figure 7: Predicted versus true output means on the test set for the trained Distributional Operator. Each panel shows one of the four mean components, with one point per test distribution; the dashed diagonal $y = x$ denotes perfect prediction.

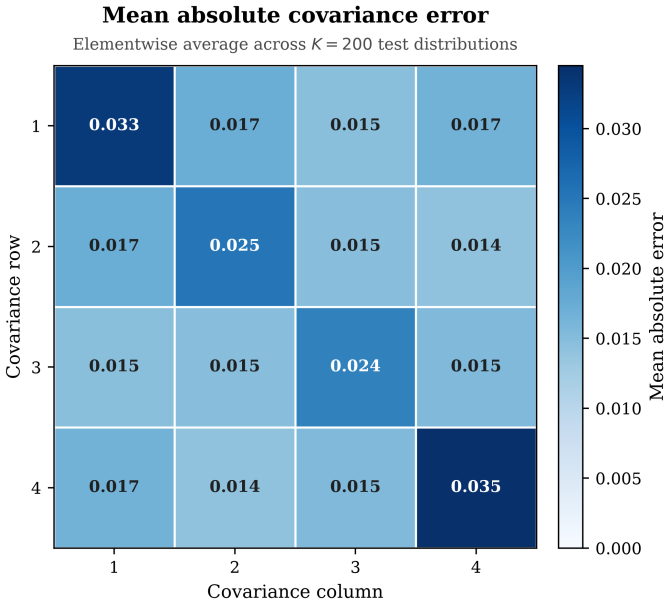

Figure 8: Elementwise mean absolute error of the predicted output covariance matrices on the test set for the trained Distributional Operator. For covariance entry (i, j), the displayed value is $N_{\text{test}}^{-1} \sum_{k=1}^{N_{\text{test}}} |\widehat{\Sigma}_{ij}^{(k)} - \Sigma_{ij}^{(k)}|$; darker errors identify components that are systematically more difficult to predict.

weights

$$\pi_{kc} = \frac{u_{kc}}{\sum_{c'=1}^{C_k} u_{kc'}}.$$

Component means and coordinatewise standard deviations are sampled independently as

$$m_{kcj} \sim \text{Unif}[-2, 2], \qquad s_{kcj} \sim \text{Unif}[0.15, 0.80],$$

giving the input law

$$P_k = \sum_{c=1}^{C_k} \pi_{kc} \mathcal{N}\big(m_{kc}, \text{diag}(s_{kc}^2)\big)\,.$$

We draw $S = 200$ independent input particles from each P_k.

The deterministic law feature $z_k \in \mathbb{R}^{24}$ concatenates the analytic mixture mean, elementwise variance, and eight sine and eight cosine Fourier features. Eight frequency vectors $\omega_r \sim \mathcal{N}(0, I_4)$ are sampled once and shared by the dataset. Specifically,

$$\begin{aligned}
\bar{m}_k &= \sum_c \pi_{kc} m_{kc}, \\
v_k &= \sum_c \pi_{kc}(s_{kc}^2 + m_{kc}^2) - \bar{m}_k^2, \\
a_{kr} &= \sum_c \pi_{kc} \exp\left(-\frac{1}{2}\sum_j s_{kcj}^2 \omega_{rj}^2\right) \sin(\omega_r^\top m_{kc}), \\
b_{kr} &= \sum_c \pi_{kc} \exp\left(-\frac{1}{2}\sum_j s_{kcj}^2 \omega_{rj}^2\right) \cos(\omega_r^\top m_{kc}), \\
z_k &= [\bar{m}_k, v_k, a_k, b_k].
\end{aligned}$$

One fixed random map is shared by all laws. Every entry of W_μ, W_d, W_o is sampled independently from $\mathcal{N}(0, 1/24)$. The output mean is $\mu_k = W_\mu z_k$. A lower-triangular matrix L_k is constructed by

$$(L_k)_{ii} = 0.25 + 0.75\{1 + \exp[-(W_d z_k)_i]\}^{-1},$$

and $(L_k)_{ij} = 0.2 \tanh((W_o z_k)_{ij})$ for $i > j$. Thus

$$\Sigma_k = L_k L_k^\top + 10^{-4} I_4, \qquad Q_k = \mathcal{N}(\mu_k, \Sigma_k).$$

Finally, 200 independent output particles are drawn from Q_k. The analytic targets support evaluation but are not used by the principal training loss.

Baseline Implementations The *Fixed-Feature MLP* computes the empirical mean and unbiased empirical covariance of each input ensemble, concatenates all four mean and sixteen covariance entries, and maps the resulting 20 features through four width-128 GELU layers. The same full-covariance Gaussian head used by the Distributional Operator returns the predicted mean and covariance. It uses the same split, objective, optimizer, scheduler, and evaluation procedure as the main model.

The parameter-free kernel estimator stores empirical input and output means and covariances for all 1,000 training laws. For a query, it computes Gaussian W_2 distances d_k between its empirical input moments and those of each training law, then assigns radial-basis weights

$$\alpha_k = \frac{\exp[-d_k^2/(2h^2)]}{\sum_{\ell=1}^{1000} \exp[-d_\ell^2/(2h^2)]}, \qquad h = 1.$$

It interpolates empirical output moments as

$$\widehat{\mu} = \sum_k \alpha_k \widetilde{m}_k^y, \qquad \widehat{\Sigma} = \sum_k \alpha_k \widetilde{C}_k^y + 10^{-5} I_4.$$

The covariance is symmetrized before stabilization and likelihood evaluation.

Evaluation Metrics A fixed random permutation defines the common train–test split. Five runs use mini-batches of 64 under the default training configuration.

For target $p = \mathcal{N}(\mu, \Sigma)$ and prediction $q = \mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$, we report empirical output-sample NLL, Gaussian W_2, target-to-prediction KL divergence, and Hellinger distance. The latter two are

$$D_{\mathrm{KL}}(p\|q) = \frac{1}{2}\left[\mathrm{tr}(\widehat{\Sigma}^{-1}\Sigma) + (\widehat{\mu}-\mu)^\top \widehat{\Sigma}^{-1}(\widehat{\mu}-\mu) - 4 + \log\frac{\det\widehat{\Sigma}}{\det\Sigma}\right],$$

$$H(p,q) = \sqrt{1 - \mathrm{BC}(p,q)},$$

$$\mathrm{BC}(p,q) = \frac{(\det\Sigma)^{1/4}(\det\widehat{\Sigma})^{1/4}}{\det((\Sigma+\widehat{\Sigma})/2)^{1/2}} \exp\left[-\frac{1}{8}(\mu-\widehat{\mu})^\top \left(\frac{\Sigma+\widehat{\Sigma}}{2}\right)^{-1}(\mu-\widehat{\mu})\right].$$

Each metric is averaged over the 200 test laws; NLL is additionally averaged over the 200 observed output particles per law.