CURVES IN CONTEXT: ZERO-SHOT REGRESSION OF FUNCTIONAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalar-on-function prediction—mapping an observed curve to a scalar response—is a central task of functional data analysis (FDA). Practice remains dominated by dataset-specific pipelines whose ingredients (basis, smoothing, model family, hyperparameters) must be re-selected for every new problem and are tied to a fixed observation grid. We introduce FunPFN, a foundation model for scalar-on-function regression. FunPFN is pretrained once, purely on synthetic episodes drawn from a generative prior over functional tasks, and then predicts on a new dataset in a single forward pass: the training set is supplied as an in-context support set and the model returns a Gaussian predictive distribution for every query curve, with no gradient updates and no hyperparameter tuning. Its encoder combines a Fourier neural operator (FNO) trunk and a support-standardized spectral readout, making predictions discretization-invariant by construction. Across ten simulation scenarios, FunPFN achieves the best average rank among fourteen methods (1.7), significantly ahead of all twelve tuned baselines (Friedman test, Holm post-hoc); the runner-up, TabPFN applied to the discretized curves (4.7), is not separated by this test, yet its normalized excess risk is 1.5 to 4.6 times that of FunPFN in every scenario. On twelve real benchmark tasks, FunPFN ranks second overall while operating over 200 times faster than TabPFN (0.025s vs. 5.983s).

1 INTRODUCTION

Many scientific measurements are curves rather than vectors: near-infrared spectra, growth trajectories, annual temperature profiles. Predicting a scalar response from such a functional covariate is *scalar-on-function* regression (Ramsay & Silverman, 2005; Ferraty & Vieu, 2006; Morris, 2015; Wang et al., 2016), a cornerstone of FDA. Its toolbox is rich: functional linear models on principal-component or penalized bases (Cardot et al., 1999; Yao et al., 2005; Cai & Hall, 2006; Hall & Horowitz, 2007; Goldsmith et al., 2011), functional partial least squares (Preda & Saporta, 2005; Reiss & Ogden, 2007; Delaigle & Hall, 2012b), index models with nonparametric links (Müller & Stadtmüller, 2005; Chen et al., 2011; Jiang & Wang, 2011) and fully nonparametric estimators (Ferraty & Vieu, 2006). Yet each new dataset calls for a bespoke pipeline: a representation, a model class matching the unknown form of the relationship, and hyperparameters tuned by cross-validation, often on only dozens to hundreds of curves. Such pipelines must be re-fitted whenever the observation grid changes, and most return point predictions without calibrated uncertainty.

A different route builds on neural processes (Garnelo et al., 2018a; Kim et al., 2019; Nguyen & Grover, 2022) and in-context learning (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023): *prior-data fitted network*s (PFNs), transformers pretrained on synthetic tasks drawn from a prior, take a training set as context and approximate the Bayesian posterior predictive in one forward pass (Müller et al., 2022; Nagler, 2023). On small tables, the resulting tabular prior-data fitted network (TabPFN) outperforms tuned pipelines at a fraction of their cost (Hollmann et al., 2023; 2025). No such model exists for curves, which are infinite-dimensional and observed on dataset-specific grids: the prior and the architecture must be designed over function space, not over a fixed feature vector.

We close this gap with FunPFN, to our knowledge the first foundation model for scalar-on-function regression. Pretrained only on synthetic data, its in-context transformer maps a support set of (curve,

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.

response) pairs to a Gaussian predictive for each query curve in one forward pass (Sections 3.1 and 3.2.2), with no fine-tuning, representation choice or hyperparameter selection. Our contributions:

- A resolution-agnostic function encoder—FNO trunk, mean and attention pooling, support-standardized spectral readout—whose output converges to a functional of the curve as the grid is refined (Section 3.2.1).
- A generative prior over 27 families of scalar-on-function mechanisms, with analytic index standardization and in-context noise augmentation (Section 3.3).
- An evaluation on ten simulation scenarios and twelve real tasks, with repeated splits, tuned baselines and rank-based tests (Demšar, 2006). Among fourteen methods, FunPFN has the best average rank in simulation and the second-best on real data, statistically tied with TabPFN at a fraction of its cost; it is the most accurate with 20–40 training curves, insensitive to the grid resolution and close to calibrated (Sections 4–5; limitations in Section 6).

2 Related work

Scalar-on-function prediction, classical and deep. The functional linear model $y = \int_0^1 x(t)\beta(t)\,\mathrm{d}t + \varepsilon$ is the workhorse of FDA (Cardot et al., 1999; Ramsay & Silverman, 2005), estimated via functional principal components (Yao et al., 2005; Cai & Hall, 2006; Hall & Horowitz, 2007), penalized bases (Goldsmith et al., 2011) or functional partial least squares (Preda & Saporta, 2005; Reiss & Ogden, 2007; Delaigle & Hall, 2012b)—the standard in chemometrics together with Savitzky–Golay derivatives (Savitzky & Golay, 1964; Rinnan et al., 2009). Nonlinear extensions include generalized functional linear and index models (Müller & Stadtmüller, 2005; Chen et al., 2011; Jiang & Wang, 2011) and fully nonparametric regression (Ferraty & Vieu, 2006); functional classification runs from penalized discriminant analysis (Hastie et al., 1995) to near-perfect classifiers (Delaigle & Hall, 2012a). Minimax theory characterizes these problems (Hall & Horowitz, 2007; Cai & Hall, 2006; Zhou et al., 2023), and our simulation scenarios follow the task taxonomy of Zhou et al. (2026). Neural approaches—functional multilayer perceptrons (MLPs) (Rossi & Conan-Guez, 2005), adaptive basis layers (Yao et al., 2021), networks with functional inputs (Thind et al., 2023; Wang et al., 2023), kernel-embedding functional networks (Shi et al., 2024)—are trained on the target dataset itself and inherit the small-sample difficulties of deep learning. All of the above are fit *per dataset*; our goal is orthogonal: amortize the entire estimation problem into a single pre-trained model, with two deep functional baselines, namely *functional direct neural network* (FDNN) and *functional basis neural network* (FBNN) (Rao & Reimherr, 2023), included in our comparison.

Neural operators. Neural operators learn maps between function spaces with parameters independent of the discretization (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023; Bartolucci et al., 2023; Raonić et al., 2023). We adopt the FNO (Li et al., 2021) as the trunk of our encoder and extend the discretization-invariance requirement to every stage of an *episodic* pipeline: derivative features, pooling, episode-level normalization, and the spectral readout.

In-context learning and foundation models. Neural processes meta-learn maps from context sets to predictive distributions (Garnelo et al., 2018a;b; Kim et al., 2019; Nguyen & Grover, 2022); PFNs sharpen the Bayesian view—trained by maximum likelihood on episodes from a prior, the network approximates the posterior predictive under that prior (Müller et al., 2022; Nagler, 2023), connecting to amortized inference (Cranmer et al., 2020) and meta-learning (Finn et al., 2017; Hospedales et al., 2022); a growing literature studies what in-context learners implement (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023; Bai et al., 2023; Raventós et al., 2023). TabPFN validated the recipe for small tables (Hollmann et al., 2023; 2025); foundation models for scientific data target partial differential equation (PDE) surrogates (Subramanian et al., 2023; McCabe et al., 2023; Hao et al., 2024; Herde et al., 2024) and in-context operator learning (Yang et al., 2023). We bring this program to functional covariates, which requires a function-space encoder and a prior over *functionals* rather than over finite-dimensional predictors.

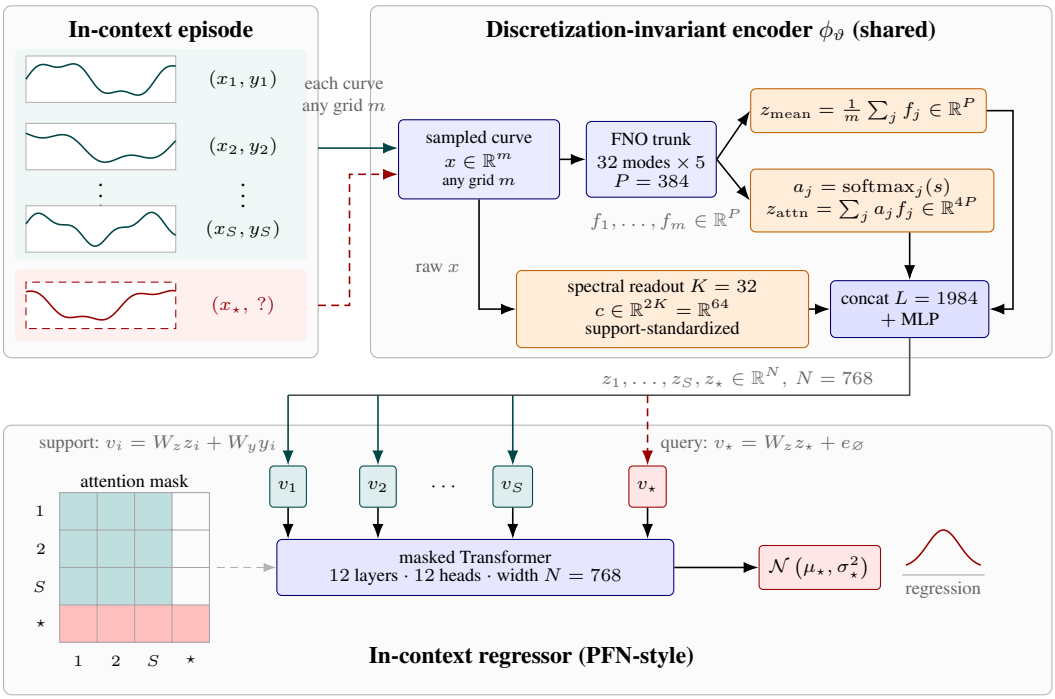


Figure 1: FunPFN overview — *Left:* an episode comprising a support set $\{(x_i, y_i)\}_{i=1}^S$ (teal) and a query curve $x_\star$ (red), observed on an arbitrary uniform grid of size m. *Center:* each curve passes through the shared discretization-invariant encoder, which combines an FNO trunk, mean and multihead attention pooling, and a support-standardized spectral readout before projecting the resulting representation to $z \in \mathbb{R}^{768}$. *Bottom:* support tokens combine each curve representation z_i with an embedding of its response y_i, whereas the query token combines $z_\star$ with a learned missing-response embedding $e_\varnothing$. The attention mask (inset; shaded cells indicate allowed attention, with rows attending to columns) makes queries conditionally independent given the support set. A Gaussian head $\mathcal{N}(\mu_\star, \sigma_\star^2)$ approximates the posterior predictive distribution under the pretraining prior, enabling inference on a new dataset in a single forward pass.

3 Method

3.1 Problem setting and in-context objective

Let $x : [0, 1] \to \mathbb{R}$ be a functional covariate and let $y \in \mathbb{R}$ be a scalar response. We assume that the data-generating mechanism is indexed by a task τ with prior distribution $p(\tau)$. A task τ specifies a curve distribution $p_\tau(x)$ and a relationship

$$y = F_\tau(x) + \varepsilon_\tau \tag{1}$$

where ε_τ is a conditionally centered, possibly heteroscedastic observation noise.

Given a support set $D_S = \{(x_i, y_i)\}_{i=1}^S$ from an unknown task τ and a query curve $x_\star$, our goal is to approximate the posterior predictive distribution $q_\theta(y_\star \mid x_\star, D_S)$.

Following the PFN recipe (Müller et al., 2022), a single model q_θ is pretrained by maximum likelihood on episodes sampled from a generative prior (Section 3.3):

$$\min_\theta \mathcal{L}(\theta)$$

with

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{\tau \sim p(\tau)} \left[\mathbb{E}_{\mathcal{D}_S, x_\star, y_\star \mid \tau} \left[-\log q_\theta \left(y_\star \mid x_\star, \mathcal{D}_S \right) \right] \right] \tag{2}$$

where $\mathcal{D}_S$ and $(x_\star, y_\star)$ are sampled from $p_\tau(x)$ followed by the synthetic model (1). The minimizer over all conditionals is the exact posterior predictive under $p(\tau)$ (Müller et al., 2022; Nagler,

2023); with a Gaussian predictor, $q_\theta(\cdot \mid x_\star, \mathcal{D}_S) = \mathcal{N}\left(\mu_\theta(x_\star, \mathcal{D}_S), \sigma^2_\theta(x_\star, \mathcal{D}_S)\right)$, exact minimization of Equation (2) over Gaussian conditionals returns $\mu_\theta(x_\star, \mathcal{D}_S) = \mathbb{E}_{\tau|x_\star,\mathcal{D}_S}[F_\tau(x_\star)]$ and $\sigma^2_\theta(x_\star, \mathcal{D}_S) = \mathrm{Var}(y_\star|x_\star, \mathcal{D}_S)$.

At deployment, the training set of a real dataset is $\mathcal{D}_S$ and the test set the queries: *one forward pass performs the entire fit-and-predict cycle*, including uncertainty. Curve values are standardized per grid point and responses globally, *using support statistics only*—the deployment regime where a scaler is fit on training data—so the task is always posed in the support's coordinate system at every context size, resolution and noise level seen in pretraining.

3.2 Architecture of the In-Context Gaussian Predictor

FunPFN consists of two components. First, a shared function encoder ϕ_ϑ maps each support and query curve to a fixed-dimensional code,

$$z_i = \phi_\vartheta(x_i) \in \mathbb{R}^N, \qquad z_\star = \phi_\vartheta(x_\star) \in \mathbb{R}^N.$$

The encoder is described in Section 3.2.1. Second, a Transformer-based in-context regressor jointly processes the support codes and responses $\{(z_i, y_i)\}_{i=1}^S$ together with the query code $z_\star$, and outputs

$$\mu_\star = \mu_\theta(x_\star, D_S), \qquad \sigma^2_\star = \sigma^2_\theta(x_\star, D_S).$$

This module is described in Section 3.2.2. We use ϑ to denote the parameters of the function encoder, while θ denotes the parameters of the complete model, including ϑ. The overall architecture is summarized in Figure 1.

3.2.1 Discretization-invariant function encoder

The encoder ϕ_ϑ maps a curve x observed on any uniform grid of size m to a code vector $z \in \mathbb{R}^N$. Every stage converges, as the grid is refined, to a well-defined functional of the underlying curve—the map therefore lives on function space rather than on $\mathbb{R}^m$.

FNO trunk The curve x, observed on the uniform grid $t_j = (j-1)/(m-1)$, $j = 1, \ldots, m$, enters an FNO (Li et al., 2021) as the two-channel signal $\big(x(t_j), t_j\big)_{j=1}^m$. Five Fourier layers of width $P = 384$, each keeping the $K = 32$ lowest modes, map it to pointwise features $f = (f_1, \ldots, f_m) \in \mathbb{R}^{P\times m}$, which we pool by averaging and by attention with four learned read-out queries,

$$z_{\mathrm{mean}} = \frac{1}{m}\sum_{j=1}^m f_j \in \mathbb{R}^P, \qquad z_{\mathrm{attn}} = \left(\sum_{j=1}^m a_{r,j} f_j\right)_{r=1}^4 \in \mathbb{R}^{4P}, \qquad a_{r,j} = \frac{\mathrm{e}^{s_{r,j}}}{\sum_{\ell=1}^m \mathrm{e}^{s_{r,\ell}}} \in \mathbb{R}, \tag{3}$$

where the scores $s_j = (s_{1,j}, \ldots, s_{4,j}) = g\big(\mathrm{LN}(f_j)\big)$ are given by a two-layer MLP g. Since the FNO is discretization-invariant, f_j and $s_{r,j}$ approximate the values $f(t_j)$ and $s_r(t_j)$ of fields on $[0,1]$ that do not depend on m. Both poolings are therefore ratios of Riemann sums: as $m \to \infty$, $z_{\mathrm{mean}} \to \int_0^1 f(t)\,\mathrm{d}t$ and $\sum_j a_{r,j} f_j \to \int_0^1 f(t)\,\mathrm{e}^{s_r(t)}\,\mathrm{d}t / \int_0^1 \mathrm{e}^{s_r(t)}\,\mathrm{d}t$, whereas sum pooling (Zaheer et al., 2017) scales with m. When s_r is sharply peaked, the r-th read-out approaches f at the maximizer of s_r, so functionals such as suprema or argmax locations can be extracted, in the spirit of pooling by multihead attention (Lee et al., 2019).

Support-standardized spectral readout. Both pooling operators integrate the input curve against a nonnegative weight, i.e., a positive measure. High-frequency linear functionals instead require signed, oscillatory test functions. The trunk does not synthesize these, so the likelihood-trained model responds by widening its predictive uncertainty on such targets, which amounts to calibrated abstention. We therefore expose the first $K = 32$ Fourier coefficients, $c_k = \Delta t \sum_{j=0}^{m-2} x(t_j)\mathrm{e}^{-2\pi i k t_j} \approx \int_0^1 x(t)\mathrm{e}^{-2\pi i k t}\,\mathrm{d}t$, which requires $m \geq 2K$. Since the decay of $|c_k|$ would bury task-relevant coordinates, each coefficient is standardized with per-episode support statistics, as are inputs and outputs, and scaled by a learnable gain, mirroring the truncated-score representation of functional linear regression (Hall & Horowitz, 2007; Zhou et al., 2023). Their real and imaginary parts $c \in \mathbb{R}^{2K}$, $z_{\mathrm{mean}} \in \mathbb{R}^P$ and the four attention read-outs $z_{\mathrm{attn}} \in \mathbb{R}^{4P}$ ($P = 384$ FNO channels) are concatenated and mapped by a linear layer, GELU and layer normalization to $z = \phi_\vartheta(x) \in \mathbb{R}^N$, with $N = 768$.

3.2.2 IN-CONTEXT REGRESSOR

The per-curve codes produced by the encoder are processed jointly by a transformer over the episode. For each support observation $(x_i, y_i) \in \mathcal{D}_S$, let $z_i = \phi_\vartheta(x_i) \in \mathbb{R}^N$. The corresponding transformer token is $v_i = W_z z_i + w_y y_i \in \mathbb{R}^N$, $i = 1, \dots, S$, where $W_z \in \mathbb{R}^{N \times N}$ and $w_y \in \mathbb{R}^N$ are trainable (bias terms are omitted for readability). For the query curve, given $z_\star = \phi_\vartheta(x_\star)$, the query token is $v_\star = W_z z_\star + e_\emptyset$, where $e_\emptyset \in \mathbb{R}^N$ is a learned missing-response embedding.

The transformer stack has twelve pre-norm layers of width $N = 768$. Each layer has twelve attention heads of dimension 64 and a feed-forward block of width $4N = 3072$ with GELU activations, and no dropout is used. There are no positional encodings, so the prediction is invariant to permutations of the support by construction. Attention follows a PFN-style mask (Müller et al., 2022): support tokens attend only to support tokens, and each query token attends to the support tokens and to itself. When Q query curves are processed in a single pass, this mask makes their predictions conditionally independent given $\mathcal{D}_S$. Each prediction therefore coincides with the one obtained for that query alone. A two-layer head is applied to the output of the query token: layer normalization, a linear map to width $N/2$, a GELU, and a linear map to $\mathbb{R}^2$. It returns $(\mu_\star, \log \sigma_\star^2)$, and $\log \sigma_\star^2$ is clipped to $[-10, 5]$.

3.3 A GENERATIVE PRIOR OVER SCALAR-ON-FUNCTION TASKS

The prior $p(\tau)\, p_\tau(x)\, p_\tau(y \mid x)$ is fully synthetic and designed for *mechanism coverage* (cf. task diversity in Raventós et al., 2023). A task τ fixes a curve process, weight functions, a target functional and a noise law; curves are drawn on a 700-point master grid, where targets are computed by quadrature, and resampled to the step resolution m. *Curves* compose a base process—a *Gaussian process* (GP) with one of 21 fixed or 12 randomly parameterized kernel types (radial basis function (RBF), Matérn, spectral-mixture (Wilson & Adams, 2013), Brownian (Rasmussen & Williams, 2006),etc.), a non-Gaussian Karhunen–Loève expansion, or spectroscopy-like, spike and piecewise generators—with task-level transformations (mean, time warping (Ramsay & Silverman, 2005), envelope, drift, jumps). A *pure-GP* regime (centered GP, no transformation) is mandatory for multiple-index and eigen-score tasks and used for 20–30% of the others: under centered Gaussian curves, $\mathbb{E}[g(\langle x, \beta\rangle)\, x] = 0$ for every even link g, so no linear proxy exists and identification must be second-order. Derivative functionals are restricted to C^2 paths. *Weights* β come from a 530-atom dictionary or nine continuous generators, e.g. decaying cosine coefficients, process eigenfunctions (Hall & Horowitz, 2007) and sparse Haar wavelets (Zhao et al., 2012). *Targets* span 27 families (Table 4, Appendix A): linear, single- and multiple-index (Chen et al., 2011), additive (Müller & Yao, 2008) and quadratic (Yao & Müller, 2010) models, and energy, extremal, level-crossing, distributional, derivative, curvature, template-distance and spectral functionals. Under the pure-GP regime, indices are standardized by their exact scale $\varsigma^2 = (w \circ \beta)^\top C_\tau (w \circ \beta)$. The noise law (Gaussian, Student-t or heteroscedastic) is fixed per task, but its level is drawn log-uniformly *per episode* between 2% and 100% of the response scale, so it must be inferred in context.

Each episode draws its task from a frozen bank of 3,130 tasks (5% held out) with probability 0.3, and otherwise from an online sampler producing fresh tasks from the same families. With probability $1/2$, input noise at a common level $\sigma_x \sim \mathcal{U}(0, 0.3) \cdot \mathrm{sd}(x)$ is added to support and queries after targets are computed, at the observed resolution so that its variance does not depend on m. The support reveals this level, and the Bayes predictive learns to shrink its high-frequency dependence accordingly—a robustness trade-off learned in context rather than fixed by preprocessing. Two generators share a parametric class with benchmark scenarios (decaying coefficients and Hall–Horowitz designs; additive and quadratic models), but their coefficients, directions and links are drawn independently (Appendix A).

3.4 PRETRAINING AND ZERO-SHOT INFERENCE

Episodes use log-uniform context sizes $S \in \{8, \dots, 512\}$ (densifying the small-context regime), Q=64 queries, and resolutions drawn log-uniformly on $\{64, \dots, 700\}$ (equal weight per octave), except that with probability 0.2 the nominal resolution m=700 is used; curves are generated and targets integrated on the master grid (700 points), then resampled to m, so targets never depend on the grid. The meta-batch adapts to two per-GPU budgets, $B(S{+}Q) \le 2{,}600$ tokens for the transformer and $B(S{+}Q)m \le 8.2{\times}10^5$ points for the encoder, which keeps cost and memory per step roughly con-

Table 1: Summary of the simulation experiments (SIM1, SIM2 & SIM3). n = number of curves, m = evaluation grid size.

Study	Aim	Setup	n	m
SIM1	Accuracy across DGPs	10 scenarios; noise $= 0.3\sigma_f$; 30 replications	150	100
SIM2	Sample efficiency	3 scenarios	$\{50, \ldots, 800\}$	100
SIM3	Noise robustness	2 scenarios; $\varsigma/\sigma_f \in \{0, 0.25, 0.5, 1\}$	150	100

Table 2: Real data benchmark regression tasks. n = curves, m = native grid.

Task	Functional covariate	Response	n	m
Tecator-fat/water/protein	near-infrared (NIR) spectrum (850–1050 nm)	fat / water / protein (%)	215	100
Growth-height18	height curve up to age 12	adult height (age 18)	93	19
Weather-logprec	daily temperature profile (Canada)	$\log_{10}$ total precipitation	35	365
Gasoline-octane	NIR spectrum (900–1700 nm, 2 nm)	octane number	60	401
Cookie-fat/sucrose/dry_flour/water	NIR reflectance of dough (1100–2498 nm, 2 nm)	fat / sucrose / dry flour / water (%)	72	700
DTI-pasat	fractional anisotropy along corpus callosum	PASAT cognitive score	100	93
Aemet-logprec	daily temperature profile (Spain)	$\log_{10}$ total precipitation	73	365

stant. Families from the functional-regression literature are over-sampled ($\times 3$ first/second derivative, curvature, multi-index, eigen-score; $\times 2$ single-index, additive-index, quadratic), and linear tasks account for about 15% of fresh tasks. We train with AdamW (Loshchilov & Hutter, 2019) (β=$(0.9, 0.95)$, lr 3×10^{-4}, weight decay 10^{-4}, gradient clipping 1.0) for 300k optimizer steps, with a linear warmup over the first 5% followed by cosine decay, in mixed precision. The model has $\approx$136M parameters (FNO encoder: 5 layers, width 384, 32 modes; transformer: 12 layers, width 768, 12 heads). At deployment, inputs and responses are standardized with support statistics and predictions de-standardized, with no gradient step.

4 Experimental setup

We evaluate FunPFN on simulated and real functional data. Three regression simulations (Table 1) assess predictive accuracy on unseen data-generating processes (DGPs) (SIM1), sample efficiency (SIM2), and robustness to response noise (SIM3). On real data, four regression experiments assess predictive accuracy (E1), invariance to the curve resolution (E2), sample efficiency (E3), and calibration of the predictive uncertainty (E4). Table 2 lists each dataset's covariates, responses, sample sizes and grid resolutions.

Simulation study. We use ten scenarios from the statistical literature: Hall–Horowitz functional linear regression (FLR) designs at two smoothness levels (Hall & Horowitz, 2007; Cai & Hall, 2006), the five-case taxonomy of Zhou et al. (2026), a sinusoidal single index (Chen et al., 2011; Jiang & Wang, 2011), a non-separable quadratic operator $\iint X(s)X(t)K(s,t)\,\mathrm{d}s\,\mathrm{d}t$, and an FLR on Brownian paths (Ferraty & Vieu, 2006). The prior (Section 3.3) draws on the same taxonomy: every target form belongs to one of its families, and only the instance differs—coefficient functions, interval cut points, index directions, covariance of X. Table 5 and Appendix B grade the overlap: identical mechanism (basis-score targets and their nonlinear transforms, squared and sinusoidal single indices), same parametric class with parameters in the support of the prior's generators (Hall–Horowitz designs, sub-interval interaction, Brownian FLR), or approximation by nearby members (non-separable operator, affine-β FLR). The study thus measures how well in-context inference *amortizes* estimation within the prior—how close one forward pass comes to the Bayes oracle at small n, relative to per-task estimators—not out-of-distribution generalization, which the real-data benchmarks address. ***SIM1*** reports the *normalized excess risk* (0 = Bayes oracle, 1 = mean predictor) over 30 replications at n=150 with target noise $0.3\,\sigma_f$; ***SIM2*** varies $n \in \{50, \ldots, 800\}$ and, for the Hall–Horowitz designs, compares log–log slopes with the minimax rate $n^{-(a+2b-1)/(a+2b)}$ (Hall & Horowitz, 2007); ***SIM3*** varies the noise in $\{0, 0.25, 0.5, 1\}\,\sigma_f$. Aggregation follows Demšar (2006): average ranks, Friedman test (Friedman, 1937), Nemenyi critical difference, and Wilcoxon tests with Holm correction (Holm, 1979).

Real-data benchmark. Twelve regression tasks from seven public datasets (Table 2) cover chemometrics—Tecator (Borggaard & Thodberg, 1992), gasoline (Kalivas, 1997) and cookie dough (Osborne et al., 1984; Brown et al., 2001) NIR spectra—, growth (Tuddenham & Snyder, 1954), climate—Canadian (Ramsay & Silverman, 2005) and Spanish Agencia Estatal de Meteorología (aemet) (Febrero-Bande & Oviedo de la Fuente, 2012) temperature profiles—and clinical neuroimaging, with Diffusion Tensor Imaging (dti) tract profiles predicting the PASAT score (Gronwall, 1977; Goldsmith et al., 2011). Data are loaded from `scikit-fda` (Ramos-Carreño et al., 2024), `refund` (Goldsmith et al., 2024) and `ppls` (Krämer et al., 2008). All tasks are small ($n \leq 215$), the regime FDA methods target, with native grids from $m = 19$ to $m = 700$. *E1*: 30 random $75/25$ splits per task, shared across methods; each baseline is tuned by inner 5-fold cross-validation (CV) on the training split (grids in Table 6, Appendix C); root mean-squared error (RMSE) is aggregated as above. *E2*: train and test curves resampled to $m \in \{100, 200, \ldots, 1500\}$ (10 repetitions; RMSE averaged over tasks and normalized by its value at m=700). *E3*: training set subsampled to $n_{\mathrm{tr}} \in \{20, \ldots, 160\}$ with the test split fixed (15 repetitions), pooling at each n_{tr} the tasks large enough to reach it. *E4*: quality of the Gaussian predictive, which deterministic competitors lack: mean negative log-likelihood (NLL) (Gneiting & Raftery, 2007), $\mathrm{RMS}(z)$ and central coverage at $\{50, 80, 90, 95\}\%$ (Kuleshov et al., 2018).

Baselines. Twelve tuned regression baselines span four methodological families. The *FDA* baselines comprise functional principal component analysis (FPCA) regression and functional partial least squares (FPLS) (Reiss & Ogden, 2007; Preda & Saporta, 2005; Delaigle & Hall, 2012b), penalized FLR on B-splines (Goldsmith et al., 2011), functional k-nearest neighbors (k-NN), and Nadaraya–Watson (NW) regression (Ferraty & Vieu, 2006); these methods are implemented using `scikit-fda` (Ramos-Carreño et al., 2024). The *chemometric* baseline applies a Savitzky–Golay second derivative followed by partial least squares (PLS) (Savitzky & Golay, 1964; Rinnan et al., 2009). The *multivariate machine learning (ML)* baselines operate on discretized curves and comprise ridge regression, PLS, random forest (RF), and a 2×32 rectified linear unit (ReLU) MLP; the latter follows the small-deep neural network (DNN) architecture of Zhou et al. (2026). The *deep functional* baselines comprise the dataset-fitted FDNN and FBNN, trained with early stopping (Rao & Reimherr, 2023). Where reported, we additionally compare with TabPFN (Hollmann et al., 2023; 2025). In contrast to the dataset-fitted baselines, FunPFN is applied strictly zero-shot, without gradient updates or hyperparameter tuning. All methods use identical data splits, and library failures are recorded transparently.

Table 7 in Appendix D summarizes the methods included in each experiment. An ablation study of the components of the model is documented in Appendix H. Experiments use PyTorch 2.10 with CUDA 12.8, `neuraloperator` (Kovachki et al., 2023), `scikit-fda` 0.10.1, `scikit-learn` 1.7.2, SciPy 1.15.3, and NumPy 2.2.6, and are run on a single graphics processing unit (GPU) with fixed random seeds. TabPFN version used is v3 (8.5.0) with eight estimators and 2000 maximum features.

5 Results

5.1 Mechanism coverage (SIM1–SIM3)

On the five functional-*linear* scenarios (Table 3), tuned linear specialists are best on four, but zero-shot FunPFN stays within 0.0015 of them and ties FPLS on the score-linear target (0.0060 vs. 0.0061). On the five *nonlinear* mechanisms, where every linear pipeline collapses to the trivial predictor (excess risk 0.91–1.16; Appendix E), FunPFN is first throughout (0.010–0.022). It has lower mean excess risk than TabPFN in all ten scenarios (ratio 0.22–0.67) and the best average rank (1.70; next best TabPFN, 4.70; Friedman $\chi^2_F(13)$=51.57, p=1.6×10^{-6}). With only N=10 scenarios, Holm-corrected rank comparisons separate FunPFN from every baseline except TabPFN (p=0.11); over the 300 scenario×replication pairs, Holm–Wilcoxon tests find it significantly better than every baseline, TabPFN included (p=4.7×10^{-49}), although against the linear specialists this pooled result is driven by the nonlinear scenarios.

In the convergence study (SIM2; Table 10, Appendix F), the excess risk of FunPFN decays with n at log–log slopes of -0.512 and -0.439 on Hall–Horowitz $(2, 2)$ and $(1.2, 1.1)$, shallower than both the minimax rates (-0.833, -0.706) and those of FPCA and ridge. The shortfall is consistent with

Table 3: SIM1 — normalized excess risk (mean ± sd over 30 replications; 0 = Bayes oracle, 1 = mean predictor). "Best other": best of the ten classical/ML baselines; all methods in Appendix E. Bold: best per row.

Scenario	FunPFN (zero-shot)	FBNN	TabPFN	Best other
FLR Hall–Horowitz (a, b)=$(2, 2)$	0.005 ± 0.002	0.027 ± 0.012	0.012 ± 0.005	**0.004** (FPLS)
FLR Hall–Horowitz $(1.2, 1.1)$	0.007 ± 0.003	0.042 ± 0.008	0.014 ± 0.004	**0.006** (FLR-pen)
Score-linear ξ_3	$\mathbf{0.006 \pm 0.002}$	0.025 ± 0.008	0.017 ± 0.004	**0.006** (FPLS)
Score-quadratic ξ_3^2	$\mathbf{0.016 \pm 0.007}$	0.093 ± 0.035	0.057 ± 0.025	0.653 (NW)
Interval interaction	$\mathbf{0.017 \pm 0.007}$	0.056 ± 0.043	0.025 ± 0.012	0.174 (NW)
FLR $\beta(t) = t - 0.3$	$\mathbf{0.005 \pm 0.004}$	0.032 ± 0.014	0.014 ± 0.004	**0.005** (FPLS)
Quadratic $(\int X\beta)^2$	$\mathbf{0.010 \pm 0.005}$	0.082 ± 0.025	0.046 ± 0.071	0.233 (RF)
Single-index sin	$\mathbf{0.022 \pm 0.009}$	0.174 ± 0.097	0.037 ± 0.009	0.256 (NW)
Non-separable quad. operator	$\mathbf{0.01 \pm 0.004}$	0.041 ± 0.025	0.025 ± 0.006	0.155 (NW)
FLR on Brownian paths	0.004 ± 0.002	0.02 ± 0.013	0.010 ± 0.003	**0.003** (FPLS)

a context-length effect. Up to n=200, FunPFN stays within a factor 1.3 of the best specialist and is best at four of nine $(n, \text{scenario})$ points; at n=800 it saturates at 0.0018–0.0035 (best specialist: 0.0007–0.0028). Under increasing target noise (SIM3; Table 11), linear specialists degrade most gracefully in the linear setting, by a small margin (at $\varsigma = \sigma_f$: ridge 0.029, FunPFN 0.037, FDNN 0.043); on the non-separable operator, where they stay near 1, FunPFN is first at every noise level (0.002–0.087, vs. 0.18–0.27 for FDNN).

5.2 Real-data benchmark, resolution invariance, sample efficiency, calibration (E1–E4)

E1. We compare $k = 14$ methods on twelve scalar-on-function regression tasks (Table 12, Appendix F), including two deliberately chosen outside the spectroscopic domain that guided the prior's design: a cognitive score from white-matter tract profiles and log-precipitation from Spanish daily temperature curves. Following Demšar (2006), each task contributes one observation, its RMSE averaged over 30 splits. The Friedman test rejects equal performance ($\chi_F^2 = 74.7$, $p = 1.1 \times 10^{-10}$; Iman–Davenport $F_F(13, 143) = 10.1$, $p = 7.8 \times 10^{-15}$), and the Nemenyi critical difference is 5.73 rank points (Figure 2).

FunPFN has the second-best average rank (4.63), behind TabPFN (3.29) and on par with Ridge (4.67). Holm-corrected Wilcoxon signed-rank tests find it significantly better than five of the thirteen baselines (MLP-2×32, RF, Nadaraya–Watson, FDNN and Func-k-NN; $p_{\text{Holm}} \leq 0.044$) and not significantly different from TabPFN (4 wins, 7 losses, 1 tie; $p_{\text{Holm}} = 0.61$). The more conservative Nemenyi test separates it from only four of these five (all but RF). We therefore claim competitiveness with the strongest baselines, not superiority, at a fraction of their cost. Averaging 0.025 s per split (Table 12), FunPFN is roughly 330×, 240×, 150× and 100× faster than FPCA-Reg (8.22 s), TabPFN (5.98 s), FLR-penalized (3.84 s) and FPLS-Reg (2.39 s), all four statistically tied with it. Unlike TabPFN, the three functional baselines also require per-dataset fitting and tuning. The only faster method is Ridge (< 0.01 s), which is, however, less robust to changes in the discretization grid (E2, below).

E2. With train and test resampled to every $m \in \{100, 200, \ldots, 1000, 1500\}$, the RMSE of FunPFN relative to its value at the pre-training resolution m=700 stays within 0.99–1.00 over the whole range—on grids 7× coarser (m=100) and over 2× finer (m=1500)—without any re-fitting (Table 13, Appendix F). Classical pipelines are equally stable (FPCA-/FPLS-Reg: 0.99–1.00; Ridge: 0.98–1.03) but are *re-fit at every resolution*; FDNN drifts monotonically from 0.95 to 1.03, its effective capacity changing with the grid.

E3. Pooled over the five tasks, FunPFN attains the lowest RMSE for $n_{\text{tr}} \leq 40$: 3.10 vs. 3.34 for the best tuned baseline and 3.91 for TabPFN at n_{tr}=20, and 2.78 vs. 3.39 and 3.05 at n_{tr}=40. Against the tuned baselines, the gap then stabilizes at about 33% (1.49 vs. $\geq$ 2.23 at n_{tr}=80; 1.28 vs. $\geq$ 1.91 at 160). TabPFN, however, overtakes FunPFN from n_{tr}=80 onward (1.33 vs. 1.49), and its lead grows with sample size (0.89 vs. 1.28 at 160). The benefit of the amortized functional prior is therefore concentrated in the smallest-sample regime. The full RMSE vs. training-set size results are reported in Table 14, Appendix F.

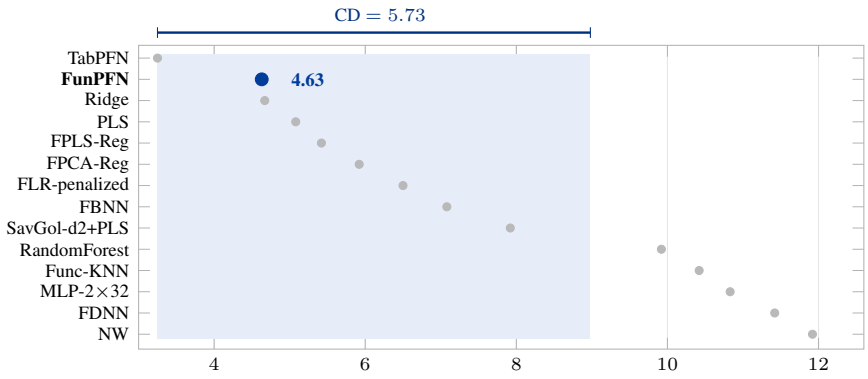


Figure 2: E1 — Results on twelve tasks with the Nemenyi critical difference 5.73 and the Friedman test $p = 1.1 \times 10^{-10}$.

E4. Aggregated over splits and tasks, the Gaussian predictive attains mean NLL 1.703, $\mathrm{RMS}(z)$=0.976, and central coverage 56.2%, 82.5%, 91.2%, 95.2% at nominal 50/80/90/95%: the predictive uncertainty is thus slightly overestimated, most visibly at the 50% level, while coverage at 95% is essentially nominal.

6 Discussion and Limitations

Without per-task tuning or basis expansion, FunPFN is competitive with tuned specialists: it has the best average rank on SIM1, ranks second on E1 (not significantly behind TabPFN; Nemenyi test), and processes a test split in 0.025 s on average, 150–330× faster than TabPFN, FPCA regression and penalized FLR. It attains the lowest pooled RMSE at $n_{\mathrm{tr}} \in \{20, 40\}$, and its FNO trunk and quadrature pooling limit RMSE variation to 1% across grid sizes $m = 100$–1500, without retraining or interpolation.

Limitations. Predictive intervals are slightly conservative, with over-coverage falling from 6.2 points at the 50% level to 0.2 at 95%; inputs are restricted to 1D curves on uniform grids over $[0, 1]$; and scarce public scalar-on-function benchmarks limit real-data evaluation.

7 Conclusion

We introduced FunPFN, a foundation model for zero-shot scalar-on-function regression that replaces fitting and tuning on a new dataset with a single forward pass. It pairs a discretization-invariant Fourier neural operator encoder with an in-context transformer that outputs calibrated Gaussian predictive distributions. Although FunPFN never sees a real curve during pretraining, it matches extensively tuned classical and deep baselines on synthetic and real-world benchmarks and outperforms them when training curves are scarce. It is also 240× faster than TabPFN, which likewise needs no per-dataset training. These results show that prior-data fitted networks extend from tabular inputs to infinite-dimensional functional covariates. The modeling effort shifts accordingly: from fitting each dataset to designing the synthetic prior, where it is spent once and amortized across datasets.

Ethics Statement

This research relies exclusively on publicly available benchmark datasets and synthetic data, containing no personally identifiable information (PII) or newly collected human subject data. We foresee no direct negative ethical or societal implications arising specifically from this methodological work. As with any statistical model providing uncertainty estimates, practitioners should validate performance and calibration within their target domain prior to deployment in high-stakes settings.

STATEMENT ON LLM AND GENERATIVE AI USE

Required disclosure. Generative AI was used in this paper strictly as an auxiliary tool for routine coding and prose refinement—specifically, assisting with the implementation of the evaluation harness, baseline scripts, and copyediting. All primary intellectual and scientific contributions remain entirely the authors' own: the research hypotheses, model architecture, generative prior, mathematical formulations, experimental design, and interpretation of results were independently conceived and executed by the authors. No synthetic data was generated by AI, and the authors assume complete responsibility for every claim and analysis presented in this paper.

Verification. All numerical claims were verified directly against primary run records, and all aggregation scripts are released alongside the data. Every AI-assisted passage was vetted line by line by the authors, who assume full responsibility for all content, code, and claims in this work

REPRODUCIBILITY STATEMENT

Complete model architectures, training procedures, and evaluation protocols are detailed in Sections 3 and 4 and the Appendix. All source code, environment configurations, random seeds, model checkpoints, and execution logs required to reproduce our results will be publicly available. All models and reported benchmarks have been verified to ensure full end-to-end reproducibility.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Francesca Bartolucci, Emmanuel de Bézenac, Bogdan Raonić, Roberto Molinaro, Siddhartha Mishra, and Rima Alaifari. Representation equivalent neural operators: a framework for alias-free operator learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Claus Borggaard and Hans Henrik Thodberg. Optimal minimal neural interpretation of spectra. *Analytical Chemistry*, 64(5):545–551, 1992.

Philip J. Brown, Tom Fearn, and Marina Vannucci. Bayesian wavelet regression on curves with application to a spectroscopic calibration problem. *Journal of the American Statistical Association*, 96(454):398–408, 2001.

T. Tony Cai and Peter Hall. Prediction in functional linear regression. *The Annals of Statistics*, 34 (5):2159–2179, 2006.

Hervé Cardot, Frédéric Ferraty, and Pascal Sarda. Functional linear model. *Statistics & Probability Letters*, 45(1):11–22, 1999.

Dong Chen, Peter Hall, and Hans-Georg Müller. Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39(3):1720–1747, 2011.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Aurore Delaigle and Peter Hall. Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B*, 74(2):267–286, 2012a.

Aurore Delaigle and Peter Hall. Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics*, 40(1):322–352, 2012b.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Manuel Febrero-Bande and Manuel Oviedo de la Fuente. Statistical computing in functional data analysis: the R package fda.usc. *Journal of Statistical Software*, 51(4):1–28, 2012.

Frédéric Ferraty and Philippe Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer, 2006.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Jimenez Rezende, and S. M. Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning (ICML)*, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Mark N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Jeff Goldsmith, Jennifer Bobb, Ciprian M. Crainiceanu, Brian Caffo, and Daniel Reich. Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4):830–851, 2011.

Jeff Goldsmith, Fabian Scheipl, Lei Huang, Julia Wrobel, Chongzhi Di, Jonathan Gellar, Jaroslaw Harezlak, Mathew W. McLean, Bruce Swihart, Luo Xiao, Ciprian Crainiceanu, and Philip T. Reiss. *refund: Regression with Functional Data*, 2024. URL `https://CRAN.R-project.org/package=refund`. R package.

D. M. A. Gronwall. Paced auditory serial-addition task: a measure of recovery from concussion. *Perceptual and Motor Skills*, 44(2):367–373, 1977.

Peter Hall and Joel L. Horowitz. Methodology and convergence rates for functional linear regression. *The Annals of Statistics*, 35(1):70–91, 2007.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *International Conference on Machine Learning (ICML)*, 2024.

Trevor Hastie, Andreas Buja, and Robert Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, 23(1):73–102, 1995.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5149–5169, 2022.

Ci-Ren Jiang and Jane-Ling Wang. Functional single index models for longitudinal data. *The Annals of Statistics*, 39(1):362–388, 2011.

John H. Kalivas. Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 37(2):255–259, 1997.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations (ICLR)*, 2019.

Nikola 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.

Nicole Krämer, Anne-Laure Boulesteix, and Gerhard Tutz. Penalized partial least squares with applications to B-spline transformations and functional data. *Chemometrics and Intelligent Laboratory Systems*, 94(1):60–69, 2008.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, 2018.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, 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 (ICLR)*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for physical surrogate models. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Jeffrey S. Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2:321–359, 2015.

Hans-Georg Müller and Ulrich Stadtmüller. Generalized functional linear models. *The Annals of Statistics*, 33(2):774–805, 2005.

Hans-Georg Müller and Fang Yao. Functional additive models. *Journal of the American Statistical Association*, 103(484):1534–1544, 2008.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning (ICML)*, 2023.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning (ICML)*, 2022.

B. G. Osborne, T. Fearn, A. R. Miller, and S. Douglas. Application of near infrared reflectance spectroscopy to the compositional analysis of biscuits and biscuit doughs. *Journal of the Science of Food and Agriculture*, 35(1):99–105, 1984.

Christopher J. Paciorek and Mark J. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 16, pp. 273–280, 2004.

Cristian Preda and Gilbert Saporta. PLS regression on a stochastic process. *Computational Statistics & Data Analysis*, 48(1):149–158, 2005.

Carlos Ramos-Carreño, José L. Torrecilla, Miguel Carbajo-Berrocal, Pablo Marcos, and Alberto Suárez. scikit-fda: A Python package for functional data analysis. *Journal of Statistical Software*, 109(2):1–37, 2024.

James O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer, 2nd edition, 2005.

Aniruddha Rajendra Rao and Matthew Reimherr. Nonlinear functional modeling using neural networks. *Journal of Computational and Graphical Statistics*, 32(4):1248–1257, 2023.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Philip T. Reiss and R. Todd Ogden. Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479):984–996, 2007.

Åsmund Rinnan, Frans van den Berg, and Søren Balling Engelsen. Review of the most common pre-processing techniques for near-infrared spectra. *TrAC Trends in Analytical Chemistry*, 28(10): 1201–1222, 2009.

Fabrice Rossi and Brieuc Conan-Guez. Functional multi-layer perceptron: a non-linear tool for functional data analysis. *Neural Networks*, 18(1):45–60, 2005.

Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

Zhongjie Shi, Jun Fan, Linhao Song, Ding-Xuan Zhou, and Johan A. K. Suykens. Nonlinear functional regression by functional deep neural network with kernel embedding. *arXiv preprint arXiv:2401.02890*, 2024.

Anuj Srivastava and Eric P. Klassen. *Functional and Shape Data Analysis*. Springer Series in Statistics. Springer, New York, 2016.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Barinder Thind, Kevin Multani, and Jiguo Cao. Deep learning with functional inputs. *Journal of Computational and Graphical Statistics*, 32(1):171–180, 2023.

Read D. Tuddenham and Margaret M. Snyder. Physical growth of California boys and girls from birth to eighteen years. *University of California Publications in Child Development*, 1:183–364, 1954.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning (ICML)*, 2023.

Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016.

Shuoyang Wang, Guanqun Cao, and Zuofeng Shang. Deep neural network classifier for multidimensional functional data. *Scandinavian Journal of Statistics*, 50(4), 2023.

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1067–1075, 2013.

Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

Fang Yao and Hans-Georg Müller. Functional quadratic regression. *Biometrika*, 97(1):49–64, 2010.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903, 2005.

Junwen Yao, Jonas Mueller, and Jane-Ling Wang. Deep learning for functional data analysis with adaptive basis layers. In *International Conference on Machine Learning (ICML)*, pp. 11898–11908, 2021.

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

Yihong Zhao, R. Todd Ogden, and Philip T. Reiss. Wavelet-based LASSO in functional linear regression. *Journal of Computational and Graphical Statistics*, 21(3):600–617, 2012.

Hang Zhou, Fang Yao, and Huiming Zhang. Functional linear regression for discretely observed data: from ideal to reality. *Biometrika*, 110(2):381–393, 2023.

Hang Zhou, Ju-Sheng Hong, Xiucai Ding, and Jane-Ling Wang. Deep neural network regression with functional covariates. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, Seoul, South Korea, 2026. PMLR.

A Prior details

Curves live on a uniform 700-point master grid of $[0, 1]$; integrals are trapezoid sums, derivatives central finite differences, extrema taken over the grid. A task fixes the curve recipe (regime, base, transformations), its weights, family parameters and noise law; curves and noise are redrawn per episode. LU denotes a log-uniform law and g_ℓ a unit-variance RBF GP path with length-scale ℓ among 16 values log-spaced in $[0.08, 0.4]$.

A.1 Gaussian-process kernels

Fixed bank (21 kernels): RBF ($\ell \in \{0.5, 0.2, 0.1, 0.05, 0.02\}$), Matérn ($\nu \in \{1/2, 3/2, 5/2\}$, $\ell \in \{0.2, 0.05\}$), four periodic or periodic-product kernels, four rational quadratic kernels, and two sums (RBF+Matérn$_{1/2}$, Matérn$_{3/2}$+periodic); the smooth regime excludes the 6 kernels with a Matérn$_{1/2}$ or $_{3/2}$ component. *Fresh kernels*: online tasks use, with probability $1/2$, an entry of a pool of 256 covariances, 32 of which are redrawn every 1,000 tasks. Each entry has $\ell \sim \mathrm{LU}(0.03, 0.6)$ and one of 12 types: RBF, Matérn $1/2, 3/2, 5/2$, rational quadratic, periodic$\times$RBF, spectral mixture with 1–3 components (Wilson & Adams, 2013), Brownian motion and bridge, integrated Brownian motion, Gibbs (Gibbs, 1997; Paciorek & Schervish, 2004), or linear trend + RBF; two entries are mixed with probability 0.3. Covariances are diagonalized in double precision and paths drawn as $x = V\Lambda^{1/2}\varepsilon$, which reproduces C exactly even at low numerical rank; the 12 leading eigenpairs are kept.

Table 4: Target families. $\langle x, \beta \rangle = \int_0^1 x\beta \, \mathrm{d}t$. $\tilde{s}_j$: index $\langle x, \beta_j \rangle$ standardized by its analytic scale ς_j under pure GP (Section 3.3), empirically otherwise. $\xi_j = v_j^\top x / \sqrt{\lambda_j} \sim \mathcal{N}(0,1)$, with (λ_j, v_j) the j-th eigenpair of the grid covariance. Regime: G = pure GP; S = smooth; L, M = (pure GP, smooth, general) with probabilities $(0.3, 0.2, 0.5)$, $(0.2, 0.15, 0.65)$; A = G (0.7) or general. w: online sampling weight (total 47).

Family	**Target** $F_\tau(x)$	**Reg.**	w
linear	$\langle x, \beta \rangle$	L	8
index-tanh/-sin/-sq	$\tanh(2\langle x, \beta \rangle)$, $\sin(2\langle x, \beta \rangle)$, $\langle x, \beta \rangle^2$	M	1 each
index-link	$g(\tilde{s}_1)$, random link g	M	2
multi-index	$\tilde{s}_1^2 - 1$, $\cos \tilde{s}_1 - e^{-1/2}$, $\tilde{s}_1 \tilde{s}_2$, $\tilde{s}_1^2 - \tilde{s}_2^2$, $\sin \tilde{s}_1 \cos \tilde{s}_2$, $\tanh(\tilde{s}_1)\tilde{s}_2$	G	3
eigen-score	ξ_j or $\xi_j^2 - 1$, $j \le 12$	G	3
additive	$\sum_{k \le q} g_k(\xi_{j_k})$ (G) or $\sum_{k \le q} g_k(\tilde{s}_k)$, $q \in \{2, 3, 4\}$	A	2
quadratic	$\langle x, \beta \rangle \langle x, \beta_2 \rangle$	M	1
quadratic-r	$b\langle x, \beta_0 \rangle + \sum_{j \le r} a_j \tilde{s}_j^2$, $r \le 3$, $b \in \{0, 1\}$, $a_j = \pm \mathcal{U}(0.3, 1)$	M	2
energy	$\langle x^2, \beta \rangle$	M	1
sup/argmax/range	$\max x$, $\arg\max x$, $\max x - \min x$	M	1 each
deriv, curv	$\langle x', \beta \rangle$, $\langle x'', \beta \rangle$	S	3 each
deriv2-energy	$\langle (x'')^2, \beta \rangle$	S	3
deriv-energy, TV	$\langle (x')^2, \beta \rangle$, $\int \lvert x' \rvert$	S	1 each
deriv-sup/-argmax	$\max x'$, $\arg\max x'$	S	1 each
occupation, passage	fraction of grid points with $x > c$; first t with $x(t) > c$ (1 if none)	M	1 each
quantile, skewness	of the grid values $\{x(t_i)\}$, order $p \sim \mathcal{U}(0.1, 0.9)$	M	1 each
template distance	$\lVert x - x_0 \rVert_{L^2}$, x_0 a path of the task's process	M	1
band energy	$\frac{1}{2} \log \sum_{k=k_1}^{k_2} \lvert c_k(x) \rvert^2$, c_k the DFT coefficients, $k_1 \le 40$, $k_2 \le 80$	M	1

A.2 Curve processes

In the *pure-GP* regime, curves are centered GP paths without transformation. The *smooth* regime draws spectroscopy-like (0.4), Karhunen–Loève (0.3) or GP (0.3) curves; the *general* regime draws GP (0.35), Karhunen–Loève (0.2), spectroscopy-like (0.2), spike (0.15) or piecewise (0.1) curves. General-regime GP curves receive 1–5 Gaussian bumps with probability 0.3. *Karhunen–Loève* curves expand on Fourier or Legendre (8–40 terms, variance decay $k^{-\alpha}$), cubic B-spline (AR(1) coefficients) or sparse Haar bases, with Gaussian, uniform, Student, skewed or bimodal standardized scores. *Spike* curves carry up to 12 Gaussian, Lorentzian or asymmetric peaks of width $\mathrm{LU}(0.004, 0.08)$; *piecewise* curves 1–6 steps or ramps; both sit on a smooth background.

Spectroscopy-like curves. The task fixes a smooth envelope $a\, g_\ell$, 1–6 absorption (or, with probability 0.3, emission) bands with centers $\mathcal{U}(0.1, 0.9)$ and widths $\mathrm{LU}(0.005, 0.06)$, and, with probability 0.6, a ripple frequency $\mathcal{U}(8, 40)$. Each sample adds three random low-order harmonics, band depths $\mathcal{U}(0.1, 1)$, a ripple of amplitude $\mathcal{U}(0.02, 0.1)$ and random phase, a baseline shift $\mathcal{N}(0, 0.3^2)$ and optionally a slope. On these curves, β is centered in the linear, index, index-link and quadratic families and in the linear term of quadratic-r, so the baseline carries no signal through these terms.

Transformations. Outside pure GP, each is activated independently per task: random time warping (Ramsay & Silverman, 2005; Srivastava & Klassen, 2016) (0.35), amplitude factor or smooth envelope (0.4), polynomial drift of degree 1–2 (0.35), 1–3 jumps (0.2, general regime only), and a task-level mean function (0.5).

A.3 Weights, links and noise

Bank tasks draw β from 530 fixed atoms: monomials, sines and cosines of frequencies 1–25, Gaussian bumps, random sine series, sigmoids, interval indicators, chirps, boundary exponentials, tents, Haar wavelets, B-splines, Legendre polynomials and localized oscillations. Online tasks draw from nine generators: dictionary atoms (0.15); coefficients $\pm k^{-b}$, $b \sim \mathcal{U}(0.6, 3)$, on 40 cosine or Fourier functions (0.2); smooth GP draws (0.12); indicators, ramps or sigmoids (0.13); B-splines with AR(1)

coefficients (0.12); sparse Haar wavelets (Zhao et al., 2012) (0.08); weights $\sum_{k\le 12} \pm k^{-b} v_k$ aligned with the covariance eigenvectors (Cai & Hall, 2006; Hall & Horowitz, 2007) (0.08); GP paths (0.06); localized chirps (0.06). Two draws are summed with probability 0.15, and the result is standardized over the grid (with a floored standard deviation) and rescaled by $\mathcal{U}(0.5, 2)$. Random links (Chen et al., 2011) are drawn among $a\tanh(bs+c)$, $\sin(bs+c)$, $s^3/3 - as$, $\mathrm{softplus}(bs+c)$, $|s-c|$, $e^{0.4as}$, $\sigma(2b(s-c))$ and a three-knot hinge, with a, b, c uniform on bounded ranges. The noise law is fixed per task: Gaussian (0.7), unit-variance Student-t_ν, $\nu \sim \mathcal{U}(3, 8)$ (0.15), or heteroscedastic with scale $0.5 + |z|$ in the standardized target z (0.15); the episode draws its level $\rho \sim \mathrm{LU}(0.02, 1)$ and sets $y = F_\tau(x) + \rho\,\mathrm{sd}(F_\tau)\,\varepsilon$.

A.4 Task bank and episodes

The frozen bank holds one linear task per atom and 100 tasks per nonlinear family (3,130 tasks, fixed kernels only); 156 (5%) are held out. Each training episode draws a fresh task (probability 0.7, family weights w) or a bank task (probability 0.3, same weights). For a task, context size S, $Q = 64$ queries and resolution m: (1) draw $S+Q$ curves on the master grid, compute noisy targets there; (2) linearly resample the curves to m points—targets never depend on m; (3) training only, with probability $1/2$, add white noise of common level $\sigma_x = u\,\mathrm{sd}(x)$, $u \sim \mathcal{U}(0, 0.3)$, at resolution m; (4) split, standardize inputs per grid point and responses globally with *support* statistics only, and clip standardized responses at ± 20. Each step shares $S \sim \mathrm{LU}\{8, \dots, 512\}$ and m (700 with probability 0.2, otherwise $\mathrm{LU}\{64, \dots, 700\}$) across its episodes. Validation episodes, one per held-out task, use $m \in \{64, 128, 256, 700\}$, $S \in \{32, 64, \dots, 512\}$, noise level 0.05 and no observation noise, and are frozen.

B Simulation scenarios

Table 5 lists the ten simulation scenarios of the SIM benchmark, with the source of each design and its response mechanism.

Table 5: The ten SIM scenarios. ξ_j denotes the j-th basis score of the scenario's eigen-design; K is an exponential kernel.

Scenario	**Source**	**Response**
FLR-HH $(2, 2)$	Hall & Horowitz (2007); Cai & Hall (2006)	$\langle X, \beta\rangle$
FLR-HH $(1.2, 1.1)$, rough regime	Hall & Horowitz (2007)	$\langle X, \beta\rangle$
Score-linear	Zhou et al. (2026, Case 1)	ξ_3
Score-quadratic	Zhou et al. (2026, Case 2)	ξ_3^2
Interval interaction	Zhou et al. (2026, Case 3)	$\big(\int_{I_1} X\big)\big(\int_{I_2} X\big)$
FLR affine, HH basis	Zhou et al. (2026, Case 4)	$\langle X,\, t - 0.3\rangle$
Quadratic	Zhou et al. (2026, Case 5)	$\langle X, \beta\rangle^2$
Single-index sin	Chen et al. (2011); Jiang & Wang (2011)	$\sin\big(\langle X, \beta\rangle\big)$
Non-separable operator	–	$\iint X(s)X(t)K(s,t)\,ds\,dt$
FLR on Brownian paths	Ferraty & Vieu (2006)	$\langle X, \beta\rangle$

C Baseline hyperparameter grids (inner 5-fold CV)

Table 6 summarizes the hyperparameter search spaces used for the regression baselines. Hyperparameters are selected independently on each training split using inner 5-fold CV, with grids adapted to the training-set size and curve resolution where required. In contrast, FunPFN is applied zero-shot and requires no dataset-specific hyperparameter tuning.

Table 6: Search spaces tuned by inner cross-validation on each training split (n_{tr} = its size; grids are capped so fold sizes remain valid). Methods below the rule are used without tuning.

Method	**Grid**
FPCA-Reg	components $\in \{2, 4, \ldots, \min(40,\ n_{\mathrm{tr}}-6,\ m)\}$
FPLS-Reg	components $\in \{1, \ldots, \min(25,\ n_{\mathrm{tr}}-6)\}$; B-spline pre-smoothing and multivariate PLS as numerical fallbacks
FLR-penalized	B-spline basis size $\in \{6, 10, 15, 20, 30\}$
Func-KNN	$k \in \{1, 3, 5, 7, 10, 15, 20\}$, weights $\in$ {uniform, distance}
Nadaraya–Watson	bandwidth $\in 8$ points linearly spaced in $[0.2, 5.0]$
SavGol-d2+PLS	window $\min(11, \cdot)$, polyorder 3, deriv 2; components $\in \{1, \ldots, \min(20, n_{\mathrm{tr}}-6)\}$
Ridge	$\alpha \in 15$ points log-spaced in $[10^{-3}, 10^{4}]$ (RidgeCV), standardized inputs
PLS	components $\in \{1, \ldots, \min(20,\ n_{\mathrm{tr}}-6)\}$, standardized inputs
RandomForest	300 trees, default depth (seeded per split)
MLP-2×32	two hidden layers of 32 ReLU units, early stopping, standardized x and y
FDNN / FBNN	deep functional baselines, early stopping; architectures as released
TabPFN	*none* (zero-shot; `tabpfn` 8.5.0, v3 default regressor checkpoint, 8 estimators, at most 2000 input features)
FunPFN	*none* (zero-shot)

Table 7: Methods included in each experiment. SIM1–SIM3 denote the simulation experiments; E1–E4 denote the real-data regression experiments. A check mark indicates that the method was evaluated.

Family	**Method**	**SIM1**	**SIM2**	**SIM3**	**E1**	**E2**	**E3**	**E4**
Proposed	FunPFN	✓	✓	✓	✓	✓	✓	✓
Pretrained	TabPFN	✓	—	—	✓	—	✓	—
FDA	FPCA regression	✓	✓	✓	✓	✓	✓	—
FDA	FPLS	✓	✓	✓	✓	✓	✓	—
FDA	Penalized FLR	✓	—	—	✓	—	—	—
FDA	Functional k-NN	✓	—	—	✓	—	—	—
FDA	NW	✓	—	—	✓	—	—	—
Chemometrics	Savitzky–Golay + PLS	✓	—	—	✓	—	—	—
ML	Ridge	✓	✓	✓	✓	✓	✓	—
ML	PLS	✓	—	—	✓	—	—	—
ML	RF	✓	—	—	✓	—	—	—
ML	2×32 ReLU MLP	✓	✓	—	✓	—	—	—
Deep functional	FDNN	✓	✓	✓	✓	✓	✓	—
Deep functional	FBNN	✓	—	—	✓	—	—	—

D Comparison methods and experimental coverage

Table 7 summarizes the methods evaluated in each simulation and real-data experiment. The complete regression suite is used in SIM1 and E1, whereas SIM2, SIM3, E2, and E3 use selected subsets. E4 evaluates only FunPFN, since the comparison methods do not provide predictive distributions.

E Full SIM1 results

Tables 8 and 9 report the complete SIM1 normalized excess-risk results for scenarios 1–5 and 6–10, respectively.

F E1, E2, E3, SIM2 and SIM3 tables

Tables 10 and 11 report the results of the SIM2 and SIM3 experiments, respectively. Tables 12, 13, and 14 report the results of the real-data experiments E1, E2, and E3, respectively.

Table 8: SIM1 — normalized excess risk (mean ± sd, 30 replications), scenarios 1–5. Methods ordered as in Table 12; best per column in bold.

Method	HH$(2, 2)$	HH$(1.2, 1.1)$	Score-lin	Score-quad	Interval
FunPFN	.0046±.0023	.0073±.0033	**.0060±.0022**	**.0159±.0065**	**.0169±.0067**
TabPFN	.0118±.0049	.0139±.0040	.0170±.0040	.0567±.0253	.0254±.0119
FBNN	.0268±.0115	.0418±.0080	.0249±.0079	.0928±.0352	.0559±.0427
PLS	.0044±.0023	.0116±.0050	.0080±.0049	1.0568±.0459	1.0577±.0441
FPLS-Reg	**.0036±.0030**	.0102±.0037	.0061±.0034	1.0549±.0548	1.0629±.0565
Ridge	.0048±.0021	.0134±.0047	.0109±.0040	1.0197±.0266	1.0219±.0272
FPCA-Reg	.0043±.0035	.0093±.0041	.0078±.0040	1.0487±.0512	1.0512±.0459
FLR-penalized	.0047±.0025	**.0058±.0038**	.0074±.0045	1.0585±.0527	1.0654±.0590
RandomForest	.0548±.0117	.1353±.0149	.2789±.0283	.9064±.0973	.2039±.0555
SavGol-d2+PLS	1.1531±.0578	1.1440±.0624	.0268±.0099	1.0977±.0494	1.1222±.0759
Func-KNN	.0444±.0088	.1107±.0183	.2983±.0347	.7062±.0779	.1863±.0557
FDNN	.0117±.0041	.0248±.0079	.0417±.0110	.8936±.0927	.1984±.0420
MLP-2×32	.1275±.0630	.1836±.0774	.1905±.2766	.7454±.3136	.4670±.2579
Nadaraya–Watson	.0594±.0095	.2117±.0286	.2848±.0304	.6527±.0713	.1741±.0407

Table 9: SIM1 — scenarios 6–10 (same conventions as in Table 8).

Method	FLR $t-0.3$	Quadratic	SI-sin	Non-sep. op.	Brownian
FunPFN	.0051±.0022	**.0101±.0048**	**.0221±.0094**	**.0099±.0039**	.0042±.0023
TabPFN	.0138±.0041	.0463±.0714	.0369±.0071	.0252±.0062	.0104±.0029
FBNN	.0318±.0138	.0813±.0251	.1743±.0972	.0412±.0249	.0196±.0131
PLS	.0056±.0023	1.0594±.0832	.9274±.0672	1.0635±.0895	.0055±.0027
FPLS-Reg	**.0049±.0035**	1.0588±.0848	.9290±.0687	1.0641±.0813	**.0031±.0018**
Ridge	.0060±.0022	1.0178±.0263	.9108±.0457	1.0216±.0473	.0052±.0037
FPCA-Reg	.0054±.0046	1.0631±.0815	.9235±.0783	1.0648±.0929	.0039±.0035
FLR-penalized	.0072±.0056	1.1001±.0674	.9350±.0636	1.0589±.0692	.0045±.0026
RandomForest	.0657±.0151	.2330±.0488	.3107±.0437	.1871±.0482	.0337±.0079
SavGol-d2+PLS	1.1411±.0549	1.1306±.0736	1.1556±.0673	1.0932±.0626	1.1716±.0740
Func-KNN	.0808±.0145	.2791±.0607	.2693±.0400	.1768±.0473	.0396±.0104
FDNN	.0134±.0049	.3092±.0487	.7245±.0833	.1962±.0455	.0153±.0127
MLP-2×32	.1050±.0493	.4842±.2220	.8744±.2946	.4406±.1887	.0989±.0591
Nadaraya–Watson	.0885±.0145	.2606±.0574	.2558±.0376	.1553±.0409	.0307±.0100

G MODEL AND OPTIMIZATION HYPERPARAMETERS

Table 15 summarizes the model architecture, optimization settings, episode construction, task sampling and noise augmentation used in the experiments.

H WHICH COMPONENTS EARN THEIR PLACE

Figure 3 reports the RMSE of each ablated model after re-adaptation, relative to the adapted full model; the shaded band is the seed-noise floor ($\pm 1.3\%$). The neural-operator trunk dominates: feeding the transformer the standardized Fourier scores alone (`spectral_only`, row "FNO trunk") multiplies the RMSE by 2.56. Since the readout excludes the zero-frequency mode and frequencies above K, this control bounds what the released network can do without its trunk, not what a transformer trained on spectral scores could achieve. The attention readouts and the grid channel come next ($+17.9\%$ and $+12.5\%$, about 14 and 10 times the noise floor; both $p_{\mathrm{Holm}} < 10^{-9}$), after raising the RMSE by 59% and 52% at lesion. Without the grid channel the trunk is equivariant to circular shifts of the curve, so its features carry no position information; the largest losses occur on position-dependent functionals, such as eigenfunction scores and weighted derivative functionals. Keeping a single attention readout instead of four (row "Query 2 to Q") limits the loss to $+5.4\%$, so one query recovers about 70% of the gain of four. The remaining components are used by the trained model but are not required: removing the spectral readout or its support standardization raises the RMSE by 6.3% and 5.2% at lesion, but only by 1.0% and 1.6% after re-adaptation, respectively inside and just outside the noise band. The mean readout is unused ($+1.6\%$ at lesion, $+0.2\%$ after re-adaptation).

Table 10: SIM2 — log–log slopes of the excess risk over the three largest n.

Scenario (minimax)	**FunPFN**	**FPLS**	**FPCA**	**Ridge**	**MLP**	**FDNN**
FLR-HH $(2, 2)$ (-0.833)	-0.512	-1.049	-1.362	-0.883	-0.746	-0.400
FLR-HH $(1.2, 1.1)$ (-0.706)	-0.439	-0.692	-0.759	-0.838	-0.766	-0.297
Score-linear	-0.445	-1.158	-1.075	-0.833	-0.685	-0.201

Table 11: SIM3 — normalized excess risk vs. target-noise ratio ς/σ_f.

Scenario	**Method**	0	0.25	0.5	1.0
FLR-HH $(2, 2)$	FunPFN	.0001	.0035	.0088	.0374
	FDNN	.0057	.0111	.0210	.0434
	FPLS-Reg	.0000	.0023	.0087	.0341
	FPCA-Reg	.0000	.0025	.0088	.0330
	Ridge	.0000	.0031	.0077	.0291
Non-separable operator	FunPFN	.0023	.0096	.0233	.0867
	FDNN	.1883	.1834	.2085	.2707
	FPLS-Reg	1.0708	1.0723	1.0940	1.0893
	FPCA-Reg	1.0726	1.0858	1.0853	1.1087
	Ridge	1.0425	1.0201	1.0187	1.0330

Table 12: E1 — Test RMSE (mean ± sd over 30 random $75/25$ splits; inner 5-fold CV when applicable) on twelve tasks. Top: Tecator, Growth, Weather and Gasoline; bottom: Cookie, DTI and Aemet. The rightmost columns summarise all twelve tasks: mean wall-clock time per split in seconds (top) and average rank (bottom; ties share the average rank). FunPFN is zero-shot. Best per column bold, second best underlined (on Growth, FunPFN and TabPFN tie at the reported precision).

	Tecator						
Method	**fat**	**water**	**protein**	**Growth**	**Weather**	**Gasoline**	**Time (s)**
TabPFN	**0.97±0.15**	**1.00±0.16**	**0.66±0.08**	**4.84±0.74**	**0.151±0.044**	0.308±0.084	5.98
FunPFN (zero-shot)	1.28±0.17	1.46±0.18	1.03±0.13	**4.84±0.62**	0.162±0.054	**0.216±0.042**	0.025
Ridge	2.69±0.29	2.37±0.24	0.78±0.07	5.23±0.64	0.178±0.058	0.228±0.023	**<0.01**
PLS	2.42±0.53	2.34±0.58	0.67±0.10	5.38±0.71	0.187±0.062	0.224±0.028	0.19
FPLS-Reg	2.47±0.55	2.45±0.72	0.68±0.10	5.27±0.66	0.194±0.066	0.238±0.026	2.39
FPCA-Reg	2.53±0.50	2.40±0.54	0.68±0.08	5.37±0.67	0.196±0.061	0.248±0.039	8.22
FLR-penalized	2.63±0.53	2.48±0.54	0.72±0.08	5.30±0.63	0.208±0.085	0.218±0.039	3.84
FBNN	1.54±0.24	1.51±0.19	1.02±0.12	5.68±1.17	0.168±0.055	0.327±0.110	0.86
SavGol-d2+PLS	2.64±0.42	2.50±0.54	0.69±0.10	7.71±0.79	0.212±0.074	0.234±0.034	0.19
RandomForest	7.59±1.00	5.78±0.72	2.16±0.22	6.53±0.65	0.172±0.038	0.638±0.208	0.38
Func-k-NN	7.76±1.02	5.83±0.72	2.21±0.25	6.92±0.65	0.176±0.049	0.822±0.227	0.19
MLP-2×32	6.55±4.31	5.64±3.16	2.39±0.69	7.54±1.56	0.197±0.057	0.523±0.214	0.057
FDNN	9.97±1.03	7.58±0.80	2.58±0.19	5.99±1.15	0.187±0.052	0.625±0.198	1.78
Nadaraya–Watson	11.10±0.99	8.45±0.74	2.88±0.18	7.00±0.70	0.181±0.047	1.485±0.133	0.073

	Cookie						
Method	**fat**	**sucrose**	**flour**	**water**	**DTI**	**Aemet**	**Rank**
TabPFN	0.516±0.164	**1.26±0.41**	1.09±0.41	0.507±0.299	12.77±1.66	**0.909±0.348**	**3.29**
FunPFN (zero-shot)	0.350±0.078	1.58±0.61	1.28±0.47	0.505±0.257	12.69±1.82	1.005±0.390	4.63
Ridge	**0.295±0.070**	1.63±0.58	1.25±0.37	0.526±0.236	12.27±1.80	0.947±0.287	4.67
PLS	0.317±0.069	1.66±0.55	1.27±0.36	0.554±0.215	**12.02±1.84**	0.986±0.257	5.04
FPLS-Reg	0.336±0.070	1.64±0.56	1.25±0.38	0.566±0.217	12.07±1.83	0.982±0.276	5.38
FPCA-Reg	0.344±0.077	1.64±0.58	1.31±0.37	0.549±0.218	12.08±1.81	0.953±0.267	5.96
FLR-penalized	0.338±0.079	1.54±0.54	1.17±0.35	0.613±0.205	12.64±1.82	1.023±0.240	6.50
FBNN	0.718±0.156	1.66±0.70	1.39±0.55	0.677±0.347	12.45±2.05	1.003±0.297	7.08
SavGol-d2+PLS	0.407±0.090	1.63±0.60	**1.07±0.38**	**0.488±0.220**	14.62±1.55	1.026±0.257	7.92
RandomForest	1.569±0.225	3.46±0.36	2.16±0.28	0.927±0.189	13.09±1.79	0.969±0.303	9.92
Func-k-NN	1.602±0.181	3.79±0.36	2.42±0.23	1.059±0.162	12.51±2.05	0.955±0.304	10.42
MLP-2×32	1.323±0.584	3.05±1.29	2.20±0.56	0.969±0.331	13.34±2.37	0.985±0.323	10.83
FDNN	1.541±0.234	3.73±0.37	2.50±0.30	1.124±0.221	12.59±1.78	1.034±0.295	11.46
Nadaraya–Watson	1.933±0.180	3.89±0.37	2.71±0.30	1.446±0.232	12.74±1.72	0.936±0.282	11.92

Table 13: E2 — RMSE relative to its value at the pre-training resolution m=700 (mean over tasks, 10 repetitions). Values ≈1.0 indicate invariance; classical baselines are re-fit at every m, FunPFN is not.

Method	m=100	200	300	400	500	600	700	800	900	1000	1500
FunPFN (zero-shot)	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FDNN	0.95	0.96	0.97	0.98	0.99	0.99	1.00	1.01	1.01	1.02	1.03
FPCA-Reg	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FPLS-Reg	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Ridge	1.03	1.03	1.02	1.01	1.00	1.01	1.00	0.99	0.99	0.99	0.98

Table 14: E3 — RMSE vs. training-set size (pooled over the five tasks, 15 repetitions; test fixed).

n_{tr}	**FunPFN**	**FPLS-Reg**	**FPCA-Reg**	**Ridge**	**FDNN**	**TabPFN**
20	**3.102**	3.344	3.421	3.383	6.360	3.907
40	**2.781**	3.572	3.587	3.386	7.321	3.047
80	1.493	2.228	2.246	2.284	7.273	**1.328**
120	1.354	2.041	2.045	2.075	6.854	**1.073**
160	1.280	1.914	1.907	1.985	6.741	**0.894**

Table 15: Hyperparameters of FunPFN and of its pretraining.

FNO encoder	5 layers, width 384, 32 modes; input $(x(t_j), t_j)$; channel MLP ratio $1/2$
Pooling	mean + 4-query attention pooling (softmax over grid points)
Spectral readout	$K = 32$ ($k = 1, \ldots, 32$), support-standardized, learnable gains
Transformer	width 768 (= embedding); 12 layers, 12 heads, FFN 4×, pre-norm, GELU, no dropout
Head	Gaussian $(\mu, \log \sigma^2)$, $\log \sigma^2 \in [-10, 5]$; NLL Huberized beyond 10σ
Parameters	136.2M (encoder 50.2M, transformer 85.9M)
Optimization	AdamW $(0.9, 0.95)$, wd 10^{-4}, clip 1.0; fp16 mixed precision
Schedule	peak lr 3×10^{-4}, linear warmup over 5%, cosine decay to 0
Steps	300K optimizer steps, 4 accumulated micro-batches per step, data-parallel
Episodes	S log-unif. on $[8, 512]$, $Q = 64$; m log-unif. on $[64, 700]$, $m = 700$ w.p. 0.2
Batching	≤8 episodes; per GPU, $B(S{+}Q) \leq 2{,}600$ tokens, $B(S{+}Q)\, m \leq 8.2\times10^5$ points
Task sampling	fresh task w.p. 0.7 / frozen bank (3,130 tasks) w.p. 0.3; family weights ×3/×2
Target noise	level log-unif. on $[0.02, 1] \cdot \mathrm{sd}(y)$; Gaussian / Student-$t$ / heterosc. (0.7/0.15/0.15)
Input noise	w.p. $1/2$, $\mathcal{U}(0, 0.3) \cdot \mathrm{sd}(x)$, added at the observed resolution

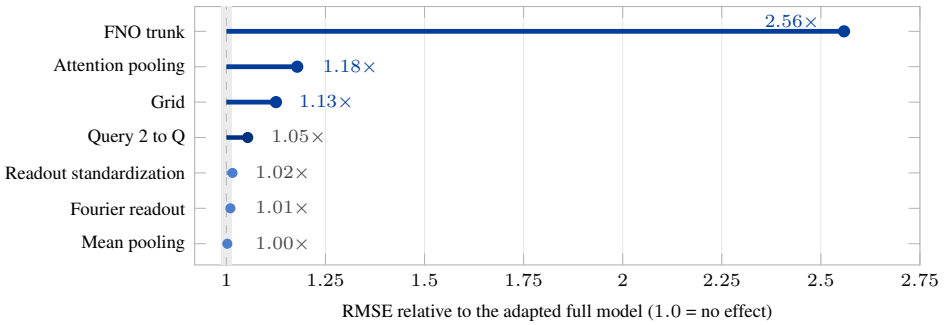


Figure 3: Ablation — Each component is removed from the full model and the resulting model is re-adapted, then evaluated on the 312-episode report set. Each dot shows the adapted score divided by that of the adapted full model; the dashed line at 1 therefore indicates no effect, and values above 1 indicate worse performance. The grey band shows $1 \pm 1.29\%$, where 1.29% is the relative standard deviation of the adapted full-model score across two replicates. It is included as a visual estimate of seed noise, not as a confidence interval. Ablations are ordered by decreasing effect from top to bottom