LEARNING LOCALITY-AWARE BASIS FUNCTIONS VIA SPLINE-WAVELET KOLMOGOROV-ARNOLD NETWORKS FOR FUNCTIONAL DATA CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Functional data analysis (FDA) is widely used to model continuous processes from discrete observations across scientific domains. However, conventional FDA methods often rely on fixed basis families, leading to inefficient representations when discriminative patterns are concentrated within localized regions. To address this limitation, this work proposes **Spline-Wavelet KAN for Adaptive Basis Learning (SKAB)**, a neural method that learns locality-aware basis functions for functional data classification. Motivated by an observed structural correspondence between FDA and Kolmogorov-Arnold Networks (KANs), SKAB formulates a KAN as a Coordinate-to-Basis Generator (**CBG**) that maps each functional-domain coordinate to basis functions. To better capture localized patterns, Spline-Wavelet KAN is introduced to parameterize the CBG using compactly supported B-spline wavelet atoms as edge functions. Each functional observation is projected onto the learned bases to obtain a coefficient representation for downstream prediction. Experiments on synthetic and real-world datasets demonstrate that SKAB achieves competitive classification performance, with particular advantages for data containing localized discriminative patterns. The code is available at this link.

1 INTRODUCTION

Functional observations arise as curves, spectra, and trajectories sampled over a continuous domain (Meng et al., 2026). Unlike conventional vectors, they are viewed as discrete observations of an underlying continuous function, so values observed at nearby points are usually closely related (Wu et al., 2024). Although each function is observed on a finite grid, its underlying representation is intrinsically infinite-dimensional. Therefore, FDA typically represents each observation by a finite coefficient vector with respect to a shared set of basis functions (Wang et al., 2024). However, the effectiveness of basis expansion depends strongly on whether the selected basis functions match the structural characteristics of the data (Basna et al., 2022; 2024). Conventional approaches typically adopt predefined families, such as Fourier bases, B-splines, wavelets, or functional principal components (Seong et al., 2026). Although mathematically convenient, these bases are fixed before learning, so their ability to adapt to heterogeneous patterns is limited. Recently, neural FDA methods have begun to learn task-adaptive basis functions jointly with downstream prediction (Shang & Zeng, 2025; Jiang et al., 2026). Yet many neural basis generators remain generic function approximators, leaving the structural bias of the learned bases weakly controlled (Yang et al., 2022).

A key missing structural bias is locality. Recent functional classification studies show that label-relevant information may appear only in selected portions of the domain rather than across the whole curve (Wang et al., 2026; Boschi et al., 2024). These localized signals commonly take the form of peaks, transient shape changes, or narrow spectral bands in biomedical signals and spectroscopic measurements (Eslami, 2024; Mishra & Bhusnur, 2023). For example, as shown in Fig. 1(a), different classes share similar global trends but differ within a narrow middle-domain region in the EthanolLevel dataset (Large et al., 2018). As illustrated in Fig. 1(b), globally supported bases, such as Fourier bases, respond across the entire domain and may underrepresent such localized patterns. B-splines provide local support but may require multiple neighboring functions to represent localized patterns under a fixed knot configuration (Shi et al., 2025). Similarly, Functional Principal

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.

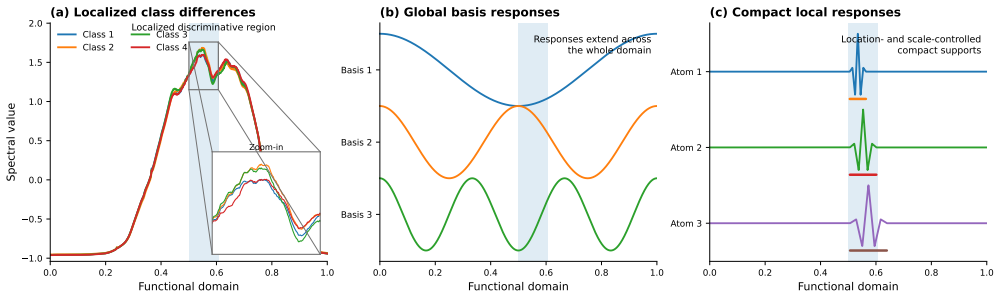


Figure 1: Motivation for adaptive basis learning in functional data with localized patterns. **(a)** Localized class differences in the EthanolLevel dataset. **(b)** Global basis responses extend over the full domain. **(c)** Localized atom responses provide compact local support.

Component Analysis (FPCA) captures dominant global variation (Shang, 2014) and may overlook localized task-relevant structures (Yang et al., 2022). Localized FPCA introduces spatially concentrated components (Chen & Lei, 2015), but it remains unsupervised and is not jointly optimized for classification. As illustrated in Fig. 1(c), these limitations call for task-adaptive basis functions with explicit localized control.

Inspired by the ability of wavelets to characterize localized variation, recent studies have explored wavelet-based representations for FDA (Amato et al., 2025; Morettin & Pinheiro, 2017). By translating and dilating a mother wavelet, these representations generate atoms at different locations and scales, enabling localized and multi-scale representation of functional patterns (Vidal & Aguilera, 2025). In particular, compactly supported spline wavelets confine each wavelet atom to a finite region of the domain (Makarov & Makarova, 2025). Wavelet-based functional classification and smoothing have consequently shown advantages for localized patterns and oscillations (Eslami, 2024). However, most wavelet-based methods rely on predefined transforms whose atoms are not learned jointly with the predictive objective (Sousa et al., 2026). Kolmogorov-Arnold Networks (KANs) offer a suitable structure for addressing this gap since they construct network mappings through learnable univariate edge functions (Liu et al., 2025). By restricting the network input to the functional coordinate and using multiple output channels, a KAN can be reformulated as a generator of shared basis functions. Parameterizing these edge functions with compactly supported spline-wavelet atoms combines the explicit locality and multi-scale structure of wavelets with task-adaptive neural basis learning.

Based on these observations, we propose **Spline-Wavelet KAN for Adaptive Basis Learning (SKAB)**, a neural method for learning task-adaptive basis functions for functional data classification. SKAB first introduces a **Coordinate-to-Basis Generator** (CBG) that maps domain coordinates to shared learned basis functions. To encode an explicit locality prior, **Spline-Wavelet KAN** (SWK) is developed, which parameterizes KAN edge functions using compactly supported spline-wavelet atoms. A data-driven **Center-Scale Dictionary Selection** (CSDS) strategy is further designed to initialize these atoms in statistically informative regions. Finally, each functional observation is projected onto the learned basis functions via ridge-regularized least squares to obtain coefficient representations. The main contributions are summarized as follows:

- We propose **SKAB**, a locality-aware basis learning method for functional data classification. Specifically, SKAB employs a **Coordinate-to-Basis Generator** (CBG), which maps a scalar coordinate to values of shared task-adaptive basis functions.
- We design **Spline-Wavelet KAN** (SWK) to parameterize the CBG with compactly supported, multi-scale spline wavelet atoms, providing an explicit inductive bias for localized functional patterns.
- We introduce **Center-Scale Dictionary Selection** (CSDS), which uses training-set statistics to initialize atom locations and scales of SWK in informative regions.
- Experiments on synthetic and real-world datasets demonstrate the effectiveness of SKAB, with particular advantages on data exhibiting localized discriminative patterns.

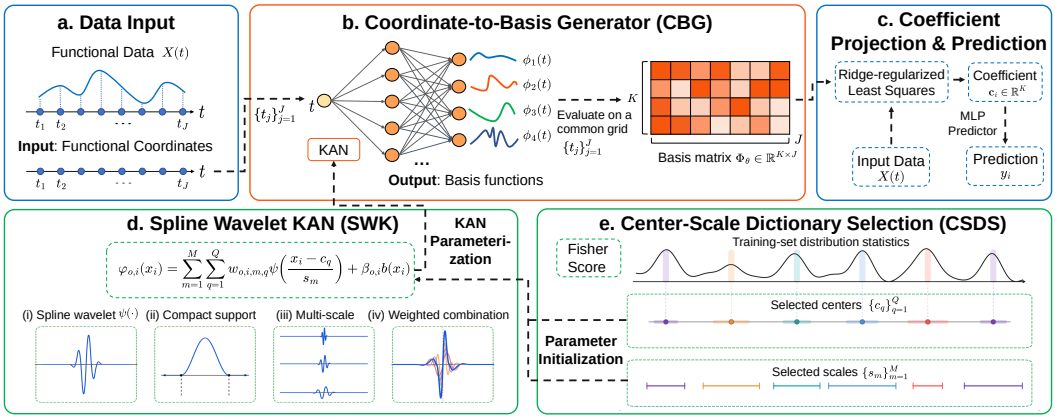


Figure 2: Overview of the proposed SKAB method. **(a)** Functional observations are discretized on a common grid $\{t_j\}_{j=1}^{J}$. **(b)** The Coordinate-to-Basis Generator (CBG) takes functional-domain coordinates as input and outputs K basis functions $\{\phi_k(\cdot)\}_{k=1}^{K}$. The shared basis matrix is then constructed through grid evaluation. The CBG is parameterized by **(d)** Spline-Wavelet KAN (SWK), whose edge functions are expressed as weighted sums of compactly supported, multi-scale spline-wavelet atoms. The atom centers and scales in SWK are initialized by **(e)** Center-Scale Dictionary Selection (CSDS) using training-set statistics derived from Fisher scores. **(c)** Ridge-regularized projection maps each input observation onto the learned bases, and the resulting coefficient representation is used for downstream classification.

2 Related Work

Neural models for functional data. A growing number of neural FDA methods have been developed to learn predictive representations from functional observations (Wang et al., 2024). AdaFNN generates adaptive bases using MLP-based micro-networks (Yao et al., 2021); FNN introduces shift-invariant models that preserve smoothness (Heinrichs et al., 2023); DeepFRC integrates functional registration with classification (Jiang et al., 2026), while FSFC investigates functional feature selection (Boschi et al., 2024). Most closely related, AdaFNN demonstrates supervised basis learning, but the bases are learned by separate micro-networks and its generic MLP parameterization does not encode a localized structural prior. In contrast, SKAB constructs task-adaptive bases using compactly supported spline-wavelet KAN edge functions.

Kolmogorov-Arnold Networks. Recent KAN studies mainly differ in their edge-function parameterizations. The original KAN employs B-splines (Liu et al., 2025); FastKAN uses Gaussian radial basis functions (Li, 2024); ReLU-KAN adopts ReLU-based constructions (Qiu et al., 2025); HRKAN introduces higher-order ReLU basis functions to retain efficient matrix operations (So & Yung, 2025), and BSRBF-KAN combines spline and radial basis functions (Ta, 2024). These variants are primarily designed for generic function approximation rather than FDA basis learning. SKAB instead reformulates a KAN as a coordinate-to-basis generator.

Wavelets and localized functional representations. Localized functional patterns have been modeled through wavelet coefficient selection (Chang et al., 2014), wavelet-based clustering (Amato et al., 2025), localized FPCA (Chen & Lei, 2015), and hybrid representations that combine FPCA with wavelet features (Yang et al., 2022). On the other hand, Wav-KAN introduces wavelets as learnable KAN edge functions (Bozorgasl & Chen, 2024). However, conventional wavelet FDA relies mainly on predefined transforms, whereas Wav-KAN primarily employs non-compactly supported wavelets, lacking explicit control over localized support regions (Fatema et al., 2026). SKAB addresses this limitation through compactly supported spline-wavelet atoms and CSDS strategy.

3 METHOD

3.1 PROBLEM DEFINITION AND MOTIVATION

We aim to represent and classify functional data. Given a functional dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^N$, each $X_i : \mathcal{T} \to \mathbb{R}$ denotes a functional observation defined on a compact domain $\mathcal{T} \subset \mathbb{R}$, and $y_i \in \{1, \ldots, C\}$ denotes its class label. In practice, X_i is observed on a common finite grid $\mathcal{T}_J = \{t_1, \ldots, t_J\}$. The resulting discretized observation vector is denoted by $\mathbf{x}_i = [X_i(t_1), \ldots, X_i(t_J)]^\top \in \mathbb{R}^J$, where its j-th entry is the observed function value $x_{i,j} = X_i(t_j)$.

A classical FDA representation expands each observation over a set of basis functions (Rossi & Conan-Guez, 2005):

$$X_i(t) \approx \sum_{k=1}^{K} c_{i,k}\phi_k(t), \tag{1}$$

where $K \ll J$, $\{\phi_k(\cdot)\}_{k=1}^K$ are shared across observations, and $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,K})^\top \in \mathbb{R}^K$ is the coefficient representation of the i-th observation.

Rather than pre-specifying the basis family, we aim to learn shared, task-adaptive basis functions while retaining this classical basis-expansion representation. The motivation arises from a structural correspondence between FDA basis expansion and the Kolmogorov-Arnold representation. For a continuous multivariate function $f : [0, 1]^n \to \mathbb{R}$, the Kolmogorov-Arnold representation theorem (Kolmogorov, 1963) states that

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Psi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{2}$$

where $\phi_{q,p} : [0, 1] \to \mathbb{R}$ and $\Psi_q : \mathbb{R} \to \mathbb{R}$ are univariate functions. For the univariate case $n = 1$, the theorem gives a scalar-input representation with $2n + 1 = 3$ terms. KANs further generalize this structure to network architectures with arbitrary widths and depths (Liu et al., 2025). Under this generalized architecture, a scalar-input two-layer KAN with hidden width K can be written as

$$f(t) = \sum_{k=1}^{K} \Psi_k(\phi_k(t)), \tag{3}$$

where $t \in \mathcal{T}$, $\phi_k : \mathcal{T} \to \mathbb{R}$ denotes the k-th inner function, and $\Psi_k : \mathbb{R} \to \mathbb{R}$ denotes its corresponding outer function. Here, K is regarded as the number of learned basis functions. For any coefficient vector $\mathbf{c} = (c_1, \ldots, c_K)^\top$, restricting the outer functions to linear mappings gives

$$\Psi_k(u) = c_k u \quad \Longrightarrow \quad f(t) = \sum_{k=1}^{K} c_k \phi_k(t), \tag{4}$$

which has the same form as the FDA basis expansion in Eq. 1.

Why KAN is a suitable basis generator for FDA. The key observation is that the inner functions $\{\phi_k\}_{k=1}^K$ of a scalar-input KAN can be interpreted as shared FDA basis functions, while $\{c_k\}_{k=1}^K$ are the corresponding coefficients. It is worth noting that, generally, any sufficiently expressive scalar-to-vector map $B_\Theta : \mathcal{T} \to \mathbb{R}^K$, including an MLP, can parameterize K shared basis functions. The key reason of choosing a KAN is that its learnable univariate edge functions provide an explicit structural interface through which inductive priors can be encoded. SKAB exploits this interface by parameterizing the edge functions with compactly supported, multi-scale spline-wavelet atoms (Section 3.3), thereby introducing a locality-aware inductive bias into the basis generator. The overall pipeline of SKAB is illustrated in Figure 2.

3.2 COORDINATE-TO-BASIS GENERATOR

We implement the Coordinate-to-Basis Generator (CBG) as an L-layer scalar-input KAN with architecture $[1, n_1, \ldots, n_{L-1}, K]$. Given a functional-domain coordinate $t \in \mathcal{T}$, it outputs

$$B_\Theta(t) = [\phi_1(t; \Theta), \ldots, \phi_K(t; \Theta)]^\top \in \mathbb{R}^K, \tag{5}$$

where Θ denotes the learnable parameters of CBG and $\phi_k(t;\Theta)$ is the value of the k-th learned basis function at coordinate t. Evaluating the CBG over the grid $\mathcal{T}_J = \{t_j\}_{j=1}^J$ gives the basis matrix $\Phi_\Theta = [B_\Theta(t_1), \ldots, B_\Theta(t_J)] \in \mathbb{R}^{K\times J}$. Each row of Φ_Θ represents one learned function evaluated over the common grid. Φ_Θ is shared across all observations, whereas each observation obtains its own coefficient vector through the projection introduced in Section 3.5.

3.3 SPLINE-WAVELET KAN

Although CBG enables task-adaptive basis generation, its ability to capture localized patterns still depends on the parameterization of the KAN edge functions. To introduce an explicit local inductive bias, we parameterize the KAN edge functions using compactly supported spline wavelet atoms. A function is compactly supported if it is identically zero outside a bounded interval, so its response is confined to a finite region of the domain. Specifically, we adopt the rescaled order-2 Chui-Wang spline wavelet ψ whose support is $[-1/2, 1/2]$ (Chui, 1992). Its complete definition is provided in Appendix D. The edge function from input node i to output node o is parameterized as

$$\varphi_{o,i}(x_i) = \sum_{m=1}^{M}\sum_{q=1}^{Q} w_{o,i,m,q}\psi\left(\frac{x_i - c_q}{s_m}\right) + \beta_{o,i}b(x_i), \tag{6}$$

where $b(x) = \mathrm{SiLU}(x)$ is the residual base function, $\beta_{o,i}$ and $w_{o,i,m,q}$ are edge-specific trainable weights, and c_q and $s_m > 0$ denote the q-th center and the m-th scale, respectively. To guarantee positivity and avoid singular support widths, an unconstrained parameter $\rho_m \in \mathbb{R}$ is introduced and s_m is optimized via $s_m = s_{\min} + \mathrm{softplus}(\rho_m)$ where $s_{\min} = 10^{-3}$. The functional domain is normalized to $[0, 1]$, so every scale is strictly positive and bounded below by $s_{\min}$.

The center and scale dictionary is shared within each layer, while different edges learn distinct linear combinations of its atoms. Multi-scale representation is realized by pairing every center with the M scales, so each edge can independently select and combine the resulting MQ localized atoms. In SKAB, the first SWK layer takes the one-dimensional functional coordinate t as input, while subsequent layers operate on learned hidden representations.

Because ψ is compactly supported, each translated and scaled atom is active only within

$$\mathrm{supp}\left[\psi\left(\frac{x - c_q}{s_m}\right)\right] = \left[c_q - \frac{s_m}{2}, c_q + \frac{s_m}{2}\right]. \tag{7}$$

Thus, the center determines the response location, while the scale controls the support width. Following (Liu et al., 2025), each SWK layer additionally retains a trainable SiLU base branch to provide a smooth global component, while the spline-wavelet branch provides controlled responses to localized patterns. The centers and scales are initialized by the CSDS strategy introduced in Section 3.4 and subsequently optimized jointly with the network.

Preference for Localized Deviations. The adopted order-2 spline wavelet has two vanishing moments:

$$\int_{\mathbb{R}} \psi(u)\, du = \int_{\mathbb{R}} u\psi(u)\, du = 0. \tag{8}$$

Consequently, for an atom whose support is fully contained within the functional domain, its inner product with a linear function over its support is zero. This property suppresses low-order local background trends and makes the spline-wavelet component more responsive to localized deviations.

3.4 CENTER-SCALE DICTIONARY SELECTION

The SWK parameterization explicitly exposes the center and scale of each local atom, but these parameters still require an appropriate initialization. Uniform initialization may allocate atoms to weakly informative regions or repeatedly cover the same local interval. Therefore, we propose **Center-Scale Dictionary Selection** (CSDS) to initialize the first-layer dictionary from training-set statistics before training. To locate class-discriminative regions of the functional domain, we assign each observed coordinate t_j a Fisher score that compares its between-class and within-class

variations (Sun et al., 2022; Fisher, 1936). Let $\widetilde{S}_j$ denote the normalized and smoothed score. These scores are converted into a relevance distribution:

$$p_j = (1-\alpha)\frac{\widetilde{S}_j}{\sum_{\ell=1}^{J}\widetilde{S}_\ell} + \frac{\alpha}{J}, \tag{9}$$

where p_j represents the relevance mass at t_j, and $\alpha \in [0,1]$ controls the uniform component used to maintain coverage of the full domain. The Q center candidates are chosen as weighted quantiles at probability levels $\xi_q = (q-\frac{1}{2})/Q$, $q = 1, \ldots, Q$. This places one candidate near the center of each equal-mass segment, concentrating centers in more discriminative regions.

The compact support of SWK atoms (Eq. 7) is further used to avoid redundant initialization. Excessively close centers would produce highly overlapping local atoms. Therefore, we retain the weighted-quantile centers when they are sufficiently separated. Otherwise, support-aware non-maximum suppression is used to select high-relevance coordinates under a minimum-distance constraint. After sorting the selected centers, we compute their median adjacent spacing $\Delta_c = \mathrm{median}_{q=1,\ldots,Q-1}(c_{q+1} - c_q)$. The M initial scales are obtained from relative multiples of this spacing $s_m = r_m \Delta_c$, where $m = 1, \ldots, M$. The selected centers and scales initialize the first SWK layer and remain trainable during optimization. CSDS is applied only to this layer because its input directly corresponds to the original functional coordinate. The remaining SWK layers are initialized with uniformly spaced centers and a predefined multiscale bank. All statistics are computed exclusively from the training split. Further details of CSDS are provided in Appendix E.

3.5 Coefficient Projection and Prediction

The learned basis matrix is shared across observations, whereas the coefficient vectors characterize their individual functional representations. Evaluating CBG on the observation grid yields the basis matrix $\Phi_\Theta \in \mathbb{R}^{K\times J}$. For each observation $\mathbf{x}_i \in \mathbb{R}^J$, we obtain its coefficient vector by ridge-regularized least squares (Ramsay, 2005):

$$\widehat{\mathbf{c}}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^K} \left\|\mathbf{x}_i - \Phi_\Theta^\top \mathbf{c}\right\|_2^2 + \lambda \left\|\mathbf{c}\right\|_2^2. \tag{10}$$

We set $\lambda = 10^{-6}$ by default and solve the resulting regularized system using a Cholesky-based solver without explicit matrix inversion. The coefficients are passed to a lightweight MLP predictor, $\widehat{\mathbf{y}}_i = g(\widehat{\mathbf{c}}_i)$, and the model is trained with cross-entropy loss. The differentiable coefficient solve allows the predictive loss to jointly optimize the basis generator and downstream prediction.

Regularization. We additionally apply a small smoothness penalty on the second differences of the learned bases, with weight $\gamma = 10^{-5}$, to discourage oscillatory basis functions. The complete definition is provided in Appendix F.

4 Experiments

We evaluate SKAB on synthetic and real-world classification tasks, focusing on predictive performance and the recovery of localized patterns. The evaluation covers both functional and time-series data, as many time series can also be treated as discrete observations of underlying functions.

4.1 Experimental Setup

We compare against four groups of baselines. **(1) Generic neural predictors**, including MLP (Rossi & Conan-Guez, 2005) and KAN (Liu et al., 2025); **(2) Classical functional representations**, including B-spline bases (Ramsay, 2005), Fourier bases (Kokoszka & Reimherr, 2017), and FPCA (Shang, 2014); **(3) Neural FDA and functional learning methods**, including AdaFNN (Yao et al., 2021), Functional Neural Networks (Heinrichs et al., 2023), FSFC (Boschi et al., 2024), FuncAE (Wu et al., 2024), and DeepFRC (Jiang et al., 2026); **(4) Recent time-series classification methods**, including TSLANet (Eldele et al., 2024),and TimeSliver (Pandey et al., 2026). These methods provide modern sequence-learning baselines that are not specifically designed for functional data.

When applying these methods, we follow their original structural design and the code provided in the papers unless otherwise specified. The number of basis functions is fixed to $K = 6$ by default.

Table 1: Classification performance on DJ-MIXTURE.

Method	ACC	AUC	Method	ACC	AUC
MLP	0.9117 ± 0.0095	0.9930 ± 0.0014	FuncAE	0.2323 ± 0.0130	0.5077 ± 0.0070
KAN	0.9270 ± 0.0033	$\underline{0.9860 \pm 0.0006}$	DeepFRC	0.6794 ± 0.0234	0.6039 ± 0.0748
B-spline	0.3983 ± 0.0150	0.5780 ± 0.0263	FSFC	0.5663 ± 0.0771	0.8203 ± 0.0330
Fourier	0.4990 ± 0.0077	0.6093 ± 0.0367	TSLANet	0.7607 ± 0.3265	0.7992 ± 0.2144
FPCA	0.7543 ± 0.0121	0.9468 ± 0.0046	TimeSliver	0.6020 ± 0.0190	0.6042 ± 0.0167
AdaFNN	0.4987 ± 0.0166	0.6987 ± 0.0145	SKAB	$\mathbf{0.9743 \pm 0.0124}$	$\mathbf{0.9981 \pm 0.0020}$
FNN	$\underline{0.9550 \pm 0.0423}$	0.9676 ± 0.0132			

Results are reported as mean $\pm$ standard deviation over ten runs. Bold and underlined entries denote the best and second-best results, with lower standard deviation breaking ties in the mean. Data preprocessing details, model configurations and training details are provided in Appendix G.

4.2 RESULTS ON SYNTHETIC DATA

We first evaluate SKAB on a controlled synthetic dataset with known task-relevant structures. This setting tests both classification performance and whether the learned basis functions respond to localized discriminative regions.

Data generation. We construct DJ-MIXTURE, a five-class functional dataset based on six localized atoms inspired by Donoho-Johnstone Bumps signals (Johnstone & Silverman, 1997). Each class combines different sparse combinations of localized atoms, while perturbations and observation noise introduce within-class variation. We use a few-shot setting with 10 training curves per class. Full generation details are provided in Appendix G.1.

Classification performance. As shown in Table 1, SKAB achieves the best ACC and AUC, exceeding the strongest competing ACC by 1.93. Its advantage over MLP, KAN, and functional baselines demonstrates the benefit of explicitly modeling localized functional structures.

Analysis of learned localized structure. Fig. 3 examines whether SKAB recovers and uses the localized structures in DJ-MIXTURE. Panel **(a)** establishes the known localized reference structures that define the synthetic task. Panel **(b)** evaluates whether the training-set Fisher score guides CSDS toward informative locations. Panel **(c)** shows whether the resulting center-scale dictionary retains localized responses after optimization. Panel **(d)** measures whether the learned basis components are used in the predictive representation through $\mathbb{E}_i[|c_{ik}\phi_k(t)|]$. The dominant contribution region of every learned basis function contains at least one localized atom center, indicating that these localized atoms contain discriminative information. Together, the panels show that CSDS allocates the dictionary to task-relevant regions and that these regions contribute to the final coefficient representation.

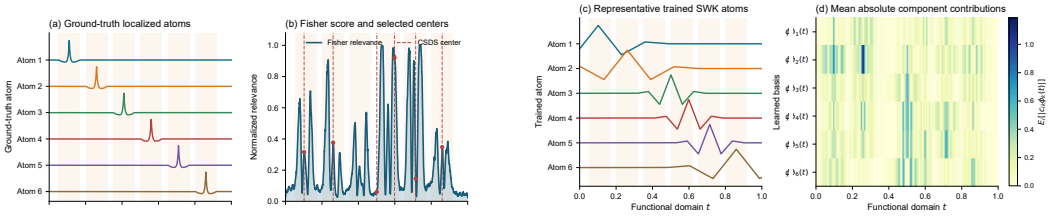


Figure 3: Analysis of learned localized structure on DJ-MIXTURE. **(a)** Six ground-truth localized atoms and their reference intervals. **(b)** Training-set Fisher score and centers selected by CSDS. **(c)** Representative trained spline-wavelet dictionary atoms, with one highest-weighted scale shown for each learned center. **(d)** Mean absolute component contributions $\mathbb{E}_i[|c_{ik}\phi_k(t)|]$.

Table 2: Classification accuracy on six real-world datasets.

Method	Phoneme	Tecator	Medflies	ECG5000	EthanolLevel	EOGHorizontalSignal
MLP	0.9271 ± 0.0101	0.8849 ± 0.1089	0.5850 ± 0.0748	0.9382 ± 0.0012	$\underline{0.5906 \pm 0.0237}$	0.3840 ± 0.0163
KAN	0.9270 ± 0.0094	0.9302 ± 0.0380	0.5523 ± 0.0372	0.9387 ± 0.0016	0.5188 ± 0.0193	0.4298 ± 0.0173
B-spline	0.8997 ± 0.0072	0.9465 ± 0.0348	$\underline{0.5869 \pm 0.0502}$	0.9190 ± 0.0037	0.3066 ± 0.0136	0.3829 ± 0.0242
Fourier	0.9096 ± 0.0060	0.9488 ± 0.0325	0.5804 ± 0.0558	0.9262 ± 0.0030	0.3370 ± 0.0178	0.3760 ± 0.0152
FPCA	0.9163 ± 0.0076	0.9581 ± 0.0286	0.5589 ± 0.0462	0.9350 ± 0.0027	0.3574 ± 0.0149	0.4157 ± 0.0161
AdaFNN	0.9216 ± 0.0085	0.8116 ± 0.1142	0.5682 ± 0.0468	0.9219 ± 0.0052	0.2912 ± 0.0222	0.3854 ± 0.0459
FNN	0.9191 ± 0.0077	0.8721 ± 0.0505	0.5748 ± 0.0374	0.6252 ± 0.0121	0.3650 ± 0.0364	0.4141 ± 0.0233
FuncAE	0.8836 ± 0.0094	0.7279 ± 0.0581	0.5766 ± 0.0425	0.9180 ± 0.0039	0.3452 ± 0.0270	0.4287 ± 0.0241
FSFC	0.8053 ± 0.0105	0.6535 ± 0.0689	0.5710 ± 0.0370	0.9029 ± 0.0021	0.2944 ± 0.0108	0.2950 ± 0.0262
DeepFRC	0.8698 ± 0.0437	$\underline{0.9623 \pm 0.0278}$	0.5159 ± 0.0440	0.9263 ± 0.0084	0.3530 ± 0.0716	0.1301 ± 0.0334
TSLANet	$\underline{0.9293 \pm 0.0053}$	0.9621 ± 0.0213	0.5514 ± 0.0460	$\underline{0.9398 \pm 0.0023}$	0.4624 ± 0.0856	$\mathbf{0.5221 \pm 0.0212}$
TimeSliver	0.7845 ± 0.0300	0.6953 ± 0.0643	0.5262 ± 0.0458	0.8908 ± 0.0095	0.2534 ± 0.0244	0.1975 ± 0.0139
SKAB	$\mathbf{0.9309 \pm 0.0072}$	$\mathbf{0.9744 \pm 0.0196}$	$\mathbf{0.6219 \pm 0.0342}$	$\mathbf{0.9435 \pm 0.0014}$	$\mathbf{0.8160 \pm 0.0148}$	$\underline{0.4624 \pm 0.0302}$

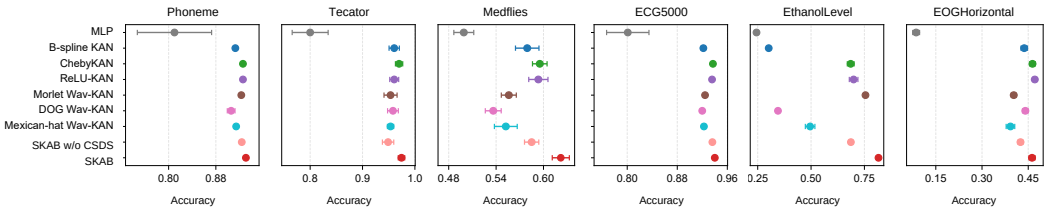


Figure 4: Accuracy comparison of SKAB and ablation variants across six datasets. Points denote mean test accuracy over ten runs, and horizontal error bars indicate the standard error.

4.3 Results on Real-World Data

We consider three established functional datasets, including Phoneme (Hastie et al., 1995), Tecator (Ferraty & Vieu, 2006), and Medflies (Chiou et al., 2003), and three datasets from the UCR Archive, including ECG5000 (Goldberger et al., 2000), EthanolLevel (Lines et al., 2016), and EOGHorizontalSignal (Fang & Shinozaki, 2018). Together, they cover spectral, biological, physiological, and movement signals containing localized class-related variation.

Overall performance. Table 2 shows that SKAB ranks first on five of the six datasets. It consistently outperforms all classical functional representations and neural FDA methods, while also surpassing recent time-series classifiers on five datasets, demonstrating that its advantage extends beyond comparisons with conventional FDA baselines. On EOGHorizontalSignal, SKAB outperforms other FDA basis-learning methods but specialized time-series classifiers with stronger full-sequence temporal modeling perform better. Additional comparisons on localized FPCA and supervised db4 wavelet bases are reported in Appendix B. Sensitivity analyses of the number of bases K, the ridge and smoothness regularization and the CSDS strategy are provided in Appendix A.

Advantages of adaptive localized representations. The improvement is most remarkable on EthanolLevel, where SKAB exceeds the strongest competing method by 38%. As illustrated in Figure 1, the class differences in EthanolLevel are concentrated within a localized spectral region. This result supports the motivation of SKAB: learning where task-relevant patterns occur and at which scales can provide a more effective representation than fixed bases.

4.4 Ablation Studies

We evaluate the two principal design choices of SKAB: the spline-wavelet parameterization of the CBG and the CSDS strategy. For the first comparison, we replace SWK with B-spline KAN, ChebyKAN, ReLU-KAN, MLP, and existing Wav-KAN variants using Morlet, derivative-of-Gaussian (DOG), or Mexican-hat wavelets. For the second comparison, SKAB w/o CSDS retains the same SWK architecture but replaces the data-driven dictionary with uniformly distributed centers and scales.

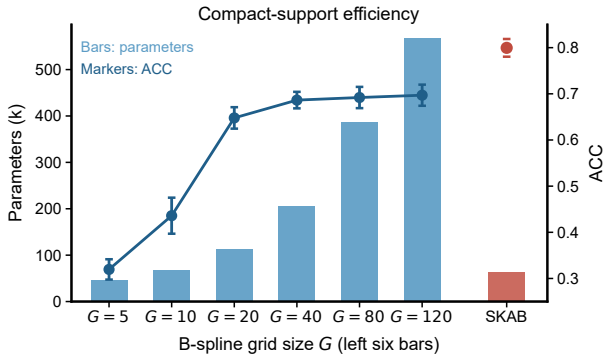


Figure 5: Compact-support efficiency on EthanolLevel. Bars show the total number of trainable parameters and markers show mean ACC with standard deviation. The first six settings are order-two B-spline KANs with increasing grid size G; the final setting is SKAB.

Fig. 4 compares the mean test accuracy of all variants over ten runs on six datasets. SKAB achieves the highest ACC on five of the six datasets, outperforming alternative KAN parameterizations, including the non-compactly supported wavelet variants. Notably, the MLP CBG is ranked last on all datasets. As illustrated in Eq. 4, KANs can encode priors on the edge-wise univariate function structure. The consistently weak MLP result provides empirical support for using this structured KAN parameterization rather than a generic MLP with coordinate inputs to learn shared basis functions. Moreover, CSDS improves ACC over uniform initialization on every dataset, supporting the benefit of allocating the center-scale dictionary to task-relevant locations and supports. Detailed comparisons are provided in Appendix H.

Analysis of compact-support efficiency. We further examine why the compactly supported spline-wavelet dictionary is effective for localized discrimination. Figure 5 compares the proposed spline-wavelet KAN dictionary with an order-2 B-spline KAN on EthanolLevel. For an order-two spline-wavelet atom, the support width is exactly s. Across ten runs, SKAB learns three mean support widths of 0.0810, 0.1967, and 0.4055. The narrowest width is particularly important because it provides responses to short-range spectral variation. Since an order-two B-spline atom has support width $3/G$, where G is the grid size, matching this finest scale requires approximately $G = 38$, with $G = 40$ as the nearest evaluated setting. At this comparable local resolution, B-spline KAN obtains 0.6864 ACC with $205{,}060$ parameters, whereas SKAB obtains 0.8160 ACC with $64{,}217$ parameters. Further increasing the B-spline resolution to $G = 120$ reduces its support width to 0.025 and raises the parameter count to $568{,}580$, but improves ACC only to 0.6970. These results show that the compact multi-scale spline-wavelet dictionary represents localized spectral information more parameter-efficiently than uniformly densifying the B-spline grids of a standard KAN.

Basis Energy Maps. Appendix C shows that SKAB concentrates its learned basis energy on the localized interval with a high Fisher score on EthanolLevel, while assigning little energy to the remaining low-score regions. Here, the energy of basis ϕ_k at coordinate t is its squared response, $\phi_k(t)^2$, normalized within each basis to reveal where that basis is active. In contrast, the competing CBG parameterizations retain broadly distributed or dispersed responses across the full domain. This selective localization is consistent with the performance advantage of SKAB on EthanolLevel.

5 Conclusion

This work addresses the limited adaptability of predefined functional bases to localized task-relevant patterns through SKAB, which uses spline-wavelet KANs to generate shared task-adaptive bases. Experiments on synthetic and real-world datasets show that SKAB is particularly effective when discriminative information is concentrated in localized functional regions. Future work will extend this method to multidimensional functional domains and broader FDA tasks.

AI USE STATEMENT

In this work, we used generative AI tools for helping develop theoretical models or conceptual frameworks, assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, assisting with translation. We have not used generative AI tools for generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, implementing methods, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results. and proposing or refining hypotheses are not applicable to this work. Additionally, we used generative AI tools for formulating questions for surveys or interviews, creating or modifying scientific figures or images, creating or editing software code, and summarizing or analysing existing literature. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey, and LLM-generated code was verified and tested for correctness by all the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Umberto Amato, Anestis Antoniadis, Italia De Feis, and Irène Gijbels. Unsupervised curve clustering using wavelets: U. amato et al. *Advances in Data Analysis and Classification*, 19(4): 1051–1085, 2025.

Rani Basna, Hiba Nassar, and Krzysztof Podgórski. Data driven orthogonal basis selection for functional data analysis. *Journal of Multivariate Analysis*, 189:104868, 2022.

Rani Basna, Hiba Nassar, and Krzysztof Podgórski. Spline-based methods for functional data on multivariate domains. *Journal of Mathematics in Industry*, 14(1):13, 2024.

Tobia Boschi, Francesca Bonin, Rodrigo Ordonez-Hurtado, Alessandra Pascale, and Jonathan P Epperlein. A new computationally efficient algorithm to solve feature selection for functional data classification in high-dimensional spaces. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4383–4402, 2024.

Zavareh Bozorgasl and Hao Chen. Wav-kan: Wavelet kolmogorov-arnold networks. *arXiv preprint arXiv:2405.12832*, 2024.

Chung Chang, Yakuan Chen, and R Todd Ogden. Functional data classification: a wavelet approach. *Computational Statistics*, 29(6):1497–1513, 2014.

Kehui Chen and Jing Lei. Localized functional principal component analysis. *Journal of the American Statistical Association*, 110(511):1266–1275, 2015.

Jeng-Min Chiou, Hans-Georg Müller, Jane-Ling Wang, and James R Carey. A functional multiplicative effects model for longitudinal data, with application to reproductive histories of female medflies. *Statistica Sinica*, 13(4):1119, 2003.

Charles K Chui. *An introduction to wavelets*, volume 1. Academic press, 1992.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. In *International Conference on Machine Learning*, pp. 12409–12428. PMLR, 2024.

Mohammad Mahdi Eslami. Spline and wavelet smoothing techniques for functional data. In *Proceedings of the 2024 8th International Conference on Graphics and Signal Processing*, pp. 36–41, 2024.

Fuming Fang and Takahiro Shinozaki. Electrooculography-based continuous eye-writing recognition system for efficient assistive communication systems. *PloS one*, 13(2):e0192684, 2018.

Kaniz Fatema, Emad A Mohammed, and Sukhjit Singh Sehra. Taylor-series expanded kolmogorov–arnold network for medical imaging classification. *Machine Learning with Applications*, pp. 100929, 2026.

Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice*. Springer, 2006.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2):179–188, 1936.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

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

Florian Heinrichs, Mavin Heim, and Corinna Weber. Functional neural networks: Shift invariant models for functional data with applications to eeg classification. In *International Conference on Machine Learning*, pp. 12866–12881. PMLR, 2023.

Siyuan Jiang, Yihan Hu, Wenjie Li, and Pengcheng Zeng. DeepFRC: An end-to-end deep learning model for functional registration and classification. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5vdw8Qmrre.

Iain M Johnstone and Bernard W Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the royal statistical society: series B (statistical methodology)*, 59(2):319–351, 1997.

Piotr Kokoszka and Matthew Reimherr. *Introduction to functional data analysis*. Chapman and Hall/CRC, 2017.

A. N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Translations American Mathematical Society*, 2(28):55–59, 1963. URL https://cir.nii.ac.jp/crid/1571698599625560832.

James Large, E Kate Kemsley, Nikolaus Wellner, Ian Goodall, and Anthony Bagnall. Detecting forged alcohol non-invasively through vibrational spectroscopy and machine learning. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 298–309. Springer, 2018.

Ziyao Li. Kolmogorov-arnold networks are radial basis function networks. *arXiv preprint arXiv:2405.06721*, 2024.

Jason Lines, Sarah Taylor, and Anthony Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1041–1046. IEEE, 2016.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, volume 2025, pp. 70367–70413, 2025.

AA Makarov and SV Makarova. Lifting modifications of spline wavelets with unshifted and shifted supports. *Journal of Mathematical Sciences*, 288(4):490–504, 2025.

Sixuan Meng, Zhixia Yang, Junyou Ye, Xue Yang, and Jieyu Yang. Advanced data augmentation methods in neural network for functional data classification. *Journal of Computational and Graphical Statistics*, pp. 1–17, 2026.

Alka Mishra and Surekha Bhusnur. A piecewise spline approach for modeling of ecg signals. *Biomedical Physics & Engineering Express*, 9(6):065017, 2023.

Pedro Morettin and Aluísio Pinheiro. *Wavelets in Functional Data Analysis*. 11 2017. ISBN 978-3319596228. doi: 10.1007/978-3-319-59623-5.

Akash Pandey, Payal Mohapatra, Wei Chen, Qi Zhu, and Sinan Keten. Timesliver: Symbolic-linear decomposition for explainable time series classification. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MDRp9XhGtS.

Qi Qiu, Tao Zhu, Helin Gong, Liming Chen, and Huansheng Ning. Relu-kan: New kolmogorov-arnold networks that only need matrix addition, dot multiplication, and relu. In *2025 IEEE Smart World Congress (SWC)*, pp. 1686–1694. IEEE, 2025.

James Ramsay. *Functional Data Analysis*. John Wiley Sons, Ltd, 2005. ISBN 9780470013199. doi: https://doi.org/10.1002/0470013192.bsa239. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/0470013192.bsa239.

Fabrice Rossi and Brieuc Conan-Guez. Functional multi-layer perceptron: A non-linear tool for functional data analysis. *Neural Networks*, 18(1):45–60, January 2005. ISSN 08936080. doi: 10.1016/j.neunet.2004.07.001.

Yeji Seong, Iris Ivy Gauran, Hyunsung Kim, Hernando Ombao, and Junyong Park. Classification of multivariate functional data with an application to adhd fmri data. *Journal of Applied Statistics*, 53(8):1515–1537, 2026.

Han Lin Shang. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98(2):121–142, 2014.

Jiayu Shang and Xiao-Jun Zeng. Classifying infinite-dimensional data with unified basis functions: An effective machine learning approach. *Neurocomputing*, 622:129245, 2025.

Chengdong Shi, Xiao-Jun Zeng, Ching-Hsun Tseng, and Wei Zhao. Mapping-to-parameter non-linear functional regression with iterative local b-spline knot placement. *Neurocomputing*, 644: 130403, 2025.

Chi Chiu So and Siu Pang Yung. Higher-order-relu-kans (hrkans) for solving physics-informed neural networks (pinns) more accurately, robustly and faster. In *2025 IEEE World AI IoT Congress (AIIoT)*, pp. 1035–1042. IEEE, 2025.

Alex Rodrigo dos Santos Sousa, João Victor Siqueira Rodrigues, Vitor Ribas Perrone, and Raul Gomes Rocha. Wavelet-based estimation in aggregated functional data with positive and correlated errors. *arXiv preprint arXiv:2603.24718*, 2026.

Lin Sun, Jiuxiao Zhang, Weiping Ding, and Jiucheng Xu. Mixed measure-based feature selection using the fisher score and neighborhood rough sets. *Applied Intelligence*, 52(15):17264–17288, 2022.

Hoang-Thang Ta. Bsrbf-kan: A combination of b-splines and radial basis functions in kolmogorov-arnold networks. In *International Symposium on Information and Communication Technology*, pp. 3–15. Springer, 2024.

Marc Vidal and Ana M Aguilera. Wavelet thresholding on independent subspace factorizations of spatially indexed wide functional data for robust estimation of cortical activity. *Mathematics and Computers in Simulation*, 232:346–361, 2025.

Shuoyang Wang, Wanyu Zhang, Guanqun Cao, and Yuan Huang. Functional data analysis using deep neural networks. *Wiley Interdisciplinary Reviews: Computational Statistics*, 16(4):e70001, 2024.

Shuoyang Wang, Guanqun Cao, and Yuan Huang. Empowering classification for multivariate functional data with simultaneous feature selection. *Statistical methods in medical research*, pp. 09622802261449366, 2026.

Sidi Wu, Cédric Beaulac, and Jiguo Cao. Functional autoencoder for smoothing and representation learning. *Statistics and Computing*, 34(6):203, 2024.

Yang Yang, Yanrong Yang, and Han Lin Shang. Feature extraction for functional time series: Theory and application to nir spectroscopy data. *Journal of Multivariate Analysis*, 189:104863, 2022.

Junwen Yao, Jonas Mueller, and Jane-Ling Wang. Deep learning for functional data analysis with adaptive basis layers. In *International conference on machine learning*, pp. 11898–11908. PMLR, 2021.

A Sensitivity Analyses

The following separate sweep contains only three runs per configuration; it cannot be described as a five-run experiment and is not used for formal significance claims. Results are preliminary pending matched repetitions, conditioning diagnostics, and validation-based hyperparameter selection.

Number of bases. On Phoneme, SKAB has the highest observed three-run mean at $K = 6$, but differences among $K = 4, 6, 8, 12$ are small relative to the run-to-run variation (Table 3). For fixed classical bases, the B-spline and Fourier means rise with K in the completed range; the three-run B-spline $K = 12$ mean is 0.9272. The attempted B-spline $K = 2$ is invalid for the chosen order-three splines, and Fourier $K = 8$ has only one completed run; neither is included as a complete result.

Table 3: Preliminary Phoneme test ACC $\pm$ SD from three runs per displayed setting.

K	SKAB	B-spline	Fourier
2	0.9098 ± 0.0082	0.8234 ± 0.0055	0.7753 ± 0.0013
4	0.9283 ± 0.0126	0.8677 ± 0.0023	0.8828 ± 0.0050
6	0.9316 ± 0.0061	0.8936 ± 0.0040	0.9087 ± 0.0061
8	0.9261 ± 0.0097	0.9091 ± 0.0011	0.9104 ± 0.0043
12	0.9272 ± 0.0078	0.9272 ± 0.0061	0.9210 ± 0.0032

Ridge regularization and smoothness. Table 4 records separate three-run Phoneme sweeps at $K = 6$. Accuracy varies modestly across the evaluated ridge and smoothness values. These descriptive results neither establish an optimal regularizer nor address conditioning of the learned basis matrix.

Table 4: Preliminary Phoneme regularization sweeps: three-run test ACC $\pm$ SD, varying one parameter at a time with $K = 6$.

λ_{ridge}	Test ACC	γ_{smooth}	Test ACC
10^{-10}	0.9291 ± 0.0020	0	0.9305 ± 0.0045
10^{-8}	0.9280 ± 0.0059	10^{-6}	0.9298 ± 0.0061
10^{-6}	0.9265 ± 0.0083	10^{-5}	0.9280 ± 0.0059
10^{-4}	0.9276 ± 0.0086	10^{-4}	0.9287 ± 0.0061
10^{-2}	0.9265 ± 0.0074	10^{-3}	0.9265 ± 0.0050

CSDS parameters. We study the sensitivity of SKAB to the size of its center-scale dictionary by varying the number of centers $Q \in \{2, 4, 6, 8\}$ and the number of scales $M \in \{2, 3, 4, 5\}$. Each configuration is trained for ten runs with different random seeds. Table 5 reports the test ACC as mean $\pm$ SD.

The two datasets exhibit different sensitivity patterns. On Phoneme, the performance is relatively stable across the tested configurations, with ACC ranging from 0.9163 to 0.9310. Increasing the number of centers generally improves the results, while changing the number of scales has a smaller effect. The default configuration $(Q, M) = (6, 3)$ achieves 0.9309, which is essentially the best performance observed on this dataset. EthanolLevel is more sensitive to the dictionary size. The best result is obtained with $(Q, M) = (4, 3)$, reaching 0.8160 ± 0.0148, whereas configurations with only two centers perform substantially worse. Increasing the number of centers to eight does

Table 5: Sensitivity of SKAB to the number of centers Q and scales M. Values are test ACC reported as mean $\pm$ SD over ten runs.

Phoneme				
$Q\backslash M$	2	3	4	5
2	0.9195 ± 0.0076	0.9190 ± 0.0078	0.9259 ± 0.0105	0.9266 ± 0.0066
4	0.9253 ± 0.0070	0.9163 ± 0.0079	0.9266 ± 0.0066	0.9249 ± 0.0053
6	0.9274 ± 0.0064	$\mathbf{0.9309 \pm 0.0072}$	0.9286 ± 0.0078	0.9293 ± 0.0070
8	$\mathbf{0.9310 \pm 0.0070}$	0.9295 ± 0.0062	0.9282 ± 0.0074	$\mathbf{0.9310 \pm 0.0059}$

EthanolLevel				
$Q\backslash M$	2	3	4	5
2	0.6700 ± 0.0844	0.5604 ± 0.0376	0.6036 ± 0.0429	0.6360 ± 0.0504
4	0.7832 ± 0.0182	$\mathbf{0.8160 \pm 0.0148}$	0.7978 ± 0.0229	0.8014 ± 0.0271
6	0.7492 ± 0.0335	0.7870 ± 0.0163	0.7658 ± 0.0241	0.7760 ± 0.0211
8	0.6966 ± 0.0535	0.7554 ± 0.0193	0.6490 ± 0.1935	0.7608 ± 0.0349

not provide further improvement and can lead to unstable results, as shown by the large variation for $(Q, M) = (8, 4)$. Overall, the results support using a moderate number of centers and a small multiscale dictionary rather than increasing both quantities indiscriminately.

This dataset-dependent behavior is expected because functional datasets can exhibit substantially different distributional shapes, signal complexity, and degrees of localized variation, so the most suitable numbers of centers and scales need not be identical across datasets. Developing a principled criterion for selecting the center-scale dictionary size from the observed data remains an important direction for future work.

B Additional Localized-Basis Comparisons

We evaluate a covariance-block localized FPCA approximation and a supervised db4 wavelet-coefficient selection baseline with the same downstream MLP head. Both feature transformations use training data only; the latter ranks db4 coefficients by a training-set Fisher score. Localized FPCA here approximates spatial blocks from the covariance structure; it does not reproduce the BD-SVD block detector of prior work. Each completed configuration below uses five repeated runs. Entries give mean test accuracy $\pm$ sample standard deviation, without selecting K on the test set.

EthanolLevel. Table 6 shows that supervised db4 selection is substantially stronger in observed accuracy than the unsupervised localized FPCA approximation at every tested K. Its highest observed five-run mean is 0.8176 ± 0.0123 at $K = 8$, close to the ten-run SKAB result in Table 2; the two results use different experimental protocols and should therefore be interpreted descriptively.

Table 6: EthanolLevel: five runs per setting; mean test ACC $\pm$ SD.

K	Localized FPCA	Supervised db4
2	0.2868 ± 0.0184	0.6096 ± 0.0207
4	0.2868 ± 0.0147	0.7984 ± 0.0089
6	0.3496 ± 0.0209	0.7948 ± 0.0050
8	0.4092 ± 0.0124	0.8176 ± 0.0123
10	0.4520 ± 0.0032	0.7960 ± 0.0251
20	0.5708 ± 0.0236	0.7684 ± 0.0055

Phoneme. Both new representations improve as the number of selected coefficients increases from two, but their observed means are similar at larger K (Table 7). Supervised db4 reaches its highest observed mean of 0.9297 ± 0.0081 at $K = 10$, while localized FPCA reaches 0.9257 ± 0.0078 at $K = 20$.

Table 7: Phoneme: five runs per setting; mean test ACC ± SD.

K	Localized FPCA	Supervised db4
2	0.7488 ± 0.0052	0.8166 ± 0.0082
4	0.8951 ± 0.0069	0.9142 ± 0.0062
6	0.9175 ± 0.0095	0.9151 ± 0.0057
8	0.9242 ± 0.0055	0.9131 ± 0.0076
10	0.9253 ± 0.0064	0.9297 ± 0.0081
20	0.9257 ± 0.0078	0.9281 ± 0.0027

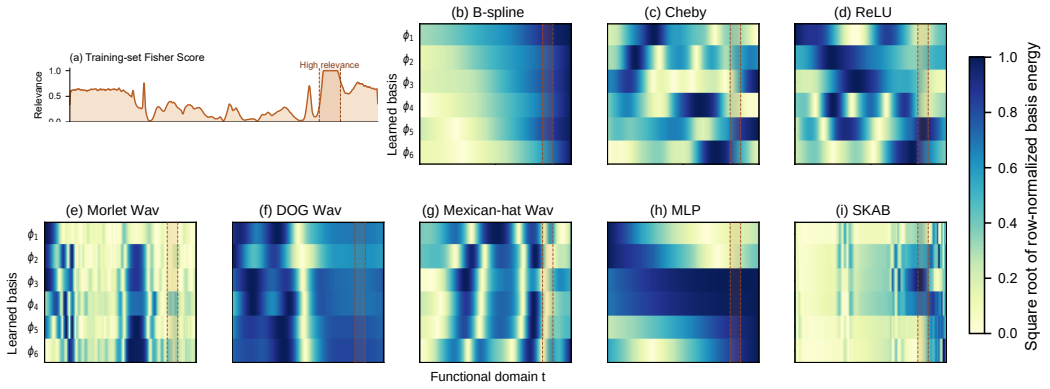


Figure 6: Basis energy maps of different CBG parameterizations on EthanolLevel. **(a)** Training-set Fisher score, where the shaded interval indicates high-relevance locations. **(b–i)** Square root of row-normalized energy for the six learned basis functions produced by different parameterization in CBG.

ECG5000. Table 8 gives the four K settings with five completed paired runs. Localized FPCA has a numerically higher mean than supervised db4 at $K = 4, 6, 8$. The $K = 10$ localized FPCA log contains only three of five repeats before a test-file parsing error; it and the unrun $K = 10$ db4 configuration are excluded rather than represented as five-run measurements.

Table 8: ECG5000: five runs per displayed setting; mean test ACC ± SD.

K	Localized FPCA	Supervised db4
2	0.9218 ± 0.0085	0.9196 ± 0.0011
4	0.9321 ± 0.0023	0.9284 ± 0.0034
6	0.9372 ± 0.0008	0.9325 ± 0.0025
8	0.9375 ± 0.0017	0.9309 ± 0.0052

C Analysis of basis energy map

The basis energy maps of different CBG parameterization are shown in Fig. 6.

D Exact Construction of the Order-2 Spline-Wavelet Atom

This section provides the exact cardinal B-spline and Chui-Wang spline-wavelet definitions used by SWK.

D.1 CARDINAL B-SPLINES

Let $(x)_+ = \max(x, 0)$. The cardinal B-spline of order m is defined by

$$N_m(x) = \frac{1}{(m-1)!} \sum_{\ell=0}^{m} (-1)^\ell \binom{m}{\ell} (x-\ell)_+^{m-1}. \tag{11}$$

Equivalently, it can be defined recursively as

$$N_1(x) = \mathbb{I}_{[0,1)}(x), \qquad N_m(x) = \int_0^1 N_{m-1}(x-u)\, du. \tag{12}$$

The support of N_m is $[0, m]$, and its integer translates form a partition of unity.

For $m = 2$, the cardinal B-spline is the linear hat function

$$N_2(x) = \begin{cases} x, & 0 \le x < 1, \\ 2 - x, & 1 \le x < 2, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

D.2 ORDER-2 CHUI-WANG SPLINE-WAVELET

The compactly supported Chui-Wang spline wavelet of order m can be constructed from cardinal B-splines chui1992compactly. For $m = 2$, the wavelet reduces to

$$\begin{aligned} \psi_{\mathrm{CW},2}(x) = \frac{1}{12} \big[& N_2(2x) - 6N_2(2x-1) + 10N_2(2x-2) \\ & - 6N_2(2x-3) + N_2(2x-4) \big]. \end{aligned} \tag{14}$$

Its explicit piecewise-linear form is

$$\psi_{\mathrm{CW},2}(x) = \begin{cases} \dfrac{x}{6}, & 0 \le x < \dfrac{1}{2}, \\ -\dfrac{7x}{6} + \dfrac{2}{3}, & \dfrac{1}{2} \le x < 1, \\ \dfrac{8x}{3} - \dfrac{19}{6}, & 1 \le x < \dfrac{3}{2}, \\ -\dfrac{8x}{3} + \dfrac{29}{6}, & \dfrac{3}{2} \le x < 2, \\ \dfrac{7x}{6} - \dfrac{17}{6}, & 2 \le x < \dfrac{5}{2}, \\ -\dfrac{x}{6} + \dfrac{1}{2}, & \dfrac{5}{2} \le x < 3, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Hence,

$$\mathrm{supp}(\psi_{\mathrm{CW},2}) = [0, 3]. \tag{16}$$

The wavelet is symmetric around $x = 3/2$, sign-changing, and has two vanishing moments:

$$\int_{\mathbb{R}} \psi_{\mathrm{CW},2}(x)\, dx = 0, \qquad \int_{\mathbb{R}} x\psi_{\mathrm{CW},2}(x)\, dx = 0. \tag{17}$$

Thus, the wavelet annihilates constant and affine components on its support.

D.3 RESCALING TO UNIT SUPPORT

To make the center and scale parameters directly interpretable on the normalized functional domain, we translate and rescale the wavelet to the support $[-1/2, 1/2]$:

$$\psi(u) = \nu_\psi \psi_{\mathrm{CW},2} \left(3u + \frac{3}{2} \right), \tag{18}$$

where $\nu_\psi = 1$ in the implementation. The support is

$$\operatorname{supp}(\psi) = \left[-\frac{1}{2}, \frac{1}{2}\right]. \tag{19}$$

The rescaled wavelet can also be written as

$$\psi(u) = \nu_\psi \begin{cases} \frac{u}{2} + \frac{1}{4}, & -\frac{1}{2} \le u < -\frac{1}{3}, \\ -\frac{7u}{2} - \frac{13}{12}, & -\frac{1}{3} \le u < -\frac{1}{6}, \\ 8u + \frac{5}{6}, & -\frac{1}{6} \le u < 0, \\ -8u + \frac{5}{6}, & 0 \le u < \frac{1}{6}, \\ \frac{7u}{2} - \frac{13}{12}, & \frac{1}{6} \le u < \frac{1}{3}, \\ -\frac{u}{2} + \frac{1}{4}, & \frac{1}{3} \le u \le \frac{1}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

D.4 Complete SWK Edge Function

For completeness, the full edge function used in an SWK layer is written as

$$\varphi_{o,i}(x) = \beta_{o,i}\,\mathrm{SiLU}(x) + \sum_{m=1}^{M}\sum_{q=1}^{Q} w_{o,i,m,q}\,\psi\left(\frac{x - c_q}{s_m}\right), \tag{21}$$

where $\beta_{o,i}$ and $w_{o,i,m,q}$ are trainable parameters. The SiLU branch is the residual base function, while the spline-wavelet branch provides localized responses.

For each atom,

$$\operatorname{supp}\left[\psi\left(\frac{x - c_q}{s_m}\right)\right] = \left[c_q - \frac{s_m}{2}, c_q + \frac{s_m}{2}\right]. \tag{22}$$

The Center-Scale dictionary is shared by all edges in one layer, while the combination weights are edge-specific.

The order-2 wavelet is piecewise linear and therefore classically differentiable away from its knots. Smoothness of the final learned bases is controlled through the discrete roughness regularizer rather than by assuming that each atom is twice continuously differentiable.

D.5 Support and Vanishing Moments

We summarize the two properties of the rescaled spline wavelet that motivate its use in SWK: compact support and vanishing moments. Recall that

$$\psi(u) = \nu_\psi \psi_{\mathrm{CW},2}\left(3u + \frac{3}{2}\right), \tag{23}$$

where $\operatorname{supp}(\psi_{\mathrm{CW},2}) = [0, 3]$. The affine transformation $x = 3u + \frac{3}{2}$ therefore gives

$$\operatorname{supp}(\psi) = \left[-\frac{1}{2}, \frac{1}{2}\right]. \tag{24}$$

The original order-2 Chui-Wang spline wavelet satisfies

$$\int_{\mathbb{R}} \psi_{\mathrm{CW},2}(x)\,dx = 0, \qquad \int_{\mathbb{R}} x\psi_{\mathrm{CW},2}(x)\,dx = 0. \tag{25}$$

Using $x = 3u + \frac{3}{2}$ and $du = dx/3$, we obtain

$$\int_{\mathbb{R}} \psi(u)\,du = \frac{\nu_\psi}{3}\int_{\mathbb{R}} \psi_{\mathrm{CW},2}(x)\,dx = 0, \tag{26}$$

and

$$\begin{aligned}\int_{\mathbb{R}} u\psi(u)\,du &= \frac{\nu_\psi}{9}\int_{\mathbb{R}}\left(x-\frac{3}{2}\right)\psi_{\mathrm{CW},2}(x)\,dx \\ &= 0.\end{aligned} \tag{27}$$

Hence, the translated and rescaled wavelet retains two vanishing moments:

$$\int_{\mathbb{R}} \psi(u)\,du = \int_{\mathbb{R}} u\psi(u)\,du = 0. \tag{28}$$

For a translated and scaled atom

$$\psi_{c,s}(t) = \psi\left(\frac{t-c}{s}\right), \qquad s>0, \tag{29}$$

the compact support of ψ implies

$$\mathrm{supp}(\psi_{c,s}) = \left[c-\frac{s}{2}, c+\frac{s}{2}\right]. \tag{30}$$

Thus, c determines the response location and s determines the support width.

Moreover, if the complete atom support is contained in the functional domain, then for any affine function $p(t)=a+bt$,

$$\begin{aligned}\int_{\mathcal{T}} p(t)\psi_{c,s}(t)\,dt &= s(a+bc)\int_{\mathbb{R}}\psi(u)\,du + bs^2\int_{\mathbb{R}} u\psi(u)\,du \\ &= 0.\end{aligned} \tag{31}$$

Therefore, each interior spline-wavelet atom is insensitive to constant and locally linear components over its support, while remaining responsive to localized deviations.

E Detailed Center-Scale Dictionary Selection

CSDS performs a one-shot, data-driven initialization of the center and scale dictionary in the first SWK layer. This layer receives the original functional coordinate as input, so its atom locations and support widths can be initialized from coordinate-wise training statistics. All quantities below are computed exclusively from the training split.

E.1 Grid Normalization

Let N training observations be measured on a common grid $\mathcal{T}_J = \{t_1,\ldots,t_J\}$, where J is the number of observed coordinates. Before initialization, the grid is normalized to $[0,1]$:

$$t_j \leftarrow \frac{t_j - \min_\ell t_\ell}{\max_\ell t_\ell - \min_\ell t_\ell + \varepsilon_t}, \qquad j=1,\ldots,J, \tag{32}$$

where $\varepsilon_t > 0$ prevents division by zero. If the supplied grid does not contain J valid coordinates, it is replaced by the uniform grid

$$t_j = \frac{j-1}{J-1}. \tag{33}$$

E.2 Coordinate Relevance Score

The relevance score at coordinate t_j measures how strongly the observed values $X_i(t_j)$ distinguish different classes. Suppose the training set contains R classes. Let N_r be the number of observations in class r, and define its empirical proportion as

$$\pi_r = \frac{N_r}{N}. \tag{34}$$

The class-wise and global means at t_j are

$$\mu_{rj} = \frac{1}{N_r}\sum_{i:y_i=r} X_i(t_j), \qquad \mu_j = \frac{1}{N}\sum_{i=1}^{N} X_i(t_j). \tag{35}$$

The within-class variance is computed as

$$\sigma_{rj}^2 = \frac{1}{N_r} \sum_{i:y_i=r} \left[X_i(t_j) - \mu_{rj}\right]^2. \tag{36}$$

The between-class and within-class variations at t_j are

$$B_j = \sum_{r=1}^{R} \pi_r \left(\mu_{rj} - \mu_j\right)^2, \qquad W_j = \sum_{r=1}^{R} \pi_r \sigma_{rj}^2, \tag{37}$$

respectively. The Fisher relevance score is

$$S_j = \frac{B_j}{W_j + \varepsilon_F}, \tag{38}$$

where $\varepsilon_F = 10^{-6}$. A large S_j indicates strong between-class differences relative to within-class variation.

E.3 Robust Normalization and Smoothing

Sharp isolated Fisher-score peaks may dominate the subsequent center allocation. We therefore normalize the scores using their empirical 95-th percentile. Let

$$q_{0.95} = \mathrm{quantile}_{0.95}\left(\{S_j\}_{j=1}^{J}\right). \tag{39}$$

The normalized score is

$$\bar{S}_j = \min\left\{\frac{S_j}{\max(q_{0.95}, \varepsilon_P)}, 1\right\}, \tag{40}$$

where $\varepsilon_P = 10^{-12}$. Before normalization, nonfinite values are replaced by zero and negative values are truncated to zero. If all scores vanish, they are replaced by a constant positive vector.

The normalized scores are then smoothed using a centered moving average. For an odd window size $h = 2a + 1$,

$$\widetilde{S}_j = \frac{1}{h} \sum_{u=-a}^{a} \bar{S}_{j+u}, \tag{41}$$

where values outside the grid are treated as zero. The current implementation uses

$$h = 9. \tag{42}$$

Here, $\widetilde{S}_j$ denotes the final relevance score used for center selection.

E.4 Relevance Distribution

The smoothed scores are converted into a probability distribution:

$$\widehat{p}_j = \frac{\widetilde{S}_j}{\sum_{\ell=1}^{J} \widetilde{S}_\ell + \varepsilon_P}. \tag{43}$$

This distribution is mixed with a uniform component:

$$p_j = (1 - \alpha)\widehat{p}_j + \frac{\alpha}{J}, \tag{44}$$

where p_j is the relevance mass assigned to coordinate t_j, and $\alpha \in [0, 1]$ controls the amount of uniform domain coverage. The reported experiments use

$$\alpha = 0.25. \tag{45}$$

Thus, discriminative regions receive greater probability mass, while the uniform component prevents the initialization from ignoring the remaining domain.

E.5 Weighted-Quantile Center Proposal

Let Q denote the number of centers in the first SWK layer. Define the cumulative relevance mass as

$$F_j = \sum_{\ell=1}^{j} p_\ell, \qquad j = 1, \dots, J. \tag{46}$$

For center index $q = 1, \dots, Q$, we use the probability level

$$\xi_q = \frac{q - \frac{1}{2}}{Q}. \tag{47}$$

The corresponding weighted-quantile center candidate is

$$\widetilde{c}_q = t_{j_q}, \qquad j_q = \min\{j : F_j \geq \xi_q\}. \tag{48}$$

These probability levels place one candidate within each equal-mass portion of the relevance distribution. Consequently, regions with greater discriminative mass receive denser center allocation.

E.6 Compact-Support-Aware Center Selection

The spline-wavelet atom with center c and scale s is

$$\psi\left(\frac{t-c}{s}\right). \tag{49}$$

Because the rescaled mother wavelet is supported on $[-1/2, 1/2]$, this atom has support

$$\operatorname{supp}\left[\psi\left(\frac{t-c}{s}\right)\right] = \left[c - \frac{s}{2}, c + \frac{s}{2}\right]. \tag{50}$$

Duplicated or excessively close centers would therefore initialize multiple atoms over nearly identical local intervals.

To reduce this redundancy, we define a minimum center separation

$$\delta_c = \frac{\rho}{Q-1}, \tag{51}$$

where ρ is the separation ratio. The current implementation uses

$$\rho = 0.45. \tag{52}$$

After sorting the weighted-quantile candidates, they are accepted directly when

$$\widetilde{c}_{q+1} - \widetilde{c}_q \geq \delta_c, \qquad q = 1, \dots, Q-1. \tag{53}$$

If this condition is violated, support-aware non-maximum suppression is applied. The grid coordinates are first ranked in descending order of their relevance masses p_j. A coordinate t_j is retained only when

$$|t_j - c| \geq \delta_c \tag{54}$$

for every previously selected center c. Selection continues until Q centers are obtained or all grid coordinates have been examined.

If fewer than Q centers are selected, unused weighted-quantile candidates are reconsidered under the same separation condition. If the dictionary remains incomplete, distinct uniformly spaced grid coordinates are added deterministically. The final fallback guarantees a complete dictionary, although it does not reapply the separation condition. The resulting centers are sorted as

$$c_1 < c_2 < \cdots < c_Q. \tag{55}$$

E.7 Spacing-Based Scale Initialization

The scale bank is initialized from the spatial distribution of the selected centers. For $Q > 1$, let

$$\Delta_c = \max\left\{\text{median}_{q=1,\dots,Q-1}(c_{q+1} - c_q), \frac{1}{J-1}\right\}, \tag{56}$$

where Δ_c is the median adjacent-center spacing. The lower bound $1/(J-1)$ prevents the reference spacing from becoming smaller than one normalized grid interval. When $Q = 1$, the implementation sets

$$\Delta_c = 0.25. \tag{57}$$

Let M be the number of scales and let r_m be the relative multiplier for the m-th scale. The initialized scales are

$$s_m = \min\left\{s_{\max}, \max\left\{s_{\min}, r_m \Delta_c\right\}\right\}, \qquad m = 1, \dots, M, \tag{58}$$

where $s_{\min}$ and $s_{\max}$ are the global lower and upper scale bounds.

The reported experiments use

$$M = 3, \qquad (r_1, r_2, r_3) = (0.75, 1.25, 2.00), \tag{59}$$

together with

$$s_{\min} = 0.01, \qquad s_{\max} = 0.60. \tag{60}$$

These multipliers provide medium-to-broad support widths relative to the typical distance between adjacent centers and avoid initializing the first layer with nearly pointwise atoms.

E.8 Dictionary Assignment and Optimization

The selected centers $\{c_q\}_{q=1}^{Q}$ and scales $\{s_m\}_{m=1}^{M}$ are shared across all spline-wavelet edge functions in the first SWK layer. Positive scales are stored through log-scale parameters:

$$\gamma_m = \log s_m, \qquad s_m = \exp(\gamma_m). \tag{61}$$

Both the centers and log-scales remain trainable after initialization. Thus, CSDS provides a task-informed starting dictionary, while gradient-based optimization can further adjust atom locations and support widths.

The remaining SWK layers are not initialized by CSDS. Their centers are uniformly spaced over predefined activation ranges, and their scales are initialized from a predefined multiscale bank. These centers and scales also remain trainable during optimization.

F Regularization

Smoothness penalty. The compact support of spline-wavelet atoms encourages localized responses, but unconstrained neural parameterization can still produce numerically oscillatory basis functions. We therefore use a small smoothness penalty on the learned basis matrix $\Phi_\Theta = [\phi_k(t_j)]_{k=1,j=1}^{K,J}$.

Let $h_j = t_{j+1} - t_j$ denote the interval lengths. For an interior grid point t_j, the discrete second derivative is approximated by

$$\begin{aligned} D_j^{(2)}\phi_k = \frac{2}{h_{j-1} + h_j}\Big[&\frac{\phi_k(t_{j+1}) - \phi_k(t_j)}{h_j} \\ &- \frac{\phi_k(t_j) - \phi_k(t_{j-1})}{h_{j-1}}\Big], \quad j = 2, \dots, J-1. \end{aligned} \tag{62}$$

For a uniform grid with spacing Δt, this reduces to the standard central second difference,

$$D_j^{(2)}\phi_k = \frac{\phi_k(t_{j+1}) - 2\phi_k(t_j) + \phi_k(t_{j-1})}{(\Delta t)^2}. \tag{63}$$

Using trapezoidal quadrature weights w_j, the penalty is

$$\mathcal{L}_{\text{smooth}} = \frac{1}{K(J-2)} \sum_{k=1}^{K} \sum_{j=2}^{J-1} w_j \left(D_j^{(2)} \phi_k \right)^2 . \tag{64}$$

The training objective adds this auxiliary term to the classification loss:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \gamma \mathcal{L}_{\text{smooth}}, \tag{65}$$

where $\gamma = 10^{-5}$ by default. This term is used only to suppress unnecessary high-frequency numerical oscillations and does not impose a fixed global basis shape.

G Experimental Details

This section provides the implementation details referenced in the main paper. Table 9 summarizes the dataset-specific representation dimensions, mini-batch sizes, sample partitions, and common-grid resolutions; Table 10 summarizes the shared training configuration; and Table 11 lists the SKAB architecture and dictionary-initialization settings. Unless otherwise specified, preprocessing and data-driven initialization statistics are estimated exclusively from the training split.

G.1 Synthetic Data Generation

Donoho-Johnstone Kernel and Localized Atoms. The synthetic dataset DJ-Mixture is generated on the normalized domain $t \in [0, 1]$ with $J = 1500$ equally spaced grid points. Our implementation is based on the rational kernel used by the standard Donoho-Johnstone Bumps test function (Johnstone & Silverman, 1997). In particular, let

$$q(u) = (1 + |u|)^{-4}. \tag{66}$$

The standard Bumps signal can be written as

$$f_{\text{Bumps}}(t) = \sum_{r=1}^{11} H_r q\left(\frac{t - P_r}{W_r} \right), \tag{67}$$

with the following locations, heights, and widths:

r	1	2	3	4	5	6
P_r	0.10	0.13	0.15	0.23	0.25	0.40
H_r	4.0	5.0	3.0	4.0	5.0	4.2
W_r	0.005	0.005	0.006	0.010	0.010	0.030

r	7	8	9	10	11
P_r	0.44	0.65	0.76	0.78	0.81
H_r	2.1	4.3	3.1	5.1	4.2
W_r	0.010	0.010	0.005	0.008	0.005

(68)

We do not use this complete 11-bump curve as a class template. Instead, we use its kernel q to construct six separated, cosine-tapered atoms on the reference intervals

$$\begin{aligned} I_1 &= [0.05, 0.17], \quad I_2 = [0.20, 0.32], \quad I_3 = [0.35, 0.47], \\ I_4 &= [0.50, 0.62], \quad I_5 = [0.65, 0.77], \quad I_6 = [0.80, 0.92]. \end{aligned} \tag{69}$$

Let c_k be the midpoint of I_k, and let

$$(h_1, \ldots, h_6) = (0.024, 0.026, 0.025, 0.027, 0.025, 0.024).$$

For $I_k = [a_k, b_k]$, we use the cosine-taper window with edge width $e = 0.018$:

$$w_k(t) = \begin{cases} \frac{1}{2} - \frac{1}{2} \cos\left(\pi \frac{t - a_k}{e} \right), & a_k \le t < a_k + e, \\ 1, & a_k + e \le t \le b_k - e, \\ \frac{1}{2} + \frac{1}{2} \cos\left(\pi \frac{t - (b_k - e)}{e} \right), & b_k - e < t \le b_k, \\ 0, & \text{otherwise.} \end{cases} \tag{70}$$

The raw generating atom is

$$\widetilde{g}_k(t) = w_k(t)\left[q\left(\frac{t-c_k}{h_k}\right) - 0.42q\left(\frac{t-c_k}{2.8h_k}\right)\right]. \tag{71}$$

The final generating atom g_k is obtained by subtracting the mean of $\widetilde{g}_k$ over $[0,1]$ and normalizing it to unit L^2 norm. Thus, the six atoms are fully specified by the intervals and widths above rather than sampled from the original 11 Bumps components.

Class Prototypes. The dataset contains five classes, indexed by $y_i \in \{1,\dots,5\}$. Each class is represented by a prototype coefficient vector over the six localized atoms:

$$A = \begin{pmatrix} 1.15 & 0.45 & 0 & 0 & 0 & 0 \\ 0 & 1.15 & 0.45 & 0 & 0 & 0 \\ 0 & 0 & 1.15 & 0.45 & 0 & 0 \\ 0 & 0 & 0 & 1.15 & 0.45 & 0 \\ 0.45 & 0 & 0 & 0 & 1.15 & 0.45 \end{pmatrix}. \tag{72}$$

For sample i with label y_i, the localized component is

$$L_i(t) = \rho \sum_{k=1}^{6} (A_{y_i,k} + \xi_{i,k})\, g_k(t - \delta_{i,k}), \tag{73}$$

where $\rho = 0.85$, $\xi_{i,k} \sim \mathcal{N}(0, 0.10^2)$, and $\delta_{i,k} \sim \mathcal{N}(0, 0.008^2)$ clipped to $[-0.024, 0.024]$. The perturbation $\xi_{i,k}$ changes the strength of each local atom, while $\delta_{i,k}$ introduces small sample-level position shifts. The shifted values $g_k(t-\delta_{i,k})$ are evaluated by linear interpolation, with zero values outside $[0,1]$.

Background, Noise, and Split. Each curve includes a class-independent smooth background

$$B_i(t) = \beta_{i,1}\sin(2\pi t) + \beta_{i,2}\cos(2\pi t) + \beta_{i,3}\sin(4\pi t + 0.35), \tag{74}$$

where $\beta_{i,r} \sim \mathcal{N}(0, 0.38^2)$. Additive observation noise is sampled independently as

$$\epsilon_i(t_j) \sim \mathcal{N}(0, 0.22^2). \tag{75}$$

The final observed curve is

$$X_i(t) = L_i(t) + B_i(t) + \epsilon_i(t). \tag{76}$$

No additional nuisance atoms are used in the final experiment.

We generate 300 samples per class, resulting in 1500 curves in total. Dataset generation uses random seed 42. A stratified split with random seed 1 follows a $7:1:2$ train-validation-test ratio, giving 1050, 150, and 300 samples. To form a few-shot setting, only 10 training samples per class are retained, resulting in an effective training set of 50 curves; the validation and test sets are kept unchanged.

G.2 Dataset-Specific Configurations

The dataset-specific representation dimensions, mini-batch sizes, sample partitions, and common-grid resolutions are provided in Table 9.

G.3 Common Training Configuration

Conventional functional datasets including Phoneme, Tecator, and Medflies are stratified into training, validation, and test sets at a $7:1:2$ ratio. For UCR datasets, we retain the official test set and reserve 25% of the official training set for validation. Input observations are standardized coordinate-wise using statistics estimated exclusively from the training split for Phoneme, Tecator, EthanolLevel, and EOGHorizontalSignal; Medflies, ECG5000, and DJ-MIXTURE retain their provided scales. Functional grids are normalized to $[0,1]$. The shared optimization, prediction, and computing configurations are provided in Table 10.

Dataset	K	Batch size	Train	Validation	Test	Grid size J
DJ-MIXTURE	6	64	50	150	300	1500
Phoneme	6	128	3156	451	902	256
Tecator	6	8	150	22	43	100
Medflies	6	32	374	53	107	30
ECG5000	6	32	375	125	4500	140
EthanolLevel	6	128	378	126	500	1751
EOGHorizontalSignal	6	64	271	91	362	1250

Table 9: Overview of the datasets used in the main classification experiments, including the representation dimension, mini-batch size, sample partition, and number of common-grid observations.

Setting	Value	Setting	Value
Optimization			
Optimizer	Adam	Initial learning rate	10^{-4}
Weight decay	10^{-5}	Maximum epochs	1000
Early-stopping criterion	Validation loss	Early-stopping patience	100
Prediction and representation			
Prediction head	$[64, 64]$ MLP	Classification loss	Cross-entropy
Ridge coefficient λ	10^{-6}	Model-selection criterion	Validation loss
Computing environment			
GPU	NVIDIA GeForce RTX 2080 Ti	GPU memory	11 GB
CPU	2× Intel Xeon Gold 6126	System memory	128 GB
Operating system	Ubuntu 22.04.4 LTS	Python	3.10.0
PyTorch	2.12.0	CUDA	13.0

Table 10: Common optimization, prediction, reproducibility, and computing configurations. For comparison methods that use different settings, we follow the configurations provided in their original implementations.

G.4 SKAB CONFIGURATION

Implementation details. We use the original model structures and released implementations of the baselines whenever available. Representation-based methods use the same coefficient dimension within each dataset. SKAB is trained with Adam for at most $1{,}000$ epochs, with an initial learning rate of 10^{-4}, weight decay 10^{-5}, and early-stopping patience of 100 epochs. A ReduceLROnPlateau scheduler monitors validation loss, halves the learning rate after 30 non-improving epochs, and uses a minimum learning rate of 10^{-6}. We use 3 scales and 6 centers, except for EthanolLevel, which uses 4 centers. The basis dimension is $K = 6$ for all reported datasets. On DJ-MIXTURE, SKAB uses a single $[1, K]$ SWK layer without hidden CBG layers. The complete SKAB architecture and CSDS initialization settings are provided in Table 11.

G.5 BASELINE IMPLEMENTATION DETAILS

All baselines use the original method designs and released implementations when available. MLP and KAN process the discretized observations directly. B-spline, Fourier, and FPCA baselines first produce coefficient features and then use the common $[64, 64]$ MLP prediction head. AdaFNN, FNN, FSFC, DeepFRC, FuncAE, TSLANet, SoftShape, and TimeSliver follow the training protocols supplied by their official implementations. For the ablation variants, only the CBG parameterization or the center-scale initialization is changed; all remaining training settings are shared with SKAB.

All reported results are the mean and standard deviation over ten runs. The same run seeds are used across comparable methods, and model selection is performed using validation loss.

Setting	Value	Setting	Value
CBG and SWK architecture			
CBG architecture	$[1, 64, 64, K]$	Number of bases	Dataset-specific, default 6
Number of SWK centers Q	Dataset-specific, default 6	Number of SWK scales M	3
Scale multipliers	$(0.75, 1.25, 2.00)$	Residual base function	SiLU
SWK mother wavelet	Order-2 Chui-Wang	Mother-wavelet support	$[-1/2, 1/2]$
CSDS initialization			
Relevance score	Fisher score	Uniform-mixture weight α	0.25
Score normalization	95-th percentile clipping	Smoothing-window size	9
Center proposal	Weighted midpoint quantiles	Separation ratio ρ	0.45
Minimum scale $s_{\min}$	0.01	Maximum scale $s_{\max}$	0.60
CSDS application	First SWK layer only	Scale parameterization	Log-scale
First-layer centers	Trainable after initialization	First-layer scales	Trainable after initialization
Hidden-layer centers	Uniformly initialized	Hidden-layer scales	Predefined multiscale bank
Hidden-layer centers/scales	Trainable	Dictionary sharing	Shared within each layer

Table 11: Architecture, dictionary initialization, and objective configurations of the proposed SKAB model.

H Additional Experiment Analysis

CBG parameterization. The CBG ablation presented in the main paper reveals different advantages over four competing parameterization families. First, SKAB outperforms B-spline KAN on all six datasets. The gain is especially large on EthanolLevel, where ACC increases from 0.3066 to 0.8160, while the gain on EOGHorizontalSignal is from 0.4370 to 0.4624. This pattern is consistent with SWK allocating a limited dictionary through adaptable centers and multiple scales, rather than relying on a uniformly refined B-spline knot grid. Second, SKAB exceeds both ChebyKAN and ReLU-KAN on five of the six datasets, including gains of 13.02 and 11.66 percentage points, respectively, on EthanolLevel. ChebyKAN uses global polynomial responses and ReLU-KAN uses a single ReLU-window construction, whereas SWK explicitly represents localized variation at multiple adaptable scales. On EOGHorizontalSignal, however, SKAB is comparable to ChebyKAN (0.4624 versus 0.4638) and below ReLU-KAN (0.4713), indicating that this advantage is not universal. Third, SKAB outperforms Morlet, DOG, and Mexican-hat Wav-KAN on every dataset; on EthanolLevel, the respective gains are 6.10, 47.10, and 32.04 percentage points. Unlike these non-compactly supported wavelets, spline-wavelet atoms restrict each response to an explicit interval, reducing interference from observations outside the selected region. Finally, the MLP CBG is lower on all six datasets by 10.28 to 57.16 percentage points, supporting the use of edge-wise KAN functions rather than a generic coordinate-to-basis MLP for generating shared basis functions.

Center-Scale selection. Relative to uniform initialization, CSDS improves ACC by 0.70, 2.56, 1.69, 0.39, 12.94, and 3.75 percentage points on Phoneme, Tecator, Medflies, ECG5000, EthanolLevel, and EOGHorizontalSignal, respectively. Since the architecture and dictionary size remain fixed, these results isolate the benefit of data-guided dictionary allocation.

H.1 Compact Support Analysis of Spline-Wavelet Atoms

This appendix examines the resolution and complexity trade-off between the compact spline-wavelet atoms used by SKAB and the B-spline branch of a standard KAN. The purpose is not to claim that B-spline KANs cannot model local patterns, but to quantify the grid resolution and resulting end-to-end parameter cost required to obtain a comparable local support width.

Support-width calculation. All coordinates are normalized to $[0, 1]$. The order-two spline-wavelet mother atom in SKAB is rescaled to have support $[-\frac{1}{2}, \frac{1}{2}]$. Hence, an atom centered at c with scale s,

$$\psi_{c,s}(t) = \psi\left(\frac{t-c}{s}\right),$$

Method	G	Support width	Params. (K)	ACC
B-spline KAN	5	0.6000	46.0	0.3196 ± 0.0220
B-spline KAN	10	0.3000	68.7	0.4360 ± 0.0390
B-spline KAN	20	0.1500	114.2	0.6478 ± 0.0232
B-spline KAN	40	0.0750	205.1	0.6864 ± 0.0179
B-spline KAN	80	0.0375	386.8	0.6920 ± 0.0230
B-spline KAN	120	0.0250	568.6	0.6970 ± 0.0229
SKAB	–	0.0810/0.1967/0.4055	64.2	$\mathbf{0.8160 \pm 0.0148}$

Table 12: Order-two B-spline grid sweep on EthanolLevel. Support width denotes nominal interior B-spline support for B-spline KAN and the three mean learned first-layer support widths for SKAB. ACC results are mean ± standard deviation over ten runs.

has support

$$\mathrm{supp}(\psi_{c,s}) = \left[c - \frac{s}{2}, c + \frac{s}{2}\right],$$

whose width is exactly s. On EthanolLevel, the three learned first-layer scale groups have mean support widths of 0.0810, 0.1967, and 0.4055 over ten runs.

For an order-two B-spline on a uniform grid with G intervals, an interior B-spline basis function spans $p + 1 = 3$ adjacent intervals. Its nominal support width on $[0, 1]$ is therefore

$$w_{\text{B-spline}} = \frac{3}{G}.$$

Matching the narrowest mean support of SKAB requires approximately $G = 38$. The nearest evaluated configuration is therefore $G = 40$, for which the nominal B-spline support width is 0.075.

Experimental protocol. We conduct a grid sweep on EthanolLevel with $G \in \{5, 10, 20, 40, 80, 120\}$, fixing the B-spline order to two and using ten repeated runs per setting. The sweep changes the grid size in all B-spline KAN layers; consequently, the reported parameter counts represent the total model cost of increasing B-spline resolution in the standard KAN backbone. The B-spline and spline-wavelet models use the same dataset split and training protocol.

Both implementations include a SiLU base branch. Accordingly, the support-width calculation refers to the localized B-spline or spline-wavelet component, whereas the parameter counts include the complete model. We further report first-layer atom supports only: although individual first-layer atoms are compactly supported, the final basis functions obtained after multiple nonlinear layers need not be compactly supported with respect to the original coordinate.

Results. Table 12 shows that increasing the B-spline grid improves performance initially, but the gain saturates well below SKAB. At the closest support-matched setting, $G = 40$, B-spline KAN uses 205.1K parameters, which is $3.19\times$ the 64.2K parameters of SKAB, while obtaining 0.6864 ± 0.0179 ACC compared with 0.8160 ± 0.0148. Increasing the resolution further to $G = 120$ reduces the nominal B-spline support width to 0.025, but raises the model size to 568.6K parameters ($8.85\times$ SKAB) and reaches only 0.6970 ± 0.0229 ACC. Thus, for this controlled EthanolLevel setting, increasing the B-spline grid resolution alone does not yield a comparably effective localized representation.

I Model Size Analysis

We analyze the model size on EthanolLevel to determine whether the performance improvement of SKAB is simply attributable to increased model capacity. As shown in Table 13, SKAB achieves the highest ACC of 0.8160 ± 0.0148 with 64,217 trainable parameters. It uses 45.0% fewer parameters than the MLP, 94.5% fewer than KAN, 88.4% fewer than TSLANet, 81.6% fewer than SoftShape, and 98.9% fewer than DeepFRC. Thus, SKAB provides a more compact architecture than several neural baselines while achieving substantially better classification performance.

Method	Parameters	Test ACC
SKAB (SWK, $Q = 4, M = 3$)	64,217	**0.8160 ± 0.0148**
MLP	116,804	0.5906 ± 0.0237
KAN	1,164,160	0.5188 ± 0.0193
B-spline decomposition	5,124	0.3066 ± 0.0136
Fourier decomposition	5,188	0.3370 ± 0.0178
FPCA	5,124	0.3574 ± 0.0149
AdaFNN	32,778	0.2912 ± 0.0222
FNN	1,440	0.3650 ± 0.0364
FuncAE	5,000	0.3452 ± 0.0270
DeepFRC	6,063,107	0.3530 ± 0.0716
FSFC	20	0.2944 ± 0.0108
TSLANet	553,990	0.4624 ± 0.0856
SoftShape	349,205	0.6530 ± 0.0783
TimeSliver	9,542	0.2534 ± 0.0244

Table 13: Comparison of model size and classification performance on EthanolLevel. #Params. denotes the number of trainable parameters, and test ACC is reported as mean ± SD over ten runs.

Some fixed-basis and feature-selection methods have considerably smaller model sizes because they use predefined functional representations or retain only a small set of selected features, rather than learning a task-adaptive coordinate-to-basis generator. However, these compact methods obtain substantially lower ACC than SKAB. Overall, the results indicate that the improvement of SKAB does not arise from an excessively large model and demonstrate a favorable trade-off between model size and classification accuracy.