DYNAMIC-GRID KOLMOGOROV-ARNOLD NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce DGKAN, a dynamic-grid Kolmogorov-Arnold Network that learns to redistribute a fixed number of grid points across the input domain. Existing KAN use uniformly spaced, predefined grids, implicitly assigning the same representational resolution to all regions. This allocation can be inefficient when function complexity and feature distributions are heterogeneous, motivating the problem of how to optimally allocate a limited grid budget across the domain. DGKAN addresses this problem through backward-guided grid adaptation: output-layer grids are relocated according to local prediction errors, while intermediate-layer grids adapt to the empirical distribution of latent features. To support non-uniform grids, we develop adaptive radial basis functions with grid-dependent bandwidths and selectively introduce variational Bayesian modeling into the Radial Basis Function (RBF) branch for localized shrinkage and uncertainty estimation. Experiments and ablation studies demonstrate the effectiveness of dynamic grid allocation, while generalization tests across diverse tasks validate the generality of DGKAN.

1 INTRODUCTION

Kolmogorov–Arnold Networks (KAN) (Liu et al., 2025b) have attracted increasing interest as a compelling alternative to traditional Multilayer Perceptrons (MLP) (Cybenko, 1989). While preserving the compositional topology of deep networks, KAN replace conventional scalar weights with learnable univariate functions, thereby moving nonlinear representational capacity from network nodes to edges. This edge-centric parameterization enables explicit localized functional modeling through structured and decoupled representations, while providing strong capacity for modeling functions. Specifically, recent studies have successfully deployed KAN variants in physics-informed neural dynamics (Rong et al., 2025; Jiang & Gou, 2025; Ren et al., 2026; Jacob et al., 2025) and continuous control (Fan et al., 2025; Kazemi et al., 2026), biomedical (Dumontet et al., 2026; Du et al., 2026; Zhu et al., 2026) and molecular discovery (Li et al., 2025), computer vision (Lou et al., 2026; Nikonorov et al., 2025) and geospatial remote sensing (Zhang et al., 2026).

Despite these promising capabilities, KAN architectures typically employ a predefined uniform grid, in which grid points are evenly distributed across the input domain (Wang et al., 2025). This design partitions the domain of each learnable univariate function into equally sized subintervals thereby assigning the same representational capacity to all subintervals, regardless of their local functional complexity. Adding grid points subdivides the input domain into smaller subintervals, improving resolution and representational capacity. However, uniform refinement also adds unnecessary subintervals to simple regions, increasing parameter, computational, and memory costs, whereas a coarse grid may fail to capture sharp variations in locally complex regions (Stier et al., 2025). Moreover, data are often unevenly distributed across grid intervals, and densely sampled regions provide richer supervisory signals, enabling the model to learn more detailed representations in those regions. Therefore, the central challenge is to allocate a limited number of grid points more effectively, assigning more grid points to regions with greater functional complexity or higher data, as illustrated in Figure 1.

To address the limitations of fixed-grid KAN, we draw inspiration from adaptive mesh refinement methods (Berger & Oliger, 1984), formulating grid adaptation as an explicit grid update problem rather than a naive parameter expansion. Grid update operates in a backward-guided manner, whereby grid relocation for the Output layer is explicitly directed by a smoothed residual monitor to pinpoint and cover high-prediction-error regimes, while intermediate grids retroactively adapt according to the empirical density distribution of latent features to systematically optimize upstream feature resolution. To prevent grid statistics from being corrupted by anomalous observations, we introduce a selective

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.

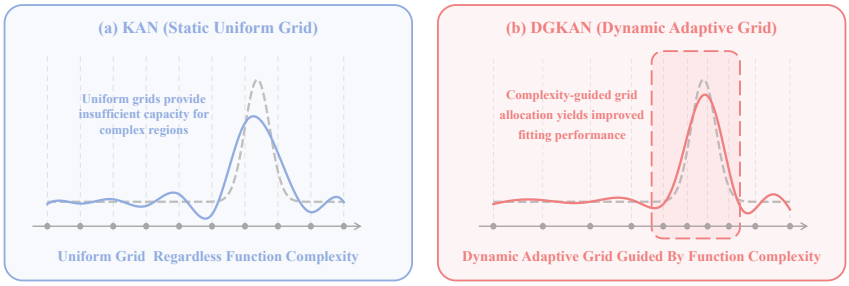


Figure 1: **Activation function fitting capacity and grid allocation.** Gray dashed curves denote the target functions, gray dots represent grid nodes, and solid curves denote the fitted functions.

Variational Bayesian (Graves, 2011) method. By retaining a deterministic backbone to capture global trends and confining Bayesian posterior inference strictly to localized Radial Basis Function (RBF) (Park & Sandberg, 1991) corrections, our model enforces localized shrinkage and quantifies epistemic uncertainty without destabilizing the overall network through full randomization. Finally, to counteract functional distortion induced by grid movement, we perform regularized functional transplantation to project the learned mean functions onto the newly relocated basis.

The main contributions of this work are summarized as follows:

- We propose DGKAN, a novel Kolmogorov-Arnold Network architecture addressing the limitations of uniform grid allocation in standard KAN. By formulating grid adaptation as a budget-constrained allocation problem, DGKAN dynamically reallocates fixed grid point budgets to high complexity regions, improving representational efficiency while preserving network composition.
- We propose a dynamic grid method that adaptively reallocates grid points during training through a layerwise adaptation scheme. Specifically, output layer grids are relocated using prediction residual equidistribution, intermediate layer grids are driven by empirical input distributions. Coupled with a damped displacement rule and a regularized least squares coefficient transfer, our method ensures stable grid relocation while maintaining functional representation continuity training.
- Extensive experiments demonstrate the effectiveness of dynamic grid allocation through diverse function fitting tasks, noise and grid sensitivity analyses, and cross-domain evaluations in scientific computing and time series forecasting, highlighting its robustness, stability, and generality, and consistent performance across diverse datasets.

2 Related Work

Inspired by the Kolmogorov-Arnold representation theorem, a Kolmogorov-Arnold Network (KAN) instantiates $f_{\boldsymbol{\Theta}}$ as a deep composition of L layers. And the forward propagation through layer $l \in \{1, \ldots, L\}$ is expressed in matrix-vector form as:

$$\mathbf{z}_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,2,1}(\cdot) & \cdots & \phi_{l,n_l,1}(\cdot) \\ \phi_{l,1,2}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,n_l,2}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,1,n_{l+1}}(\cdot) & \phi_{l,2,n_{l+1}}(\cdot) & \cdots & \phi_{l,n_l,n_{l+1}}(\cdot) \end{pmatrix}}_{\boldsymbol{\Phi}_l} \mathbf{z}_l, \tag{1}$$

where n_l represents the number of nodes in the l-th layer, $\mathbf{z}_l \in \mathbb{R}^{n_l}$ is the intermediate representation vector, and $\phi_{l,i,j} : \mathbb{R} \to \mathbb{R}$ denotes the univariate activation function mapping the i-th node in layer l to the j-th node in layer $l+1$. Although KAN possess strong functional representation capabilities, their edge activation functions typically rely on fixed grids defined over predefined continuous domains, which severely restricts their representation capacity and spatial adaptability.

Existing studies have extended KAN to established architectures and domain-specific settings. Recent work incorporates KAN layers into Transformers (Yang & Wang, 2025), graph attention networks (Fang et al., 2025), time-series decomposition frameworks (Huang et al., 2025), and frequency-domain operators (Lee et al., 2026). These extensions combine edge-wise learnable functions with

attention, graph aggregation, temporal decomposition, and spectral processing to model nonlinear relationships in structured data. For structured and high-dimensional inputs, FS-KAN (Elbaz et al., 2026) introduces permutation-equivariant representations that respect input symmetries. These methods broaden the applicability of KAN beyond standard feed-forward architectures. However, they mainly modify network topology or information flow, while grid-point allocation across each edge function's input domain remains largely predetermined.

Other studies explore alternative parameterizations and functional operators for KAN edges. KAN 2.0 (Liu et al., 2025a) augments the architecture with multiplication nodes to support explicit symbolic and compositional reasoning. KAFN (Zhang et al., 2025) replaces the conventional grid-based representation with Fourier basis functions, providing spectral parameterization for modeling smooth functional patterns. lmKAN (Pozdnyakov & Schwaller, 2026) employs lookup-based multivariate functions to reduce the cost of evaluating high-dimensional interactions. These methods address limitations of the original KAN formulation by introducing additional operators, alternative function bases, or efficient evaluation mechanisms. Nevertheless, these methods focus on redesigning the functional representation rather than adapting the spatial allocation of a fixed representational budget.

Beyond architectural design, efforts have focused on the optimization dynamics and theoretical properties of KAN. From a theoretical perspective, Zhang & Zhou (2025); Li et al. (2026) derive generalization bounds, Wang et al. (2025) analyze the spectral bias of KAN, and Eshtehardian et al. (2026) establish convergence results for two-layer KAN. These studies provide theoretical foundations for understanding the generalization, expressiveness, and convergence properties of KAN, but mainly consider fixed parameterizations or restricted model settings. From an algorithmic perspective, Rigas et al. (2026) propose a stable grid initialization strategy, whereas De Luca et al. (2026) introduce Gaussian parameterizations to improve training stability and parameter efficiency. In addition, Canevaro et al. (2025) exploit the local parameter plasticity of KAN for out-of-distribution detection. Overall, these studies improve aspects of KAN but generally retain fixed grid layouts. How to jointly optimize edge-function parameters and grid locations therefore remains largely unexplored.

3 METHOD

3.1 PROBLEM SETUP

We target the supervised learning setup over a continuous input domain $\mathcal{X} \subset \mathbb{R}^{d_0}$ mapped to a target space $\mathcal{Y} \subset \mathbb{R}^{d_L}$. Given an empirical dataset $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$ of N observations, where $\boldsymbol{x}_n \in \mathcal{X}$ denotes the input vector and $\boldsymbol{y}_n \in \mathcal{Y}$ represents the ground-truth target, the goal is to parameterize a composite function $f_{\boldsymbol{\Theta}} : \mathcal{X} \to \mathcal{Y}$ with learnable parameters $\boldsymbol{\Theta}$ such that the predicted output $\widehat{\boldsymbol{y}} = f_{\boldsymbol{\Theta}}(\boldsymbol{x})$ accurately fits the target $\boldsymbol{y}$.

3.2 DYNAMIC GRID

To overcome the allocation mismatch of fixed-grid KAN, we introduce a dynamic-grid mechanism inspired by r-adaptive mesh methods. For the edge activation function $\phi_{l,i,j}(x)$, we denote its associated spatial grid by $\mathbf{G}_{l,i,j} = [g_1, \ldots, g_K]^\top$, where $g_1 < \cdots < g_K$ and K is the number of grid points. The grid points serve explicitly as the spatial centers of localized basis functions $\psi(u; g_k)$, parameterizing the edge function as $\phi_{l,i,j}(u) = a \cdot \mathrm{SiLU}(u) + \sum_{k=1}^{K} w_k \psi(u; g_k)$. While K remains constant throughout training, the grid coordinates $\mathbf{G}_{l,i,j}$ are periodically updated to reallocate resolution dynamically.

Dynamic grid in output layer. For the output layer, grid relocation is guided by prediction residuals to dynamically refine the edge parameterization. Let $z_{L,n}^{(i)}$ denote the i-th component of the input representation to the output layer, and let $\hat{\mathbf{y}}_n, \mathbf{y}_n \in \mathbb{R}^{d_{\text{out}}}$ denote the corresponding prediction and ground-truth target for sample n, respectively. We then quantify the prediction discrepancy using the mean absolute residual over the output dimensions, $e_n = \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_1 / d_{\text{out}}$, which is subsequently aggregated within each grid interval to guide grid adaptation.

The grid partitions the active domain into $K-1$ intervals, with the m-th interval defined as $I_m = [g_m, g_{m+1}]$, for $m = 0, \ldots, K-2$. The index set of samples whose corresponding output-layer input component lies within the m-th interval is defined as $\mathcal{S}_m = \{n \mid z_{L,n}^{(i)} \in I_m\}$. Based on the

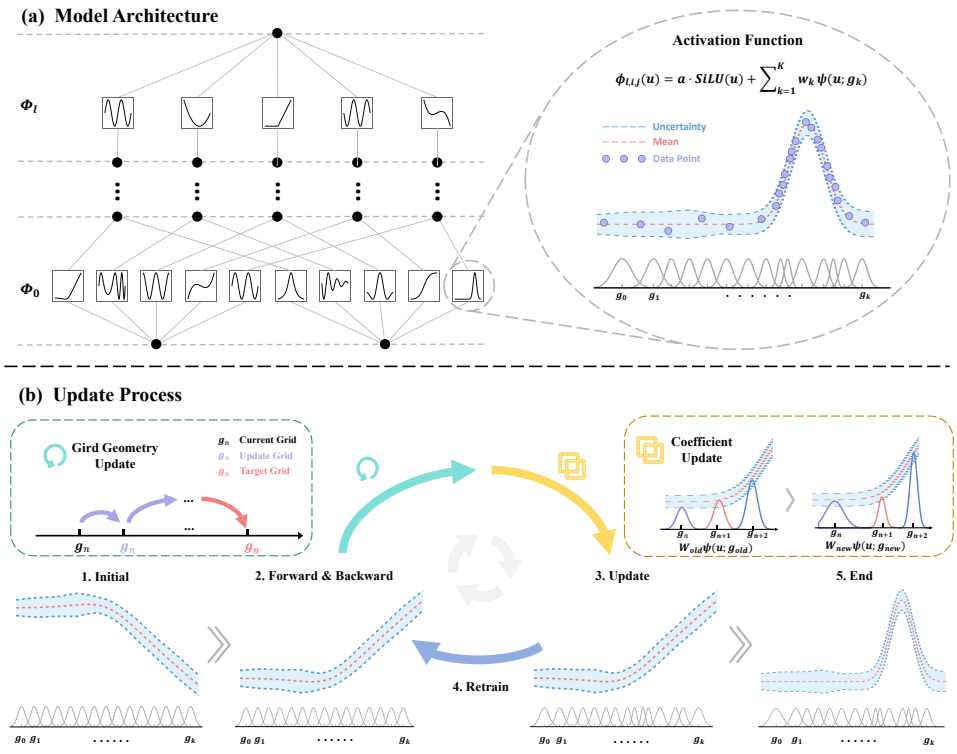


Figure 2: **Overall Architecture and Grid Adaptation Workflow of DGKAN. (a) Model Architecture:** Shows the deep network structure, whose edge activation function combines a deterministic base branch and a Bayesian variational ARBF branch (formulation shown on the right). **(b) Grid Adaptation:** The adaptation process sequentially computes the target grid, performs damped grid relocation with adaptive bandwidth adjustment, and completes regularized coefficient re-projection before continuing retraining.

prediction residuals, we define the local error measure for each grid interval as

$$M_m = \begin{cases} \max\left\{\sum_{n\in\mathcal{S}_m} e_n \,/\, |\mathcal{S}_m|,\ \varepsilon\right\}, & |\mathcal{S}_m| > 0, \\ \varepsilon, & |\mathcal{S}_m| = 0, \end{cases} \tag{2}$$

where $\varepsilon > 0$ enforces the lower bound $M_m \geq \varepsilon$ for every grid interval, including intervals containing no samples. To reduce local fluctuations in the interval-wise error measure, we smooth the discrete sequence $\{M_m\}$ by a three-point weighted average. For each interior interval ($m = 1, \ldots, K-3$), the smoothed measure is defined as

$$\widetilde{M}_m = M_{m-1}/4 + M_m/2 + M_{m+1}/4. \tag{3}$$

The two boundary intervals are treated separately according to the prescribed boundary smoothing rule. The weight associated with the m-th grid interval is defined as $\omega_m = \widetilde{M}_m(g_{m+1} - g_m)$. By accumulating these interval weights, we define $C_m = \sum_{q=0}^{m} \omega_q$ denote the cumulative weight up to the m-th grid interval, and define the total weight over the domain covered by the grid as $C_{\text{total}} = C_{K-2}$. Let $C_{-1} = 0$. Based on the cumulative interval weights, we define a piecewise-linear cumulative function $C(x)$. For $x \in [g_m, g_{m+1}]$,

$$C(x) = C_{m-1} + \widetilde{M}_m(x - g_m). \tag{4}$$

Following the equidistribution principle, the candidate grid points are obtained by uniformly partitioning the total cumulative weight:

$$\widetilde{g}_k = C^{-1}\left(\frac{k}{K-1} C_{\text{total}}\right), \qquad k = 0, \ldots, K-1. \tag{5}$$

Dynamic grid in intermediate layer. For the intermediate layers, grid adaptation is driven by the empirical distribution of the layer inputs rather than by prediction residuals. Let $z_{l,n}^{(i)}$ denote the i-th component of the input to layer l for sample n, where $1 \le l < L$. For a fixed input component i, we omit the index i from the distribution and grid notation below. Given the observed values $\{z_{l,n}^{(i)}\}_{n=1}^{N_l}$, the corresponding empirical cumulative distribution function (ECDF) is defined as

$$\widehat{F}_l(u) = \sum\nolimits_{n=1}^{N_l} \mathbb{I}\Big(z_{l,n}^{(i)} \le u\Big) / N_l, \tag{6}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. The candidate grid is then defined by evaluating the generalized inverse of $\widehat{F}_l$ at uniformly spaced probability levels:

$$\widetilde{g}_{k,l} = \widehat{F}_l^{-1}(k/(K-1)), \qquad k = 0, \dots, K-1. \tag{7}$$

By construction, adjacent grid intervals contain approximately equal empirical probability mass, yielding finer resolution in regions of higher empirical concentration.

Once the candidate grid is obtained, let $\mathbf{G}$, $\widetilde{\mathbf{G}}$, and $\mathbf{G}^{\mathrm{new}}$ denote the current, candidate, and updated grids, respectively. The grid is then updated as

$$\mathbf{G}^{\mathrm{new}} = \mathcal{R}_{\mathrm{grid}}\Big[\mathbf{G} + \mathcal{P}_{[-\delta_{\max},\, \delta_{\max}]}\Big(\lambda(\widetilde{\mathbf{G}} - \mathbf{G})\Big)\Big]. \tag{8}$$

where $\lambda \in (0, 1]$ controls the relocation rate and $\delta_{\max} > 0$ denotes the maximum displacement allowed for each grid point in a single adaptation step. The operator $\mathcal{P}_{[-\delta_{\max},\, \delta_{\max}]}(\cdot)$ restricts each grid-point displacement to the prescribed range, while $\mathcal{R}_{\mathrm{grid}}(\cdot)$ keeps the relocated grid points within the grid domain and enforces strict ordering and minimum spacing.

3.3 Adaptive Radial Basis Functions

For the one-dimensional activation functions, we employ Gaussian radial basis functions centered at the grid points. To account for grid nonuniformity, we introduce *Adaptive Radial Basis Functions* (ARBF), whose bandwidths are determined by the local grid geometry. We first define the lower-bounded spacing between adjacent grid points as

$$\Delta_k = \max\{g_{k+1} - g_k,\, h_{\min}\}, \qquad k = 0, \dots, K-2, \tag{9}$$

where $h_{\min} > 0$ provides a numerical lower bound. The local width associated with the k-th grid point is then defined as

$$\widetilde{h}_k = \begin{cases} \Delta_0, & k = 0, \\ (\Delta_{k-1} + \Delta_k)/2, & 0 < k < K-1, \\ \Delta_{K-2}, & k = K-1. \end{cases} \tag{10}$$

The corresponding width is defined as $h_k = \max\{\kappa_h \widetilde{h}_k,\, h_{\min}\}$, where $\kappa_h > 0$ is a scaling factor. With the adaptive width h_k determined above, the k-th Gaussian basis function is defined as

$$\rho_k(u) = \exp\big[-(u - g_k)^2/(2h_k^2)\big], \qquad k = 0, \dots, K-1. \tag{11}$$

To mitigate magnitude fluctuations induced by non-uniform node distributions, we apply a softmax-like partition-of-unity normalization across the basis functions:

$$\psi_k(u) = \rho_k(u) / \max\{\textstyle\sum_{q=0}^{K-1} \rho_q(u),\, \epsilon_{\mathrm{norm}}\}. \tag{12}$$

where $\epsilon_{\mathrm{norm}} > 0$ is a small constant preventing division by zero.

Grid relocation changes the centers and widths of the RBF basis functions. Consequently, retaining the original coefficients after a grid update may alter the represented function. To approximately preserve the function across grid updates, we transfer the RBF coefficients from the old basis to the updated basis. This transfer is formulated as a regularized least-squares problem.

Let Φ_{old} and Φ_{new} denote the RBF design matrices evaluated on the same calibration inputs using the old and updated grids, respectively. Given the original coefficient matrix $\mathbf{U}_{\mathrm{old}}$, the updated coefficients are obtained as

$$\mathbf{U}_{\mathrm{new}} = \arg\min_{\mathbf{U}} \|\Phi_{\mathrm{new}}\mathbf{U} - \Phi_{\mathrm{old}}\mathbf{U}_{\mathrm{old}}\|_F^2 + \lambda_{\mathrm{proj}}\|\mathbf{U}\|_F^2. \tag{13}$$

where $\lambda_{\mathrm{proj}} > 0$ is a regularization parameter that improves numerical stability and discourages excessively large coefficients.

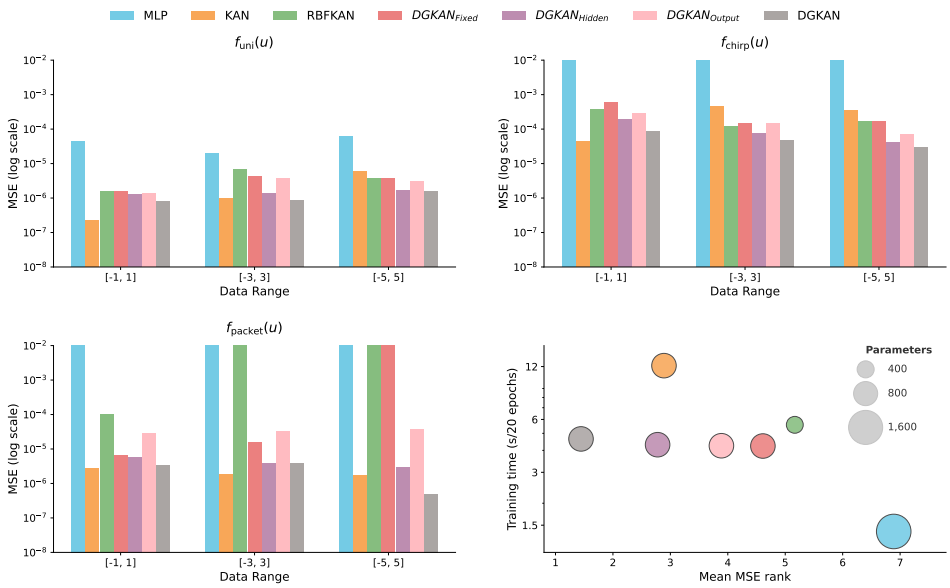


Figure 3: **Function approximation performance and ablation analysis across target functions and input domains.** Approximation errors (MSE, log scale) of MLP, KAN, RBFKAN, and DGKAN variants on f_{uni}, f_{chirp}, and f_{packet} over three data ranges, $[-1, 1]$, $[-3, 3]$, and $[-5, 5]$. DGKAN achieves the lowest MSE in most settings, with larger gains on functions with spatially varying or localized high-frequency structures and wider data ranges; the bottom-right panel summarizes mean MSE rank, training time, and trainable parameters.

3.4 Bayesian Variational RBF Modeling

To suppress overfitting and numerical instability induced by potential near-singular eigenvalues during adaptive grid updates, as well as to quantify the prediction uncertainty of localized RBF corrections, we model the RBF coefficients as random variables. Importantly, the Bayesian treatment is applied exclusively to the RBF branch, whereas the base branch remains deterministic.

We model the RBF coefficients using a variational Gaussian distribution $q(w) = \mathcal{N}(\mu, \sigma^2)$, under an isotropic zero-mean Gaussian prior $p(w) = \mathcal{N}(0, \sigma_p^2)$. The shrinkage effect of the Gaussian prior effectively dampens abnormal eigenvalues and suppresses overfitting. The overall training objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{loss}} + (\beta / N_{\text{data}})\, \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{reg}}, \tag{14}$$

where $\mathcal{L}_{\text{loss}}$ varies depending on the task. The Kullback-Leibler (KL) divergence between the variational posterior and the prior is analytically evaluated as:

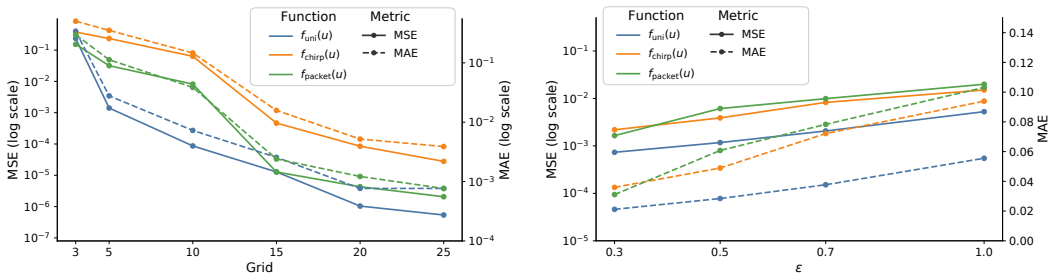
$$\mathcal{L}_{\text{KL}} = \sum_k \left[\log \frac{\sigma_p}{\sigma_k} + \frac{\sigma_k^2 + \mu_k^2}{2\sigma_p^2} - \frac{1}{2} \right]. \tag{15}$$

At inference time, predictive output and epistemic uncertainty are estimated by taking the sample mean and variance over Monte Carlo predictions drawn from the variational posteriors.

To promote parameter sparsity, we impose penalties on the layer parameters. The total regularization loss is defined as

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{edge}} \left(\overline{|W_{\text{base}}|} + \overline{|W_{\text{rbf}}|} \right) + \lambda_{\text{basis}} \overline{|\boldsymbol{\mu}|}, \tag{16}$$

where $\overline{|\cdot|}$ denotes mean absolute value, W_{base} and W_{rbf} denote deterministic base linear weights and edge-wise RBF scaling factors, respectively, and $\boldsymbol{\mu}$ represents variational posterior means of the RBF coefficients. The hyperparameters λ_{edge} and λ_{basis} control the edge- and basis-level regularization strengths, respectively.



(a) **Sensitivity analysis of grid size.** (b) **Sensitivity analysis of noise.**

Figure 4: (a) **Sensitivity analysis of grid size.** Model performance across three target functions under varying grid sizes $G \in \{3, 5, 10, 15, 20, 25\}$. (b) **Sensitivity analysis of noise.** Evaluation of model performance across three benchmark functions under different observational noise levels $\varepsilon \in \{0.3, 0.5, 0.7, 1.0\}$. MSE is plotted on the left logarithmic axis (solid lines), while MAE is on the right linear axis (dashed lines).

4 Experiments

4.1 Function Approximation

To compare dynamic and fixed grids under different spatial-frequency distributions and localized features, we construct three target functions with increasing representation difficulty, namely $f_{\text{uni}}(u) = \sin(2\pi \cdot 3u)$, $f_{\text{chirp}}(u) = \sin\big(2\pi\left[u + 4(u+1)^2\right]\big)$, and $f_{\text{packet}}(u) = \sin(2\pi \cdot 2u) + 0.8\exp\left[-\frac{1}{2}\left(\frac{u-0.35}{0.08}\right)^2\right]\sin(2\pi \cdot 18u)$.

For each function, we generate 3,000 training, 1,000 validation, and 10,000 test samples over the ranges $[-1, 1]$, $[-3, 3]$, and $[-5, 5]$. Models are trained with MSE loss and evaluated using MSE and MAE. We compare DGKAN with a standard MLP (Cybenko, 1989), B-spline KAN (Liu et al., 2025b), and RBFKAN (Li, 2024). We further consider three ablations: $\text{DGKAN}_{\text{Fixed}}$, which uses fixed uniform grids in all layers; $\text{DGKAN}_{\text{Hidden}}$, which adapts grids only in hidden layers; and $\text{DGKAN}_{\text{Output}}$, which adapts grids only in the output layer.

As shown in Figure 3, dynamic grid adaptation consistently improves approximation accuracy, especially for functions with heterogeneous local complexity. Although KAN performs well on the uniformly oscillatory function, DGKAN achieves lower MSE across all data ranges, with larger gains on the chirp and packet functions. For f_{packet}, DGKAN maintains low error as the input range expands, whereas fixed-grid models degrade substantially. This indicates that DGKAN reallocates grid points toward complex regions instead of wasting resolution in simpler ones.

The ablation results demonstrate that the two adaptation mechanisms are complementary. $\text{DGKAN}_{\text{Fixed}}$ consistently produces higher errors, while $\text{DGKAN}_{\text{Hidden}}$ and $\text{DGKAN}_{\text{Output}}$ provide only partial improvements. Full DGKAN achieves the lowest error across all functions and input ranges, with the largest advantage on the localized packet function as the range expands.

The lower-right panel of Figure 3 summarizes the trade-off among error, training cost, and model size. Full DGKAN achieves the lowest mean MSE rank with moderate parameter count and training time, suggesting that its gains arise from complexity-aware grid allocation rather than increased model capacity. Under a fixed grid-point budget, allocating resolution according to function complexity is therefore more effective than using uniformly distributed grids.

4.2 Grid Sensitivity Analysis

To investigate the impact of grid density on fitting precision, we conduct a grid sensitivity analysis on DGKAN. In KAN-derived architectures, the grid size K serves as a critical hyperparameter governing model fitting performance. We evaluate the model's accuracy across a range of grid numbers $K \in \{3, 5, 10, 15, 20, 25\}$. Specifically, the input domain of the data is defined over the interval $[-5, 5]$. All other experimental setups remain identical to those in Section 4.1.

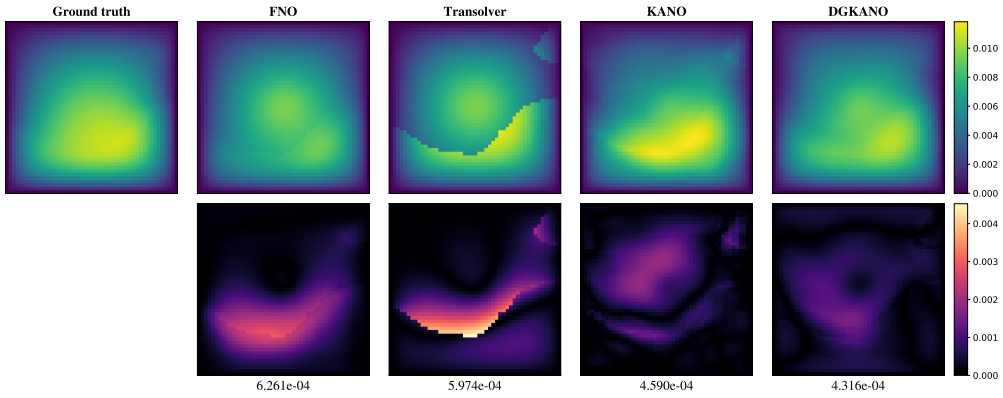


Figure 5: **Generalization and qualitative evaluation on 2D Darcy Flow.** The top row displays the target solution (Ground Truth) and the predicted solution fields from FNO, Transolver, KANO, and DGKANO (ours) for a specific problem instance. The bottom row visualizes the magnitude of the absolute error between each model's prediction and the ground truth.

As depicted in Figure 4a, where MSE (solid lines) and MAE (dashed lines) are shown on logarithmic scales, increasing the total grid budget initially yields substantial accuracy gains, particularly in coarse-grid regimes ($K \leq 10$) where errors decrease by several orders of magnitude. However, as the grid budget expands further ($K \geq 15$), the performance trajectories quickly enter an asymptotic plateau, showing pronounced diminishing returns. For instance, increasing K from 15 to 25 offers only marginal reconstruction improvements while incurring significantly higher computational and memory burdens. This clear saturation phenomenon demonstrates that unconstrained global grid expansion is both computationally inefficient and unnecessary. Instead, it strongly validates our core premise: once a reasonable total budget is provided, performance bottle-necks are better addressed by adaptively allocating grid nodes to high-complexity or rapidly changing regions rather than naively increasing the total node count across all evaluated settings.

4.3 Noise Sensitivity Analysis

To systematically evaluate the performance of DGKAN under noisy conditions, we construct training and validation datasets by introducing synthetic Gaussian noise with varying noise levels $\eta \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$, defined as $y_{\text{noise}} = y + \eta \cdot \text{std}(y) \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ and the input range is constrained to $[-5, 5]$. All other experimental setups remain identical to those in Section 4.1.

As depicted in Figure 4b, where MSE (solid lines) is plotted on the left log-scale axis and MAE (dashed lines) on the right axis, the model exhibits stable convergence and robust error scaling across all benchmark functions. Crucially, as the noise intensity grows more than three-fold (from 0.3 to 1.0), both MSE and MAE metrics increase in a predictable, sub-linear trajectory without catastrophic performance degradation. Simple dynamics like $f_{\text{uni}}(u)$ consistently yield lower absolute reconstruction errors (MSE $< 10^{-2}$ at $\varepsilon = 1.0$), whereas high frequency functions such as $f_{\text{chirp}}(u)$ and $f_{\text{packet}}(u)$ exhibit slightly higher error baselines due to target spectral complexity. Furthermore, the parallel linear slopes observed across all functions in the log-scale MSE curves confirm that the model's inherent noise robustness scales consistently regardless of signal non-linearity.

4.4 Generality Evaluation

To assess the versatility and plug-and-play capability of DGKAN in scientific computing, we replace only the KAN components in KANO with DGKAN, denoting the model as DGKANO, while keeping all other neural components, optimization strategies, and evaluation settings unchanged. We compare DGKANO with FNO (Li et al., 2021), Transolver (Wu et al., 2024), and the original KANO. As shown in Figure 5, all methods capture the global field structure, but FNO oversmooths peak values, Transolver introduces severe boundary artifacts (> 0.004), and KANO, despite reducing boundary errors, exhibits diffuse residuals around the central active region with an absolute error of 4.590×10^{-4}. In contrast, DGKANO closely matches the ground truth, removes localized high-

Table 1: Long-term forecasting results. The input sequence length is fixed at 96 for all methods, and the forecasting horizons are $F \in \{96, 192, 336, 720\}$. *Avg* denotes the average performance across the four forecasting horizons. Results marked with "*" are reported in the original TimeKAN paper (Huang et al., 2025). TimeDGKAN is obtained by replacing the KAN modules in TimeKAN with DGKAN, while keeping all other components unchanged.

Models		**TimeDGKAN Ours**		TimeKAN 2025		TimeMixer* 2024		iTransformer* 2024b		Time-FFM* 2024a		FreTS* 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.367**	0.399	**0.367**	**0.395**	0.385	0.402	0.386	0.405	0.385	0.400	0.395	0.407
	192	0.420	0.421	**0.414**	**0.420**	0.443	0.430	0.441	0.436	0.439	0.430	0.490	0.477
	336	**0.444**	**0.430**	0.445	0.434	0.512	0.470	0.487	0.458	0.480	0.449	0.510	0.480
	720	**0.440**	**0.455**	0.444	0.459	0.497	0.476	0.503	0.491	0.462	0.456	0.568	0.538
	Avg	**0.417**	**0.426**	**0.417**	0.427	0.459	0.444	0.454	0.447	0.442	0.434	0.491	0.475
ETTh2	96	0.293	0.343	0.290	**0.340**	**0.289**	0.342	0.297	0.349	0.301	0.351	0.332	0.387
	192	**0.369**	0.395	0.375	**0.392**	0.378	0.397	0.380	0.400	0.378	0.397	0.451	0.457
	336	**0.388**	**0.414**	0.423	0.435	0.432	0.434	0.428	0.432	0.422	0.431	0.466	0.473
	720	0.428	**0.443**	0.443	0.449	0.464	0.464	**0.427**	0.445	**0.427**	0.444	0.485	0.471
	Avg	**0.370**	**0.399**	0.383	0.404	0.390	0.409	0.383	0.407	0.382	0.406	0.433	0.446
ETTm1	96	0.319	0.360	0.322	0.361	**0.317**	**0.356**	0.334	0.368	0.336	0.369	0.337	0.374
	192	0.361	0.384	**0.357**	**0.383**	0.367	0.384	0.377	0.391	0.378	0.389	0.382	0.398
	336	**0.381**	0.405	0.382	**0.401**	0.391	0.406	0.426	0.420	0.411	0.410	0.420	0.423
	720	0.450	0.441	**0.445**	**0.435**	0.454	0.441	0.491	0.459	0.469	0.441	0.490	0.471
	Avg	0.378	0.398	**0.376**	**0.395**	0.382	0.397	0.407	0.410	0.399	0.402	0.407	0.417
ETTm2	96	**0.172**	**0.254**	0.174	0.255	0.175	0.257	0.180	0.264	0.181	0.267	0.186	0.275
	192	**0.235**	**0.298**	0.239	0.299	0.240	0.302	0.250	0.309	0.247	0.308	0.259	0.323
	336	0.303	0.342	**0.301**	**0.340**	0.303	0.343	0.311	0.348	0.309	0.347	0.349	0.386
	720	0.393	**0.395**	0.395	0.396	**0.392**	0.396	0.412	0.407	0.406	0.404	0.559	0.511
	Avg	**0.276**	**0.322**	0.277	**0.322**	0.277	0.324	0.288	0.332	0.286	0.332	0.339	0.347

gradient artifacts, and achieves the lowest absolute error of 4.316×10^{-4}, a 6.0% reduction relative to KANO. These results support the generalizability of DGKAN to challenging operator-learning tasks.

To test whether this plug-and-play capability extends to long-term time-series forecasting, we replace only the KAN modules in TimeKAN with DGKAN, while keeping the remaining components, optimization procedures, and evaluation settings unchanged. As shown in Table 1, we compare TimeDGKAN with TimeKAN, TimeMixer (Wang et al., 2024), iTransformer (Liu et al., 2024b), Time-FFM (Liu et al., 2024a), and FreTS (Yi et al., 2023) across four datasets and four forecasting horizons. TimeDGKAN achieves the best or tied-best performance in a substantial fraction of dataset-horizon settings and obtains the best average results on ETTh2 and ETTm2. Its competitive performance on the remaining datasets shows that dynamic grid adaptation can be transferred to existing KAN architectures without modifying their surrounding components or training procedures.

5 Conclusion

In this work, we proposed DGKAN, a dynamic-grid KAN that adaptively reallocates a fixed number of grid points according to function complexity. Experiments across function approximation, time-series forecasting, and scientific computing demonstrate the effectiveness, robustness, and generality of adaptive grid allocation, showing that DGKAN can better capture complex regions while avoiding inefficient uniform resolution under a fixed grid-point budget.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript polishing, grammar checking, and assistance with code optimization and debugging. We did not use generative AI tools to generate core research ideas or original research content from scratch. Non-text asset generation, including image creation, dataset construction, and synthesis of experimental data, was not applicable to this work. Additionally, generative AI tools assisted the authors in identifying key points and summarizing content from manually identified background and related work. All AI-assisted outputs were reviewed by the authors. In particular, LLM-assisted language edits were verified to preserve the original mathematical and technical precision, while AI-assisted code was rigorously tested, benchmarked, and validated by the authors to ensure correctness. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Marsha J Berger and Joseph Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3):484–512, 1984.

Alessandro Canevaro, Julian Schmidt, Sajad Marvi, Hang Yu, Georg Martius, and Julian Jordan. Advancing out-of-distribution detection via local neuroplasticity. In *International Conference on Learning Representations*, volume 2025, pp. 51063–51087, 2025.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Pasquale De Luca, Emanuel Di Nardo, Livia Marcellino, and Angelo Ciaramella. Stable and efficient gaussian-based kolmogorov–arnold networks. *Mathematics*, 14(3):513, 2026.

Zhenshun Du, Zhiju Wang, Yu Chen, Boyou Li, Xin Wan, Tianyi Ren, Haowei Chen, Lei Liu, Qing Jin, Yongle Zhang, et al. Kan-prospect: a kolmogorov–arnold networks–integrated framework for predicting the effects and adverse reactions of natural products via transfer learning. *Scientific Reports*, 2026.

Louis Dumontet, So-Ra Han, Axel Prouvost, Jun Hyuck Lee, Tae-Jin Oh, and Mingon Kang. Interpretable kolmogorov-arnold networks for enzyme commission number prediction. *npj Artificial Intelligence*, 2(1):11, 2026.

Ran Elbaz, Guy Bar-Shalom, Yam Eitan, Fabrizio Frasca, and Haggai Maron. Fs-kan: Permutation equivariant kolmogorov-arnold networks via function sharing. In *International Conference on Learning Representations*, volume 2026, pp. 99537–99564, 2026.

Seyed Mohammad Eshtehardian, Mohammad Hossein Yassaee, and Babak Khalaj. On the convergence of two-layer kolmogorov–arnold networks with first-layer training. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuqian Fan, Yi Li, Chong Yan, Yaqi Liang, Ye Yuan, Zihang Li, Meng Sun, Lixin Wang, Xiaoying Wu, Zhiwei Ren, et al. A physics-enhanced hybrid kolmogorov–arnold network with dynamic coupling for interpretable battery state-of-charge estimation. *Applied Energy*, 400:126533, 2025.

Taoran Fang, Tianhong Gao, Chunping Wang, Wei Chow, Lei CHEN, and Yang Yang. Kaa: Kolmogorov-arnold attention for enhancing attentive graph neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 1517–1541, 2025.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 28514–28529, 2025.

Bruno Jacob, Amanda A Howard, and Panos Stinis. Spikans: separable physics-informed kolmogorov–arnold networks. *Machine Learning: Science and Technology*, 6(3):035060, 2025.

Quan Jiang and Zhiyong Gou. Solutions to two-and three-dimensional incompressible flow fields leveraging a physics-informed deep learning framework and kolmogorov–arnold networks. *International Journal for Numerical Methods in Fluids*, 97(4):665–673, 2025.

Aneseh Kazemi, Alireza Afzal Aghaei, and Alireza Nazemi. Physics-informed kolmogorov-arnold networks for granular fuzzy fractional optimal control problems. *IEEE Transactions on Fuzzy Systems*, 2026.

Jin Lee, Ziming Liu, Xinling Yu, Yixuan Wang, Haewon Jeong, Murphy Niu, and Zheng Zhang. Kano: Kolmogorov-arnold neural operator. In *International Conference on Learning Representations*, volume 2026, pp. 71003–71031, 2026.

Longlong Li, Yipeng Zhang, Guanghui Wang, and Kelin Xia. Kolmogorov–arnold graph neural networks for molecular property prediction. *Nature Machine Intelligence*, 7(8):1346–1354, 2025.

Pengqi Li, Lizhong Ding, Jiarun Fu, Chunhui Zhang, Guoren Wang, and Ye Yuan. Generalization bounds for kolmogorov-arnold networks (kans) and enhanced kans with lower lipschitz complexity. *Advances in Neural Information Processing Systems*, 38:77098–77132, 2026.

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

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

Qingxiang Liu, Xu Liu, Chenghao Liu, Qingsong Wen, and Yuxuan Liang. Time-ffm: Towards lm-empowered federated foundation model for time series forecasting. *Advances in Neural Information Processing Systems*, 37:94512–94538, 2024a.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024b.

Ziming Liu, Max Tegmark, Pingchuan Ma, Wojciech Matusik, and Yixuan Wang. Kolmogorov-arnold networks meet science. *Physical Review X*, 15(4):041051, 2025a.

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, 2025b.

Shimeng Lou, Yubin Shao, and Qingzhi Du. Kolmogorov-arnold optimized unet: An enhanced image segmentation model based on kolmogorov-arnold network and convolutional kolmogorov-arnold network. *Engineering Applications of Artificial Intelligence*, 173:114405, 2026.

Artem Nikonorov, Georgy Perevozchikov, Andrei Korepanov, Nancy Mehta, Mahmoud Afifi, Egor Ershov, and Radu Timofte. Color matching using hypernetwork-based kolmogorov-arnold networks. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7099–7109. IEEE, 2025.

Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

Sergey Pozdnyakov and Philippe Schwaller. Lookup multivariate kolmogorov-arnold networks. In *International Conference on Learning Representations*, volume 2026, pp. 114701–114734, 2026.

Yi Ren, Ligang Wang, and Haitao Ma. Physics-informed kolmogorov-arnold networks with residual-based adaptive distribution for scattered acoustic field prediction. *Applied Intelligence*, 56(5):143, 2026.

Spyros Rigas, Dhruv Verma, Georgios Alexandridis, and Yixuan Wang. Initialization schemes for kolmogorov–arnold networks: An empirical study. In *International Conference on Learning Representations*, volume 2026, pp. 153078–153113, 2026.

Desheng Rong, Zhongbao Lin, and Guomin Xie. Recurrent fourier-kolmogorov arnold networks for photovoltaic power forecasting. *Scientific reports*, 15(1):4684, 2025.

Noah Stier, Alex Rich, Pradeep Sen, and Tobias Höllerer. Anigrad: Anisotropic gradient-adaptive sampling for 3d reconstruction from monocular video. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21814–21823. IEEE, 2025.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024.

Yixuan Wang, Jonathan Siegel, Ziming Liu, and Thomas Hou. On the expressiveness and spectral bias of kans. In *International Conference on Learning Representations*, volume 2025, pp. 27492–27511, 2025.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. 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. 53681–53705. PMLR, 21–27 Jul 2024.

Xingyi Yang and Xinchao Wang. Kolmogorov-arnold transformer. In *International Conference on Learning Representations*, volume 2025, pp. 76063–76086, 2025.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in neural information processing systems*, 36:76656–76679, 2023.

Jusheng Zhang, Yijia Fan, Kaitong Cai, Keze Wang, and Wenhao Wang. Kolmogorov-arnold fourier networks. *arXiv preprint arXiv:2502.06018*, 2025.

Weizhen Zhang, Zhihui Li, and Tong Zhen. Hybrid-kanet: a hyperspectral remote sensing crop classification method based on the kolmogorov–arnold network. *Geo-spatial Information Science*, 29(2):1121–1143, 2026.

Xianyang Zhang and Huijuan Zhou. Generalization bounds and model complexity for kolmogorov–arnold networks. In *International Conference on Learning Representations*, volume 2025, pp. 18581–18603, 2025.

Junan Zhu, Zhizhe Tang, Zheng Liang, Ping Ma, and Chuanjian Wang. Kanseg: An efficient medical image segmentation model based on kolmogorov-arnold networks for multi-organ segmentation. *Digital Signal Processing*, 168:105472, 2026.