Oregon State University

COLLEGE OF ENGINEERING

MetaCluster: Enabling Deep Compression of Kolmogorov-Arnold Networks

Matthew Raffel*, Adwaith Renjith*, Lizhong Chen

Oregon State University, Corvallis, OR

NeurIPS 2026

NEURAL INFORMATION PROCESSING SYSTEMS

* Equal contribution

Oregon State University
College of Engineering

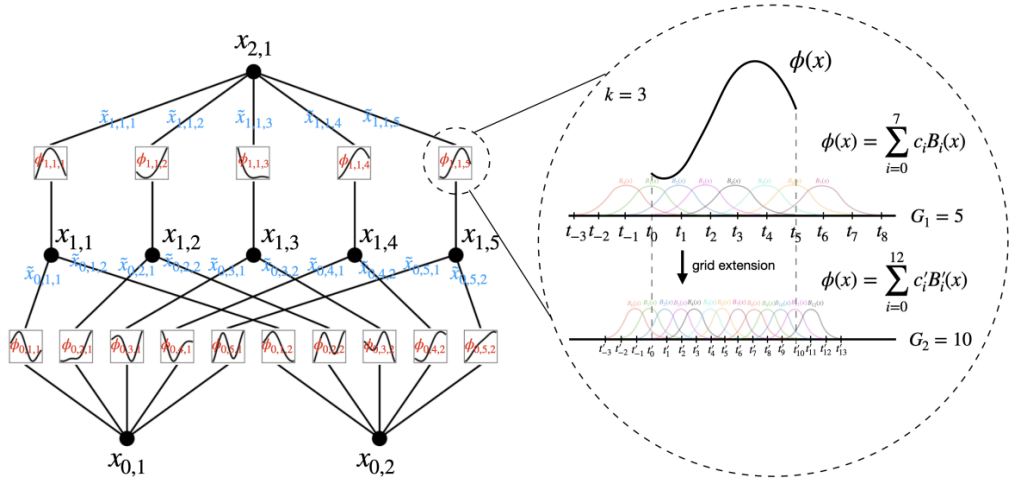
In a Nutshell…

- KANs store a represents each edge with a function represented as a learnable vector of basis coefficients, which increases storage costs compared to an MLP.
- We propose **MetaCluster**, a framework that makes KANs highly compressible by mapping the coefficient vectors to a lower dimensional manifold to facilitate easier weight clustering.
- We reduce storage by about **80× on image classification tasks** and **124.1× on 1,000-dimensional equation modeling** with **minimal impacts to accuracy**.

Oregon State University
College of Engineering

Kolmogorov-Arnold Networks (KANs)

- A conventional MLP learns **one scalar weight** per connection, but a KAN learns **a function on each connection** as a weighted sum of basis functions.
- Each connection therefore stores **a vector of coefficients**.
- With 9 coefficients per edge, a KAN has **9× as many edge parameters** as an MLP with the same connections.



Oregon State University
College of Engineering

How Do We Compress KANs, and Why Is It Hard?

- **Weight sharing** compresses a model by grouping similar weights, storing one representative per group in a codebook, and giving each connection an index.
- Weight sharing is especially useful for KANs because **one short index can replace an entire coefficient vector**.
- High-dimensional coefficient vectors are **difficult to cluster tightly**, so replacing them with a shared representative can distort the learned edge functions and hurt accuracy.
- To our knowledge, **there is no existing approach to cluster KAN edges for compression**.

Oregon State University
College of Engineering

Method: MetaCluster

1. Train

$$M_\theta(\mathbf{z}_i) = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{z}_i + \mathbf{b}_1) + \mathbf{b}_2 = \mathbf{w}_i$$

A jointly trained meta-learner maps small embeddings to a low-dimensional manifold of coefficient vectors for easier clustering.

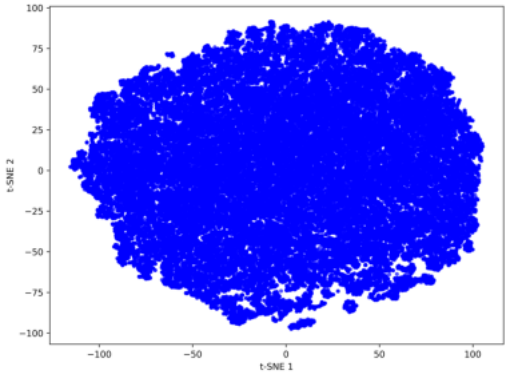
2. Cluster

Run K-means within each layer. Store one centroid index per edge.

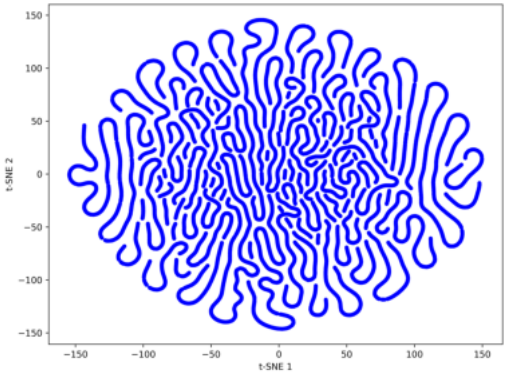
3. Fine-tune

Discard the meta-learner and embeddings. Refine the centroids with indices fixed with fine-tuning.

Standard KAN



With meta-learner

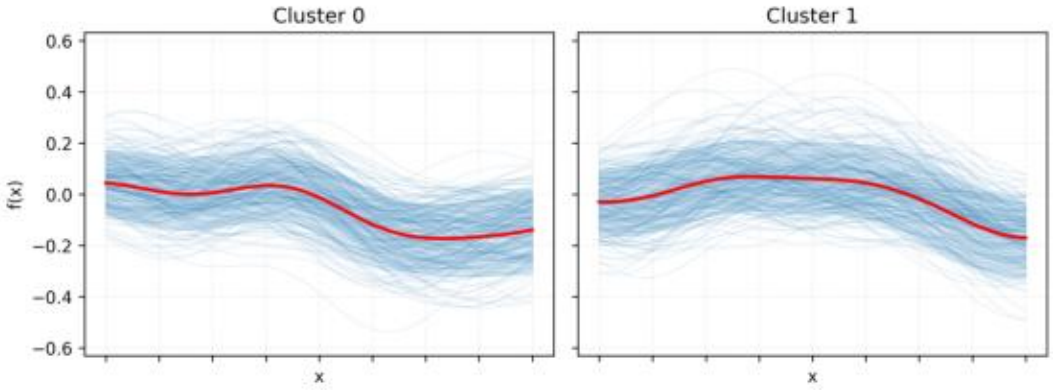


Oregon State University
College of Engineering

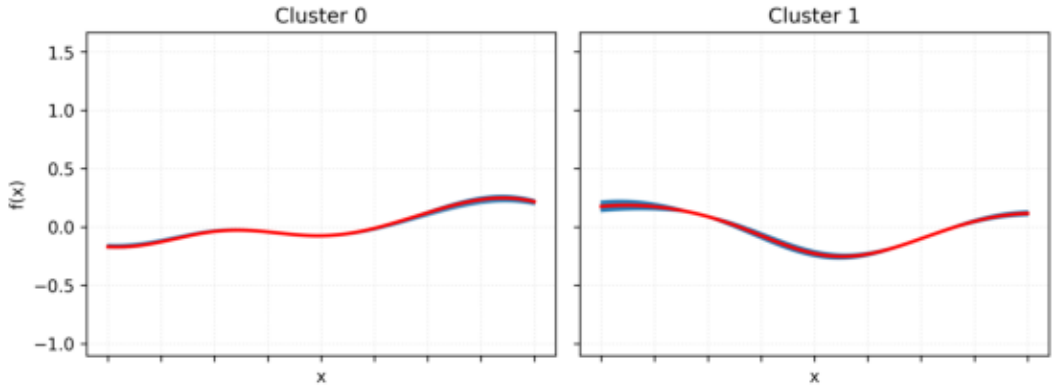
Edge Functions Cluster More Tightly

- With the meta-learner, edge functions in these examples stay closer to their shared centroid.

FastKAN



MetaFastKAN



Blue = edge functions Red = centroid function

Oregon State University
College of Engineering

Image Classification Results

- On CIFAR-100, MetaCluster cuts storage from 3,177.45 KB to 39.75 KB, about **80×**.
- At 39.75 KB, accuracy is **19.16%** with MetaCluster and 11.04% with direct clustering after fine-tuning.

Fully connected KAN on CIFAR-100

Model	Storage (KB)	Accuracy
Uncompressed KAN	3,177.45	17.80%
Direct clustering + fine-tuning	39.75	11.04%
MetaCluster + fine-tuning	**39.75**	**19.16%**

Oregon State University
College of Engineering

Equation Modeling Results

- On 1,000-dimensional equations, MetaCluster cuts storage by **124.1×**.
- After fine-tuning, it has **lower MSE than direct clustering on all three equations**.

$$f_1(\mathbf{x}) = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\sin^2\left(\frac{\pi x_i}{2}\right)\right)$$

$$f_2(\mathbf{x}) = \sum_{i=1}^{n}\left(x_i^2 + x_i^3\right)$$

$$f_3(\mathbf{x}) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}x_i^2\right)$$

Mean squared error by equation (lower is better)

Model	KB	f_1	f_2	f_3
Uncompressed KAN	1,303.6	5.06×10^{-5}	0.270	5.30×10^{-7}
Direct clustering + fine-tuning	10.5	2.24×10^{-4}	0.892	7.82×10^{-6}
MetaCluster + fine-tuning	**10.5**	**2.70×10^{-5}**	**0.504**	**7.55×10^{-7}**

Oregon State University
College of Engineering

Conclusion

- MetaCluster trains KAN edge vectors to be **easier to cluster and share** before compression.
- It reduces model storage by **about 80× on CIFAR-100 and 124.1× for equation modeling**.
- At the same compressed size, it delivers **higher CIFAR-100 accuracy and lower error on all three test functions** than direct clustering with fine-tuning.

Code