FSKA

Flow-based Spectral Kernel Learning for Nonstationary Attention

Learning token similarity with linear aggregation

Zi Yang · Ying Li · Alejandro Lancho
Darío Cabezas Larese · Pablo M. Olmos · Michael Minyi Zhang

The University of Hong Kong · National University of Singapore
Universidad Carlos III de Madrid

NeurIPS 2026

Motivation

Flexible similarities

Fixed or stationary kernels restrict input-dependent token interactions.

Scalable attention

Dense pairwise attention costs quadratically in sequence length L.

Our approach

Learn a nonstationary spectral density and factorize attention with random features.

Kernel expressiveness can simplify attention.

Nonstationary Spectral Features

A bivariate density $p\theta(\omega_1, \omega_2)$ models similarities beyond functions of $x - x'$.

Theorem 1. Unbiased signed features

For a real-valued kernel with a normalized spectral probability density, draw M/2 independent frequency pairs (M even).

$$\phi_\theta(x) = \frac{1}{\sqrt{2M}} \begin{bmatrix} \cos(\Omega_1 x) + \cos(\Omega_2 x) \\ \sin(\Omega_1 x) + \sin(\Omega_2 x) \end{bmatrix} \in \mathbb{R}^M.$$

$$\mathbb{E}_\Omega \left[\phi_\theta(x)^\top \phi_\theta(x') \right] = k_\theta(x, x').$$

The feature inner product is an unbiased estimator of $k\theta(x, x')$.

Flow-based Spectral Density Learning

RealNVP flows map a Gaussian base to paired frequencies and provide a tractable spectral density.

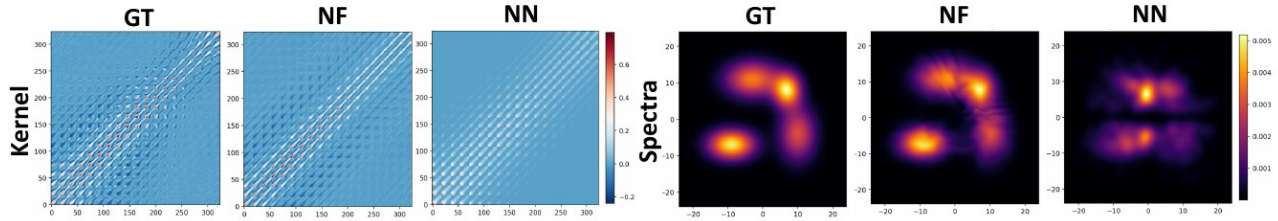
$$z \sim \mathcal{N}(0, I), \qquad (\omega_1, \omega_2) = F_\theta(z),$$

$$p_\theta(\omega_1, \omega_2) = p_0(z) \left| \det \frac{\partial F_\theta(z)}{\partial z} \right|^{-1}.$$

Training combines the task loss with a spectral regularizer.

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{spec}} \mathrm{KL}(p_\theta \| p_{\text{NGSM}}).$$

Learned kernels and spectral densities



GT: ground truth; NF: normalizing flow; NN: implicit neural model. NF recovers more of the target spectral structure and kernel pattern.

Fixed V and Diverse Attention Heads

Q/K controls similarity. V defines the aggregation space.

Raw X values give all heads the same representation space. Fixed orthogonal projections retain complementary value subspaces.

Each head learns its own kernel, which sets its attention weights.

$$\alpha_{ij}^{(h)} = \frac{\widetilde{k}_{\theta_h}(x_i, x_j)}{\sum_{j'} \widetilde{k}_{\theta_h}(x_i, x_{j'})}, \qquad (W_V^{(h)})^\top W_V^{(h')} = 0 \ (h \neq h'),$$

Positive Features, Linear Aggregation

A positive feature map avoids signed cancellation.

$$\psi_h = \mathrm{ELU}(\phi_{\theta_h}) + 1$$

Aggregate feature–value products and feature sums once per head.

$$S_h = \sum_j \psi_h(x_j) v_j^{(h)}, \qquad z_h = \sum_j \psi_h(x_j),$$

$$\widetilde{k}_{\theta_h}(x, x') = \psi_h(x)^\top \psi_h(x'), \qquad y_i^{(h)} = \frac{\psi_h(x_i)^\top S_h}{\psi_h(x_i)^\top z_h}.$$

Aggregation cost grows linearly with sequence length

Theorem 1 applies to signed features before ELU+1. It does not apply to the transformed affinity or normalized attention ratio.

Sentence Classification

Macro-F1 (%) · mean ± SD over three runs

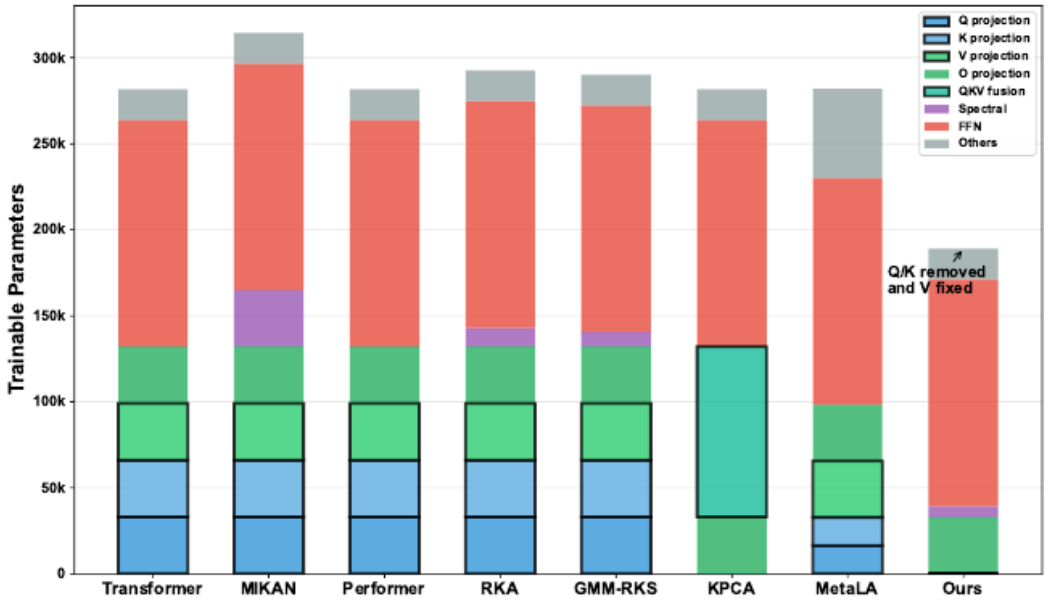
Model	DBPedia	AGNews	TREC	RT-Pol.	SST-2	SST-5
Transformer	98.13 ± 0.15	90.43 ± 0.07	86.14 ± 0.93	70.52 ± 0.45	76.07 ± 1.10	35.11 ± 1.46
MIKAN	97.74 ± 0.01	91.15 ± 0.24	85.47 ± 1.07	71.30 ± 0.92	76.04 ± 0.86	32.84 ± 0.39
Performer	98.12 ± 0.09	90.23 ± 0.31	85.70 ± 0.57	70.88 ± 0.62	77.24 ± 0.41	35.92 ± 0.96
RKA	98.26 ± 0.01	90.80 ± 0.28	85.56 ± 0.75	71.15 ± 0.72	77.08 ± 1.46	36.10 ± 0.49
GMM-RKS	96.76 ± 0.21	90.58 ± 0.41	85.36 ± 0.91	71.05 ± 1.02	76.81 ± 0.89	36.42 ± 0.84
KPCA	97.59 ± 0.14	90.20 ± 0.49	85.90 ± 0.96	72.16 ± 1.58	77.62 ± 1.13	36.97 ± 1.36
MetaLA	97.24 ± 0.15	91.14 ± 0.36	85.07 ± 1.02	70.28 ± 1.71	77.03 ± 0.69	35.27 ± 1.41
FSKA	**98.33 ± 0.13**	**91.20 ± 0.37**	**86.21 ± 1.02**	**72.08 ± 1.44**	**78.74 ± 1.15**	**38.16 ± 1.40**

Highest mean macro-F1 on five of six datasets

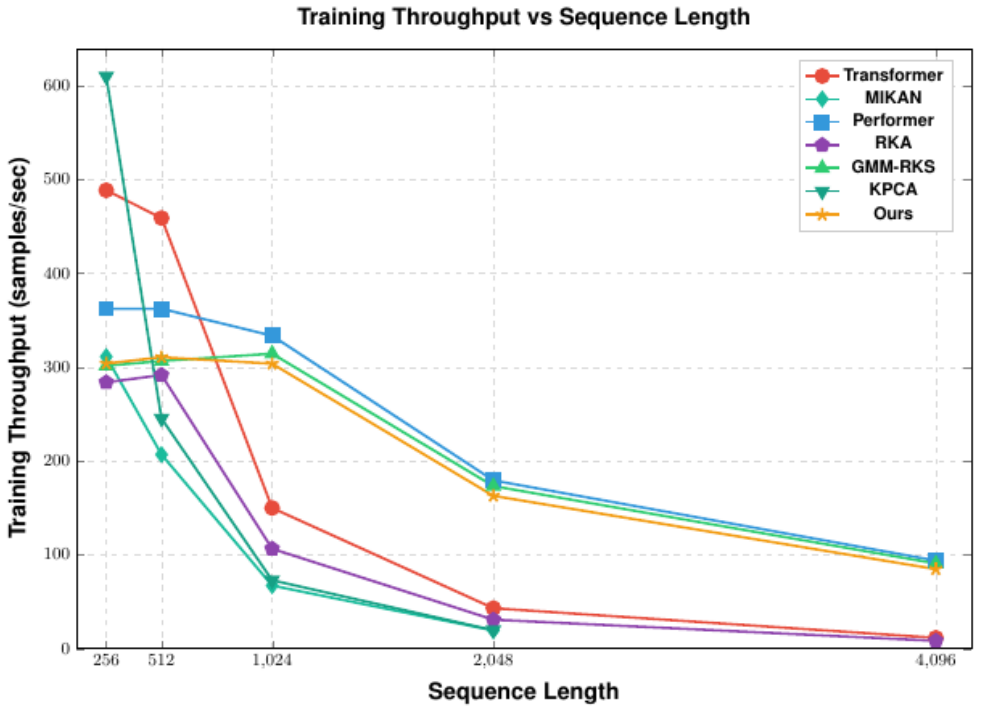
Versus Transformer: SST-2 improves from 76.07 to 78.74, SST-5 from 35.11 to 38.16.

Parameter and Runtime Efficiency

Trainable parameters



Training throughput



Left: original parameter breakdown. Removing learned Q/K and fixing V reduces projection parameters.

Right: higher throughput is faster. Synthetic LRA-style benchmark, lengths 256–4096; successful measurements only.

Removing learned Q/K and fixing V reduces projection parameters.

Conclusion

Richer token similarities with a simpler projection design

Learned spectral density and head-specific kernels
Fixed value projections and linear aggregation
Highest mean macro-F1 on five of six datasets

Project page: https://dariocb.github.io/fska-project-page/

Paper: openreview.net/forum?id=aP71hsUN95
Code: github.com/ziyang18/FSKA

Correspondence
pamartin@ing.uc3m.es
mzhang18@hku.hk