# DEEP KERNEL LEARNING NETWORKS WITH MULTIPLE LEARNING PATHS

*Ping Xu* *Yue Wang* *Xiang Chen* *Zhi Tian*

Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 22030, USA

## ABSTRACT

This paper proposes deep kernel learning networks with multiple learning paths (DKL-MLP) for nonlinear function approximation. Leveraging the random feature (RF) mapping technique, kernel methods can be implemented as a two-layer neural network, at drastically reduced workload on weight training. Motivated by the representation power of the deep architecture in deep neural networks, we devise a vanilla deep kernel learning network (DKL) by applying RF mapping at each layer and learn the last layer only. To improve the learning performance of DKL, we add multiple trainable paths to DKL and develop the DKL-MLP method so that some implicit information from earlier hidden layers to the output layer can be learned. We prove that both DKL and DKL-MLP permit universal representation of a wide variety of interesting functions with arbitrarily small error and have no bad local minimum. Numerical experiments on both regression and classification tasks are provided to demonstrate the learning performance and computational efficiency of the proposed methods.

***Index Terms*—** Kernel methods, random feature mapping, deep kernel networks, multiple learning paths.

## 1. INTRODUCTION

Kernel methods and neural networks (NNs) are both attractive in various learning tasks such as regression, classification, as well as reinforcement learning [1–3], thanks to their abilities to model the (complex) nonlinear function $f$ in the learning tasks. With different representations of $f$, kernel methods and NNs enjoy (suffer) different strengths (weaknesses).

Kernel methods approximate $f$ as a linear combination of *pre-selected* kernels (nonlinear basis functions), and learn $f$ by optimizing the combination coefficients according to certain metrics. The learning process is usually done through convex optimization, hence kernel methods are amenable to theoretical analysis, permit globally optimal parameters, and enjoy rigorous statistical learning guarantees [4]. However, the curse of dimensionality issue prevents their applications in large-scale learning scenarios. On the other hand, an NN approximates $f$ as a composite of functions of concatenated layers, where each layer consists of a linear operation and a nonlinear activation function [5]. The representation power of NNs can be adjusted by the depth and width of their architecture, as well as the activation functions. However, optimizing NN models involves solving nonconvex functions; as a result, they rely much on intuition, heuristics, and trial-and-error, and our theoretical understanding of them is still incomplete [6].

Motivated by these observations, we propose a method that possesses strengths of both methods and overcomes their weaknesses. Specifically, we adopt the random feature (RF) mapping [7] method to circumvent the curse of dimensionality issue in kernel methods and implement kernel learning using a two-layer NN (termed RF-KL). Then, we extend RF-KL to deep kernel learning (DKL) by performing RF mapping at each layer and train the last output layer only. To increase the representation power of DKL, we add multiple trainable paths to connect each hidden layer (except for the last hidden layer) with the output layer. It leads to a deep kernel learning network structure with multiple learning paths (DKL-MLP), whose output functional is linear in the trainable multi-path model parameters. In this way, DKL-MLP benefits not only from the depth of DKL, but also from the (implicit) flexibility and computational advantage of RF-based multi-kernel learning.

**Related work:** To mitigate the computational complexity of kernel methods in deep structures, RF mapping is usually adopted. For example, [8] utilizes RF mapping to address the scalability issue in phone recognition and speech understanding tasks by stacking the RF-mapped kernel modules to form a deep architecture. [9] proposes a deep hybrid NN structure where a trainable layer and a fixed RF mapping layer are concatenated. In [10], the random Fourier feature layer is initialized from a Gaussian distribution and trained end-to-end through back-propagation, thus the final optimized RF related parameters may not follow Gaussian distribution any more. A deep semi-random network is proposed in [11], where each layer consists of both trainable and fixed parameters, which differs from [9]. More recently, generative models and their extension to multi-layer structures are employed to jointly solve the learning task and learn the random Fourier features [12]. Note that among all recent works, the deep semi-random kernel method shares some similarity with our proposed method. However, compare with our method, [11] requires much more computational resources for their trainable weights to get updated due to back-propagation through all layers.

**Our contributions:** Relative to prior art, our work has

This work was supported in part by the US NSF grants #1939553, #1741338, and #2003211.

three contributions. Firstly, we leverage the RF mapping method to implement DKL as a randomized deep neural network with only last layer trainable. Compared with extreme learning machine [13], the randomized parameters are generated from specific distributions related with pre-selected kernels. Statistical accuracy guarantees of standard kernel methods are applicable to RF-based kernel methods. Secondly, we add multiple trainable paths to DKL to increase its expressiveness and develop the DKL-MLP method. These paths directly connect the hidden layers with the output layer in a linear manner and do not change the convexity of the learning problem if the original DKL is convex. Moreover, updating all trainable parameters in DKL-MLP does not involve back-propagation, thus gradient diminishing problem is avoided and the computational complexity is much reduced compared with [11]. Finally, we provide theoretical analysis in terms of universality to show that the developed algorithm can represent a wide variety of interesting functions with arbitrarily small error and have no bad local minimum, which is lacking in most of the current deep kernel learning work. In addition, we test the performance of our proposed algorithm on real datasets to solve both the classification and regression tasks. The results corroborate that DKL-MLP enjoys both good generalization performance and low computational complexity.

**Notations.** $\mathbb{R}$ denotes the set of real numbers. $\|\cdot\|_2$ denotes the Euclidean norm of vectors. $\mathbf{A}$, $\mathbf{a}$, and $a$ represent a matrix, vector and scalar, respectively.

## 2. PROBLEM STATEMENT AND PRELIMINARIES

This section states the learning problem, reviews standard kernel methods and introduces the RF-based kernel method (RF-KL). We show that RF-KL can be implemented as a two-layer NN, which leads to the development of the deep kernel learning (DKL) framework and the deep kernel learning with multiple learning path (DKL-MLP) method.

### 2.1. Problem statement

The learning task is to find a nonlinear function $f \in \Omega$ such that $y_t = f(\mathbf{x}_t) + e_t$ for the data sample set $\{\mathbf{x}_t, y_t\}_{t=1}^T$, with $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \mathbb{R}$, and $e_t$ being an error term. The optimal $f$ is obtained by minimizing the following total cost function:

$$\min_{f\in\Omega} \quad \hat{R}(f) := \tfrac{1}{T}\textstyle\sum_{t=1}^T \ell(f(\mathbf{x}_t), y_t) + \lambda\|f\|_\Omega^2, \qquad (1)$$

where $\ell(\cdot,\cdot)$ is a loss function, $\|\cdot\|_\Omega$ is the norm associated with the function space $\Omega$, and $\lambda > 0$ is a regularization parameter that controls over-fitting. Depending on different tasks, $\ell(\cdot,\cdot)$ can be selected as least-squares in regression tasks or the logistic or the hinge loss in classification tasks.

### 2.2. RF-KL implemented using a two-layer NN

Assume $f \in \mathcal{H}$, where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) $\mathcal{H} := \{f | f(\mathbf{x}) = \sum_{t=1}^\infty \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t)\}$ induced by a shift-invariant positive semidefinite kernel $\kappa(\mathbf{x}, \mathbf{x}_t) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Then, by the Representer theorem [4], the optimal solution of (1) admits

$$\hat{f}_\kappa(\mathbf{x}) = \textstyle\sum_{t=1}^T \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t) := \boldsymbol{\alpha}^\top \boldsymbol{\kappa}(\mathbf{x}), \qquad (2)$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_T]^\top \in \mathbb{R}^T$ is the coefficient vector to be learned, and $\boldsymbol{\kappa}(\mathbf{x}) = [\kappa(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}, \mathbf{x}_T)]^\top$. However, both evaluating $\boldsymbol{\alpha}$ and updating it from a new sample requires large computation, thus incurs the curse of dimensionality when the data size is very large.

To make kernel methods scalable for large datasets, RF mapping [7] is adopted, which approximates the kernel function $\kappa(\mathbf{x}, \mathbf{x}_t)$ in (2) by the sample average

$$\hat{\kappa}_M(\mathbf{x}, \mathbf{x}_t) := \tfrac{1}{M}\textstyle\sum_{m=1}^M \phi(\mathbf{x}; \boldsymbol{\omega}_m, b_m)\phi(\mathbf{x}_t, ; \boldsymbol{\omega}_m, b_m), \quad (3)$$

where $\phi(\mathbf{x}; \boldsymbol{\omega}, b) = \sqrt{2}\cos(\boldsymbol{\omega}^\top \mathbf{x} + b)$. Here $\{\boldsymbol{\omega}_m\}_{m=1}^M$ are randomly drawn from $p_\kappa(\boldsymbol{\omega})$, which is the inverse Fourier transform of $\kappa$, and $\{b_m\}_{m=1}^M$ are drawn uniformly from $[0, 2\pi]$. For a Gaussian kernel $\kappa(\mathbf{x}_t, \mathbf{x}_\tau) = \exp(-\|\mathbf{x}_\mathrm{t} - \mathbf{x}_\tau\|_2^2/(2\sigma^2))$, we have $p_\kappa(\boldsymbol{\omega}) \sim \mathcal{N}(\mathbf{0}, \sigma^{-2}\mathbf{I})$.

Then, the function $\hat{f}_\kappa$ in (2) can be expressed as

$$\hat{f}_{\mathrm{RF}}(\mathbf{x}) = \textstyle\sum_{t=1}^T \alpha_t \boldsymbol{\phi}_M^\top(\mathbf{x}_t)\boldsymbol{\phi}_M(\mathbf{x}) := \boldsymbol{\theta}^\top \boldsymbol{\phi}_M(\mathbf{x}), \qquad (4)$$

where $\boldsymbol{\phi}_M(\mathbf{x}) = \sqrt{\frac{1}{M}}[\phi(\mathbf{x}; \boldsymbol{\omega}_1, b_1), \ldots, \phi(\mathbf{x}; \boldsymbol{\omega}_M, b_M)]^\top$ and $\boldsymbol{\theta}^\top := \sum_{t=1}^T \alpha_t \boldsymbol{\phi}_M^\top(\mathbf{x}_t)$ denotes the new decision vector to be learned in the RF space. Note that the size of $\boldsymbol{\theta}$ is fixed and does not vary with the number of data samples.

The optimization problem (1) can then be reformulated as

$$\min_{\boldsymbol{\theta}} \quad \hat{R}(\boldsymbol{\theta}) := \tfrac{1}{T}\textstyle\sum_{t=1}^T \ell(\boldsymbol{\theta}^\top \boldsymbol{\phi}_M(\mathbf{x}_t), y_t) + \lambda\|\boldsymbol{\theta}\|_2^2. \quad (5)$$

Depending on the loss function, there may exist a closed form solution for $\boldsymbol{\theta}$ [14]. Otherwise, the model parameters can be updated through gradient descent.

**Remark 1 (Equivalence to a shallow NN).** Alternatively, we can rewrite (4) as

$$\hat{f}_M(\mathbf{x}) := \hat{f}_{\mathrm{RF}}(\mathbf{x}) = \textstyle\sum_{m=1}^M \theta_m \hat{\phi}(\boldsymbol{\omega}_m^\top \mathbf{x} + b_m), \qquad (6)$$

where $\hat{\phi}(\boldsymbol{\omega}_m^\top \mathbf{x} + b_m) := \sqrt{\frac{2}{M}}\cos(\boldsymbol{\omega}_m^\top \mathbf{x} + b_m)$. Notice that with RF mapping, a kernel method that models $f$ as (6) plays as a special case of a two-layer NN, where the random features $\{\boldsymbol{\omega}\}_{m=1}^M$ and the parameters $\boldsymbol{\theta}$ to be learned are equivalent to the weights in the first layer and the weights in the second layer of a standard NN, respectively, whereas the activation function of the first layer is cosine.

**Remark 2 (Complexity and representation power).** Note that in RF-KL, the first layer's weights are fixed and do not need to be trained, only the weights in the second/output layer need to trained if there does not exist a closed form solution. Therefore, compared to the standard NN where weights from

all layers need to be trained, RF-KL can afford to have a larger network size given the same training overload and computation complexity. Moreover, RF-KL does not need back-propagation to train the weights since they are in the output layer, hence bypassing the diminishing gradient issue. The simplicity of RF-KL coming from the fixed weights and fixed activation function of the first layer is a double-sword. In the extreme case where only one kernel is applied in a NN with one hidden layer, RF-KL has limited representation power and results in degraded learning performance compared to the standard NN.

## 3. DEEP KERNEL LEARNING NETWORKS WITH MULTIPLE LEARNING PATHS

To balance the representation power and computational complexity, we devise deep kernel learning networks with multiple learning paths. With RF mapping, a vanilla deep kernel learning (DKL) network of $N$ layers is designed and modeled by

$$\hat{f}_{M_N}(\mathbf{x}) = \boldsymbol{\vartheta}^\top \boldsymbol{\phi}^{N-1}(\cdots(\boldsymbol{\phi}^1(\mathbf{x}))), \tag{7}$$

where $\boldsymbol{\phi}^1(\mathbf{x}) = \sqrt{\frac{1}{M_1}}[\phi(\mathbf{x};\boldsymbol{\omega}_1^1, b_1^1), \ldots, \phi(\mathbf{x};\boldsymbol{\omega}_{M_1}^1, b_{M_1}^1)]^\top$, $\boldsymbol{\phi}^2(\boldsymbol{\phi}^1(\mathbf{x})) = \sqrt{\frac{1}{M_2}}[\cdots, \phi(\boldsymbol{\phi}^1(\mathbf{x});\boldsymbol{\omega}_m^2, b_m^2), \cdots]^\top$, until the last hidden layer, and $\boldsymbol{\vartheta} \in \mathbb{R}^{M_N}$ are trainable parameters that connect the last hidden layer and the output layer. Each hidden layer has $M_n$ neurons that does not have to be the same across all layers.

**Theorem 1** *(Universal approximation of DKL.) Let $\mathcal{C} \neq \{0\}$ be any fixed nonempty subset of $\mathbb{R}^d$, then, for any $f \in L^2(\mathcal{C})$ and $\epsilon > 0$, there exists a positive integer $M_N$ such that with probability greater than $1-\delta$, we can find coefficients $\boldsymbol{\vartheta} \in \mathbb{R}^{M_N}$ that satisfy*

$$\|\hat{f}_{M_N} - f\|_{L^2(\mathcal{C})} \leq \epsilon. \tag{8}$$

*Proof.* Ref. [15] has proved that the composed kernel of two positive definite kernels is also positive definite. Therefore, by mathematic induction the composed kernel $\kappa_{N-1}$ of $N-1$ layers is also positive definite if the deep kernel model is equipped with positive definite kernels across all layers. Specifically, for Gaussian kernels adopted in this paper, the composed kernel $\kappa_{N-1}$ is also a Gaussian kernel and its corresponding RKHS $\mathcal{H}_{N-1}$ is dense, which indicates that $\mathcal{H}_{N-1}$ is universal [16]. Then, for $f \in \mathcal{H}_{N-1}$ and by the mapping equivalence between the Gaussian kernel and its random features, we conclude that $f$ can be approximated by $\hat{f}_{M_N}$ with high probability by random choices of $\boldsymbol{\omega}_m^n$ from the distribution $p_{\kappa_m}(\boldsymbol{\omega})$, $\forall m, n$. ■

Compared with a standard DNN of same network structure, DKL enjoys drastically reduced workload on weight training since $\boldsymbol{\omega}_m^n$ and $b_m^n$, $\forall m, n$, are fixed once the kernel functions of all layers are fixed, and it only needs to train the last layer's parameters $\boldsymbol{\vartheta}$. However, the simplicity coming from the fixed weights is a double-sword, and DKL may not have enough representation power for complex learning tasks.

To further enhance the learning ability of DKL, we add multiple paths to connect the output of each hidden layer with the output layer and form the deep kernel learning with multiple learning path (DKL-MLP) model. Compared with DKL, DKL-MLP has more trainable weights resulting from the multiple paths. The network structure is shown in Figure 1 and can be modeled as

$$\tilde{f}_{M_N}(\mathbf{x}) = \hat{f}_{M_{N-1}} + \hat{f}_{M_{N-2}} + \cdots + \hat{f}_{M_1} = \textstyle\sum_{n=1}^{N-1} \hat{f}_{M_n}, \tag{9}$$

where $\hat{f}_{M_n} := \boldsymbol{\vartheta}^{n\top}\boldsymbol{\phi}^n(\cdots(\boldsymbol{\phi}^1(\mathbf{x}))$ with learnable parameters $\boldsymbol{\vartheta}^n \in \mathbb{R}^{M_n}$ associated with each hidden layer.

The following theorem shows the universality of (9).

**Theorem 2** *(Universal approximation of DKL-MLP.) Let $\mathcal{C} \neq \{0\}$ be any fixed nonempty subset of $\mathbb{R}^d$, then, for any $f \in L^2(\mathcal{C})$ and $\epsilon > 0$, there exists a positive integer $M_\Sigma = M_1 + \cdots + M_{N-1}$ such that with probability greater than $1-\delta$, we can find coefficients $\boldsymbol{\Theta} \in \mathbb{R}^{M_\Sigma}$ that satisfy*

$$\|\tilde{f}_{M_N} - f\|_{L^2(\mathcal{C})} \leq \epsilon, \tag{10}$$

*where $\boldsymbol{\Theta} = [\boldsymbol{\vartheta}^{N-1}; \cdots; \boldsymbol{\vartheta}^2; \boldsymbol{\vartheta}^1] \in \mathbb{R}^{M_\Sigma}$.*

*Proof.* To prove that (9) admits universality, we can view it as a multi-kernel learning problem. Theorem 5.7 in [15] states that the space induced by the summation of two reproducing kernels is also a RKHS. Hence, by mathematical induction, the summation of $N-1$ reproducing kernels also induces a RKHS. For our model where all kernels are Gaussian, the composed kernel is also Gaussian, and its corresponding RKHS is dense. Thus, by the mapping equivalence between the Gaussian kernel and its random Fourier features, we conclude that $f$ can be approximated by $\tilde{f}_{M_N}$ with high probability by random choices of $\boldsymbol{\omega}_m^n$ that follows the distribution $p_{\kappa_m}(\boldsymbol{\omega})$, $\forall m, n$. ■

**Remark 3 (No bad local minimum).** With $\boldsymbol{\Theta}$ defined in Theorem 2, we can rewrite (9) as

$$\tilde{f}_{\boldsymbol{\Theta}}(\mathbf{x}) = \boldsymbol{\Theta}^\top \boldsymbol{\Phi}(\mathbf{x}), \tag{11}$$

where $\boldsymbol{\Phi}(\mathbf{x}) = [\boldsymbol{\phi}^{N-1}(\cdots(\boldsymbol{\phi}^1(\mathbf{x})));\cdots;\boldsymbol{\phi}^2(\boldsymbol{\phi}^1(\mathbf{x}));\boldsymbol{\phi}^1(\mathbf{x})]$. Then, (1) becomes

$$\min_{\boldsymbol{\Theta}} \quad \hat{R}(\boldsymbol{\Theta}) := \tfrac{1}{T}\textstyle\sum_{t=1}^T \ell(\tilde{f}_{\boldsymbol{\Theta}}(\mathbf{x}_t), y_t) + \lambda\|\boldsymbol{\Theta}\|_2^2. \tag{12}$$

It is clear that the globally optimal parameters can be found via convex optimization. Therefore, our model has no bad local minimum by Theorem 2.

**Remark 4 (Difference between DKL-MLP and deep residual networks).** The multiple-learning-path (MLP) model differs from the deep residual network [17] in that short cuts in a deep residual network skip learning for some layers to avoid vanishing gradient problem. In contrast, we employ $\boldsymbol{\vartheta}^1, \ldots, \boldsymbol{\vartheta}^{N-2}$ as trainable parameter vectors to enhance the learning performance.

**Remark 5 (Difference between DKL-MLP and deep semi-random networks).** It should be noted that our work differs

from the deep semi-random network proposed in [11], which formulates a nonconvex learning problem for the parameter estimation. Moreover, updating all trainable parameters in DKL-MLP does not involve back-propagation. On the other hand, [11] requires much more computational resources for their trainable weights to get updated due to back-propagation through all layer.

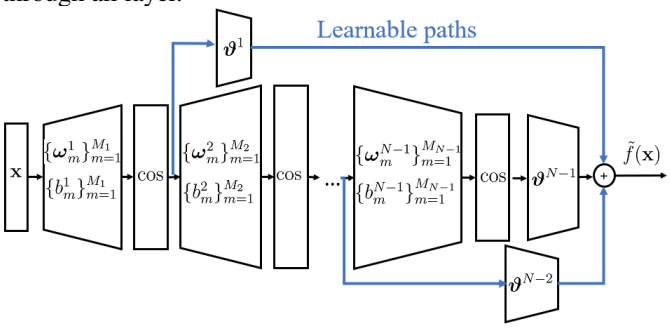


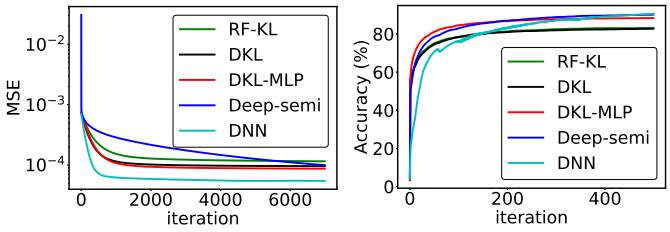

**Fig. 1**. A general structure of DKL-MLP with $N$ layers.

## 4. NUMERICAL TESTS

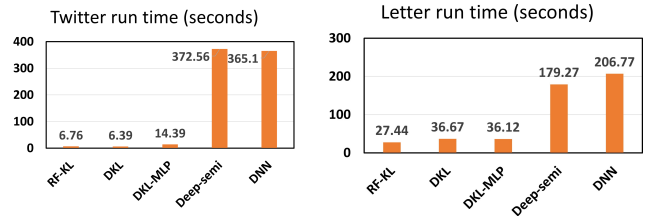
In this section, we conduct experiments to compare DKL-MLP with several benchmarks using real datasets from UCI repository [18] for regression and classification tasks. For all tests, the regularization parameter is set to be $\lambda = 10^{-4}$ and we use 75% of the data for training and the remaining data for testing for all datasets. **RF-KL** and **DKL** are described by (4) and (7), respectively. **Deep-semi** is the work proposed in [11] and **DNN** is the vanilla deep neural network that all parameters need to be trained. All kernel-based methods use Gaussian kernels with their best kernel bandwidth optimized through grid search for each dataset for each algorithm, whereas DNN employs Relu functions for all hidden layers. Depending on the tasks, the output layer equips with a sigmoid function, a softmax function, or directly outputs the results. All methods are trained using gradient descent with their learning rate optimized through grid search for each dataset for each method.

We have conducted extensive simulations on six datasets, 2 (Twitter and Toms Hardware) for regression tasks and 4 (Adult, Sensor, Human activity, and Letter) for classification tasks. Due to the page limit, we select the results for one regression (Twitter) and one classification (Letter) problems, considering the simulations results on the other datasets present the similar trend as shown in Fig. 2. Twitter dataset consists of $T = 13800$ samples with $\mathbf{x}_t \in \mathbb{R}^{77}$ and $y_t \in \mathbb{R}$ representing the average number of active discussions on a certain topic. The learning task is to predict the popularity of these topics. Letter dataset consists of $T = 20000$ samples with $\mathbf{x}_t \in \mathbb{R}^{16}$ and $y_t$ represents letters from A to Z. The learning task is to identify the capital letters. We set the number of layers to accommodate the complexity of learning tasks. In our simulations, since the regression and classification problems are not too complicate with mild-size datasets, we apply 2 hidden layers which is sufficient for good learning performance.

RF-KL has 50 neurons for Twitter dataset and 200 for Letter dataset. All other deep networks have 50 neurons of each hidden layer for Twitter dataset and 200 neurons of each hidden layer for Letter dataset. Fig. 2(a) and 2(b) show that eventually DNN and Deep-semi methods perform better than the other methods, which makes sense. Take Twitter dataset as an example, Deep-semi has 6400 parameters and DNN has 6501 trainable parameters, while DKL-MLP only has 100 trainable parameters. Also, DKL-MLP performs better than DKL and RF-KL since the latter two only have 50 trainable parameters, respectively. The running time of all algorithms on the two datasets is presented in Fig. 2(c) and 2(d), and suggests that RF-based kernel methods enjoy low complexity. The experimental results corroborate that our proposed DKL-MLP achieves a good trade-off of learning performance and computational complexity compared with the benchmark methods. Note that if the training resource is limited, DNN and Deep-semi methods may not be able to train a large model and their representation power would degrade. On the other hand, if the training time is limited, which means DNN and Deep-semi methods may not have enough time to converge, their learning performance will also be affected.




(a) MSE on Twitter dataset. (b) Accuracy on Letter dataset.




(c) Run time on Twitter dataset. (d) Run time on Letter dataset.

**Fig. 2**. Performance comparison.

## 5. CONCLUSION

In this paper, we have developed a deep kernel learning network with multiple learning paths. Leveraging the theory from traditional kernel methods, we have proved that our proposed deep kernel learning networks admit universality. The learnability of our method can be improved by the trainable paths and the computational complexity is greatly reduced via random feature mapping. It should be noted that the multiple-learning-path scheme can also be applied to other deep networks such as deep extreme learning machines and deep neural networks. Future efforts will be devoted to analyzing the generalization bound of the proposed method.

## 6. REFERENCES


[1] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola, "Kernel methods in machine learning," *The Annals of Statistics*, pp. 1171–1220, 2008.

[2] Xin Xu, Dewen Hu, and Xicheng Lu, "Kernel-based least squares policy iteration for reinforcement learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 973–992, 2007.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[4] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 416–426.

[5] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein, "Training neural networks without gradients: A scalable ADMM approach," in *International Conference on Machine Learning*, 2016, pp. 2722–2731.

[6] Gilad Yehudai and Ohad Shamir, "On the power and limitations of random features for understanding neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 6598–6608.

[7] Ali Rahimi and Benjamin Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, 2008, pp. 1177–1184.

[8] Po-Sen Huang, Li Deng, Mark Hasegawa-Johnson, and Xiaodong He, "Random features for kernel deep convex network," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 3143–3147.

[9] Siamak Mehrkanoon and Johan AK Suykens, "Deep hybrid neural-kernel networks using random Fourier features," *Neurocomputing*, vol. 298, pp. 46–54, 2018.

[10] Jiaxuan Xie, Fanghui Liu, Kaijie Wang, and Xiaolin Huang, "Deep kernel learning via random Fourier features," *arXiv preprint arXiv:1910.02660*, 2019.

[11] Kenji Kawaguchi, Bo Xie, and Le Song, "Deep semi-random features for nonlinear function approximation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32.

[12] Kun Fang, Xiaolin Huang, Fanghui Liu, and Jie Yang, "End-to-end kernel learning via generative random Fourier features," *arXiv preprint arXiv:2009.04614*, 2020.

[13] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[14] Ping Xu, Yue Wang, Xiang Chen, and Zhi Tian, "COKE: Communication-censored decentralized kernel learning," *Journal of Machine Learning Research*, vol. 22, no. 196, pp. 1–35, 2021.

[15] Vern I Paulsen and Mrinal Raghupathi, *An introduction to the theory of reproducing kernel Hilbert spaces*, vol. 152, Cambridge university press, 2016.

[16] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang, "Universal kernels.," *Journal of Machine Learning Research*, vol. 7, no. 12, 2006.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] Arthur Asuncion and David Newman, "UCI machine learning repository," 2007.