Demystifying the Random Feature-Based Online Multi-Kernel Learning

Songnam Hong, *Member, IEEE* and Jeongmin Chae *Student Member, IEEE*

Abstract—The random feature-based online multi-kernel learning (RF-OMKL) is a promising framework in functional learning tasks. This framework is necessary for an online learning with continuous streaming data due to its low-complexity and scalability. In RF-OMKL framework, numerous algorithms can be presented according to an underlying online learning and optimization techniques. The best known algorithm (termed Raker) has been proposed with the lens of the famous online learning with expert advice, where each kernel from a kernel dictionary is viewed as an expert. Harnessing this relation, it was proved that Raker yields a sublinear *expert* regret bound, in which as the name implies, the best function is further restricted as the expert-based framework. Namely, it is not an actual sublinear regret bound under RF-OMKL framework. In this paper, we propose a novel algorithm (named BestOMKL) for RF-OMKL framework and prove that it achieves a sublinear regret bound under a certain condition. Beyond our theoretical contribution, we demonstrate the superiority of our algorithm via numerical tests with real datasets. Notably, BestOMKL outperforms the state-of-the-art kernel-based algorithms (including Raker) on various online learning tasks, while having a lower complexity as Raker. These suggest the practicality of BestOMKL.

Index Terms—Functional learning, Multiple kernel learning, online learning, reproducing kernel Hilbert space (RKHS).

I. INTRODUCTION

In Internet-of-Things (IoT) systems, a massive number of machine-type (or mobile) devices can be used to monitor and analyze various cyber-physical systems such as smart city, connected cars, smart factory, intelligent energy management, and so on [1], [2]. Machine learning plays a key role in accomplishing such sophisticated tasks. In particular, nonlinear function learning plays a crucial role in various machine learning tasks such as classification, regression, clustering, dimensionality reduction [3], [4], [5], [6]. Supervised functional learning tasks, which are closely related to the subject of this paper, are formally defined as follows. Given data samples $\{(\mathbf{x}_t, y_t) : t = 1, \dots, T\}$ with features $\mathbf{x}_t \in \mathbb{R}^d$ and labels $y_t \in \mathbb{R}$, the objective is to learn a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that minimizes the cumulative loss $\sum_{t=1}^T \mathcal{L}(f(\mathbf{x}_t), y_t)$, where $\mathcal{L}(\cdot, \cdot)$ denotes a loss (or cost) function. Such optimization over an arbitrary function space is quite challenging. In a kernel-based function learning, the function space is restricted as a well-structured reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by a kernel κ [3]. Naturally the accuracy of a kernel-based learning relies on a preselected kernel (i.e., the “shape”

of RKHS). However, it is generally demanding to find a proper single kernel.

Multiple kernel learning (MKL), using a preselected set of kernels (called a kernel dictionary), is more powerful as it can produce a larger function space compared with a single kernel while maintaining the good properties of RKHS [7], [8], [9]. As shown in [7], given a preselected set of n kernels $\{\kappa_i : i = 1, \dots, n\}$, we can generate a large RKHS $\mathcal{H}_{\mathbf{q}}$ induced by a kernel $\bar{\kappa}_{\mathbf{q}} \in \mathcal{K}$, where

$$\mathcal{K} = \{\bar{\kappa}_{\mathbf{q}} = q_1 \kappa_1 + \dots + q_n \kappa_n : \mathbf{q} = [q_1, \dots, q_n]^T \in \Delta_n\}, \quad (1)$$

and $\Delta_n \triangleq \{\mathbf{q} \in \mathbb{R}_+^n, \sum_{i=1}^n q_i = 1\}$ denotes the $(n-1)$ -simplex (a.k.a. the probability simplex). Here, the combination weight $\mathbf{q}$ can be also optimized according to datasets. Hence, the set of all possible functions generated by the given n kernels is defined as

$$\bar{\mathcal{H}} = \left\{ f : \sum_{i=1}^n q_i f_i : f_i \in \mathcal{H}_i, i = 1, \dots, n, \mathbf{q} \in \Delta_n \right\}. \quad (2)$$

Since $\bar{\mathcal{H}}$ can be much larger function space than a RKHS $\mathcal{H}$ induced by a single kernel κ , MKL potentially finds a more accurate function than single kernel-based learning.

In many real-world applications, functional learning tasks are expected to be performed in an *online* fashion. Particularly, online learning is necessary when data arrive sequentially such as online spam detection and time series prediction [10], [11], [12]. For such cases, online MKL (OMKL) has been proposed, which seeks the optimal combination of a set of single kernel functions sequentially. However, OMKL suffers from a high computational complexity as the dimension of the optimization variables increases linearly with the time indices (i.e., the number of incoming data T). Namely, in online learning with continuous incoming data, this method is not applicable. This problem is known as the curse of dimensionality. In [13], this problem has been overcome successfully by proposing the random feature (RF) approximation for kernels. Throughout the paper, this RF-based OMKL framework is referred to as RF-OMKL framework, which is the main subject of this paper.

In RF-OMKL framework, numerous algorithms can be constructed according to the choices of underlying online learning and optimization techniques. In [14], Raker has been presented by viewing multi-kernel learning as the famous online learning with expert advice (in short, expert-based learning), where each kernel from a kernel dictionary is considered as an expert. This shows the best performance under RF-OMKL framework as well as the other types of kernel-based algorithms [14]. Very recently, AMKL has been proposed in [15] by introducing an

S. Hong is with the Department of Electronic Engineering, Hanyang University, Seoul, 04763, Korea (e-mail: snhong@hanyang.ac.kr)

J. Chae is with the Department of Electrical Engineering, University of Southern California, CA, 90089, USA (e-mail: chaej@usc.edu)

active learning strategy suitable for Raker. Also, it was shown in [15] that AMKL yields the same learning accuracy as Raker while using little labeled data. In both Raker and AMKL, the optimization techniques for n single kernel functions and combination weights are equivalent and closely related to the expert-based learning framework [16], [17], [18]. Especially, the optimization for combination weights is based on *Exp strategy*, which has been used to find the best expert in the context of expert-based online learning [16]. The Exp strategy can produce the best kernel among the preselected n kernels according to incoming data samples, i.e., the best single kernel (denoted by $\mathcal{H}_{p^*}, p^* \in \{1, \dots, n\}$) is determined as the part of an online learning. Thus, Raker and AMKL show much better performances than single kernel-based algorithms [14], [15].

Our contributions: Focusing on RF-OMKL framework, we propose an efficient algorithm with an analytical performance guarantee. We first show that the expert-based learning algorithms such as Raker and AMKL should not be optimal for RF-OMKL framework. In [14], [15], the following cumulative regret is considered:

$$\text{regret}_T^E \triangleq \sum_{t=1}^T \mathcal{L}(\hat{f}_t(\mathbf{x}_t), y_t) - \min_{f \in \mathcal{S}} \sum_{t=1}^T \mathcal{L}(f(\mathbf{x}_t), y_t), \quad (3)$$

where $\{\hat{f}_t, t = 1, \dots, T\}$ denotes a sequence of learned functions and $\mathcal{S} \triangleq \{f_i \in \mathcal{H}_i : i = 1, \dots, n\} \subset \bar{\mathcal{H}}$. We remark that the best function is chosen from a single best kernel rather than our function space $\bar{\mathcal{H}}$ in (2). In the context of online learning [16], the upper bound on regret_T^E is called *expert regret bound* as the best function in $\mathcal{S}$ should belong to the best single kernel (i.e., the best expert). In general, this function may not be the best with respect to the original function space $\mathcal{H}$ as $\mathcal{S} \subset \bar{\mathcal{H}}$. Definitely, the cumulative regret in (3) is not precise for RF-OMKL framework as a sequence of learned functions in this framework is optimized from $\bar{\mathcal{H}}$. For this reason, we in this paper consider the following cumulative regret for RF-OMKL framework:

$$\text{regret}_T \triangleq \sum_{t=1}^T \mathcal{L}(\hat{f}_t(\mathbf{x}_t), y_t) - \min_{f \in \mathcal{H}} \sum_{t=1}^T \mathcal{L}(f(\mathbf{x}_t), y_t). \quad (4)$$

The upper bound on regret_T is referred to as *regret bound*. Throughout the paper, the expert regret bound and regret bound follow the definitions in (3) and (4), respectively. In [14], [15], it was only proved that Raker and AMKL achieve a sublinear expert regret bound, i.e., $\text{regret}_T^E \leq \mathcal{O}(\sqrt{T})$. Definitely, it does not imply that they can achieve a sublinear regret bound. Motivated by this, we present a novel algorithm (named BestOMKL) for RF-OMKL framework and prove that it can achieve a sublinear regret bound under a certain condition, i.e., $\text{regret}_T \leq \mathcal{O}(\sqrt{T})$. This analysis reveals that when T is sufficiently large, the proposed BestOMKL asymptotically performs almost as well as the best function from the function space $\mathcal{H}$. Namely, in an asymptotic case, our algorithm is optimal for RF-OMKL framework, whereas Raker and AMKL are not. Via numerical tests with real datasets, we demonstrate that BestOMKL outperforms the existing kernel-based algorithms, while having a lower complexity as Raker. These suggest the practicality of BestOMKL.

The remaining part of this paper is organized as follows. In Section II, we provide useful backgrounds on kernel-based learning and define RF-OMKL framework. In Section III, we identify the limitation of the present algorithm for this framework and pose an open problem. In Section IV, the proposed BestOMKL is described and its asymptotic optimality under a certain condition is proved. Numerical tests with real datasets are performed in Section V, showing the superiority of our algorithm on various online learning tasks. Some concluding remarks are provided in Section VI.

Notations: Bold lowercase letters denote the column vectors. For any vector $\mathbf{x}$, $\mathbf{x}^\top$ denotes the transpose of $\mathbf{x}$ and $\|\mathbf{x}\|$ denote the ℓ_2 -norm of $\mathbf{x}$. Also, $\mathbb{E}[\cdot]$ represents the expectation over an associated probability distribution. To simplify the notations, we let $[m] \triangleq \{1, \dots, m\}$ for any positive integer m . Also, t , and i will be used to indicate the indices of time and kernel, respectively. The number of kernels in a kernel dictionary is denoted as n and the total number of incoming data is denoted as T .

II. PROBLEM STATEMENT

In this section, we define the problem setting of our online learning framework. To this end, we first give some backgrounds on a kernel-based learning. Given data samples $(\mathbf{x}_t, y_t), t \in [T]$ with features $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$ and $y_t \in \mathcal{Y} \subset \mathbb{R}$, a kernel-based function learning task is to seek a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that the cumulative loss is minimized:

$$f^* = \underset{f \in \mathcal{H}}{\text{argmin}} \sum_{t=1}^T \mathcal{L}(f(\mathbf{x}_t), y_t), \quad (5)$$

where $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ denotes a loss (or cost) function and $\mathcal{H}$ denotes a reproducing kernel Hilbert space (RKHS) induced by a kernel κ . Formally, the function space RKHS $\mathcal{H}$ is defined as

$$\mathcal{H} = \left\{ f : f(\mathbf{x}) = \sum_{t=1}^{\infty} \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t) \right\}, \quad (6)$$

where $\kappa(\mathbf{x}, \mathbf{x}_t) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a symmetric positive semidefinite basis function (called kernel) [19]. This basis (kernel) function can measure the similarity between $\mathbf{x}$ and $\mathbf{x}_t$. One representative example is a Gaussian kernel with a parameter σ^2 , given as

$$\kappa(\mathbf{x}, \mathbf{x}_t) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_t\|^2}{2\sigma^2}\right). \quad (7)$$

When the number of incoming data (or data samples) is finite (e.g., T samples), the representer theorem in [20] has proved that an optimal solution of (5) subject to $f \in \mathcal{H}$ can be represented using T basis functions only, i.e.,

$$f(\mathbf{x}) = \sum_{t=1}^T \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t). \quad (8)$$

This kernel-based learning can suffer from the curse of dimensionality as the number of optimization variables α_t 's grows linearly with the number of data samples T . Hence, it cannot

be applicable for an online learning with continuous streaming data (e.g., time-series data).

The above problem has been addressed in [13] by introducing the random feature (RF) approximation for kernels. Specifically, a function $f(\mathbf{x})$ in (8) can be well-approximated as the following parameterized function:

$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{z}(\mathbf{x}), \quad (9)$$

with a parameter $\mathbf{w} \in \mathbb{R}^s$, where $\{\mathbf{v}_i : i \in [s/2]\}$ denotes an independent and identically distributed (i.i.d.) samples from the Fourier transform of a given kernel function $\kappa(\cdot, \cdot)$ (denoted by $\pi_\kappa(\mathbf{v})$), and the so-called feature mapping $\mathbf{z} : \mathbb{R}^d \rightarrow \mathbb{R}^s$ is given as

$$\mathbf{z}(\mathbf{x}) = \frac{1}{\sqrt{s/2}} [\sin \mathbf{v}_1^\top \mathbf{x}, \dots, \sin \mathbf{v}_{s/2}^\top \mathbf{x}, \cos \mathbf{v}_1^\top \mathbf{x}, \dots, \cos \mathbf{v}_{s/2}^\top \mathbf{x}]^\top. \quad (10)$$

Here, a hyper-parameter s can be chosen irrespective of the number of data samples (e.g., T). Also, it can control the tradeoff between an approximation accuracy and an optimization complexity. In the example of a Gaussian kernel in (7), we have $\pi_{\kappa_p}(\mathbf{v}) = \mathcal{N}(0, \sigma^{-2}\mathbf{I})$ [21]. As seen in (9), the RF approximation can transform a function learning into a vector optimization problem.

The above RF-based kernel learning can be naturally extended into a multiple kernel framework [8], [9]. To this end, a set of n kernels (called a kernel dictionary) are preselected, which are respectively denoted as κ_i for $i \in [n]$. Accordingly, we can define the RKHS $\mathcal{H}_i$ induced by the kernel κ_i for $i \in [n]$. From (1) and (2), a function (or search) space for our RF-based MKL is defined as

$$\bar{\mathcal{H}}_{\text{RF}} \triangleq \left\{ f(\mathbf{x}; \{q_i, \mathbf{w}_i\}) = \sum_{i=1}^n q_i f(\mathbf{x}; \mathbf{w}_i) \right\}, \quad (11)$$

where $f(\mathbf{x}; \mathbf{w}_i) \in \mathcal{H}_i$, $i \in [n]$, and $(q_1, \dots, q_n) \in \Delta_n$. Using the above function space, the optimization problem in (5) can be formulated as

$$\begin{aligned} \{\mathbf{w}_{[\star, i]}, q_{[\star, i]}\} = & \underset{\{\mathbf{w}_i, q_i : i \in [n]\}}{\operatorname{argmin}} \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_i \mathbf{w}_i^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ \text{subject to } & [q_1, \dots, q_n]^\top \in \Delta_n. \end{aligned} \quad (12)$$

From now on, we formally define our online learning framework, called RF-based online multi-kernel learning (in short, *RF-OMKL framework*). In this framework, the objective is to learn a sequence of functions $\hat{f}(\mathbf{x} : \{\hat{q}_{[t, i]}, \hat{\mathbf{w}}_{[t, i]}\})$, $t \in [T]$ in an online fashion. As in other online learning frameworks [16], [14], [15], the performance is measured by the cumulative regret:

$$\begin{aligned} \text{regret}_T = & \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n \hat{q}_{[t, i]} \hat{\mathbf{w}}_{[t, i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ & - \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_{[\star, i]} \mathbf{w}_{[\star, i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right), \end{aligned} \quad (13)$$

where the optimal parameters $\{\mathbf{w}_{\star, i}, q_{[\star, i]} : i \in [n]\}$ are defined in (12). Note that this metric compares the cumulative

loss of our algorithm to the cumulative loss of the *static* best function from the function space $\bar{\mathcal{H}}_{\text{RF}}$ (equivalently, the optimal solutions of (12)).

The main purpose of this paper is to construct an efficient algorithm which generates the parameters $\{\hat{q}_{[t, i]}, \hat{\mathbf{w}}_{[t, i]}\}$ in an online fashion such that a sublinear regret bound is achieved, i.e., $\text{regret}_T \leq \mathcal{O}(\sqrt{T})$.

III. LIMITATION OF THE EXISTING ALGORITHMS

We review the best known algorithm (termed Raker) proposed in [14] for RF-OMKL framework and then discuss the inherent limitation of Raker. We note that the same limitation is straightforwardly applied to AMKL [15]. In [14], [15], the biconvex optimization problem in (12) has not been solved exactly. Instead, the objective function in (12) has been approximated using the convexity of a loss function, yielding the following optimization problem:

$$\begin{aligned} \{\bar{\mathbf{w}}_{[\star, i]}, \bar{q}_{[\star, i]}\} = & \underset{\{\mathbf{w}_i, q_i : i \in [n]\}}{\operatorname{argmin}} \sum_{t=1}^T \sum_{i=1}^n q_i \mathcal{L}(\mathbf{w}_i^\top \mathbf{z}_i(\mathbf{x}), y) \\ \text{subject to } & [q_1, \dots, q_n]^\top \in \Delta_n. \end{aligned} \quad (14)$$

The above objective function is an upper bound of that in (12) and the equality only holds when a linear loss function is used. As noticed in [14], the key advantage of such approximation is that the optimization problem in (14) can be *decomposed* with respect to kernels. Namely, for any fixed kernel $i \in [n]$, the optimal parameter $\bar{\mathbf{w}}_{[\star, i]}$ can be obtained irrespective of the other parameters, by taking the solution of

$$\bar{\mathbf{w}}_{[\star, i]} = \underset{\mathbf{w}_i}{\operatorname{argmin}} \sum_{t=1}^T \mathcal{L}(\mathbf{w}_i^\top \mathbf{z}_i(\mathbf{x}_t), y_t). \quad (15)$$

Given $\{\bar{\mathbf{w}}_{[\star, i]} : i \in [n]\}$, then, the optimization for combination weights becomes trivial as in this case, the associated objective function is linear. The optimal weights are simply obtained as

$$\bar{q}_{[\star, i]} = \begin{cases} 1, & i = p^\star \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

where

$$p^\star = \underset{1 \leq i \leq n}{\operatorname{argmin}} \sum_{t=1}^T \mathcal{L}(\mathbf{w}_{[\star, i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t). \quad (17)$$

Here, $p^\star$ stands for the index of the best kernel for the given data samples $\{(\mathbf{x}_t, y_t) : t \in [T]\}$. Also, $\hat{f}(\mathbf{x}; \bar{\mathbf{w}}_{[\star, p^\star]})$ denotes the best function belonging to the best *single* kernel $\mathcal{H}_{p^\star}$.

The above decomposed problem can be viewed as the famous online learning framework, known as *online learning with expert advice* [22], [23], [17]. In this expert-based learning framework, the best expert is determined in an online fashion. Based on this, Raker has been constructed in [14], which performs with the following two steps. At every time t , each kernel i generates its own best function $\hat{f}_i(\mathbf{x}; \bar{\mathbf{w}}_{[t, i]})$ where the parameter $\bar{\mathbf{w}}_{[t, i]}$ is optimized by solving the sub-optimization problem in (15). In [14], it is solved via online gradient descent (OGD) as

$$\bar{\mathbf{w}}_{[t+1, i]} = \bar{\mathbf{w}}_{[t, i]} - \eta_l \nabla \mathcal{L}(\bar{\mathbf{w}}_{[t, i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t), \quad (18)$$

where $\nabla \mathcal{L}(\bar{\mathbf{w}}_{t,i}^\top \mathbf{z}_i(\mathbf{x}_t), y_t)$ denotes the gradient at the point $\bar{\mathbf{w}}_{t,i}$. Note that this optimization is performed within each kernel, independently from the other kernels. The second step is to optimize combination weights on the basis of the optimized kernel functions. Since this problem is equivalent to the expert-based online learning, the *Exp strategy* (a.k.a. Hedge algorithm) in the expert-based online learning [16] can be immediately applied, in which the combination weights are computed as

$$\bar{q}_{[t,i]} = \frac{\exp\left(-\eta_g \sum_{\tau=1}^{t-1} \mathcal{L}(\hat{f}_i(\mathbf{x}_\tau; \bar{\mathbf{w}}_{[\tau,i]}), y_\tau)\right)}{\sum_{i=1}^n \exp\left(-\eta_g \sum_{\tau=1}^{t-1} \mathcal{L}(\hat{f}_i(\mathbf{x}_\tau; \bar{\mathbf{w}}_{[\tau,i]}), y_\tau)\right)}. \quad (19)$$

The Exp strategy ensures that as t increases, the weight of the best kernel p^* approaches to 1. Also, leveraging the analysis of OGD and Exp strategy properly, it was proved in [14], [15] that Raker and AMKL can achieve a sublinear regret bound compared with the best solution of the approximated optimization problem in (14), i.e., a sublinear *expert* regret bound is achieved:

$$\begin{aligned} \text{regret}_T^E &= \sum_{t=1}^T \mathcal{L}\left(\sum_{i=1}^n \bar{q}_{[t,i]} \bar{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t\right) \\ &\quad - \min_{1 \leq i \leq n} \sum_{t=1}^T \mathcal{L}\left(\bar{\mathbf{w}}_{[\star,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t\right) \leq \mathcal{O}(\sqrt{T}). \end{aligned} \quad (20)$$

Definitely this analysis does not guarantee that Raker and AMKL can achieve a sublinear regret bound as to (13). A natural question arises: *can we achieve the optimal performance of (12) within a sublinear gap in an online fashion?* We will contribute to this open problem in Section IV.

IV. MAIN RESULTS

In this section, we propose a novel algorithm (named BestOMKL) for RF-OMKL framework. We then prove that it can achieve a sublinear regret bound under a certain condition, i.e., $\text{regret}_T \leq \mathcal{O}(\sqrt{T})$. Beyond this asymptotic optimality, the superiority of our algorithm on practical online learning tasks will be demonstrated in Section V.

A. The Proposed Algorithm

Following RF-OMKL framework, the proposed BestOMKL seeks a sequence of functions $f(\mathbf{x}; \{\hat{q}_{[t,i]}, \hat{\mathbf{w}}_{[t,i]} : i \in [n]\}) \in \bar{\mathcal{H}}_{\text{RF}}$ for $t \in [T]$ in an online fashion. From (11), the learned function with the parameters $\{\hat{q}_{[t,i]}, \hat{\mathbf{w}}_{[t,i]}\}$ is represented as

$$\hat{f}(\mathbf{x}; \{\hat{q}_{[t,i]}, \hat{\mathbf{w}}_{[t,i]}\}) = \sum_{i=1}^n \hat{q}_{[t,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}). \quad (21)$$

BestOMKL is completely defined by describing how to optimize the parameters in a sequential fashion. We first give some useful backgrounds to describe our optimization method. Note that our loss function $\mathcal{L}\left(\sum_{i=1}^n \hat{q}_{[t,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t\right)$ is *biconvex*. Specifically, it is a convex function with respect to $\{\hat{\mathbf{w}}_{[t,i]}\}$ (resp. $\{\hat{q}_{[t,i]}\}$) when $\{\hat{q}_{[t,i]}\}$ (resp. $\{\hat{\mathbf{w}}_{[t,i]}\}$) are fixed. For ease of exposition, we let $\nabla_{[\mathbf{w},t]} \in \mathbb{R}^{n \times s \times 1}$ and $\nabla_{[q,t]} \in \mathbb{R}^{n \times 1}$ be the gradients of the biconvex loss function with respect to

$\{\hat{\mathbf{w}}_{[t,i]}\}$ and $\{\hat{q}_{[t,i]}\}$, respectively. In the example of a least-square loss function, defined as

$$\mathcal{L}\left(\sum_{i=1}^n q_{[t,i]} \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}), y\right) = \left[\sum_{i=1}^n q_{[t,i]} \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}) - y\right]^2,$$

the corresponding gradients are computed as

$$\nabla_{[\mathbf{w},t]} = c_t [q_{[t,1]} \mathbf{z}_1(\mathbf{x})^\top, \dots, q_{[t,n]} \mathbf{z}_n(\mathbf{x})^\top]^\top \in \mathbb{R}^{n \times s \times 1} \quad (22)$$

$$\nabla_{[q,t]} = c_t [\mathbf{w}_{[t,1]}^\top \mathbf{z}_1(\mathbf{x}), \dots, \mathbf{w}_{[t,n]}^\top \mathbf{z}_n(\mathbf{x})]^\top \in \mathbb{R}^{n \times 1}, \quad (23)$$

where the scalar c_t is given as

$$c_t = 2 \sum_{i=1}^n [q_{[t,i]} \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}) - y]. \quad (24)$$

The above gradients will be used for our numerical tests in Section V.

We are now ready to state our algorithm. Unlike the expert-based approach in [14], [15], we solve the biconvex optimization in (12) directly without the approximation for decomposition. In our algorithm, the parameters $\hat{\mathbf{w}}_t = [\hat{\mathbf{w}}_{[t,1]}^\top, \dots, \hat{\mathbf{w}}_{[t,n]}^\top]^\top$ and $\hat{\mathbf{q}}_t = [\hat{q}_{[t,1]}, \dots, \hat{q}_{[t,n]}]^\top$ will be optimized alternatively with the following two steps (see **Algorithm 1**).

Step 1: Fixing the weights $\{\hat{q}_{[t,i]}\}$, the parameters $\{\hat{\mathbf{w}}_{[t,i]}\}$ are optimized via group-OGD as

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t - \eta_l \nabla_{[\mathbf{w},t]}. \quad (25)$$

We remark that the impact of the combination weights is captured by the gradient $\nabla_{[\mathbf{w},t]}$. Intuitively, the group-OGD aims at finding the best function from the RKHS $\mathcal{H}_{\hat{\mathbf{q}}_t}$ induced by the kernel $\bar{\kappa}_t = \sum_{i=1}^n \hat{q}_{[t,i]} \kappa_i$. It is expected that a sequence of optimized functions via group-OGD can converge to the best function from $\bar{\mathcal{H}}$ (i.e., the best function of (12)), provided that the weights $\{\hat{q}_{[t,i]} : i \in [n]\}$ converge to the optimal ones $\{q_{[\star,i]} : i \in [n]\}$. In contrast, the goal of OGD update in Raker, defined in (18), is totally different. For any fixed $i \in [n]$, the OGD update aims at finding the best kernel function in the *single* RKHS $\mathcal{H}_i$ induced by κ_i . While this optimization can be suitable for the expert-based learning framework, $\{\bar{\mathbf{w}}_{[t,i]}\}$ cannot converge to the optimal function in our function space $\bar{\mathcal{H}}$. For this reason, Raker only achieves a sublinear *expert* regret bound, i.e., it is failed to attain a sublinear regret bound.

Step 2: Our weight optimization is constructed such that a sequence of kernels $\bar{\kappa}_t = \sum_{i=1}^n \hat{q}_{[t,i]} \kappa_i$ can converge to the best kernel $\bar{\kappa}^\star = \sum_{i=1}^n \hat{q}_{[\star,i]} \kappa_i$ with a sublinear rate. From our theoretical analysis (see Lemma 1), it is proved that the above goal can be attained by choosing $\{\hat{q}_{[t+1,i]}\}$ in the following ways: Given $\{\hat{\mathbf{w}}_{[t,i]}\}$ (equivalently, $\nabla_{[q,t]}$ is fixed), the weights $\{\hat{q}_{[t+1,i]}\}$ are computed as

$$\hat{q}_{[t+1,i]} = \frac{\exp\left(-\eta_g \sum_{\tau=1}^{t-1} \nabla_{[q,\tau]}^\top \mathbf{e}_i\right)}{\sum_{i=1}^n \exp\left(-\eta_g \sum_{\tau=1}^{t-1} \nabla_{[q,\tau]}^\top \mathbf{e}_i\right)}, \quad i \in [n] \quad (26)$$

where $\mathbf{e}_i \in \mathbb{R}^{n \times 1}$ denotes a unit vector whose elements are all zeros except for the i -th position. This approach is called *gradient-Exp strategy*. We would like to emphasize that although the name is similar to the Exp strategy in Raker,

their purposes are completely different. The Exp strategy in (19) is customized for the expert-based learning, namely, it aims at finding the best single kernel (i.e., the best expert) κ_{p^*} in an online fashion. Clearly, the optimized function space in our algorithm (denoted by $\bar{\mathcal{H}}_{\mathbf{q}_*}$) should be more suitable for the given data samples than that in Raker (denoted by $\mathcal{H}_{p^*}$). This is because the combination weights in the former are optimized over the probability simplex Δ_n , whereas in the latter, they are optimized over the standard basis $\mathbb{R}^n$ only (i.e., the extreme points in Δ_n only). Therefore, the proposed BestOMKL can have a potential gain over Raker, which will be verified in Section IV-B and Section V.

Algorithm 1 The Proposed BestOMKL

- 1: **Input:** A preselected set of n kernels $\{\kappa_i : i \in [n]\}$, hyper-parameters (η_l, η_g) , and the number of random features s .
- 2: **Output:** A sequence of functions $\hat{f}_t(\mathbf{x})$, $t \in [T + 1]$.
- 3: **Initialization:** $\hat{\mathbf{w}}_{[1,i]} = \mathbf{0}$ and $\hat{q}_{1,i} = 1/n$, $\forall i \in [n]$.
- 4: **Iteration:** $t = 1, \dots, T$
 - Receive a streaming data $(\mathbf{x}_t, y_t)$.
 - Update $\{\hat{\mathbf{w}}_{[t+1,i]} : i \in [n]\}$ via group-OGD in (25).
 - Update $\{\hat{q}_{[t+1,i]} : i \in [n]\}$ via gradient-Exp in (26).
 - Construct an learned function as

$$\hat{f}(\mathbf{x}; \{\hat{q}_{[t+1,i]}, \hat{\mathbf{w}}_{[t+1,i]}\}) = \sum_{i=1}^n \hat{q}_{[t+1,i]} \hat{\mathbf{w}}_{[t+1,i]}^\top \mathbf{z}_i(\mathbf{x}).$$

B. Regret Analysis

We analyze the cumulative regret of the proposed BestOMKL in **Algorithm 1**. For consistency, we follow the notations and definitions in Section III and Section IV. For example, $\{\mathbf{w}_{[t,i]}, \hat{q}_{[t,i]}\}$ are the parameters generated by BestOMKL (see Algorithm 1), and $\nabla_{[\mathbf{w},t]}$ and $\nabla_{[q,t]}$ are the gradients of our biconvex loss function, which are respectively used for group-OGD update and gradient-Exp strategy. For our theoretical analysis, we make the following assumptions, which are usually assumed for the analysis of online convex optimization and online learning [14], [15], [16], [24].

Assumption 1. The loss function $\mathcal{L}(\sum_{i=1}^n q_i \mathbf{w}_i^\top \mathbf{z}_i(\mathbf{x}), y)$ is differentiable and biconvex with respect to the parameters $\mathbf{w}$ and $\mathbf{q}$. Also, it is bounded as $\mathcal{L}(\cdot, \cdot) \in [0, 1]$.

Assumption 2. The optimal parameter $\mathbf{w}_*$ is assumed to be bounded as $\|\mathbf{w}_*\|^2 \leq C$.

Assumption 3. The loss function (with respect to $\mathbf{w}$) is L -Lipschitz continuous, i.e., $\|\nabla_{[\mathbf{w},t]}\|^2 \leq L$ for any $t \in [T]$.

Before stating the main result, we provide some useful notations and definitions that will be used throughout the section. Taking our two-step approach into account, the optimal parameters $\mathbf{w}_* = [\mathbf{w}_{[*],1}^\top, \dots, \mathbf{w}_{[*],n}^\top]^\top$ and $\mathbf{q}_* = (q_{[*],1}, \dots, q_{[*],n})$,

which are the solutions of (12), can be represented as

$$\{q_{[*],i}\} = \operatorname{argmin}_{\mathbf{q} \in \Delta_n} \mathcal{L} \left(\sum_{i=1}^n q_i \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \quad (27)$$

$$\{\mathbf{w}_{[*],i}\} = \operatorname{argmin}_{\{\mathbf{w}_i : i \in [n]\}} \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_{[*],i} \mathbf{w}_i^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right). \quad (28)$$

We give some intuitions for our two-step approach. Once the optimal weight $\mathbf{q}_*$ (equivalently, the RKHS $\bar{\mathcal{H}}_{\mathbf{q}_*}$) is fixed, a sequence of parameters $\hat{\mathbf{w}}_{[t,i]}$ should be optimized from the RKHS $\bar{\mathcal{H}}_{\mathbf{q}_*}$ so that it can converge to the optimal parameter $\mathbf{w}_*$. Namely, for this optimization, the following loss function defined as $(\mathbf{q}_*, \mathbf{x}_t, y_t)$ should be considered:

$$\mathcal{L} \left(\sum_{i=1}^n q_{[*],i} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right). \quad (29)$$

We call it a *true* loss function. Accordingly, group-OGD should be performed with the *true* gradient, defined as

$$\nabla_{[\mathbf{w}_*,t]} = \nabla \mathcal{L} \left(\sum_{i=1}^n q_{[*],i} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right). \quad (30)$$

However, in BestOMKL, the following gradient is used:

$$\nabla_{[\mathbf{w},t]} = \nabla \mathcal{L} \left(\sum_{i=1}^n \hat{q}_{[t,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right). \quad (31)$$

Following the analysis of online convex optimization, it is immediately proved that a sequence of the parameters obtained by group-OGD with $\nabla_{[\mathbf{w}_*,t]}$ can achieve a sublinear regret bound compared with the best function (parameterized by $\mathbf{w}_*$) from the RKHS $\bar{\mathcal{H}}_{\mathbf{q}_*}$. However, this is not applicable under our online learning framework, because $\mathbf{q}_*$ is unknown and should be also optimized as the part of the learning process. In the proposed BestOMKL, the above process can be mimicked via gradient-Exp strategy, namely, $\hat{\mathbf{q}}_t = [\hat{q}_{[t,1]}, \dots, \hat{q}_{[t,n]}]^\top$ can converge to the optimal weight $\mathbf{q}_*$ as t grows within a negligible gap. Consequently, the following can be happened: as t grows,

- our function space $\bar{\mathcal{H}}_{\hat{\mathbf{q}}_t}$ can converge to the best RKHS $\bar{\mathcal{H}}_{\mathbf{q}_*}$;
- our gradient $\nabla_{[\mathbf{w},t]}$ can be closer to the true one $\nabla_{[\mathbf{w}_*,t]}$.

When t is sufficiently large, we can expect that $\hat{\mathbf{q}}_t$ and $\hat{\mathbf{w}}_t$ can approach to the optimal parameters $\hat{\mathbf{q}}_*$ and $\mathbf{w}_*$, respectively. In our analysis, it is proved that $\hat{\mathbf{q}}_t$ can converge to $\hat{\mathbf{q}}_*$ within a sublinear gap (see Lemma 1). Then, the performance of BestOMKL relies on the accuracy of our gradient $\nabla_{[\mathbf{w},t]}$, which can be considered as an estimated gradient of the true $\nabla_{[\mathbf{w}_*,t]}$. In Lemma 2, it is proved that when the cumulative estimation error is bounded, BestOMKL can achieve a sublinear regret bound. To measure such error formally, we define:

$$\zeta_t \triangleq (\nabla_{[\mathbf{w}_*,t]} - \nabla_{[\mathbf{w},t]})^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*). \quad (32)$$

Note that ζ_t goes to zero, provided that $\nabla_{[\mathbf{w}_*,t]} - \nabla_{[\mathbf{w},t]}$ is orthogonal to the estimation error $\hat{\mathbf{w}}_t - \mathbf{w}_*$. This makes sense as in this case, using $\nabla_{[\mathbf{w}_*,t]}$ can only move our estimated parameter $\{\hat{\mathbf{w}}_{[t,i]}\}$ into an orthogonal direction from the optimal parameter $\mathbf{w}_*$, which cannot improve the

accuracy. Likewise, when $\zeta_t \neq 0$, there is a room to improve our accuracy with the true gradient. To achieve our desired sublinear regret, some mismatches of our (estimated) gradients from the corresponding true ones can be allowed, which is defined as

$$\sum_{t=1}^T \zeta_t \leq \mathcal{O}(\sqrt{T}). \quad (33)$$

Unfortunately, it is demanding to analytically prove that BestOMKL obeys the condition in (33). The hardness of this analysis is due to the use of our two-step approach, whereas this makes the complexity of our algorithm lower as Raker (or AMKL). Nevertheless, as explained above, one can expect that our (estimated) gradient in (30) becomes very similar (under our measure) to the true gradient in (30) as t grows. This expectation has been verified numerically, i.e., we have confirmed that the condition in (33) can be satisfied for online learning tasks with real datasets. This is also reflected in our numerical tests in Fig. 1, Table II, and Table III. Based on the above intuition, we derive our main result, which is stated in Theorem 1 below.

Theorem 1: Under the assumptions (a1)-(a3), the proposed BestOMKL in Algorithm 1 achieves the following cumulative regret bound:

$$\text{regret}_T \leq \frac{C}{2\eta_l} + \frac{\eta_l TL}{2} + \frac{\log n}{\eta_g} + \frac{\eta_g TL}{2} + \sum_{t=1}^T \zeta_t,$$

where regret_T and ζ_t are defined in (13) and (32), respectively.

Setting $\eta_l = \eta_g = \mathcal{O}(1/\sqrt{T})$ and when the condition in (33) holds, the sublinear regret bound $\mathcal{O}(\sqrt{T})$ is achieved, i.e.,

$$\text{regret}_T \leq \mathcal{O}(\sqrt{T}).$$

Proof: The proof will be completed using the following two key lemmas:

Lemma 1: Given $\{\hat{\mathbf{w}}_{[t,i]} : i \in [n], t \in [T]\}$, the gradient-Exp strategy in (26) achieves the following regret bound:

$$\begin{aligned} & \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n \hat{q}_{[t,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ & - \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_{[\star,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ & \leq \frac{\log n}{\eta_g} + \frac{\eta_g TL}{2}. \end{aligned} \quad (34)$$

Proof: The proof is provided in Appendix A. ■

Lemma 2: Given the optimal weights $\mathbf{q}_\star \in \Delta_n$, the group-OGD in (25) can achieve the following regret bound:

$$\begin{aligned} & \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_{[\star,i]} \hat{\mathbf{w}}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ & - \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_{[\star,i]} \mathbf{w}_{[\star,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ & \leq \frac{C}{2\eta_l} + \frac{\eta_l TL}{2} + \sum_{t=1}^T \zeta_t, \end{aligned} \quad (35)$$

where ζ_t is defined in (32).

Proof: The proof is provided in Appendix B. ■

Combining (34) and (35), we can complete the proof. ■

V. EXPERIMENTAL RESULTS

In this section, we verify the effectiveness of the proposed BestOMKL via experiments on various online regression and time-series prediction tasks with real datasets. We first provide the simulation settings, benchmark algorithms, and hyper-parameters. Regarding a loss function, regularized least-square loss function with a parameter $\lambda > 0$ is used as in [14], [15]. Let $\hat{y}_t$ and y_t denote an estimated label and a true label at time t , respectively. As in [14], [15], the accuracy of a function learning is measured by the mean-square-error (MSE) as

$$\text{MSE}(t) = \frac{1}{t} \sum_{\tau=1}^t (\hat{y}_\tau - y_\tau)^2. \quad (36)$$

As benchmark algorithms, we consider various kernel-based algorithms, which are also used in the most related papers [14], [15]:

- **RBF:** The online *single* kernel learning algorithm using Gaussian kernels with the parameters $\sigma^2 = [0.1, 1, 10]$ (e.g., KL-RBF(σ^2)).
- **POLY:** The online *single* kernel learning algorithm using polynomial kernels with degree $d = \{2, 3\}$ (e.g., POLY2 and POLY3).
- **LINEAR:** The online *single* kernel learning algorithm using linear kernel.
- **OMKR:** The famous online multiple kernel learning algorithm without RF approximation [25].
- **OMKL-B:** The online multiple kernel learning algorithm on a budget [19].
- **Raker:** The online multiple kernel learning algorithm based on RF approximation [14].

Due to the randomness of the RF-based algorithms caused by random features $\mathbf{z}_i(\mathbf{x})$ in (10), the averaged MSE performances over 100 trials are evaluated. Also, the following hyper-parameters will be used throughout the experiments. For all MKL algorithms, OMKR, OMKL-B, Raker, and BestOMKL, we use the kernel dictionary consisting of 7 Gaussian kernels whose parameters are given as

$$\sigma_i^2 = 10^{i-9}, i = 1, 2, \dots, 17. \quad (37)$$

In Raker and BestOMKL, the associated parameters are set by

$$\eta_l = \eta_g = \frac{1}{\sqrt{T}}, s = 50, \text{ and } \lambda = 0.01. \quad (38)$$

The budget size of OMKL-B is chosen as $B = 50$. The real datasets to be used for our experiments are given in Section V-A and the experimental results and some discussions are provided in Section V-B

A. Descriptions of online learning tasks and datasets

We describe online regression and time-series prediction tasks and real-world datasets for the experiments.

Fig. 1. Comparisons of MSE performances of various learning algorithms on online regression tasks.

TABLE I
SUMMARY OF REAL DATASETS FOR EXPERIMENTS

Datasets	# of features	# of data	feature type
Online regression tasks			
Twitter	77	98704	real & integer
Conductivity	81	11000	real
Tom's hardware	96	9725	real& integer
Air quality	13	7322	real
Wave energy	48	9500	real
Weather	21	7750	real
Naval propulsion plants	16	11934	real
Time-series prediction tasks			
Temperature	5	5500	real
Traffic	5	6500	real
NASDAQ Index	5	1977	real
S&P Index	5	1977	real
Power consumption	10	100000	real
Parking occupancy	10	21500	real

1) *Real datasets for online regression:* For the experiments of online regressions, the following real datasets from UCI Machine Learning Repository are considered, which are also

summarized in Table I.

- **Twitter [26]:** Data contains buzz events from Twitter, in which each attributes is used to predict the popularity of a topic. Higher value indicates more popularity.
- **Conductivity [27]:** Data contains the samples of extracted from superconductors, where each feature represents the critical information to construct superconductor (e.g., density and mass of atoms). The task is to predict the critical temperature to create superconductor.
- **Tom's hardware [26]:** Data consists of samples acquired from a forum, where each feature represents the number of times that a content is displayed to visitors. The learning task is to predict the average number of display about a certain topic
- **Air quality [28]:** Data includes samples, of which feature includes hourly response from an array of chemical sensors. The goal is to predict the concentration of polluting chemicals in the air.

TABLE II
COMPARISONS OF MSE ($\times 10^{-3}$) ON ONLINE REGRESSIONS TASKS

	Conductivity	Wave	Weather	Air	Tom's	Plant	Twitter
Raker	19.9	0.29	1.22	2.11	3.66	2.38	3.06
BestOMKL	16.6	0.17	0.85	1.18	1.17	0.54	2.47

TABLE III
COMPARISONS OF MSE ($\times 10^{-3}$) ON TIME-SERIES PREDICTION TASKS

	Temperature	NASDAQ	S&P	Power	Parking	Traffic
Raker	0.20	1.20	1.10	1.90	3.20	8.33
BestOMKL	0.17	0.56	0.54	1.65	1.90	8.26

- **Wave energy [29]:** Data contains the samples consisting of positions and absorbed power obtained from wave energy converters. The task is to predict the total power energy of the farm.
- **Weather [30]:** Data contains the samples obtained from LDAPS model during 2015-2017, where each feature shows the geographical variables of Seoul. The task is to predict the minimum temperatures of next day.
- **Naval Propulsion Plants [31]:** Data contains the samples with 16 features such as ship speed and fuel flow. The goal is to determine turbine decay state coefficient.

2) *Real datasets for time-series prediction:* We consider time-series prediction tasks which estimate the future values in online fashion. As considered in [15], the famous time-series prediction method called Autogressive (AR) model is considered. An AR(p) model predicts the future value y_t assuming the linear dependency on its s values, i.e.,

$$y_t = \sum_{i=1}^p \gamma_i y_{t-i} + z_t, \quad (39)$$

where γ_i denotes the weight for y_{t-i} and z_t denotes a Gaussian noise at time t . Similarly in the RF-OMKL framework, AR(p) model can be represented as

$$y_t = f(\mathbf{x}_t : \{\hat{\mathbf{w}}_{[t,i]}, \hat{q}_{[t,i]} : i \in [n]\}) + z_t, \quad (40)$$

where $\mathbf{x}_t = [y_{t-1}, \dots, y_{t-p}]^T$. The proposed BestOMKL and the state-of-the-art Raker are tested with the following univariate time-series datasets from UCI Machine Learning Repository, which are also summarized in Table I.

- **Temperature [32]:** Data consists of the time-series hourly temperature data, which is collected from the midway between Minneapolis and St Paul, MN.
- **Traffic [32]:** Data consists of the time-series hourly traffic data obtained from the same source as above.
- **NASDAQ index [33]:** Data contains the total 1984 data of daily features of NASDAQ indices from 2010 to 2017.
- **S&P index [33]:** Data contains the total 1984 data of daily features of S&P 500 indices from 2010 to 2017.
- **Power consumption [34]:** Data contains samples, each of which represents the active energy consumed every minute (in watt per hour) in the household by electrical equipment.

- **Parking occupancy [35]:** Data contains samples obtained from the parking lot in Birmingham, each of which indicates the car park occupancy rate.

B. Performance Evaluations

We compare the MSE performances of the proposed BestOMKL with the benchmark algorithms in Section V-A. As functional learning tasks, online regression and time-series prediction are considered. Fig. 1 shows the MSE performances of various algorithms on online regression tasks with real datasets. We observe that RF-OMKL based algorithms (e.g., Raker and BestOMKL) significantly outperform the famous (O)MKL algorithms as well as single kernel-based algorithms (e.g., Gaussian (RBF), Poly, and Linear algorithms). These are well coincided with the previous results in [14], [15]. Moreover, due to the RF approximation, RF-OMKL based algorithms has much lower computational complexity than the other kernel-based algorithms, which have been already verified via CPU execution times. The corresponding results are provided in [14], [15]. These advantages suggest the practicality of RF-OMKL based algorithms such as Raker and BestOMKL.

Our experiments more focus on the comparisons of Raker (the best known algorithm) [14] and the proposed BestOMKL. We remark that AMKL proposed in [15] achieves the same MSE performances as Raker with little labeled data. Namely, the advantage of AMKL over Raker is to save the labeling cost. Also, we emphasize that an active learning strategy proposed in [15] can be applied to BestOMKL. Since we are focusing on the MSE performances for comparisons, Raker is only considered, while the same performance gaps are kept for AMKL. Table II and Table III show the MSE performances of Raker and BestOMKL on online regression and time-series prediction tasks, respectively. The proposed BestOMKL can show better MSE performances than Raker for all the real datasets in our experiments. Notably, in the online regression task with the Plant data, our algorithm has 4 times better MSE performance than Raker. Also, in the time-series prediction tasks with the NASDAQ and S&P data, BestOMKL has 2 times better MSE performances than Raker. These show that our asymptotic analysis is quite accurate even with not-so-large number of incoming data. Moreover, the numerical results lead us to conclude that BestOMKL indeed enjoys the advantage of using multiple kernels in a better way.

VI. CONCLUSIONS

In this paper, we investigated a promising kernel-based online multi-kernel learning framework due to its lower-complexity and scalability. This is referred to as RF-OMKL framework. For this framework, the best known algorithm (termed Raker), which has been constructed based on the expert-based online learning framework, is failed to achieve a sublinear regret bound. We proposed a novel algorithm, named BestOMKL for RF-OMKL framework and proved that it can achieve a sublinear regret bound under a certain condition. In addition to our theoretical contribution, we demonstrated the effectiveness of our algorithm in practical online learning tasks. Remarkably, the proposed BestOMKL outperforms the state-of-the-art kernel-based algorithms whiling enjoying the advantages of RF-OMKL based algorithms. An interesting future work is extend our principle to deal with multiple kernels into decentralized networks such as fully decentralized learning, federated learning, and so on.

APPENDIX A PROOF OF LEMMA 1

For the ease of expression, the $\{\hat{\mathbf{w}}_{[t,i]}, \mathbf{z}_i(\mathbf{x}_t) : i \in [n]\}$ and y_t , which are fixed at time t , are captured by a new variable $\mathbf{z}_t$. Then, our loss function with respect to the weight variables $\hat{\mathbf{q}}_t = [\hat{q}_{[t,1]}, \dots, \hat{q}_{[t,n]}]^\top$ are represented as

$$h(\hat{\mathbf{q}}_t; \mathbf{z}_t) \triangleq \mathcal{L} \left(\sum_{i=1}^n \hat{q}_{[t,i]} \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right). \quad (41)$$

Likewise, $h(\mathbf{q}; \mathbf{z}_t)$ can be defined for any fixed $\mathbf{q} \in \Delta_n$. Then, the optimal weights are defined as

$$\mathbf{q}_\star = \underset{\mathbf{q} \in \Delta_n}{\operatorname{argmin}} h(\mathbf{q}; \mathbf{z}_t). \quad (42)$$

Recall that $\nabla_{[q,t]}^\top$ denotes the gradient to be used in our algorithm (see (23)). Leveraging them, the cumulative regret can be rewritten as

$$\begin{aligned} \sum_{t=1}^T h(\hat{\mathbf{q}}_t; \mathbf{z}_t) - \sum_{t=1}^T h(\mathbf{q}_\star; \mathbf{z}_t) &\stackrel{(a)}{\leq} \sum_{t=1}^T \nabla_{[q,t]}^\top (\hat{\mathbf{q}}_t - \mathbf{q}_\star) \quad (43) \\ &= \sum_{t=1}^T \nabla_{[q,t]}^\top \hat{\mathbf{q}}_t - \nabla_{[q,t]}^\top \mathbf{q}_\star, \end{aligned} \quad (44)$$

where (a) is from the first order characterization of a convex function [24]. Focusing on the upper bound, we can define a *linear* loss function as

$$\ell(\mathbf{q}, \mathbf{z}_t) = \nabla_{[q,t]}^\top \mathbf{q}. \quad (45)$$

We remark that the new loss function is also bounded as

$$\ell(\mathbf{q}, \mathbf{z}_t) \in [-\sqrt{L}, \sqrt{L}], \quad (46)$$

where the inequality is from the Hölder's inequality and Assumption 3. In the remaining part, the proof will be completed by showing that

$$\begin{aligned} \sum_{t=1}^T h(\hat{\mathbf{q}}_t; \mathbf{z}_t) - \sum_{t=1}^T h(\mathbf{q}_\star; \mathbf{z}_t) &\leq \sum_{t=1}^T [\ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) - \ell(\mathbf{q}_\star, \mathbf{z}_t)] \\ &\leq \frac{\log P}{\eta_g} + \frac{\eta_g T}{2}. \end{aligned} \quad (47)$$

In order to prove (47), we let

$$g_{[t,i]} = \exp \left(-\eta_g \sum_{\tau=1}^{t-1} \ell(\mathbf{e}_i, \mathbf{z}_\tau) \right), \quad (48)$$

and $G_t = \sum_{i=1}^n g_{[t,i]}$. By definition, we have that $g_{[1,i]} = 1$ and $G_1 = n$. The proof will be completed by deriving the lower and upper bounds of ζ , defined as

$$\zeta = \log \frac{G_{T+1}}{G_1}. \quad (49)$$

We first focus on the lower bound:

$$\zeta = \log \frac{G_{T+1}}{G_1} = \log \left[\sum_{i=1}^n g_{[T+1,i]} \right] - \log n \quad (50)$$

$$\geq \log \left[\max_{1 \leq i \leq n} g_{[T+1,i]} \right] - \log n \quad (51)$$

$$= -\eta_g \min_{1 \leq i \leq n} \sum_{t=1}^T \ell(\mathbf{e}_i, \mathbf{z}_t) - \log n. \quad (52)$$

Also, from the telescoping, we have:

$$\zeta = \log \frac{G_{T+1}}{G_1} = \sum_{t=1}^T \log \frac{G_{t+1}}{G_t}. \quad (53)$$

We next derive the upper bound of ζ . First we have:

$$\log \frac{G_{t+1}}{G_t} = \log \left[\sum_{i=1}^n \frac{g_{[t,i]}}{G_t} \exp(-\eta_g \ell(\mathbf{e}_i, \mathbf{z}_t)) \right] \quad (54)$$

$$\stackrel{(a)}{=} \log [\mathbb{E}[\exp(-\eta_g \ell(\mathbf{e}_I, \mathbf{z}_t))]] \quad (55)$$

$$\stackrel{(b)}{\leq} -\eta_g \mathbb{E}[\ell(\mathbf{e}_I, \mathbf{z}_t)] + \frac{\eta_g^2 L}{2} \quad (56)$$

$$\stackrel{(c)}{\leq} -\eta_g \ell(\mathbb{E}[\mathbf{e}_I], \mathbf{z}_t) + \frac{\eta_g^2 L}{2} \quad (57)$$

$$= -\eta_g \ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) + \frac{\eta_g^2 L}{2} \quad (58)$$

where the expectation in (a) is with respect to $\mathbb{P}(I = i) = \frac{g_{[t,i]}}{G_t} = \hat{q}_{[t,i]}$ (see (26)), (b) is due to the Hoeffding's lemma with a bounded random variable in $[-\sqrt{L}, \sqrt{L}]$ (see (46) for the boundness) [36] and Assumption 1, and (c) is from the Jensen's inequality. From (53) and (58), we have:

$$\zeta = \log \frac{G_{T+1}}{G_1} \leq \sum_{t=1}^T \left[-\eta_g \ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) + \frac{\eta_g^2 L}{2} \right]. \quad (59)$$

Leveraging the lower bound in (52) and upper bound in (59) of ζ , we have:

$$\begin{aligned} \sum_{t=1}^T \left[-\eta_g \ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) + \frac{\eta_g^2 L}{2} \right] \\ \geq -\eta_g \min_{1 \leq i \leq n} \sum_{t=1}^T \ell(\mathbf{e}_i, \mathbf{z}_t) - \log n. \end{aligned} \quad (60)$$

Rearranging the above terms, we can get the following inequality:

$$\sum_{t=1}^T \ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) - \min_{1 \leq i \leq n} \sum_{t=1}^T \ell(\mathbf{e}_i, \mathbf{z}_t) \leq \frac{\log n}{\eta_g} + \frac{\eta_g T L}{2}. \quad (61)$$

Due to the linearity of a loss function $\ell(\mathbf{q}, \mathbf{z}_t)$, we have:

$$\sum_{t=1}^T \ell(\mathbf{q}_*, \mathbf{z}_t) = \min_{\mathbf{q} \in \Delta_n} \sum_{t=1}^T \ell(\mathbf{q}, \mathbf{z}_t) = \min_{1 \leq i \leq n} \sum_{t=1}^T \ell(\mathbf{e}_i, \mathbf{z}_t). \quad (62)$$

This is because in the minimization of a linear function over a probability simplex, the optimal point is attained only at an extreme point. From (61) and (62), we obtain:

$$\sum_{t=1}^T [\ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) - \ell(\mathbf{q}_*, \mathbf{z}_t)] \leq \frac{\log n}{\eta_g} + \frac{\eta_g T L}{2}. \quad (63)$$

Going back to the original problem and form (41), we have:

$$\begin{aligned} \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n \hat{q}_{[t,i]} \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ - \min_{[q_1, \dots, q_n]^\top \in \Delta_n} \sum_{t=1}^T \mathcal{L} \left(\sum_{i=1}^n q_i \mathbf{w}_{[t,i]}^\top \mathbf{z}_i(\mathbf{x}_t), y_t \right) \\ \leq \sum_{t=1}^T [\ell(\hat{\mathbf{q}}_t, \mathbf{z}_t) - \ell(\mathbf{q}_*, \mathbf{z}_t)] \\ \leq \frac{\log n}{\eta_g} + \frac{\eta_g T L}{2}. \end{aligned}$$

We can complete the proof.

APPENDIX B PROOF OF LEMMA 2

Recall that the best combination weights and the parameters are respectively denoted as $\mathbf{q}_* = [q_{[*],1}, \dots, q_{[*],n}]^\top \in \Delta_n$ and $\{\mathbf{w}_{[*],i} : i \in [n]\}$. For ease of exposition, we let

$$\begin{aligned} \mathbf{w}_* &= [\mathbf{w}_{[*],1}^\top, \dots, \mathbf{w}_{[*],n}^\top]^\top \\ \mathbf{z}_{[t,*]}(\mathbf{x}) &= [q_{[1,*]} \mathbf{z}_1(\mathbf{x}_t)^\top, \dots, q_{[n,*]} \mathbf{z}_n(\mathbf{x}_t)^\top]^\top. \end{aligned} \quad (64)$$

We follow the notations for the two gradients $\nabla_{[\mathbf{w}_*, t]}$ and $\nabla_{[\mathbf{w}, t]}$ in (30) and (31), respectively.

First of all, from the group OGD update in (25), we have:

$$\begin{aligned} \|\hat{\mathbf{w}}_{t+1} - \mathbf{w}_*\|^2 &= \|\hat{\mathbf{w}}_t - \eta_l \nabla_{[\mathbf{w}, t]} - \mathbf{w}_*\|^2 \\ &= \|\hat{\mathbf{w}}_t - \mathbf{w}_*\|^2 + \eta_l^2 \|\nabla_{[\mathbf{w}, t]}\|^2 - 2\eta_l \nabla_{[\mathbf{w}_*, t]}^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*) \\ &\quad - 2\eta_l (\nabla_{[\mathbf{w}, t]} - \nabla_{[\mathbf{w}_*, t]})^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*). \end{aligned} \quad (65)$$

From the convexity of a loss function, we have:

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{w}}_t^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) - \mathcal{L}(\mathbf{w}_*^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) \\ \leq \nabla_{[\mathbf{w}_*, t]}^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*). \end{aligned} \quad (66)$$

From (65) and (66), we have the following upper bound:

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{w}}_t^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) - \mathcal{L}(\mathbf{w}_*^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) \\ \leq \frac{1}{2\eta_l} (\|\hat{\mathbf{w}}_t - \mathbf{w}_*\|^2 - \|\hat{\mathbf{w}}_{t+1} - \mathbf{w}_*\|^2) + \frac{L\eta_l}{2} \\ - (\nabla_{[\mathbf{w}, t]} - \nabla_{[\mathbf{w}_*, t]})^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*). \end{aligned} \quad (67)$$

Using the telescoping sum over $t = 1, \dots, T$, we can get:

$$\begin{aligned} \sum_{t=1}^T \mathcal{L}(\hat{\mathbf{w}}_t^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) - \sum_{t=1}^T \mathcal{L}(\mathbf{w}_*^\top \mathbf{z}_{[t,*]}(\mathbf{x}_t), y_t) \\ \leq \frac{1}{2\eta_l} \|\mathbf{w}_*\|^2 + \frac{\eta_l T L}{2} + \sum_{t=1}^T (\nabla_{[\mathbf{w}_*, t]} - \nabla_{[\mathbf{w}, t]})^\top (\hat{\mathbf{w}}_t - \mathbf{w}_*) \\ \stackrel{(a)}{\leq} \frac{C}{2\eta_l} + \frac{\eta_l T L}{2}, \end{aligned} \quad (68)$$

where (a) is from Assumption 2 and Assumption 3. This completes the proof.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2020R1A2C1099836).

REFERENCES

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [2] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang, "Review of internet of things (iot) in electric power and energy systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 847–870, 2018.
- [3] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- [4] J. Shawe-Taylor, N. Cristianini *et al.*, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [5] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1147–1160, 2010.
- [6] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2011.
- [7] A. J. Smola and B. Schölkopf, *Learning with kernels*. Citeseer, 1998, vol. 4.
- [8] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
- [9] M. Gönen and E. Alpaydm, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
- [10] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2008.
- [11] O. Anava, E. Hazan, S. Mannor, and O. Shamir, "Online learning for time series prediction," in *Conference on learning theory*. PMLR, 2013, pp. 172–184.
- [12] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *arXiv preprint arXiv:1802.02871*, 2018.
- [13] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

- [14] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 773–808, 2019.
- [15] S. Hong and J. Chae, "Active learning with multiple kernels," *accepted to IEEE Transactions on neural networks and learning systems*. [Online] *arXiv preprint arXiv:2005.03188*, 2020.
- [16] S. Bubeck, "Introduction to online optimization," *Lecture Notes*, vol. 2, 2011.
- [17] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.
- [18] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," in *Conference on Learning Theory*. PMLR, 2015, pp. 23–35.
- [19] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2165–2176, 2004.
- [20] G. Wahba, *Spline models for observational data*. Siam, 1990, vol. 59.
- [21] F. Liu, X. Huang, Y. Chen, and J. A. Suykens, "Random features for kernel approximation: A survey in algorithms, theory, and beyond," *arXiv preprint arXiv:2004.11154*, 2020.
- [22] V. Vovk, "A game of prediction with expert advice," *Journal of Computer and System Sciences*, vol. 56, no. 2, pp. 153–173, 1998.
- [23] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.
- [24] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.
- [25] D. Sahoo, S. C. Hoi, and B. Li, "Online multiple kernel regression," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 293–302.
- [26] E. G. François Kawala, Ahlame Douzal-Chouakria and E. Dimert, "Prédictions d'activité dans les réseaux sociaux en ligne," *4i^{eme} Conférence sur les Modèles et l'Analyse des Réseaux: Approches Mathématiques et Informatiques*, 2013.
- [27] K. Hamidieh, "A data-driven statistical model for predicting the critical temperature of a superconductor," *Computational Materials Science*, pp. 346–354, 2018.
- [28] M. P. L. M. Saverio De Vito, Ettore Massera and G. D. Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.
- [29] B. M. M. Neshat and Y. Xia, "A detailed comparison of meta-heuristic methods for optimising wave energy converter placements," in *Proceedings of the Genetic and Evolutionary Computation Conference*, p. 1318–1325.
- [30] I. J. D. Cho, C. Yoo and D. Cha, "Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas," *Earth and Space Science*, 2020.
- [31] O. L. G. A. S. S. A. D. Coraddu, A. and M. Figari, "Machine learning approaches for improving condition based maintenance of naval propulsion plants," *Journal of Engineering for the Maritime Environment*, 2014.
- [32] "<https://archive.ics.uci.edu/ml/datasets/metro+interstate+traffic+volume>."
- [33] E. Hoseinzade and S. Haratizadeh, "Cnnpred: Cnn-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.
- [34] "Georges hebrail. uci machine learning repository, url <https://archive.ics.uci.edu/ml/datasets/>."
- [35] D. H. S. A. Camero, J. Toutouh and E. Alba, "Evolutionary deep learning for car park occupancy prediction in smart cities," *International Conference on Learning and Intelligent Optimization*, 2019.
- [36] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019, vol. 48.