Journal of Machine Learning Research 24 (2023) 1-44 Submitted 8/21; Published 1/23

Graph-Aided Online Multi-Kernel Learning

Pouya M. Ghari PMOLLAEB@UCI.EDU
Department of Electrical Engineering and Computer Science
University of California
Irvine, CA 92697, USA

Yanning Shen YANNINGS@UCI.EDU
Department of Electrical Engineering and Computer Science
University of California
Irvine, CA 92697, USA

Editor: Karsten Borgwardt

Abstract

Multi-kernel learning (MKL) has been widely used in learning problems involving function learning tasks. Compared with single kernel learning approach which relies on a pre-selected kernel, the advantage of MKL is its flexibility results from combining a dictionary of kernels. However, inclusion of irrelevant kernels in the dictionary may deteriorate the accuracy of MKL, and increase the computational complexity. Faced with this challenge, a novel graph-aided framework is developed to select a subset of kernels from the dictionary with the assistance of a graph. Different graph construction and refinement schemes are developed based on incurred losses or kernel similarities to assist the adaptive selection process. Moreover, to cope with the scenario where data may be collected in a sequential fashion, or cannot be stored in batch due to the massive scale, random feature approximation are adopted to enable online function learning. It is proved that our proposed algorithms enjoy sub-linear regret bounds. Experiments on a number of real datasets showcase the advantages of our novel graph-aided algorithms compared to state-of-the-art alternatives. [1]

Keywords: Multi-Kernel Learning, Graphs, Random Features, Function Approximation, Online Learning

1. Introduction

The need for function approximation arises in many machine learning studies including regression, classification, and reinforcement learning, see e.g., (Chung et al., 2019). This paper studies supervised function approximation where given data samples $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^{T}$, the goal is to find the function $f(\cdot)$, such that the difference between $f(\boldsymbol{x}_t)$ and y_t is minimized. In this context, kernel learning methods exhibit reliable performance. Specifically, the function approximation problem becomes tractable under the assumption that $f(\cdot)$ belongs to a reproducing kernel Hilbert space (Scholkopf and Smola, 2001). In some cases, it is imperative to perform function approximation task in an online fashion. For instance, when the volume of data is large and is collected in a sequential fashion, it is impossible to store

[1] Preliminary results of this work were presented in part at the 2020 International Conference on Machine Learning (Ghari and Shen, 2020). The work in this paper is supported by NSF ECCS 2207457. Corresponding author: Yanning Shen.

or process it in batch. Furthermore, suffering from the well-known problem of 'curse of dimensionality' (Bengio et al., 2006; Shawe-Taylor and Cristianini, 2004), kernel learning methods are not suitable for sequential settings. This has motivated studies on online single kernel learning (Lu et al., 2016; Ding et al., 2017; Bouboulis et al., 2018; Zhang and Liao, 2019) to address the curse of dimensionality. Specifically, approximating kernels by finite-dimensional feature representations such as random Fourier feature by Rahimi and Recht (2007) and Nyström method by Williams and Seeger (2000), function approximation task becomes scalable. Furthermore, kernel approximation with finite-dimensional features has been extensively studied by e.g. Sriperumbudur and Szabó (2015); Rudi and Rosasco (2017); Shahrampour and Tarokh (2018); Ding et al. (2020).

Most of prior studies rely on a pre-selected kernel; however, such selection requires prior information which may not be available. By contrast, utilizing a dictionary of multiple kernel in lieu of a pre-selected kernel provides more flexible approach to obtain more accurate function approximations as it can learn combination of kernels (Sonnenburg et al., 2006; Kloft et al., 2011). Multiple kernel learning successfully has been employed in many learning methods as well as practical applications including cross domain learning (Duan et al., 2012) and computer vision applications (Bucak et al., 2014). To utilize the merits of multi-kernel learning several algorithms have been emerged (see e.g. (Rakotomamonjy et al., 2008; Cortes et al., 2009; Gönen and Alpaydin, 2011)) which exhibit well-documented advantages compared to their single kernel learning counterparts. However, the mentioned algorithms are suitable to apply to batch kernel learning cases and are less efficient or even intractable when it comes to performing kernel learning in an online fashion. In order to make multiple kernel learning amenable for online function approximation, several algorithms have been proposed in the literature (see e.g. (Hoi et al., 2013; Sahoo et al., 2014)). However, the aforementioned algorithms suffer from the curse of dimensionality and are not scalable to deal with large volume of data. Enabled by the random feature approximation by Rahimi and Recht (2007), scalable online multi-kernel learning algorithms have been developed by Sahoo et al. (2019); Shen et al. (2019); Ghari and Shen (2020). In particular, the aforementioned algorithms perform function approximation by learning the linear combination of random feature kernel approximations.

One of the most important challenges of MKL is the proper selection of kernels in the dictionary, which influences both computational complexity and accuracy of the function approximation significantly. However, selecting an appropriate kernel dictionary requires prior information. When such information is not available, one solution is to include a large number of kernels in the dictionary. In this case, employing all kernels in the dictionary may not be a feasible choice. Data-driven selection of subset of kernels in a given dictionary can alleviate the computational complexity. Furthermore, data-driven subset selection of kernels can enhance the accuracy of function approximation by pruning irrelevant kernels. The goal of the present paper is to select a subset of kernels in a given dictionary at each time instant in order to alleviate the computational complexity and improve function approximation accuracy. To this end, our proposed algorithms construct a graph whose vertices represent kernels such that a subset of kernels is selected based on the structure of the graph. In this case, function approximations given by the chosen subset of kernels can be viewed as feedback collected from a graph which is called *feedback graph*. Relative to existing online multi-kernel learning approaches, our novelty can be summarized as follows:

c1) Different from existing works which employ all kernels in the dictionary, while only learning the combination coefficients, our proposed algorithms only use a subset of kernels at each time instant according to a feedback graph.

c2) An adaptive and disciplined framework is developed to construct a feedback graph at each time instant according to the loss incurred by kernel-based approximants. A novel OMKL algorithm is proposed to select kernels according to the **g**raph-structured **f**eedback (OMKL-GF) which achieves sublinear regret.

c3) Construction of the feedback graph at each time instant increases the computational burden of multi-kernel learning. To address this issue, a similarity feedback graph is constructed based on the similarity among kernels, which does not require observing the data samples beforehand. The resulting algorithm is called Online Multi-Kernel Learning with Similarity-based Feedback Graph (OMKL-SFG). It is proved that the proposed OMKL-SFG achieves a sub-linear regret.

c4) A novel algorithm called OMKL-SFG-R is proposed to adaptively refine the structure of the similarity-based feedback graph 'on the fly.' It is proved that the OMKL-SFG-R enjoys the sublinear regret tighter than OMKL-SFG and OMKL-GF.

c5) Experiments on real datasets showcase the effectiveness of our proposed algorithms in comparison with other state-of-the-art OMKL baselines.

The remainder of this paper is organized as follows. Section 2 discusses preliminaries of random-feature based multi-kernel learning. Section 3 presents the proposed OMKL-GF algorithm and its regret analysis. Furthermore, section 4 presents the OMKL-SFG and OMKL-SFG-R algorithms along with their theoretical performance in terms of regret. Experimental results are provided in section 5 to study performance of MKL algorithms on several real datasets. Finally, section 6 concludes the paper.

2. Preliminaries

Given samples $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_T, y_T)$, with $\boldsymbol{x}_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$, the function approximation problem can be written as the following optimization problem

$$\min_{f \in \mathbb{H}} \frac{1}{T} \sum_{t=1}^{T} \left(\mathcal{C}(f(\boldsymbol{x}_t), y_t) + \lambda \Omega(\|f\|_{\mathbb{H}}^2) \right) \tag{1}$$

where $\mathcal{C}(\cdot, \cdot)$ denotes the cost function, which is specified according to the learning task. For example, in regression task $\mathcal{C}(\cdot, \cdot)$ can be least square function. In (1), λ denotes the regularization coefficient and $\Omega(\cdot)$ represents a non-decreasing function, which is used to prevent over-fitting and control model complexity.

2.1 Function Approximation with Reproducing Hilbert Kernel Space

Let $\kappa(\boldsymbol{x}, \boldsymbol{x}_t) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ represent a symmetric positive definite kernel function which measures the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}_t$. In the context of kernel based learning, it is assumed that the sought $f(\cdot)$ belongs to the reproducing Hilbert kernel space (RHKS) $\mathbb{H} :=$

$\{f|f(\boldsymbol{x}) = \sum_{t=1}^{\infty} \alpha_t \kappa(\boldsymbol{x}, \boldsymbol{x}_t)\}$. A kernel is reproducing if it satisfies $\langle \kappa(\boldsymbol{x}, \boldsymbol{x}_t), \kappa(\boldsymbol{x}, \boldsymbol{x}_{t'}) \rangle_{\mathbb{H}} = \kappa(\boldsymbol{x}_t, \boldsymbol{x}_{t'})$ where $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ denotes vector inner product in Hilbert space, with the RKHS norm defined as $\|f\|_{\mathbb{H}}^2 := \sum_t \sum_{t'} \alpha_t \alpha_{t'} \kappa(\boldsymbol{x}_t, \boldsymbol{x}_{t'})$. The representer theorem states that the optimal solution of (1) can be expressed as follows given finite data samples (Wahba, 1990)

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

where $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_T]^\top$ denotes the vector of unknown coefficients to be estimated, and $\boldsymbol{\kappa}(\boldsymbol{x}) := [\kappa(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \kappa(\boldsymbol{x}, \boldsymbol{x}_T)]^\top$. Furthermore, it can be observed that the dimension of $\boldsymbol{\alpha}$ increases with the number of data samples T. This is known as 'curse of dimensionality' (Wahba, 1990) and arises as a major challenge for solving (1) in an online fashion.

2.2 Random Fourier Feature Approximation

One way to cope with the increasing number of variables to be estimated is to employ random feature (RF) approximation (Rahimi and Recht, 2007). As in Rahimi and Recht (2007), we will approximate $\kappa(\cdot)$ in (2) using shift-invariant kernels which satisfy $\kappa(\boldsymbol{x}_t, \boldsymbol{x}_{t'}) = \kappa(\boldsymbol{x}_t - \boldsymbol{x}_{t'})$. However, relying on a pre-selected kernel often requires prior information that may not be available. To cope with this, multi-kernel learning can be exploited which learns the kernel as a combination of a sufficiently rich dictionary of kernels $\{\kappa_i\}_{i=1}^N$. The kernel combination is itself a kernel (Scholkopf and Smola, 2001). Let $\kappa_i(\boldsymbol{x}_t - \boldsymbol{x}_{t'})$ be the i-th kernel in the dictionary of N absolutely integrable kernels. In this case, its Fourier transform $\pi_{\kappa_i}(\boldsymbol{\psi})$ exists and can be viewed as probability density function (PDF) if the kernel is normalized such that $\kappa_i(\mathbf{0}) = 1$. Specifically, it can be written as

$$\kappa_i(\boldsymbol{x}_t - \boldsymbol{x}_{t'}) = \int \pi_{\kappa_i}(\boldsymbol{\psi}) e^{j\boldsymbol{\psi}^\top (\boldsymbol{x}_t - \boldsymbol{x}_{t'})} d\boldsymbol{\psi} := \mathbb{E}_{\pi_{\kappa_i}(\boldsymbol{\psi})}[e^{j\boldsymbol{\psi}^\top (\boldsymbol{x}_t - \boldsymbol{x}_{t'})}]. \tag{3}$$

Let $\{\boldsymbol{\psi}_{i,j}\}_{j=1}^D$ be a set of vectors which are independently and identically distributed (i.i.d) samples from $\pi_{\kappa_i}(\boldsymbol{\psi})$. Hence, $\kappa_i(\boldsymbol{x}_t - \boldsymbol{x}_{t'})$ can be approximated by the ensemble mean $\hat{\kappa}_{i,c}(\boldsymbol{x}_t - \boldsymbol{x}_{t'}) := \frac{1}{D} \sum_{j=1}^D e^{j\boldsymbol{\psi}_{i,j}^\top (\boldsymbol{x}_t - \boldsymbol{x}_{t'})}$. Furthermore, the real part of $\hat{\kappa}_{i,c}(\boldsymbol{x}_t - \boldsymbol{x}_{t'})$ also constitutes an unbiased estimator of $\kappa_i(\boldsymbol{x}_t - \boldsymbol{x}_{t'})$ which can be written as $\hat{\kappa}_i(\boldsymbol{x}_t - \boldsymbol{x}_{t'}) = \mathbf{z}_i^\top(\boldsymbol{x}_t)\mathbf{z}_i(\boldsymbol{x}_{t'})$ (Rahimi and Recht, 2007), where

$$\mathbf{z}_i(\boldsymbol{x}_t) := \frac{1}{\sqrt{D}}[\sin(\boldsymbol{\psi}_{i,1}^\top \boldsymbol{x}_t), \cdots, \sin(\boldsymbol{\psi}_{i,D}^\top \boldsymbol{x}_t), \cos(\boldsymbol{\psi}_{i,1}^\top \boldsymbol{x}_t), \cdots, \cos(\boldsymbol{\psi}_{i,D}^\top \boldsymbol{x}_t)].$$

Replacing $\kappa_i(\boldsymbol{x}, \boldsymbol{x}_t)$ with $\hat{\kappa}_i(\boldsymbol{x} - \boldsymbol{x}_t)$, $\hat{f}(\boldsymbol{x})$ in (2) can be approximated as

$$\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t \hat{\kappa}_i(\boldsymbol{x} - \boldsymbol{x}_t) = \sum_{t=1}^{T} \alpha_t \mathbf{z}_i^\top(\boldsymbol{x})\mathbf{z}_i(\boldsymbol{x}_t) = \boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{\theta}_i \in \mathbb{R}^{2D}$ is a vector whose dimension does not increase with the number of data samples. Therefore, utilizing RF approximation can make the function approximation problem amenable for online implementation. Furthermore, the loss of the i-th kernel can be calculated as

$$\mathcal{L}(\boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) = \mathcal{C}(\boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) + \lambda \Omega(\|\boldsymbol{\theta}_i\|^2). \tag{5}$$

2.3 Online MKL as Online Learning with Expert Advice

Online learning with expert advice studies the problem where a learner performs the online learning task by interacting with a set of experts, see e.g. (Cesa-Bianchi and Lugosi, 2006)). At each time instant, the learner observes the advice given by experts, and then utilize the received advice to make a decision in real time (Auer et al., 2003; Hazan, 2016; Mannor and Shamir, 2011). In multi-kernel learning, each kernel can be viewed as an expert. Specifically, the approximation obtained by the i-th kernel, can be viewed as the advice given by $\kappa_i(\cdot)$. In particular, when multiple kernels are employed, function approximation can be performed by functions in the form $f(\boldsymbol{x}) = \sum_{i=1}^{N} \bar{w}_i f_i(\boldsymbol{x})$ where $\sum_{i=1}^{N} \bar{w}_i = 1$ (Scholkopf and Smola, 2001). Also, $f_i(\boldsymbol{x}) \in \mathbb{H}_i$ where $\mathbb{H}_i$ is an RKHS induced by the kernel κ_i. Replacing $f_i(\boldsymbol{x})$ with $\hat{f}_{\text{RF},i}(\boldsymbol{x})$, the function $f(\boldsymbol{x})$ can be approximated as

$$\hat{f}_{\text{RF}}(\boldsymbol{x}) = \sum_{i=1}^{N} \bar{w}_i \hat{f}_{\text{RF},i}(\boldsymbol{x}), \sum_{i=1}^{N} \bar{w}_i = 1 \tag{6}$$

where the approximation $\hat{f}_{\text{RF}}(\boldsymbol{x})$ is a linear combination of approximations (advice) given by kernels in the dictionary. When all kernels are involved in function approximation at every time instants, the multi-kernel learning problem with RF approximation can be formulated as

$$\min_{\{\bar{w}_i, \boldsymbol{\theta}_i\}} \sum_{t=1}^{T} \left(\mathcal{C} \left(\sum_{i=1}^{N} \bar{w}_i \boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t \right) + \lambda \Omega(\|\boldsymbol{\theta}_i\|^2) \right) \tag{7a}$$

$$\text{s.t.} \sum_{i=1}^{N} \bar{w}_i = 1, \quad \bar{w}_i \geq 0, \quad \forall 1 \leq t \leq T. \tag{7b}$$

In online MKL where data samples come sequentially, the optimization problem cannot be solved in batch. Online convex optimization methods can be utilized to update $\{\bar{w}_i\}_{i=1}^N$, $\{\boldsymbol{\theta}_i\}_{i=1}^N$ upon receiving new datum $\boldsymbol{x}_t$ at each time instant t (Sahoo et al., 2019; Shen et al., 2019). Let $\bar{w}_{i,t}$ and $\boldsymbol{\theta}_{i,t}$ denote the values of $\bar{w}_i$ and $\boldsymbol{\theta}_i$ at time t. Upon receiving new datum $\boldsymbol{x}_t$ and computing the loss $\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)$, using the online gradient descent, $\boldsymbol{\theta}_{i,t}$ can be updated as

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) \tag{8}$$

where η is the learning rate. Furthermore, using multiplicative update, the value of $\bar{w}_{i,t}$ can be updated as

$$w_{i,t+1} = w_{i,t} \exp\left(-\eta \mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) \right) \tag{9a}$$

$$\bar{w}_{i,t+1} = \frac{w_{i,t+1}}{\sum_{j=1}^{N} w_{j,t+1}}. \tag{9b}$$

Employing the update rules in (8) and (9) $\bar{w}_{i,t}$ and $\boldsymbol{\theta}_{i,t}$ can be updated in an online fashion without storing data in batch.

2.4 Assumptions

In order to analyze the proposed algorithms, we apply stochastic regret (Hazan, 2016) to measure the difference between expected cumulative loss of the proposed algorithms and the best function approximant in the hindsight. Let $f^*(\cdot)$ denote the best function approximant in hindsight which can be obtained as

$$f^*(\cdot) \in \arg \min_{f_i^*, i\in\{1,\ldots,N\}} \sum_{t=1}^{T} \mathcal{L}(f_i^*(\boldsymbol{x}_t), y_t) \tag{10a}$$

$$f_i^*(\cdot) \in \arg \min_{f\in\mathbb{H}_i} \sum_{t=1}^{T} \mathcal{L}(f(\boldsymbol{x}_t), y_t). \tag{10b}$$

Hence, the stochastic regret is defined as

$$\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(f^*(\boldsymbol{x}_t), y_t) \tag{11}$$

where $\mathbb{E}_t[\cdot]$ denotes the expected value at time instant t given the loss observations in prior times. Furthermore, the performance of proposed algorithms is analyzed under the following assumptions:

(as1) The loss function $\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)$ is convex with respect to $\boldsymbol{\theta}_{i,t}$ at each time instant t.

(as2) For $\boldsymbol{\theta}$ in a bounded set $\mathbb{T}$ which satisfies $\|\boldsymbol{\theta}\| \leq C_{\boldsymbol{\theta}}$ the loss and its gradient are bounded as $\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) \in [0, 1]$ and $\|\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)\| \leq L$, respectively.

(as3) Kernels $\{\kappa_i\}_{i=1}^N$ are shift-invariant (i.e. $\kappa_i(\boldsymbol{x}, \boldsymbol{x}') = \kappa_i(\boldsymbol{x} - \boldsymbol{x}')$, $\forall i$: $i = 1, \ldots, N$), standardized and bounded. Each datum $\|\boldsymbol{x}_t\| \leq 1$.

Note that (as1) can be satisfied by many convex loss functions such as least squares loss and logistic loss. Moreover, (as2) states that the losses are bounded and L-Lipschitz continuous. (as3) states that kernels are assumed to be shift-invariant, standardized and bounded, meaning that $|\kappa_i(\boldsymbol{x})| \leq 1$, $\forall i, \forall \boldsymbol{x}$. In what follows, we introduce a general *graph-aided* OMKL framework, where only a subset of kernels in the dictionary are chosen at each time instant.

3. Online Multi-Kernel Learning with Bipartite Feedback Graph

The present section introduces an OMKL approach which selects a subset of kernels using a bipartite graph constructed adaptively at each time instant based on the observed losses.

3.1 Data-driven Graph-based Kernel Selection

Instead of combining the entire dictionary of the kernels, in the present paper, we will consider combining a subset of kernels $\{\kappa_i(\cdot), i \in \mathbb{S}_t\}$ at time instant t instead, where $\mathbb{S}_t$ is the index set of the chosen subset of kernels at time instant t. Hence, the original function approximation problem boils down to

$$\min_{\{\bar{w}_{i,t},\boldsymbol{\theta}_i\}} \sum_{t=1}^{T} \left(\mathcal{C}\left(\sum_{i\in\mathbb{S}_t} \bar{w}_{i,t}\boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t \right) + \lambda\Omega(\|\boldsymbol{\theta}_i\|^2) \right) \tag{12a}$$

$$\text{s.t.} \sum_{i\in\mathbb{S}_t} \bar{w}_{i,t} = 1, \quad \bar{w}_{i,t} \geq 0, \quad \forall 1 \leq t \leq T. \tag{12b}$$

Upon defining the normalized weights $\bar{w}_{i,t} = \frac{w_{i,t}}{\sum_{j\in\mathbb{S}_t} w_{j,t}}$, (12) can be re-written as

$$\min_{\{w_{i,t}\},\{\boldsymbol{\theta}_i\}} \sum_{t=1}^{T} \mathcal{L}\left(\sum_{i\in\mathbb{S}_t} \frac{w_{i,t}}{\sum_{j\in\mathbb{S}_t} w_{j,t}} \boldsymbol{\theta}_i^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t \right) \tag{13a}$$

$$\text{s.t.} w_{i,t} \geq 0, \quad \forall 1 \leq i \leq N, \quad \forall 1 \leq t \leq T. \tag{13b}$$

However, (13) assumes that $\{\mathbb{S}_t\}_{t=1}^T$ are preselected sets. In this section, we study data-driven scheme which can adaptively select a subset of kernels ‘on the fly’ upon receiving new data samples. In order to adaptively choose the subset of kernels, the present section models the pruned kernel combination as feedback collected from a graph, that is constructed in an online fashion. By doing this, the proposed approach trims irrelevant kernels in the dictionary to both improve the function approximation accuracy and reduce the computational complexity of MKL.

Consider a time varying bipartite graph (Asratian et al., 1998) $\mathcal{B}_t$ at time t, which consists of two sets of nodes: N kernel nodes $\{v_{k,1}, ..., v_{k,N}\}$ and J selective nodes $\{v_{s,1}, ..., v_{s,J}\}$ where $v_{k,i}$ and $v_{s,j}$ are the i-th kernel node and j-th selective node, respectively. And the edges of the graph represents the association between the kernel nodes and the selective nodes. Specifically, an edge between $v_{k,i}$ and $v_{s,j}$ exists at time t if the i-th kernel is assigned to j-th selective node. The construction of the graph will be discussed in Section 3.2 .

At each time instant, one of the selective nodes $v_{s,j}$ is chosen, and the subset of kernel nodes connected to $v_{s,j}$ will be used for the instantaneous function approximation at time t, [c.f. (13)]. Then, the loss $\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)$ is observed for every kernel in the chosen subset and $\boldsymbol{\theta}_{i,t}$ is updated as

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} - \eta \frac{\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}} \mathbf{1}_{i\in\mathbb{S}_t}, \tag{14}$$

where $q_{i,t}$ is the probability that the loss of associated kernel is observed and $\mathbf{1}_{i\in\mathbb{S}_t}$ denotes the indicator function such that $\mathbf{1}_{i\in\mathbb{S}_t} = 1$ if $i \in \mathbb{S}_t$ and $\mathbf{1}_{i\in\mathbb{S}_t} = 0$ otherwise. The value of $q_{i,t}$ depends on how the bipartite graph is generated. Upon receiving a new datum $\boldsymbol{x}_t$, the value of the weight w_i is updated ‘on the fly’. Let $w_{i,t}$ denote the weighting coefficient w_i at time instant t. We leverage multiplicative update for weights $w_{i,t}$ as

$$w_{i,t+1} = w_{i,t} \exp(-\eta\ell_{i,t}) \tag{15}$$

where $\ell_{i,t}$ denotes the importance sampling loss estimates (Alon et al., 2017) associated with the i-th kernel as follows

$$\ell_{i,t} = \frac{\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)}{q_{i,t}} \mathbf{1}_{i\in\mathbb{S}_t} \tag{16}$$

Algorithm 1 Data Driven (Bipartite) Graph-based Kernel Selection

```
**Input:** Shift-invariant kernels $\kappa_i$, $i = 1, \ldots, N$, step size $\eta > 0$, weights $\{w_{i,t}\}_{i=1}^N$, $\{u_{j,t}\}_{j=1}^J$, bipartite graph $\mathcal{B}_t$ and datum $\boldsymbol{x}_t$.
Set $u_{j,t} = \sum_{\forall i: v_{k,i} \rightarrow v_{s,j}} w_{i,t}$.
Obtain $p_{j,t}$ via (19).
Choose one selective node $v_{s,j}$ according to PMF $\boldsymbol{p}_t = (p_{1,t}, ..., p_{J,t})$.
Predict $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t) = \sum_{i \in \mathbb{S}_t} \frac{w_{i,t}}{\sum_{m \in \mathbb{S}_t} w_{m,t}} \hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t)$ with $\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t)$ in (4).
Obtain loss $\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t)$ for all $i \in \mathbb{S}_t$.
Update $\boldsymbol{\theta}_{i,t+1}$ via (14) for all $i \in \mathbb{S}_t$.
Update $w_{i,t+1}$ via (15).
**Output:** $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t)$, $\{w_{i,t+1}\}_{i=1}^N$, $\{\boldsymbol{\theta}_{i,t+1}\}_{i=1}^N$
```

which is the observed loss $\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)$ divided by the probability $q_{i,t}$. The function approximation can be obtained as

$$\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t) = \sum_{i \in \mathbb{S}_t} \frac{w_{i,t}}{\sum_{m \in \mathbb{S}_t} w_{m,t}} \hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t) \tag{17}$$

where $\hat{f}_{\mathrm{RF},i}(\cdot)$ is defined in (4).

Then the selective nodes are assigned to weights $\{u_{j,t+1}\}$ according to the the kernel nodes' weights $\{w_{i,t+1}\}$. Indeed, each selective node's weight $\{u_{j,t+1}\}$ is the total summation of weights of kernel nodes which are connected to this selective node. Specifically the weight of $v_{s,j}$ is obtained via

$$u_{j,t+1} = \sum_{\forall i: v_{k,i} \rightarrow v_{s,j}} w_{i,t+1}. \tag{18}$$

Note that the weights of the selective nodes are determined by its adjacent kernel nodes, which indicates the reliability of the corresponding kernel-based function estimate. Hence, the probability according to which a selective node is chosen in the next time instant can be updated as

$$p_{j,t+1} = (1 - \eta_e) \frac{u_{j,t+1}}{U_{t+1}} + \frac{\eta_e}{J} \tag{19}$$

where $U_{t+1} := \sum_{j=1}^J u_{j,t+1}$, and $0 < \eta_e \leq 1$ is the exploration rate. The term $\frac{\eta_e}{J}$ is introduced to tradeoff between exploitation and exploration. The first term on the right hand side of (19) implies choosing a selective node with larger weight $u_{j,t+1}$ with higher probability. And the term $\frac{\eta_e}{J}$ is used to to promote exploration. Algorithm 1 summarizes the data driven kernel selection scheme presented in this section.

To sum up, each kernel is viewed as an expert and at each time instant a subset of function approximations provided by these experts is combined. In this regard, the RF approximation $\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t)$ can be viewed as the feedback provided by i-th kernel node, and the proposed framework models the pruned kernel combination as feedback collected from a graph, where the feedback are combined only if the corresponding kernel node is connected

Algorithm 2 Generating Bipartite Feedback Graph

```
Input: Shift-invariant kernels $\kappa_i$, weighting coefficient $w_{i,t}$, $i = 1, \ldots, N$, exploration coefficient $\eta_e$ and the maximum degree of each selective node $M$.
Initialize: Sub-adjacency matrix $\boldsymbol{A}_{t+1} = \mathbf{0}_{N \times J}$.
for $j = 1, ..., J$ do
    for $i = 1, ..., N$ do
        Set $\pi_{ij,t+1} = (1 - {\eta_e}^j) \frac{w_{i,t+1}}{\sum_{i=1}^{N} w_{i,t+1}} + \frac{{\eta_e}^j}{N}$.
    end for
    for $k = 1, ..., M$ do
        Choose one of nodes $v_{k,i}$ drawn according to the probability mass function (PMF) $\boldsymbol{\pi}_{j,t+1} = (\pi_{1j,t+1}, ..., \pi_{Nj,t+1})$.
        Set $\boldsymbol{A}_{t+1}(i, j) = 1$.
    end for
end for
Output: Bipartite feedback graph $\mathcal{B}_{t+1}$ with adjacency matrix $\boldsymbol{A}_{t+1}$
```

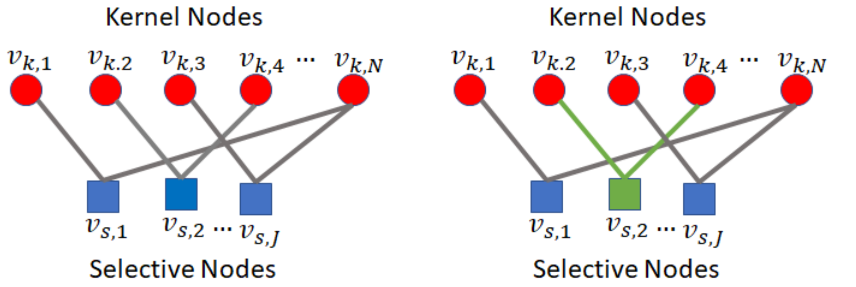
to the chosen selective node. By doing this, the proposed approach trims irrelevant kernels in the dictionary to both improve the function approximation accuracy and reduce the computational complexity of MKL. The graph construction approach will be proposed in the ensuing subsection.

3.2 Online Bipartite Feedback Graph Construction

Construction of the time varying graph is of utmost importance, as it affects both function approximation accuracy and computational complexity. In this regard, a graph is successful if it can provide a subset of kernels which results in smallest possible loss. Indeed, considering computational complexity, the graph should provide a limited number of kernels which obtain minimum loss. To this end, we aim to propose a generating method for graph. Specifically, using the weights $\{w_{i,t+1}\}_{i=1}^{N}$ obtained via (15), the structure of the graph is reconstructed in a stochastic manner stated below to be leveraged in the next time instant.

Increasing the number of kernel nodes connected to $v_{s,j}$, increases the computational complexity of performing function approximation by choosing $v_{s,j}$. Therefore, the graph generation algorithm should be designed to limit the number of kernel nodes connected to each selective node. Let M denote the maximum number of kernel nodes connected to each selective node. The procedure to generate the graph $\mathcal{B}_{t+1}$ is presented in Algorithm 2. Let $\boldsymbol{A}_{t+1}$ represent the $N \times J$ sub-adjacency matrix between two disjoint subsets $\{v_{s,1}, ..., v_{s,J}\}$ and $\{v_{k,1}, ..., v_{k,N}\}$. Notation $\boldsymbol{A}_{t+1}(i, j)$ represents the element in i-th row and j-th column of the sub-adjacency matrix $\boldsymbol{A}_{t+1}$ and it is equal to 1 if $v_{k,i}$ is connected to $v_{s,j}$, and 0 otherwise.

Each selective node $v_{s,j}$ draws kernel nodes $v_{k,i}$ in M independent trials and in each trial selective node draws only one kernel node. We put more weight on kernels which obtain less loss in a sense that the probability that selective node $v_{s,j}$ draws the kernel node $v_{k,i}$ in a



(a) A bipartite feedback graph consists of J selective nodes and N kernel nodes.

(b) The chosen selective node ($v_{s,2}$ as an example) and edges associated with it are highlighted.

Figure 1: A bipartite feedback graph generated by Algorithm 2.

trial at time $t+1$ is

$$\pi_{ij,t+1} = (1-{\eta_e}^j)\frac{w_{i,t+1}}{\sum_{k=1}^N w_{k,t+1}} + \frac{{\eta_e}^j}{N} \tag{20}$$

Note that the first term in (20) discriminates between kernels based on their weights which is determined by their loss in function approximation [c.f. (15)]. Furthermore, the second term allows exploration over all kernel nodes. Specially, the selective node $v_{s,j}$ draws kernel nodes according to uniform distribution if $\eta_e = 1$. Furthermore, note that ${\eta_e}^j$ is a non-increasing function of j for $0 < \eta_e \leq 1$. The selective node $v_{s,1}$ puts more weight on exploration in comparison with others while $v_{s,J}$ considers more exploitation than all the other selective nodes. Therefore, the selective nodes entail different level of exploration and exploitation.

Based on the definition of $\pi_{ij,t+1}$ in (20), the probability that the i-th kernel node is connected to $v_{s,j}$ is $1-(1-\pi_{ij,t+1})^M$, where $(1-\pi_{ij,t+1})^M$ is the probability that the i-th kernel node is chosen by $v_{s,j}$ in none of M trials. Therefore, the probability of observing the loss of the i-th kernel at time $t+1$ is given by

$$q_{i,t+1} = \sum_{j=1}^{J} p_{j,t+1}\left(1-(1-\pi_{ij,t+1})^M\right) \tag{21}$$

for $1 \leq i \leq N$. The value of $q_{i,t+1}$ is computed and used for importance sampling loss estimate in (16). The graph generation framework is summarized in Algorithm 2. And Figure 1 illustrates an example of the constructed bipartite feedback graph.

At each time instant t, the bipartite graph $\mathcal{B}_t$ is used for choosing a selective node, and henceforth subset of the kernels. Then the weights of the selected kernels are updated according to the loss [c.f.(14) and (15)]. And the graph can be constructed using Algorithm 2, based on which, the function approximation can be carried out by choosing one of the selective nodes which leads to selecting a subset of kernels. Our proposed online multi-kernel learning with graph-structured feedback (OMKL-GF) is summarized in Algorithm 3.

Computational Complexity. At time instant t, OMKL-GF needs to store a real $2D$ random feature vector in addition to a weighting vector for each kernel in conjunction with

Algorithm 3 OMKL with (Bipartite) Graph Feedback (OMKL-GF)

```
**Input:** Shift-invariant kernels $\kappa_i$, $i = 1, \ldots, N$, step size $\eta > 0$ and the number of random features $D$.
**Initialize:** $\boldsymbol{\theta}_{i,1} = \mathbf{0}$, $w_{i,1} = 1$, $i = 1, ..., N$, generate $\mathcal{B}_1$ using Algorithm 2 given $w_{i,1}$, $\forall i$
**for** $t = 1, ..., T$ **do**
    Receive one datum $\boldsymbol{x}_t$.
    Obtain $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t)$, $\{w_{i,t+1}\}_{i=1}^N$, $\{\boldsymbol{\theta}_{i,t+1}\}_{i=1}^N$ using Algorithm 1 given $\mathcal{B}_t$, $\{w_{i,t}\}_{i=1}^N$, $\{\boldsymbol{\theta}_{i,t}\}_{i=1}^N$.
    Generate $\mathcal{B}_{t+1}$ using Algorithm 2 with $\{w_{i,t+1}\}_{i=1}^N$.
**end for**
```

a weighting vector for each selective node. As the number of kernels is in general larger than the number of selective nodes, the required memory is of order $\mathcal{O}(dDN)$. The per-iteration complexity of our OMKL-GF (e.g. calculating inner products) is $\mathcal{O}(dDM + JN)$. In comparison, the per-iteration complexity of OMKR developed by Sahoo et al. (2014) is $\mathcal{O}(tdN)$, while more contemporary online RF-based OMKL approaches proposed by Shen et al. (2019); Sahoo et al. (2019) both have per-iteration complexity $\mathcal{O}(dDN)$. Hence, OMKL-GF can significantly reduce the per iteration complexity especially when $J \leq M << N$.

3.3 Regret Analysis

This subsection presents the regret analysis of OMKL-GF. In order to analyze the regret for OMKL-GF, we first establish an intermediate result in the following lemma.

Lemma 1 *The regret of the proposed OMKL-GF under (as1) and (as2) with respect to a preselected kernel κ_i where $\mathcal{F}_i = \{\hat{f}_i | \hat{f}_i(\boldsymbol{x}) = \boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}), \forall \boldsymbol{\theta} \in \mathbb{R}^{2D}\}$ satisfies the following bound*

$$\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t), y_t) \\ < \frac{\ln N}{\eta} + \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)} + \frac{\eta L^2 NJT}{2\eta_e^2} \tag{22}$$

where $\boldsymbol{\theta}_i^*$ is associated with the best RF function approximant $\hat{f}_i^*(\boldsymbol{x}_t) = \boldsymbol{\theta}_i^{*\top} \mathbf{z}_i(\boldsymbol{x}_t)$ and the expectation at time t is taken with respect to PMFs $\boldsymbol{p}_t$ and $\boldsymbol{\pi}_{j,t}$ in (19) *and* (20), *respectively.*

Proof see Appendix B. ■

The next theorem further characterizes the difference between the loss of OMKL-GF relative to the best functional estimator in the RKHS.

Theorem 2 *The following bound holds with probability at least $1 - 2^8(\frac{\sigma_i}{\epsilon})^2 \exp(-\frac{D\epsilon^2}{4d+8})$ under (as1)–(as3) for $\epsilon > 0$ and with f_i^* belonging to RKHS $\mathbb{H}_i$ as in* (10b)

$$\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t), y_t)] - \min_{i \in \{1,\ldots,N\}} \sum_{t=1}^{T} \mathcal{L}(f_i^*(\boldsymbol{x}_t), y_t) \\ < \frac{\ln N}{\eta} + \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \eta_e JT + \epsilon LTC + \frac{\eta NT}{2(1-\eta_e)} + \frac{\eta L^2 NJT}{2\eta_e^2} \tag{23}$$

where C is a constant, and σ_i^2 is the second order moment of the RF vector norm which can be defined as $\sigma_i^2 := \mathbb{E}_{\pi_{\kappa_i}(\boldsymbol{\psi})}[\|\boldsymbol{\psi}\|^2]$. The expectation at time t is taken with respect to the randomness in $\boldsymbol{p}_t$ and $\boldsymbol{\pi}_{j,t}$ defined in (19) *and* (20)*, respectively.*

Proof see Appendix C. ■

According to Theorem 2, by setting

$$\eta = \mathcal{O}\left(\sqrt{\frac{\ln N}{N}}T^{-\frac{3}{4}}\right), \eta_e = \mathcal{O}\left(N^{\frac{1}{6}}T^{-\frac{1}{4}}\right), J = \mathcal{O}\left(N^{\frac{1}{3}}\right), \epsilon = \mathcal{O}\left(T^{-\frac{1}{4}}\right) \tag{24}$$

in (23), the stochastic static regret in (11) satisfies $\mathcal{O}\left(\sqrt{N\ln N}T^{\frac{3}{4}}\right)$. Thus, by selecting appropriate parameters, our proposed OMKL-GF achieves sublinear regret in expectation with respect to the best static function approximant in (11). Note that while proper settings of ϵ and η relies on the knowledge of T, such information may not be necessary, via employing, e.g., doubling trick (Cesa-Bianchi and Lugosi, 2006). Considering (23), the probability $1 - 2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$ is an increasing function of D such that for a fixed ϵ, always there are some values for D which result in positive probability. Furthermore, (23) shows that by setting $\epsilon = \mathcal{O}(T^{-\frac{1}{4}})$ and $D = \mathcal{O}(\sqrt{T}\ln T)$, the sublinear regret of $\mathcal{O}(\sqrt{N\ln N}T^{\frac{3}{4}})$ can be obtained with high probability of $1 - \mathcal{O}(\frac{1}{\sqrt{T}})$.

The computational complexity of kernel learning algorithms play an important role in their applicability. Bipartite feedback graph construction at each time instant increases the computational complexity of OMKL-GF. To further alleviate the computational burden of multi-kernel learning the ensuing section investigates the problem of choosing a subset of kernels using a feedback graph while the graph is not constructed at every time instant.

4. Online Multi-Kernel Learning with Similarity-based Feedback Graph

Note that OMKL-GF is a data-driven kernel selection scheme where a bipartite feedback graph is constructed at every time instant. However, online feedback graph construction increases the computational complexity of OMKL-GF. This section proposes a novel algorithmic framework to construct the feedback graph in an offline fashion such that the proposed algorithms do not need to construct the feedback graph at every time instant. Moreover, the bipartite feedback graph $\mathcal{B}_t$ constructed by Algorithm 2 do not exploit the relationship among kernels. Hence, in this section, the similarity among kernels is measured which will facilitate constructing the feedback graph in an offline fashion in such a way that at each time instant a subset of dissimilar kernels are chosen to avoid unnecessary computation. The present section first introduces a disciplined way to construct feedback graph based on kernel similarities in an offline fashion. Based on the constructed feedback graph, a novel online MKL algorithm (called OMKL-SFG) is developed to select a subset of kernels which is proved to obtain sub-linear regret. Furthermore, to obtain tighter regret bound a modification of OMKL-SFG algorithm (called OMKL-SFG-R) is proposed which refines the structure of the feedback graph to choose a subset of kernels. OMKL-SFG-R entails more computation than OMKL-SFG.

4.1 Offline Similarity-based Feedback Graph Construction

The similarity between two shift invariant kernels κ_i and κ_j is measured through divergence between κ_i and κ_j. As κ_i and κ_j has smaller divergence, they are considered to be more similar. The present paper measures the divergence between a pair of kernels using the Bregman divergence.

Let Ω be a d-dimensional convex set. Bregman divergence defined for a strictly convex and differentiable function $F(\cdot) : \Omega \rightarrow \mathbb{R}$ as (see, e.g. (Bregman, 1967; Banerjee et al., 2005))

$$B_F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = F(\boldsymbol{\omega}_1) - F(\boldsymbol{\omega}_2) - \nabla F(\boldsymbol{\omega}_2)^\top (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2). \tag{25}$$

Based on Bregman divergence, the divergence between two shift invariant kernels κ_i and κ_j can be measured through the function $\Delta(\kappa_i, \kappa_j)$ which is defined as

$$\Delta(\kappa_i, \kappa_j) = \int B_F\left(\kappa_i(\boldsymbol{\rho}), \kappa_j(\boldsymbol{\rho})\right) d\boldsymbol{\rho} \tag{26}$$

where $\boldsymbol{\rho} \in \mathbb{R}^d$. As it can be inferred from (26), $\Delta(\kappa_i, \kappa_j)$ measures the divergence between two kernels κ_i and κ_j using the aggregation of Bregman divergence on every point $\boldsymbol{\rho}$ in the input space. As $\Delta(\kappa_i, \kappa_j)$ decreases, kernels κ_i and κ_j are considered to be more similar. Note that instead of defining the divergence as in (26), one can define the divergence function $\Delta(\kappa_i, \kappa_j)$ as the expected Bregman divergence over points $\boldsymbol{\rho}$. However, taking the expectation requires knowing the distribution of input data samples which may not be available priori. Furthermore, the distribution of input space may change over time and as a result calculating the expected value of Bregman divergence in an offline fashion over the input space may not be feasible. Moreover, squared Euclidean divergence $B_F\left(\kappa_i(\boldsymbol{\rho}), \kappa_j(\boldsymbol{\rho})\right) = \|\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})\|^2$ is generated by the function $F(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2$. Let $\Delta_S(\cdot, \cdot)$ denotes the function $\Delta(\cdot, \cdot)$ when the Bregman divergence is obtained by $F(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2$. In this case, we have

$$\Delta_S(\kappa_i, \kappa_j) = \int |\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})|^2 d\boldsymbol{\rho}. \tag{27}$$

The following Lemma states that the function $\Delta_S(\kappa_i, \kappa_j)$ exists for each pair of absolutely integrable kernels.

Lemma 3 *Under the assumption that kernels $\{\kappa_i\}_{i=1}^N$ are absolutely integrable, bounded and normalized such that $\kappa_i(\mathbf{0}) = 1$, $\forall i : 1 \leq i \leq N$, the function $\Delta_S(\kappa_i, \kappa_j)$ is bounded and exists for each pair of kernels $\kappa_i(\cdot)$ and $\kappa_j(\cdot)$.*

Proof Since kernels $\{\kappa_i(\boldsymbol{\rho})\}_{i=1}^N$ are assumed to be bounded as $0 \leq \kappa_i(\boldsymbol{\rho}) \leq 1$, $\forall \boldsymbol{\rho}, 1 \leq i \leq N$, it can be concluded that $|\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})|^2 \leq |\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})|$. Thus, it can be inferred that

$$\int |\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})|^2 d\boldsymbol{\rho} \leq \int |\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})| d\boldsymbol{\rho}. \tag{28}$$

Furthermore, based on the Triangular inequality, it can be written that

$$\int |\kappa_i(\boldsymbol{\rho}) - \kappa_j(\boldsymbol{\rho})| d\boldsymbol{\rho} \leq \int |\kappa_i(\boldsymbol{\rho})| d\boldsymbol{\rho} + \int |\kappa_j(\boldsymbol{\rho})| d\boldsymbol{\rho}. \tag{29}$$

Based on (28), (29) and the fact that kernels are absolutely integrable, we can conclude that the function $\Delta_S(\kappa_i, \kappa_j)$ is bounded and exists for each pair of kernels $\kappa_i(\cdot)$ and $\kappa_j(\cdot)$. ■

Furthermore, the following lemma states that the average difference between function approximations given by each pair of kernels is bounded above in accordance with the function $\Delta_S(\cdot, \cdot)$ defined in (27).

Lemma 4 *Let $C_m := \max_i \sum_{t=1}^T |\alpha_{i,t}|^2$ where $\{\alpha_{i,t}\}_{t=1}^T$ are weights for (2) associated with the i-th kernel $\kappa_i(\cdot)$. Also, let $\boldsymbol{x}$ is bounded as $\|\boldsymbol{x}\| \leq 1$ and kernels are absolutely integrable. Then, the average difference between function approximations given by $\kappa_i(\cdot)$ and $\kappa_j(\cdot)$ is bounded above as*

$$\frac{1}{\mathcal{U}_d} \int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x} \leq \frac{2C_m}{\mathcal{U}_d} \sum_{t=1}^{T} (\Delta_S(\kappa_i, \kappa_j) + 2\mathcal{U}_d) \tag{30}$$

where $\hat{f}_i(\boldsymbol{x})$ denotes the function approximation given by $\kappa_i(\cdot)$ as in (2) and $\mathcal{U}_d$ represents d-dimensional Euclidean unit norm ball volume.

Proof See Appendix D. ■

Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be a directed graph with vertex $v_i \in \mathcal{V}$ which represents the i-th kernel κ_i. In this case, there is a self-loop for each $v_i \in \mathcal{V}$ which means $(i, i) \in \mathcal{E}$. Furthermore, an edge from v_i to v_j is appended to $\mathcal{E}$ if

$$\frac{1}{|\mathbb{N}_i^{\text{out}}|} \sum_{m \in \mathbb{N}_i^{\text{out}}} \Delta(\kappa_m(\boldsymbol{\rho}), \kappa_j(\boldsymbol{\rho})) \geq \gamma_i \tag{31}$$

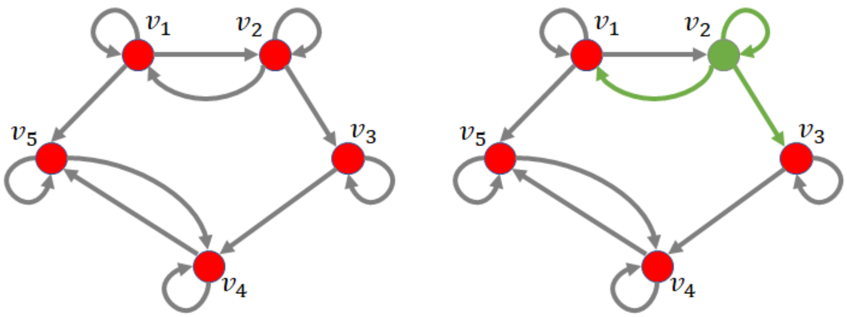
where γ_i is a threshold for v_i and $\mathbb{N}_i^{\text{out}}$ denote the current out-neighborhood set of v_i which means $j \in \mathbb{N}_i^{\text{out}}$ if $(i, j) \in \mathcal{E}$. Using the (31) to append edges to the graph $\mathcal{G}$, a vertex v_j associated with the kernel κ_j is added to the out-neighborhood set of v_i if it is dissimilar to the current out-neighbors of v_i. Therefore, the subset of vertices which are out-neighbors of v_i, are jointly dissimilar. Since a subset of function approximations associated with kernels will be chosen using the graph $\mathcal{G}$, the chosen subset of function approximations can be viewed as feedback collected from the graph $\mathcal{G}$ and as a result the graph $\mathcal{G}$ is called *feedback graph*. Specifically in order to restrict the number of out-neighbors for each node to M, the value of γ_i is obtained as

$$\gamma_i = \arg\max_{\gamma} \{\gamma \big| |\mathbb{N}_i^{\text{out}}| = M\}. \tag{32}$$

Note that M is a preselected parameter in the algorithm and increasing the value of M increases the connectivity of the feedback graph. At each time instant, one of the nodes are drawn and the function approximation is carried out using the combination of a subset of kernels which are out-neighbors of the chosen node. Therefore, increase in M can increase the exploration in the approximation task while it increases the computational complexity. The feedback graph construction procedure is summarized in Algorithm 4. It can be observed

Algorithm 4 Generating Similarity based Feedback Graph

```
**Input:** Shift-invariant kernels $\kappa_i$, $i = 1, ..., N$.
for $i = 1, ..., N$ do
    Append $(i, i)$ to $\mathcal{E}$.
    Obtain $\gamma_i$ via (32).
    Append $(i, j)$ to $\mathcal{E}$ if $\frac{1}{|\mathbb{N}_i^{\text{out}}|} \sum_{m \in \mathbb{N}_i^{\text{out}}} \Delta(\kappa_m(\boldsymbol{\rho}), \kappa_j(\boldsymbol{\rho})) \geq \gamma_i$.
end for
**Output:** Similarity-based Feedback Graph $\mathcal{G}$.
```



(a) A similarity-based feedback graph with 5 kernel nodes.

(b) As an example, the chosen node (v_2) and its out-neighbors are highlighted.

Figure 2: An example of similarity-based feedback graph generated by Algorithm 4.

from (26) that the function $\Delta(\kappa_i, \kappa_j)$ can be considered as a measure of divergence, and henceforth dissimilarity between kernels $\kappa_i(\cdot)$ and $\kappa_j(\cdot)$ without knowing data samples $\{\boldsymbol{x}_t\}_{t=1}^T$. This helps reduction of computational complexity of the function approximation since (dis)similarity among kernels in the dictionary can be measured offline before observing data samples and as a result the computation required to perform Algorithm 4 can be performed offline.

At each time instant, one of the vertices of the feedback graph is drawn according to a PMF as it will be explained in the next section. Then, the function approximation is performed using the kernels associated with out-neighbors of the chosen vertex. Therefore, based on the feedback graph construction in Algorithm 4, at each time instant a subset of dissimilar kernels is chosen to avoid unnecessary computation since it is expected that similar kernels provide comparatively similar approximations. See also Figure 2 for an example of similarity based feedback graph where the number of kernels is 5 and v_i represents a Gaussian kernel with bandwidth of 10^{i-3}. For each node v_i, γ_i is selected so that the number of out-neighbors of v_i is 3.

4.2 Kernel Selection with Offline Feedback Graph

The present section studies how to select a subset of kernels using the feedback graph $\mathcal{G}$ and prior observations of losses associated with kernels. Assume that each kernel is associated with a set of weights $\{w_{i,t}\}_{i=1}^N$ where $w_{i,t}$ is the weight associated with the i-th kernel κ_i. The weight $w_{i,t}$ indicates the accuracy of the function approximation given by the κ_i at time t and its value can be updated when more and more information is being revealed. Furthermore, a set of weights $\{u_{i,t}\}_{i=1}^{N}$ is assigned to $\mathcal{V}$ such that $u_{i,t}$ is the weight associated with $v_i \in \mathcal{V}$ at time instant t, which indicates the accuracy of function approximation when the node v_i is drawn. In order to choose a subset of kernels at time t, one of the vertices in $\mathcal{V}$ is drawn according to the PMF p_t

$$p_{i,t} = (1-\xi)\frac{u_{i,t}}{U_t} + \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}, i = 1, \ldots, N \tag{33}$$

where ξ is the exploration rate and $U_t := \sum_{i=1}^{N} u_{i,t}$. $\mathbb{D}$ represents the dominating set of $\mathcal{G}$, and $|\mathbb{D}|$ denotes the cardinality of $\mathbb{D}$. Let $\mathbb{S}_t$ denote the subset of kernel indices chosen at time t, and I_t denote the index of the kernel drawn according to the PMF p_t in (33). Therefore, $i \in \mathbb{S}_t$ if $i \in \mathbb{N}_{I_t}^{\text{out}}$, which means that the loss associated with the i-th kernel is calculated if the i-th kernel is an out-neighbor of the I_t-th node. In turn, the RF-based function approximation can be obtained as

$$\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t) = \sum_{i\in\mathbb{N}_{I_t}^{\text{out}}} \frac{w_{i,t}}{\sum_{j\in\mathbb{N}_{I_t}^{\text{out}}} w_{j,t}} \hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t). \tag{34}$$

Furthermore, the importance sampling loss estimate $\ell_{i,t}$ at time instant t is defined as

$$\ell_{i,t} = \frac{\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}\mathbf{1}_{\mathrm{i\in\mathbb{S}_t}}, i = 1, \ldots, N \tag{35}$$

where $q_{i,t}$ is the probability that $i \in \mathbb{S}_t$ and it can be computed as

$$q_{i,t} = \sum_{j\in\mathbb{N}_i^{\text{in}}} p_{j,t} \tag{36}$$

where $\mathbb{N}_i^{\text{in}}$ denote the in-neighborhood set of v_i which means $j \in \mathbb{N}_i^{\text{in}}$ if $(j, i) \in \mathcal{E}$. In addition, the importance sampling function approximation estimate $\hat{\ell}_{i,t}$ at time instant t associated with $v_i \in \mathcal{V}$ is defined as

$$\hat{\ell}_{i,t} = \frac{\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)}{p_{i,t}}\mathbf{1}_{\mathrm{I_t=i}}. \tag{37}$$

Using the importance sampling loss estimate in (36), $\boldsymbol{\theta}_{i,t}$ can be updated as

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} - \eta\nabla\ell_{i,t} = \boldsymbol{\theta}_{i,t} - \eta\frac{\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}\mathbf{1}_{\mathrm{i\in\mathbb{S}_t}}, \tag{38}$$

where η is the learning rate. Moreover, the multiplicative update is employed to update $w_{i,t}$ and $u_{i,t}$ based on importance sampling loss estimates in (35) and (37) as follows

$$w_{i,t+1} = w_{i,t}\exp(-\eta\ell_{i,t}), i = 1, \ldots, N \tag{39a}$$

$$u_{i,t+1} = u_{i,t}\exp(-\eta\hat{\ell}_{i,t}), i = 1, \ldots, N. \tag{39b}$$

Algorithm 5 OMKL with Similarity-based Feedback Graph (OMKL-SFG)

```
Input: Shift-invariant kernels $\kappa_i$, $i = 1, \ldots, N$, learning rate $\eta$, exploration rate $\xi$, the number of random features $D$.
Initialize: $\boldsymbol{\theta}_{i,1} = \mathbf{0}$, $w_{i,1} = 1$, $i = 1, \ldots, N$, Construct the feedback graph $\mathcal{G}$ via Algorithm 4.
for $t = 1, \ldots, T$ do
    Receive one datum $\boldsymbol{x}_t$.
    Draw one of nodes $v_i \in \mathcal{V}$ according to the PMF $p_t = (p_{1,t}, ..., p_{N,t})$ in (33).
    Predict $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t)$ via (34).
    Calculate loss $\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)$ for all $i \in \mathbb{S}_t$.
    Update $\boldsymbol{\theta}_{i,t+1}$ via (38).
    Update $w_{i,t+1}$ and $u_{i,t+1}$ via (39).
end for
```

The procedure to choose a subset of kernels at each time instant for function approximation is summarized in Algorithm 5. This algorithm is called OMKL-SFG which stands for Online Multi Kernel Learning with Similarity based Feedback Graph. Figure 2(b) illustrates the case when v_2 is drawn by the Algorithm 5 as an example. Then the function approximation is performed using kernels associated with out-neighbors of v_2, which are v_1, v_2 and v_3 highlighted in Figure 2(b).

The following Theorem presents the upper bound for cumulative stochastic regret of OMKL-SFG.

Theorem 5 *Under (as1) and (as2), let* $j^* = \arg\min_{\forall j: 1 \leq j \leq N} \sum_{t=1}^{T} \mathcal{L}(f_j^*(\boldsymbol{x}_t), y_t)$*. Then for any* $i \in \mathbb{N}_{j^*}^{in}$*, the stochastic regret of OMKL-SFG satisfies*

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(f^*(\boldsymbol{x}_t), y_t) \\ \leq &\frac{\ln N |\mathbb{N}_i^{out}|}{\eta} + \frac{(1+\epsilon)C^2}{2\eta} + \epsilon LTC + (\xi + \frac{\eta N}{2} - \frac{\eta \xi}{2})T + \frac{\eta}{2} \sum_{t=1}^{T} (\frac{1}{\bar{q}_{i,t}} + \frac{L^2}{q_{j^*,t}}) \end{aligned} \tag{40}$$

with probability at least $1 - 2^8 (\frac{\sigma_{j^*}}{\epsilon})^2 \exp(-\frac{D\epsilon^2}{4d+8})$ *under (as1)-(as3) for any* $\epsilon > 0$*. Furthermore,* $\frac{1}{\bar{q}_{i,t}} = \sum_{j \in \mathbb{N}_i^{out}} \frac{w_{j,t}}{q_{j,t} W_{i,t}}$*,* C *is a constant and* $\sigma_{j^*}^2$ *is the second moment of* $\pi_{\kappa_{j^*}}(\boldsymbol{\psi})$*.*

Proof The proof is deferred to Appendix E. ■

The regret bound in (40) depends on $\frac{1}{\bar{q}_{i,t}}$ and $\frac{1}{q_{j^*,t}}$. Since, there is a self-loop for all $v_k \in \mathcal{V}$, it can be written that $q_{k,t} \geq p_{k,t}$. In addition, based on (33), we can conclude that $p_{k,t} > \frac{\xi}{|\mathbb{D}|}$, $\forall v_k \in \mathcal{V}$ and as a result $q_{k,t} > \frac{\xi}{|\mathbb{D}|}$, $\forall v_k \in \mathcal{V}$. Therefore, in the worst case where $q_{j^*,t} = \mathcal{O}(\frac{\xi}{|\mathbb{D}|})$, considering

$$\eta = \mathcal{O}\left(\sqrt{\frac{\ln N}{N}} T^{-\frac{2}{3}}\right), \epsilon = \xi = \mathcal{O}\left(T^{-\frac{1}{3}}\right), D = \mathcal{O}\left(T^{\frac{2}{3}} \ln T\right) \tag{41}$$

OMKL-SFG can achieve regret bound of $\mathcal{O}(\sqrt{N \ln N} T^{\frac{2}{3}})$ with high probability of $1-\mathcal{O}(T^{-\frac{1}{3}})$. Moreover, comparing regret bound of OMKL-SFG with that of OMKL-GF, it can be observed that OMKL-SFG obtains tighter regret than OMKL-GF. The reason behind this is that the regret bound of both OMKL-SFG and OMKL-GF depend on $1/q_{j^*,t}$ (c.f. (40) and (90)) and OMKL-SFG performs more exploration than OMKL-GF in the sense that using OMKL-SFG the lower bound for the probability $q_{i,t}$, $\forall i$ is larger than that of OMKL-GF. Specifically, using OMKL-GF, $q_{i,t} > \eta_e^2/NJ$ (c.f. (91)). Setting $\eta_e = \mathcal{O}(N^{\frac{1}{6}} T^{-\frac{1}{4}})$ and $J = \mathcal{O}(N^{\frac{1}{3}})$ as it is specified in (24), it can be concluded that

$$q_{i,t} > \mathcal{O}\left(\frac{1}{N\sqrt{T}}\right), \forall i \in \{1, \ldots, N\}, \forall t \in \{1, \ldots, T\} \tag{42}$$

when OMKL-GF is employed. Moreover, since using OMKL-SFG, $q_{i,t} > \frac{\xi}{|\mathbb{D}|}$, choosing $\xi = \mathcal{O}(T^{-\frac{1}{3}})$ as in (41) and considering the fact that $|\mathbb{D}| \leq N$, the lower bound of $q_{i,t}$ when OMKL-SFG is employed is obtained as

$$q_{i,t} > \mathcal{O}\left(\frac{1}{NT^{\frac{1}{3}}}\right), \forall i \in \{1, \ldots, N\}, \forall t \in \{1, \ldots, T\}. \tag{43}$$

Comparing (42) with (43), it can be concluded that OMKL-SFG can provide larger lower bound for $q_{j^*,t}$ than that of OMKL-GF. This enables OMKL-SFG to obtain tighter regret upper bound than OMKL-GF. Since the regret bound of $\mathcal{O}(\sqrt{T})$ is more satisfactory than the regret bound of $\mathcal{O}(T^{\frac{2}{3}})$, in what follows the structure of the feedback graph is refined at each time instant so that the regret of $\mathcal{O}(\sqrt{T})$ can be achieved.

4.3 OMKL with Similarity-based Feedback Graph Refinement

This subsection further improves the OMKL-SFG by refining the offline feedback graph 'on the fly', so that the resulting algorithm achieves a tighter sub-linear regret of $\mathcal{O}(\sqrt{T})$. To this end, at each time instant the offline feedback graph $\mathcal{G}$ constructed by the Algorithm 4 is refined to a feedback graph $\mathcal{G}'_t$ based on the observed losses. To begin with, let's define set $\mathbb{D}'_t$ as

$$\mathbb{D}'_t := \left\{ i \middle| \frac{u_{i,t}}{U_t} \geq \frac{1}{1-\xi}(\beta - \frac{\xi}{N}) \right\} \tag{44}$$

where β is a pre-selected constant. According to (33), it can be inferred that $p_{i,t} \geq \beta, \forall i \in \mathbb{D}'_t$. Let $\mathcal{G}'_t = (\mathcal{V}, \mathcal{E}'_t)$ be a graph such that $\mathbb{D}'_t$ is a dominating set of $\mathcal{G}'_t$. Suppose at each time instant t, $\mathcal{G}'_t$ is employed as the feedback graph instead of $\mathcal{G}$. In this case, it is ensured that there is at least one edge from $\mathbb{D}'_t$ to each $v_i \in \mathcal{V} \setminus \mathbb{D}'_t$, i.e., $\mathbb{D}'_t$ is a dominating set of $\mathcal{G}'_t$. In this case, we have $q_{i,t} \geq \beta$, $\forall v_i \in \mathcal{V}$. To this end, at each time instant t, $\mathcal{G}'_t$ can be constructed based on $\mathcal{G}$ by expanding $\mathcal{E}$ to $\mathcal{E}'_t$ such that $\mathbb{D}'_t$ would be a dominating set of $\mathcal{G}'_t$. Specifically, at each time instant t, the edge $(d_{i,t}, i)$ is appended to $\mathcal{E}'_t$, if there is not any edge from $\mathbb{D}'_t$ to v_i, where

$$d_{i,t} = \arg\max_{j \in \mathbb{D}'_t} \Delta(\kappa_i, \kappa_j). \tag{45}$$

Algorithm 6 OMKL with Similarity Feedback Graph Refinement (OMKL-SFG-R)

```
Input: Shift-invariant kernels $\kappa_i$, $i = 1, \ldots, N$, learning rate $\eta > 0$, exploration rate $\xi > 0$, the number of random features $D$ and the constant $\beta > 0$.
Initialize: $\boldsymbol{\theta}_{i,1} = \mathbf{0}$, $w_{i,1} = 1$, $i = 1, \ldots, N$, Construct the feedback graph $\mathcal{G}$ via Algorithm 4.
for $t = 1, \ldots, T$ do
    Receive one datum $\boldsymbol{x}_t$.
    Set $\mathcal{E}'_t = \mathcal{E}$ and obtain $d_{i,t}$, $\forall i \in \mathcal{V} \setminus \mathbb{D}'_t$ by (45).
    For all $i \in \mathcal{V} \setminus \mathbb{D}'_t$, append $(d_{i,t}, i)$ to $\mathcal{E}'_t$ if $(d_{i,t}, i) \notin \mathcal{E}$.
    Draw one of nodes $v_i \in \mathcal{V}$ according to the PMF $p_t = (p_{1,t}, ..., p_{N,t})$ in (46).
    Predict $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t)$ via (47).
    Calculate loss $\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)$ for all $i \in \mathbb{S}_t$.
    Update $\boldsymbol{\theta}_{i,t+1}$ via (38).
    Update $w_{i,t+1}$ and $u_{i,t+1}$ via (39).
end for
```

Hence, there is at least one edge from $\mathbb{D}'_t$ to $v_i \in \mathcal{V} \setminus \mathbb{D}'_t$, meaning $\mathbb{D}'_t$ is a dominating set for $\mathcal{G}'_t$. Then one of the vertices in $\mathcal{V}$ is drawn according to the PMF p_t, with

$$p_{i,t} = (1 - \xi)\frac{u_{i,t}}{U_t} + \frac{\xi}{|\mathbb{D}'_t|}\mathbf{1}_{\mathrm{i} \in \mathbb{D}'_{\mathrm{t}}}, i = 1, \ldots, N. \tag{46}$$

Let $\mathbb{N}^{\mathrm{out}}_{i,t}$ and $\mathbb{N}^{\mathrm{in}}_{i,t}$ denote sets of out-neighbors and in-neighbors of v_i in $\mathcal{G}'_t$, respectively. According to (44) and (46), we have $q_{i,t} \geq \beta$, $\forall v_i \in \mathcal{V}$ where $q_{i,t} = \sum_{j \in \mathbb{N}^{\mathrm{in}}_{i,t}} p_{j,t}$. The RF-based function approximation can be written as

$$\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t) = \sum_{i \in \mathbb{N}^{\mathrm{out}}_{I_t}} \frac{w_{i,t}}{\sum_{j \in \mathbb{N}^{\mathrm{out}}_{I_t}} w_{j,t}} \hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t). \tag{47}$$

According to (47), $\boldsymbol{\theta}_{i,t}$, $w_{i,t}$ and $u_{i,t}$ can be updated using (38), (39a) and (39b), respectively. The procedure is summarized in Algorithm 6, which is called OMKL-SFG-R, and its performance in terms of regret analysis is presented in the following Theorem.

Theorem 6 *Under (as1)–(as3), the stochastic regret of OMKL-SFG-R satisfies*

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(f^*(\boldsymbol{x}_t), y_t) \\ \leq & \frac{2 \ln N}{\eta} + \frac{(1+\epsilon)C^2}{2\eta} + \epsilon LTC + (\xi + \frac{\eta}{2}\frac{L^2 + N\beta + 1}{\beta} - \frac{\eta\xi}{2})T\end{aligned} \tag{48}$$

with probability at least $1 - 2^8(\frac{\sigma_{j^*}}{\epsilon})^2 \exp(-\frac{D\epsilon^2}{4d+8})$ *for any* $\epsilon > 0$ *and any* $\beta \leq \frac{1}{N}$.

Proof The proof can be found in Appendix F. ■

According to Theorem 6, by setting

$$\eta = \mathcal{O}\left(\sqrt{\frac{\ln N}{NT}}\right), \epsilon = \xi = \mathcal{O}\left(\sqrt{\frac{1}{T}}\right), D = \mathcal{O}(T \ln T) \tag{49}$$

and $\beta = \mathcal{O}(1)$ such that $\beta \leq \frac{1}{N}$, OMKL-SFG-R can achieve regret bound of $\mathcal{O}(\sqrt{TN \ln N})$ using the feedback graph $\mathcal{G}'_t$ with probability of $1 - \mathcal{O}(1)$. According to Theorem 6, larger D leads to larger probability that the regret bound in (48) holds true. Thus, in order to achieve regret of $\mathcal{O}(\sqrt{TN \ln N})$ with high probability, OMKL-SFG-R should set sufficiently large value of order $\mathcal{O}(T \ln T)$ for D. However, note that since some edges may be added to $\mathcal{G}$ to construct $\mathcal{G}'_t$, using $\mathcal{G}'_t$ instead of $\mathcal{G}$ may cause increase in computational complexity of function approximation.

Computational Complexity. Both OMKL-SFG and OMKL-SFG-R need to store a set of d-dimensional vectors $\{\boldsymbol{\psi}_{i,j}\}_{j=1}^{D}$ per kernel in addition to two weighting coefficients $\{w_{i,t}\}_{i=1}^{N}$ and $\{u_{i,t}\}_{i=1}^{N}$. Furthermore, both OMKL-SFG and OMKL-SFG-R need to store adjacency matrix of $\mathcal{G}$ which is $N \times N$ matrix. In order to perform required computation for (45), OMKL-SFG-R needs to store the divergence $\Delta(\kappa_i, \kappa_j)$ between any pair of kernels in the dictionary. Therefore, the memory requirement for both algorithms are $\mathcal{O}(dDN + N^2)$. Consider the case where the number of out-neighbors of each node v_i in $\mathcal{G}$ satisfies $M < N$, the per-iteration complexity of OMKL-SFG including calculation of inner products is $\mathcal{O}(dDM)$. Therefore, it can be inferred that OMKL-SFG incurs less computational complexity than OMKL-GF since recall that the per-iteration complexity of OMKL-GF is $\mathcal{O}(dDM + JN)$. Therefore, it can be concluded that the offline feedback graph construction indeed can alleviate the computational complexity compared with OMKL-GF in section 3. Due to the graph refinement procedure, the complexity of OMKL-SFG-R is higher than OMKL-SFG. According to Algorithm 6, there is a possibility that all nodes in the graph are out-neighbors of one node in $\mathbb{D}'_t$. Therefore, the worst-case per-iteration computational complexity of OMKL-SFG-R is $\mathcal{O}(dDN)$. Furthermore, the computational complexities of OMKL-SFG and OMKL-SFG-R are lower than state-or-art multi-kernel learning algorithms provided that the per-iteration computational complexity of OMKR is $\mathcal{O}(tdN)$ Sahoo et al. (2014), and the per-iteration computational complexity of RF-based online multi-kernel learning algorithms proposed in Sahoo et al. (2019) and Shen et al. (2019) are $\mathcal{O}(dDN)$.

Regret Bounds Comparison. The algorithm OMKR (Sahoo et al., 2014) achieves regret of $\mathcal{O}(\sqrt{T \ln N})$. However, since per iteration computational complexity of OMKR is $\mathcal{O}(tdN)$, OMKR requires much more computations than the proposed graph-aided OMKL algorithms. Furthermore, Raker (Shen et al., 2019) obtains regret of $\mathcal{O}(\sqrt{T \ln N})$ with high probability when the number of random features D is set to $\mathcal{O}(T \ln T)$. Therefore, employing all kernels in the dictionary, Raker obtains tighter regret bound than those of the proposed graph-aided OMKL algorithms. However, the proposed graph-aided OMKL algorithms require less computations than Raker.

Comparison with Online Learning. In online learning with expert advice, there is a learner interacts with a set of experts such that at each round of learning the learner choose one of the experts and takes its advice for either decision making or prediction (Cesa-Bianchi and Lugosi, 2006; Auer et al., 2003). The learner may observe the loss associated with a subset of experts after decision making and in this regard this can be modeled by a graph which is called feedback graph (see e.g. (Mannor and Shamir, 2011; Cohen et al., 2016; Alon et al., 2017; Ghari and Shen, 2022)). In all algorithms proposed in this paper, each kernel can be viewed as an expert. However, there are two major innovative differences compared with online learning with feedback graph: i) the proposed algorithm constructs and refines the feedback graph to improve the performance of learning task while in online learning, the feedback graph is generated in an adversarial manner. ii) in this paper, each expert (kernel) is a learner itself and experts implement an online scheme for self-improvement.

5. Experiments

This section presents experimental results over real datasets downloaded from UCI Machine Learning Repository (Dua and Graff, 2017). The accuracy of different approaches are evaluated using mean square error (MSE). Due to the randomness in the random features extracted for function approximation, we average the MSE over $R = 20$ different sets of random features. The MSE at time t is computed as

$$\mathrm{MSE}_t = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{t}\sum_{\tau=1}^{t}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_\tau) - y_\tau)^2. \tag{50}$$

The number of random features is $D = 50$. The kernel dictionary contains 76 kernels including 51 RBF kernels and 25 Laplacian kernels. The bandwidth of the i-th $(1 \leq i \leq 51)$ RBF kernel is 10^{σ_i} with $\sigma_i = \frac{2i-52}{25}$. And the value of the i-th $(1 \leq i \leq 25)$ Laplacian kernel's parameter is 10^{λ_i} where $\lambda_i = \frac{i-13}{6}$. For fairness of evaluation, parameters ξ, η and η_e are set to $\frac{0.1}{\sqrt{t}}$ for all algorithms at time step t. More precise results can be obtained with more extensive parameter tuning. The performance of kernel learning algorithms is evaluated through several real datasets:

Airfoil: This dataset comprises $1,503$ different size airfoils at various wind tunnel speeds and each data sample $\boldsymbol{x}_t$ includes 5 features such as frequency and chord length. The output y_t is scaled sound pressure level in decibels (Brooks et al., 1989).

Bias: This dataset contains $7,750$ samples. Each sample has 21 features including maximum and minimum temperatures of present-day, and geographic auxiliary variables for the purpose of bias correction of next-day minimum air temperatures. The goal is to predict next-day minimum air temperature (Cho et al., 2020).

Concrete: This dataset contains $1,030$ samples of 8 features, such as the amount of cement or water in a concrete. The goal is to predict concrete compressive strength (Yeh, 1998).

Naval: contains $11,934$ samples of 15 features of a naval vessel characterized by a gas turbine propulsion plant including the ship speed and gas turbine shaft torque. The goal is to predict the lever position (Coraddu et al., 2016).

Parameter λ is set to 10^{-3} for all proposed algorithms OMKL-GF, OMKL-SFG and OMKL-SFG-R. Generating the bipartite graph $\mathcal{B}_t$ at every time instant can increase the computational complexity of the proposed OMKL-GF while it cannot improve MSE considerably. To further decrease the computational complexity of our proposed OMKL-GF, we terminate generating $\mathcal{B}_t$ after 300 time instants, meaning that $\mathcal{B}_t = \mathcal{B}_{300}$ if $t > 300$. The number of selective nodes for OMKL-GF is set to be 2. Furthermore, the feedback graph $\mathcal{G}$ in Algorithm 4 is generated with the divergence function $\Delta(\cdot,\cdot)$ defined using Bregman divergence in (25) when $F(\boldsymbol{\omega}) = \|\omega\|^2$. The greedy set cover algorithm by Chvatal (1979) is utilized to find the dominating set $\mathbb{D}$ of the feedback graph $\mathcal{G}$. Moreover, in order to speed up OMKL-SFG and OMKL-SFG-R, after 300 time instants, OMKL-SFG-R chooses $I_t = \arg\max_i \frac{u_{i,t}}{U_t}$. All experiments were carried out using Intel(R) Core(TM) i7-10510U CPU

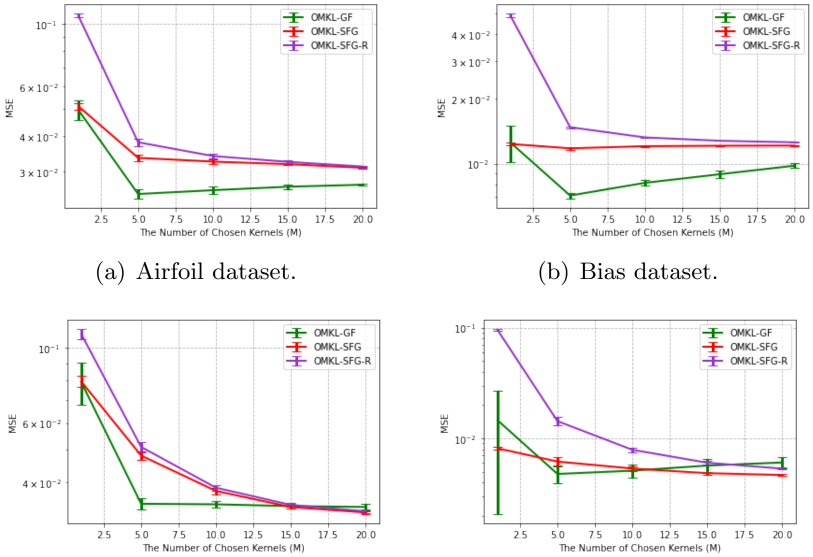
@ 1.80 GHz 2.30 GHz processor with a 64-bit Windows operating system. Codes are available at `https://github.com/pouyamghari/Graph-Aided-Online-Multi-Kernel-Learning`.

5.1 Number of Selected Kernels

The present subsection studies the effect of the maximum number of selected kernels M on the performance of the proposed OMKL-GF, OMKL-SFG and OMKL-SFG-R. Figure 3 illustrates the MSE and its standard deviation of the proposed OMKL-GF, OMKL-SFG and OMKL-SFG-R with the change in the number of selected kernels. The standard deviation and the MSE are obtained over $R = 20$ sets of i.i.d random features (c.f. (50)). Figure 3 shows the advantage of data-driven kernel selection by OMKL-GF since increasing M from $M = 10$ to $M = 20$ does not result in lower MSE for Airfoil, Bias and Naval datasets. Figure 3 indicates that for OMKL-SFG and OMKL-SFG-R choosing moderate value for M such as $M = 10$ obtains MSE comparatively close to the MSE associated with optimal M. Figure 3 illustrates that OMKL-GF can obtain lower MSE than OMKL-SFG although OMKL-SFG achieves tighter regret upper bound than that of OMKL-GF. Regret upper bounds deal with worst cases. According to Theorems 2 and 5, the worst cases for OMKL-GF and OMKL-SFG happen when the probability of observing the loss of the best kernel (i.e. $q_{j^*,t}$ where j^* defined in Theorem 5) is close to its minimum value for almost all t. For both algorithms $q_{j^*,t}$ is close to its minimum value for almost all t if the probability of choosing the best kernel for function approximation is small for all t, which is unlikely to happen although it is not impossible. In addition, both OMKL-GF and OMKL-SFG choose a subset of kernels using a trade-off between exploitation and exploration. OMKL-SFG performs more exploration in choosing a subset of kernels than OMKL-GF in the sense that using OMKL-SFG lower bound for $q_{j^*,t}$ is larger than that of OMKL-GF (see (42) and (43)). OMKL-GF performs more exploitation than OMKL-SFG since using OMKL-GF the structure of the graph is changing every time instant to enable OMKL-GF to choose optimal subset of kernels while using OMKL-SFG the structure of the graph is fixed and independent of prior loss observations. Results in Figure 3 show that the proper selection of M along with data-driven kernel selection can enable OMKL-GF to choose a more powerful subset of kernels than that of OMKL-SFG which leads to better accuracy of OMKL-GF. Moreover, since using OMKL-SFG-R, $q_{i,t} \geq \beta$, choosing $\beta = \mathcal{O}(1)$, it can be concluded that $q_{j^*,t} \geq \mathcal{O}(1)$ when OMKL-SFG-R is employed. Therefore, OMKL-SFG-R performs more exploration than OMKL-SFG which leads to tighter regret upper bound for OMKL-SFG-R. However, employing OMKL-SFG-R increases the probability that the kernels with comparatively poor prior performance being among the chosen subset of kernels. This is against exploitation and degrades the MSE performance of OMKL-SFG-R compared to OMKL-SFG. Furthermore, Figure 4 depicts that the run time of the proposed graph-aided OMKL algorithms with the change in the number of selected kernels. Figure 4 shows that a larger M leads to longer run time which is expected as larger M increases the complexity. Moreover, numerical results associated with Figures 3 and 4 are also presented in Tables 3, 4, 5 in Appendix A.

5.2 MSE and Run Time Performance Compared to Baselines

The performance of the proposed algorithms OMKL-GF, OMKL-SFG and OMKL-SFG-R are compared with the following kernel learning benchmarks:



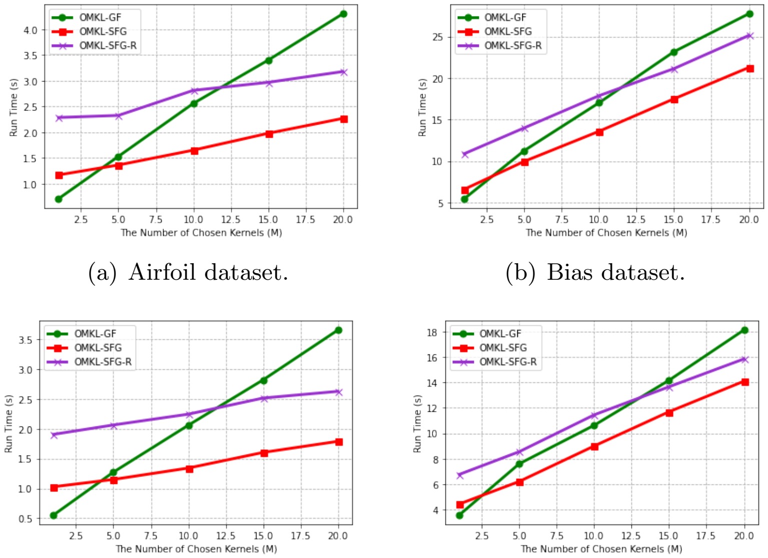
(a) Airfoil dataset.

(b) Bias dataset.

(c) Concrete dataset.

(d) Naval dataset.

Figure 3: MSE and standard deviation performance of Graph-aided MKL algorithms on real datasets with the change in the number of selected kernels.



(a) Airfoil dataset.

(b) Bias dataset.

(c) Concrete dataset.

(d) Naval dataset.

Figure 4: Run Time performance of Graph-aided MKL algorithms on real datasets with the change in the number of selected kernels.

Table 1: MSE($\times 10^{-3}$) of MKL algorithms on real datasets.

Algorithms	Airfoil	Bias	Concrete	Naval
OMKR	32.68	15.54	41.72	9.22
RBF-1	33.17	15.96	41.73	11.99
POLY-2	355.67	23.36	50.06	90.06
RFOMKR	350.52	405.44	210.42	347.06
Raker	28.64	12.70	35.22	11.35
OMKL-GF	**25.73**	**8.15**	**34.45**	**5.11**
OMKL-SFG	32.49	12.06	37.75	5.35
OMKL-SFG-R	33.99	13.24	38.58	7.89

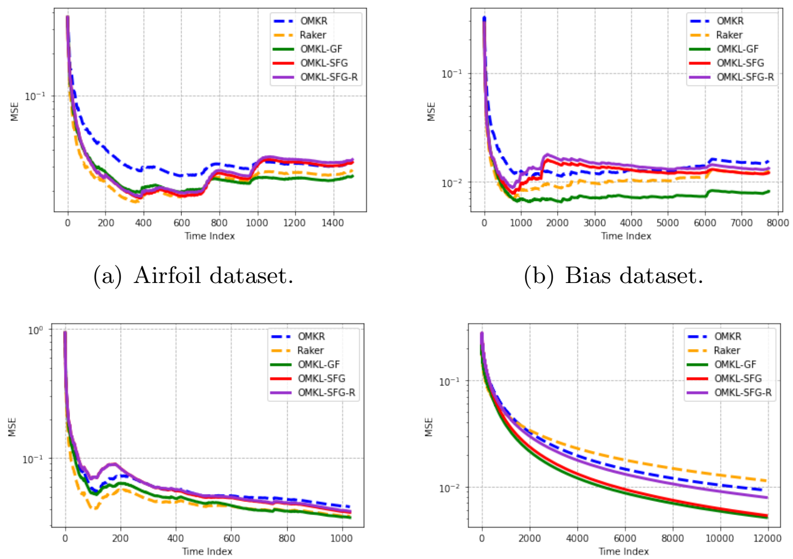
Table 2: Run time(s) of MKL algorithms on real datasets.

Algorithms	Airfoil	Bias	Concrete	Naval
OMKR	889.36	80010.03	547.93	163688.82
RBF-1	14.39	3914.67	6.60	1499.78
POLY-2	7.22	195.19	3.33	951.93
RFOMKR	2.17	27.72	1.53	18.37
Raker	3.81	46.28	2.71	31.24
OMKL-GF	2.56	17.01	2.06	10.63
OMKL-SFG	**1.65**	**13.58**	**1.34**	**9.00**
OMKL-SFG-R	2.81	17.86	2.24	11.45

- **OMKR**: online multi-kernel regression approach proposed by Sahoo et al. (2014).
- **RBF-1**: online single kernel regression approach (Sahoo et al., 2014) using a radial basis function (RBF) with bandwidth of 1.
- **POLY-2**: online single kernel regression approach (Sahoo et al., 2014) using a polynomial kernel with degree of 2.
- **RFOMKR**: online multi-kernel learning approach utilizes RF approximation (Sahoo et al., 2019).
- **Raker**: RF-based online multi-kernel learning (Shen et al., 2019).

The maximum number of kernels chosen by OMKL-GF at each time instant is 10. In addition, for OMKL-SFG, to determine the value of γ_i for each vertex $v_i \in \mathcal{V}$, the number of out-neighbors for each node is set to be 10. For OMKL-SFG-R, at each time, β is set to $\beta = (1-\xi)\bar{u}_{[10,t]} + \frac{\xi}{N}$ where $\bar{u}_{[10,t]}$ denote the tenth greatest value in the sequence $\{\frac{u_{i,t}}{U_t}\}_{i=1}^{N}$.

Tables 1 and Table 2 list MSE and run time performance of alternative algorithms on real datasets, respectively. It can be observed from Table 1 that the proposed OMKL-GF significantly outperforms all benchmark algorithms, which corroborate the effectiveness of



(a) Airfoil dataset.

(b) Bias dataset.

(c) Concrete dataset.

(d) Naval dataset.

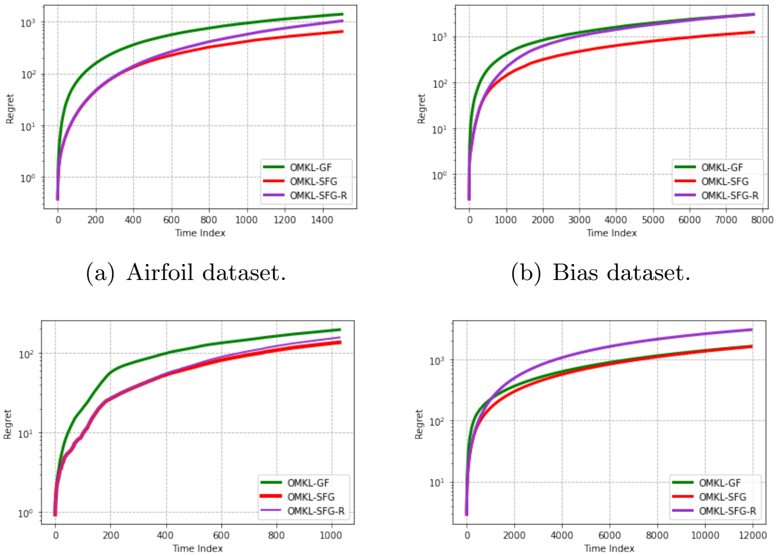
Figure 5: MSE performance of MKL algorithms on real datasets.

data-driven feedback graph based kernel pruning. Furthermore, MSEs reported in Table 1 indicates that the accuracy of OMKL-SFG is comparable with that of Raker which employs all kernels in the dictionary while OMKL-SFG chooses a subset of kernels at each time instant. Table 2 shows that OMKL-GF and OMKL-SFG are more efficient than all other alternatives, since thanks to the graph-aided pruning only a subset of kernels instead of all kernels in the dictionary are employed at each time instant. In addition, OMKL-SFG is the fastest due to the offline graph construction. Although OMKL-SFG-R enjoys tighter sub-linear regret than those of OMKL-GF and OMKL-SFG by including a larger number kernels in the selected subset, employing OMKL-SFG-R requires more computation and increases the run time which can be inferred from the results in the Table 2.

Figure 5 illustrates the MSE of OMKR, Raker and proposed algorithms over time. It can be seen that as time goes, performance gain of OMKL-GF becomes more remarkable. This confirms the effectiveness of the data-driven kernel selection in a sense that the proposed OMKL-GF learns the optimal subset of kernels in the dictionary 'on the fly'.

5.3 Regret Performance

The present section presents the regret performance of the proposed OMKL-GF, OMKL-SFG and OMKL-SFG-R. The maximum number of kernels chosen by OMKL-GF at each time instant is 10. In addition, using OMKL-SFG, at each time instant 10 kernels are chosen to perform the function approximation task. For OMKL-SFG-R, at each time, β is set to $\beta = (1-\xi)\bar{u}_{[10,t]} + \frac{\xi}{N}$. Figure 6 illustrates the regret of the proposed algorithms over time.



(a) Airfoil dataset.

(b) Bias dataset.

(c) Concrete dataset.

(d) Naval dataset.

Figure 6: Regret of proposed OMKL algorithms on real datasets.

6. Conclusion

This paper develops online multi-kernel learning algorithms for non-linear function learning. By constructing a bipartite feedback graph at every time instant, OMKL-GF chooses a subset of kernels to both prune irrelevant kernels and decrease the computational complexity. It is proved that OMKL-GF can obtain regret of $\mathcal{O}(T^{\frac{3}{4}})$. To further alleviate the computational burden of multi-kernel learning, a feedback graph is constructed in an offline fashion based on the similarities among kernels. Using the similarity-based feedback graph, a subset of kernels is chosen and the resulting algorithm is called OMKL-SFG. It is proved that OMKL-SFG can achieve sub-linear regret of $\mathcal{O}(T^{\frac{2}{3}})$. Furthermore, refining the similarity-based feedback graph structure at each time instant, OMKL-SFG-R is proposed, which enjoys sub-linear regret of $\mathcal{O}(\sqrt{T})$. Moreover, experiments on real datasets demonstrate that by choosing a subset of kernels, OMKL-GF can obtain lower MSE in comparison with other online kernel learning algorithms including OMKR and Raker. Furthermore, experiments show that OMKL-GF and OMKL-SFG have considerably lower run time compared to online multi-kernel learning algorithms OMKR and Raker.

Acknowledgement

This work is supported by NSF ECCS 2207457. PI: Yanning Shen (yannings@uci.edu).

References

Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM*

Journal on Computing, 46(6):1785–1826, 2017.

Armen S. Asratian, Tristan M. J. Denley, and Roland Häggkvist. *Bipartite Graphs and Their Applications*. Cambridge University Press, 1998.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, Jan 2003.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005.

Yoshua Bengio, Olivier Delalleau, and Nicolas L. Roux. The curse of highly variable functions for local kernel machines. In *Advances in Neural Information Processing Systems*, pages 107–114, May 2006.

Pantelis Bouboulis, Symeon Chouvardas, and Sergios Theodoridis. Online distributed learning over networks in rkh spaces using random fourier features. *IEEE Transactions on Signal Processing*, 66(7):1920–1932, Apr 2018.

L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Thomas F. Brooks, D. Stuart Pope, and Micheal A. Marcolini. Airfoil self-noise and prediction. Technical report, Jul 1989.

Serhat S. Bucak, Rong Jin, and Anil K. Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369, Jul 2014.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, USA, 2006.

Dongjin Cho, Cheolhee Yoo, Jungho Im, and Dong-Hyun 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*, 7 (4), Mar 2020.

Wesley Chung, Somjit Nath, Ajin Joseph, and Martha White. Two-timescale networks for nonlinear value function approximation. In *International Conference on Learning Representations*, May 2019.

Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, Aug 1979.

Alon Cohen, Tamir Hazan, and Tomer Koren. Online learning with feedback graphs without the graphs. In *Proceedings of International Conference on Machine Learning*, page 811–819, Jun 2016.

Andrea Coraddu, Luca Oneto, Aessandro Ghio, Stefano Savio, Davide Anguita, and Massimo Figari. Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 230(1):136–153, 2016.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, page 109–116, Jun 2009.

Liang Ding, Rui Tuo, and Shahin Shahrampour. Generalization guarantees for sparse kernel approximation with entropic optimal features. In *Proceedings of the International Conference on Machine Learning*, volume 119, pages 2545–2555, Jul 2020.

Yi Ding, Chenghao Liu, Peilin Zhao, and Steven C. H. Hoi. Large scale kernel methods for online AUC maximization. In *IEEE International Conference on Data Mining*, pages 91–100, Nov 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

Lixin Duan, Ivor W. Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, Mar 2012.

Pouya M Ghari and Yanning Shen. Online multi-kernel learning with graph-structured feedback. In *Proceedings of the International Conference on Machine Learning*, volume 119, pages 3474–3483, Jul 2020.

Pouya M. Ghari and Yanning Shen. Online learning with probabilistic feedback. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4183–4187, May 2022.

Mehmet Gönen and Ethem Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(64):2211–2268, 2011.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Steven C. H. Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. Online multiple kernel classification. *Machine Learning*, 90:289–316, Feb 2013.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, Jul 2011.

Jing Lu, Steven C.H. Hoi, Jialei Wang, Peilin Zhao, and Zhi-Yong Liu. Large scale online kernel learning. *Journal of Machine Learning Research*, 17(47):1–43, 2016.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Proceedings of International Conference on Neural Information Processing Systems*, pages 684–692, 2011.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of International Conference on Neural Information Processing Systems*, pages 1177–1184, Dec 2007.

Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9(83):2491–2521, 2008.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Proceedings of International Conference on Neural Information Processing Systems*, page 3218–3228, 2017.

Doyen Sahoo, Steven C.H. Hoi, and Bin Li. Online multiple kernel regression. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 293–302, 2014.

Doyen Sahoo, Steven C. H. Hoi, and Bin Li. Large scale online multiple kernel regression with application to time-series prediction. *ACM Transactions on Knowledge Discovery from Data*, 13(1), Jan 2019.

Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

Shahin Shahrampour and Vahid Tarokh. Learning bounds for greedy approximation with explicit feature maps from multiple kernels. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 4695–4706, Dec 2018.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA, 2004.

Yanning Shen, Tianyi Chen, and Georgios B. Giannakis. Random feature-based online multi-kernel learning in environments with unknown dynamics. *Journal of Machine Learning Research*, 20(1):773–808, Jan 2019.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, Dec 2006.

Bharath K. Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 1144–1152, Dec 2015.

Grace Wahba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, 1990.

Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 661–667, Jan 2000.

I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797 – 1808, Dec 1998.

Table 3: MSE($\times 10^{-3}$) and run time of OMKL-GF with different M on real datasets.

	MSE($\times 10^{-3}$)					Run time (s)				
Datasets	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Airfoil	49.53	24.95	25.73	26.47	26.91	0.70	1.53	2.56	3.40	4.30
Bias	12.56	7.11	8.15	8.93	9.79	5.44	11.24	17.01	23.17	27.74
Concrete	79.27	34.58	34.45	34.04	33.84	0.55	1.27	2.06	2.82	3.65
Naval	14.60	4.78	5.11	5.69	6.06	3.58	7.61	10.63	14.16	18.11

Table 4: MSE($\times 10^{-3}$) and run time of OMKL-SFG with different M on real datasets.

	MSE($\times 10^{-3}$)					Run time (s)				
Datasets	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Airfoil	51.03	33.49	32.49	31.82	30.89	1.17	1.36	1.65	1.98	2.27
Bias	12.35	11.78	12.06	12.11	12.13	6.58	9.94	13.57	17.49	21.26
Concrete	79.60	48.03	37.75	33.83	32.59	1.03	1.15	1.34	1.60	1.79
Naval	8.15	6.18	5.35	4.85	4.68	4.45	6.22	9.00	11.69	14.09

Xiao Zhang and Shizhong Liao. Incremental randomized sketching for online kernel learning. In *Proceedings of International Conference on Machine Learning*, pages 7394–7403, Jun 2019.

Appendix A. Additional Experimental Results

This section presents more detailed results on MSE and Run time of proposed algorithms with the change in the number of chosen kernels. Table 3 lists the MSE and run time of OMKL-GF with the change in the value of M which is the maximum number of kernels selected at each time instant by OMKL-GF. For OMKL-SFG and OMKL-SFG-R, the value of γ_i is chosen such that for each vertex $v_i \in \mathcal{V}$, the number of out-neighbors for each node is M. Tables 4 and 5 show both MSE and run time of OMKL-SFG and OMKL-SFG-R respectively with different values of M.

Appendix B. Proof of Lemma 1

In order to prove Lemma 1, we first establish the following lemma as a step stone.

Lemma 7 *Let $\hat{f}_{RF,i}(.)$ denote the sequence of estimates generated with a preselected kernel κ_i where $\mathcal{F}_i = \{\hat{f}_i | \hat{f}_i(\boldsymbol{x}) = \boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}), \forall \boldsymbol{\theta} \in \mathbb{R}^{2D}\}$. Then, under assumptions (as1) and (as2)*

Table 5: MSE($\times 10^{-3}$) and run time of OMKL-SFG-R with different M on real datasets.

	MSE($\times 10^{-3}$)					Run time (s)				
Datasets	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$	$M = 1$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Airfoil	106.66	38.03	33.99	32.42	31.21	2.28	2.32	2.81	2.97	3.18
Bias	48.81	14.77	13.24	12.78	12.57	10.87	13.98	17.86	21.13	25.13
Concrete	110.08	50.96	38.58	34.32	32.85	1.91	2.06	2.24	2.51	2.63
Naval	95.37	14.32	7.89	6.03	5.33	6.77	8.55	11.45	13.65	15.84

the following bound holds

$$\sum_{t=1}^{T} \mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t), y_t) \leq \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \sum_{t=1}^{T} \frac{\eta L^2}{2q_{i,t}} \tag{51}$$

where L is the Lipschitz constant in (as2) and $\boldsymbol{\theta}_i^*$ is the parameter vector associated with the best estimator $\hat{f}_i^*(\boldsymbol{x}) = (\boldsymbol{\theta}_i^*)^\top \mathbf{z}_i(\boldsymbol{x})$.

Proof For $\boldsymbol{\theta}_{i,t+1}$ and any fixed $\boldsymbol{\theta}$, it can be written that

$$\begin{aligned}\|\boldsymbol{\theta}_{i,t+1} - \boldsymbol{\theta}\|^2 &= \|\boldsymbol{\theta}_{i,t} - \eta\nabla\ell_{i,t} - \boldsymbol{\theta}\|^2 \\ &= \|\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}\|^2 - 2\eta\nabla^\top\ell_{i,t}(\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}) + \|\eta\nabla\ell_{i,t}\|^2. \end{aligned} \tag{52}$$

Furthermore, from the convexity of the loss function with respect to $\boldsymbol{\theta}$ in (as1), we can conclude that

$$\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) - \mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) \leq \nabla^\top \mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)(\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}). \tag{53}$$

Therefore, from (53), it can be inferred that

$$\left(\frac{\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}} - \frac{\mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}\right)\mathbf{1}_{\mathrm{i}\in\mathbb{S}_\mathrm{t}} \leq \frac{\nabla^\top \mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}(\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta})\mathbf{1}_{\mathrm{i}\in\mathbb{S}_\mathrm{t}}. \tag{54}$$

Based on (35), (54) is equivalent to

$$\ell_{i,t} - \frac{\mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}\mathbf{1}_{\mathrm{i}\in\mathbb{S}_\mathrm{t}} \leq \nabla^\top \ell_{i,t}(\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}). \tag{55}$$

Combining (52) with (55), we get

$$\ell_{i,t} - \frac{\mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}}\mathbf{1}_{\mathrm{i}\in\mathbb{S}_\mathrm{t}} \leq \frac{\|\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}\|^2 - \|\boldsymbol{\theta}_{i,t+1} - \boldsymbol{\theta}\|^2}{2\eta} + \frac{\eta}{2}\|\nabla\ell_{i,t}\|^2. \tag{56}$$

Taking the expectation of $\ell_{i,t}$ and $\|\nabla\ell_{i,t}\|^2$ with respect to $\mathbf{1}_{\mathrm{i}\in\mathbb{S}_\mathrm{t}}$, we arrive at

$$\mathbb{E}_t[\ell_{i,t}] = \sum_{j\in\mathbb{N}_i^{\mathrm{in}}} p_{j,t}\frac{\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)}{q_{i,t}} = \mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t) \tag{57a}$$

$$\mathbb{E}_t[\|\nabla\ell_{i,t}\|^2] = \sum_{j\in\mathbb{N}_i^{\mathrm{in}}} p_{j,t}\frac{\|\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)\|^2}{q_{i,t}^2} = \frac{\|\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^\top \mathbf{z}_i(\boldsymbol{x}_t), y_t)\|^2}{q_{i,t}}. \tag{57b}$$

Therefore, taking the expectation with respect to $\mathbf{1}_{\mathrm{i}\in\mathbb{S}_{\mathrm{t}}}$ from both sides of (56), we obtain

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t) - \mathcal{L}(\boldsymbol{\theta}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t)\\ \le&\frac{\|\boldsymbol{\theta}_{i,t}-\boldsymbol{\theta}\|^2 - \|\boldsymbol{\theta}_{i,t+1}-\boldsymbol{\theta}\|^2}{2\eta} + \frac{\eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t)\|^2}{2q_{i,t}}.\end{aligned} \tag{58}$$

Based on (as2), we have $\|\nabla\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t)\|^2 \le L^2$. Therefore, summing (58) over time from $t = 1$ to $t = T$ it can be concluded that

$$\sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t) \le \frac{\|\boldsymbol{\theta}\|^2 - \|\boldsymbol{\theta}_{i,T+1}-\boldsymbol{\theta}\|^2}{2\eta} + \sum_{t=1}^{T}\frac{\eta L^2}{2q_{i,t}}. \tag{59}$$

Putting $\boldsymbol{\theta} = \boldsymbol{\theta}_i^*$ in (59) and taking into account that $\|\boldsymbol{\theta}_{i,T+1}-\boldsymbol{\theta}\|^2 \ge 0$, we can write

$$\sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}_{i,t}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}^{\top}\mathbf{z}_i(\boldsymbol{x}_t), y_t) \le \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \sum_{t=1}^{T}\frac{\eta L^2}{2q_{i,t}} \tag{60}$$

which completes the proof of Lemma 7. ∎

Lemma 8 *Under (as1) and (as2), the following holds*

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t) \le \frac{\ln N}{\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)} \tag{61}$$

where η is the learning rate, η_e is the exploration rate, $q_{i,t} = \sum_{j=1}^{J} p_{j,t}\left(1-(1-\pi_{ij,t})^M\right)$ and N denotes the number of kernels.

Proof Let $W_t = \sum_{n=1}^{N} w_{n,t}$. For any t we find

$$\frac{W_{t+1}}{W_t} = \sum_{j=1}^{J} p_{j,t}\frac{W_{t+1}}{W_t} = \sum_{j=1}^{J} p_{j,t}\sum_{i=1}^{N}\frac{w_{i,t+1}}{W_t} = \sum_{j=1}^{J} p_{j,t}\sum_{i=1}^{N}\frac{w_{i,t}}{W_t}\exp(-\eta\ell_{i,t}). \tag{62}$$

Based on (3), we have

$$\frac{w_{i,t}}{W_t} = \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}, \forall j \in \{1, ..., J\}. \tag{63}$$

Combining (62) with (63) obtains

$$\frac{W_{t+1}}{W_t} = \sum_{j=1}^{J} p_{j,t}\sum_{i=1}^{N}\frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}\exp(-\eta\ell_{i,t}). \tag{64}$$

Using the inequality $e^{-x} \le 1 - x + \frac{1}{2}x^2, \forall x \ge 0$, (64) leads to

$$\frac{W_{t+1}}{W_t} \le \sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}\Big(1 - \eta\ell_{i,t} + \frac{1}{2}(\eta\ell_{i,t})^2\Big). \tag{65}$$

Taking logarithm from both sides of inequality (65), and use the fact that $1 + x \le e^x$, we have

$$\ln\frac{W_{t+1}}{W_t} \le \sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}\Big(-\eta\ell_{i,t} + \frac{1}{2}(\eta\ell_{i,t})^2\Big). \tag{66}$$

Summing (66) over t from 1 to T results in

$$\ln\frac{W_{T+1}}{W_1} \le \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}\Big(-\eta\ell_{i,t} + \frac{1}{2}(\eta\ell_{i,t})^2\Big). \tag{67}$$

Furthermore, recall the updating rule of $w_{i,T+1}$ in (37), for any i we have

$$\ln\frac{W_{T+1}}{W_1} \ge \ln\frac{w_{i,T+1}}{W_1} = -\ln N - \sum_{t=1}^{T}\eta\ell_{i,t}. \tag{68}$$

Combining (67) with (68) results in

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t}}{1-\eta_e^j}(\eta\ell_{i,t}) - \sum_{t=1}^{T}\eta\ell_{i,t}\\ &\le \ln N + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\frac{\eta_e^j}{N}}{1-\eta_e^j}(\eta\ell_{i,t}) + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}\Big(\frac{1}{2}(\eta\ell_{n,t})^2\Big).\end{aligned} \tag{69}$$

Multiplying both sides by $\frac{1-\eta_e^J}{\eta}$, we arrive at

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \pi_{ij,t}\frac{1-\eta_e^J}{1-\eta_e^j}\ell_{i,t} - \sum_{t=1}^{T}(1-\eta_e^J)\ell_{i,t}\\ \le&\frac{1-\eta_e^J}{\eta}\ln N + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e^j(1-\eta_e^J)}{N(1-\eta_e^j)}\ell_{i,t}\\ &+\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{(1-\eta_e^J)(\pi_{ij,t} - \frac{\eta_e^j}{N})}{1-\eta_e^j}(\frac{\eta}{2}\ell_{i,t}^2).\end{aligned} \tag{70}$$

Also, using the fact that $0 < \eta_e \le 1$ we can conclude that $1 - \eta_e^J < 1$ and for all $j \ge 1$, $\eta_e^j \le \eta_e$, the RHS of (70) can be upper bounded by

$$\begin{aligned}&\frac{1-\eta_e^J}{\eta}\ln N + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e^j(1-\eta_e^J)}{N(1-\eta_e^j)}\ell_{i,t} + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{(1-\eta_e^J)(\pi_{ij,t} - \frac{\eta_e^j}{N})}{1-\eta_e^j}(\frac{\eta}{2}\ell_{i,t}^2)\\ &\le \frac{\ln N}{\eta} + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e(1-\eta_e^J)}{N(1-\eta_e)}\ell_{i,t} + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\pi_{ij,t} - \frac{\eta_e^j}{N}}{1-\eta_e^j}(\frac{\eta}{2}\ell_{i,t}^2).\end{aligned} \tag{71}$$

Since $1-\eta_e^J=(1-\eta_e)(1+\ldots+\eta_e^{J-1})$ and $\eta_e\leq 1$, the following bound holds for the second term on the RHS of (71)

$$\begin{aligned}\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\eta_e(1-\eta_e^J)}{N(1-\eta_e)}\ell_{i,t}=&\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\eta_e(1+\ldots+\eta_e^{J-1})}{N}\ell_{i,t}\\ \leq&\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\eta_e J}{N}\ell_{i,t}.\end{aligned}\tag{72}$$

Meanwhile, as $\eta_e^J\leq\eta_e^j$ for all j, $1\leq j\leq J$, the LHS of (70) can be bounded by

$$\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\pi_{ij,t}\frac{1-\eta_e^J}{1-\eta_e^j}\ell_{i,t}-\sum_{t=1}^{T}(1-\eta_e^J)\ell_{i,t}\geq\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\pi_{ij,t}\ell_{i,t}-\sum_{t=1}^{T}\ell_{i,t}.\tag{73}$$

Combining (70), (71), (72) and (73), we can conclude that

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\pi_{ij,t}\ell_{i,t}-\sum_{t=1}^{T}\ell_{i,t}\\ \leq&\frac{\ln N}{\eta}+\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\eta_e J}{N}\ell_{i,t}+\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{1-\eta_e^j}(\frac{\eta}{2}\ell_{i,t}^2).\end{aligned}\tag{74}$$

Recall the probability of observing the loss of the i-th kernel at time t given in (21), the expected first and second moments of $\ell_{i,t}$ in (16) given the losses incurred up to time instant $t-1$, i.e., $\{\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_\tau),y_\tau)\}_{\tau=1}^{t-1}$ can be written as

$$\mathbb{E}[\ell_{i,t}]=\sum_{j=1}^{J}p_{j,t}\left(1-(1-\pi_{ij,t})^M\right)\frac{\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)}{q_{i,t}}=\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)\tag{75a}$$

$$\mathbb{E}[\ell_{n,t}^2]=\sum_{j=1}^{J}p_{j,t}\left(1-(1-\pi_{ij,t})^M\right)\frac{\mathcal{L}^2(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)}{q_{i,t}^2}=\frac{\mathcal{L}^2(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)}{q_{i,t}}\leq\frac{1}{q_{i,t}}.\tag{75b}$$

Based on (75b), the third term in the right hand side of (74) can be bounded as follows

$$\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{1-\eta_e^j}(\frac{\eta}{2}\ell_{n,t}^2)\leq\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{1-\eta_e^j}(\frac{\eta}{2q_{i,t}}).\tag{76}$$

Taking the expected value of (74) at each time t given$\{\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_\tau),y_\tau)\}_{\tau=1}^{t-1}$ and combining with (75a) and (76) we have

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\pi_{ij,t}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)-\sum_{t=1}^{T}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)\\ \leq&\frac{\ln N}{\eta}+\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\eta_e J}{N}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t)+\sum_{t=1}^{T}\sum_{j=1}^{J}p_{j,t}\sum_{i=1}^{N}\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{1-\eta_e^j}(\frac{\eta}{2q_{i,t}}).\end{aligned}\tag{77}$$

Since $\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{(1-\eta_e^j)q_{i,t}} \leq \frac{\pi_{ij,t}}{(1-\eta_e)q_{i,t}}$, replace $\frac{\pi_{ij,t}-\frac{\eta_e^j}{N}}{q_{i,t}(1-\eta_e^j)}$ by $\frac{\pi_{ij,t}}{(1-\eta_e)q_{i,t}}$, the inequality in (77) still holds

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \pi_{ij,t}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t)\\ \leq&\frac{\ln N}{\eta} + \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e J}{N}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t), y_t) + \frac{\eta}{2(1-\eta_e)}\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N}\frac{\pi_{ij,t}}{q_{i,t}}. \end{aligned} \tag{78}$$

Moreover, based on (21), the probability $q_{i,t}$ can be bounded from below as

$$\begin{aligned} q_{i,t} &= \sum_{j=1}^{J} p_{j,t}\left(1-(1-\pi_{ij,t})^M\right)\\ &= \sum_{j=1}^{J} p_{j,t}\pi_{ij,t}\left(1+(1-\pi_{ij,t})+\ldots+(1-\pi_{ij,t})^{M-1}\right) > \sum_{j=1}^{J} p_{j,t}\pi_{ij,t} \end{aligned} \tag{79}$$

Therefore, according to (79), for the third term in the right hand side of (78) we have

$$\frac{\eta}{2(1-\eta_e)}\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N}\frac{\pi_{ij,t}}{q_{i,t}} < \frac{\eta NT}{2(1-\eta_e)}. \tag{80}$$

Furthermore, based on that $\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq 1$ in (as2), the following inequality holds

$$\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e J}{N}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq \sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \frac{\eta_e J}{N} = \eta_e JT. \tag{81}$$

From (78), (80) and (81), we can conclude that

$$\sum_{t=1}^{T}\sum_{j=1}^{J} p_{j,t} \sum_{n=1}^{N} \pi_{ij,t}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq \frac{\ln N}{\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)}. \tag{82}$$

According to the procedure of generating the graph $\mathcal{B}_t$ which is presented in Algorithm 2, for each selective node $v_{s,j}$ a subset of kernels is chosen using PMF $\pi_{ij,t}$ in M independent trials. In fact, a subset of kernels is assigned to each node $v_{s,j}$ in M independent trials and in each trial one kernel is assigned and its associated entry in the sub-adjacency matrix A becomes 1. Now, let b_i represents the frequency that the i-th kernel is chosen in M independent trials. Thus, $\{b_i\}_{i=1}^N$ can be viewed as the solution to the following linear equation

$$b_1 + \ldots + b_N = M, \quad \text{s.t.} \quad b_i \geq 0, b_i \in \mathbb{N} \tag{83}$$

where $\mathbb{N}$ denotes the set of natural numbers. There are $\binom{N+M-1}{N}$ different solutions for (83). Let, $\{b_{i,k}\}_{i=1}^N$ denotes the k-th set of solution for (83). Based on Jensen's inequality, for the expected value of $\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)$ we have

$$\begin{aligned}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] &= \sum_{j=1}^{J} p_{j,t} \sum_{k=1}^{\binom{N+M-1}{N}} \left(\prod_{i=1}^{N} (\pi_{ij,t})^{b_{i,k}}\right) \mathcal{L}(\sum_{i\in\mathcal{S}_t} \bar{w}_{i,t}\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \\ &\leq \sum_{j=1}^{J} p_{j,t} \sum_{k=1}^{\binom{N+M-1}{N}} \left(\prod_{i=1}^{N} (\pi_{ij,t})^{b_{i,k}}\right) \sum_{i\in\mathcal{S}_t} \bar{w}_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t). \end{aligned} \tag{84}$$

Also, considering (84) and the fact that $\bar{w}_{i,t} \leq 1$, we can conclude that

$$\begin{aligned}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] &\leq \sum_{j=1}^{J} p_{j,t} \sum_{k=1}^{\binom{N+M-1}{N}} \left(\prod_{i=1}^{N} (\pi_{ij,t})^{b_{i,k}}\right) \sum_{i\in\mathcal{S}_t} \bar{w}_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \\ &\leq \sum_{j=1}^{J} p_{j,t} \sum_{k=1}^{\binom{N+M-1}{N}} \left(\prod_{i=1}^{N} (\pi_{ij,t})^{b_{i,k}}\right) \sum_{i\in\mathcal{S}_t} \mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t). \end{aligned} \tag{85}$$

Note that the number of ways to solve (83) when the i-th kernel is chosen for at least one time equals to the number of ways to solve the following problem

$$\tilde{b}_{1,i} + \ldots + \tilde{b}_{N,i} = M - 1, \quad \text{s.t. } \tilde{b}_{m,i} \geq 0,\ \tilde{b}_{m,i} \in \mathbb{N}. \tag{86}$$

There are $\binom{N+M-2}{N}$ different solutions for (86). Let $\{\tilde{b}^{(k)}_{m,i}\}_{i=1}^{N}$ denotes k-th set of solution for (86). Therefore, based on this, from (85) we can conclude the following equality

$$\begin{aligned}&\sum_{j=1}^{J} p_{j,t} \sum_{k=1}^{\binom{N+M-1}{N}} \left(\prod_{i=1}^{N} (\pi_{ij,t})^{b_{i,k}}\right) \sum_{i\in\mathcal{S}_t} \mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \\ =&\sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \pi_{ij,t} \sum_{k=1}^{\binom{N+M-2}{N}} \left(\prod_{m=1}^{N} (\pi_{mj,t})^{\tilde{b}^{(k)}_{m,i}}\right) \mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \end{aligned} \tag{87}$$

where $\sum_{k=1}^{\binom{N+M-2}{N}} \left(\prod_{m=1}^{N} (\pi_{mj,t})^{\tilde{b}^{(k)}_{m,i}}\right)$ is the total probability of all $\binom{N+M-2}{N}$ possible solutions of (86). Therefore, $\sum_{k=1}^{\binom{N+M-2}{N}} \left(\prod_{m=1}^{N} (\pi_{mj,t})^{\tilde{b}^{(k)}_{m,n}}\right) = 1$. Substituting (87) into (84), we obtain

$$\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] \leq \sum_{j=1}^{J} p_{j,t} \sum_{i=1}^{N} \pi_{ij,t}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t). \tag{88}$$

Combining (82) with (88) leads to

$$\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq \frac{\ln N}{\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)} \tag{89}$$

which concludes to proof of Lemma 8. ■

From Lemma 7 and Lemma 8, we conclude that for any $i : 1 \leq i \leq N$ we have

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t), y_t)\\ \leq & \frac{\ln N}{\eta} + \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)} + \frac{\eta}{2}\sum_{t=1}^{T}\frac{L^2}{q_{i,t}}.\end{aligned} \tag{90}$$

From (79) and the facts that $p_{j,t} > \frac{\eta_e}{J}$ and $\pi_{ij,t} > \frac{\eta_e^j}{N}$, the following inequality can be concluded

$$q_{i,t} \geq \sum_{j=1}^{J} p_{j,t}\pi_{ij,t} > p_{1,t}\pi_{i1,t} > \frac{\eta_e^2}{NJ}. \tag{91}$$

Therefore, we find $q_{i,t} > \frac{\eta_e^2}{NJ}$, $\forall i : 1 \leq i \leq N$, $\forall t : 1 \leq t \leq T$. Combining (90) and (91) we can conclude that

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t), y_t)\\ < & \frac{\ln N}{\eta} + \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \eta_e JT + \frac{\eta NT}{2(1-\eta_e)} + \frac{\eta L^2 NJT}{2\eta_e^2}.\end{aligned} \tag{92}$$

Hence, Lemma 1 is proved.

Appendix C. Proof of Theorem 2

To prove Theorem 2, the following lemma is exploited.

Lemma 9 *For the optimal function estimator in $\mathbb{H}_i$ expressed as $f_i^*(\boldsymbol{x}) := \sum_{t=1}^{T}\alpha_{i,t}^*\kappa_i(\boldsymbol{x}, \boldsymbol{x}_t)$ and its RF-based approximant $\hat{f}_i^*(\boldsymbol{x}, \boldsymbol{x}_t) = \sum_{t=1}^{T}\alpha_{i,t}^*\mathbf{z}_i^\top(\boldsymbol{x})\mathbf{z}_i(\boldsymbol{x}_t)$, the following bound holds with probability at least $1 - 2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$*

$$\left|\sum_{t=1}^{T}\mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(f_i^*(\boldsymbol{x}_t), y_t)\right| \leq \epsilon LTC \tag{93}$$

where the equality happens if we have $C := \max_i \sum_{t=1}^{T}|\alpha_{i,t}^*|$.

Proof For a given shift invariant kernel κ_i, the maximum point-wise error of the random feature kernel approximant is uniformly bounded with probability at least $1-2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$, by Rahimi and Recht (2007)

$$\sup_{\boldsymbol{x}_j, \boldsymbol{x}_k \in \mathcal{X}} |\mathbf{z}_i^\top(\boldsymbol{x}_j)\mathbf{z}_i(\boldsymbol{x}_k) - \kappa_i(\boldsymbol{x}_j, \boldsymbol{x}_k)| < \epsilon \tag{94}$$

where σ_i^2 is the second moment of the Fourier transform of κ_i. Therefore, under (a3) this implies that $\sup_{\boldsymbol{x}_\tau,\boldsymbol{x}_t\in\mathcal{X}} \mathbf{z}_i^\top(\boldsymbol{x}_\tau)\mathbf{z}_i(\boldsymbol{x}_t) \leq 1+\epsilon$ holds with probability at least $1 - 2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$. Let $C := \max_n \sum_{t=1}^T |\alpha_{n,t}^*|$. Hence, $\|\boldsymbol{\theta}_j^*\|^2$ can be bounded from above as

$$\|\boldsymbol{\theta}_j^*\|^2 \leq \|\sum_{t=1}^T \alpha_{j,t}^*\mathbf{z}_j(\boldsymbol{x}_t)\|^2 \leq |\sum_{t=1}^T\sum_{\tau=1}^T \alpha_{j,t}^*\alpha_{j,\tau}^*\mathbf{z}_j^\top(\boldsymbol{x}_t)\mathbf{z}_j(\boldsymbol{x}_\tau)| \leq (1+\epsilon)C^2 \tag{95}$$

with probability at least $1 - 2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$. Moreover, using the triangle inequality yields

$$\left|\sum_{t=1}^T \mathcal{L}(\hat{f}_j^*(\boldsymbol{x}_t),y_t) - \sum_{t=1}^T \mathcal{L}(f_j^*(\boldsymbol{x}_t),y_t)\right| \leq \sum_{t=1}^T \left|\mathcal{L}(\hat{f}_j^*(\boldsymbol{x}_t),y_t) - \mathcal{L}(f_j^*(\boldsymbol{x}_t),y_t)\right|. \tag{96}$$

According to Lipschitz continuity of the loss function, it can be concluded that

$$\begin{aligned}&\sum_{t=1}^T \left|\mathcal{L}(\hat{f}_j^*(\boldsymbol{x}_t),y_t) - \mathcal{L}(f_j^*(\boldsymbol{x}_t),y_t)\right| \\ &\leq \sum_{t=1}^T L\left|\sum_{\tau=1}^T \alpha_{j,\tau}^*\mathbf{z}_j^\top(\boldsymbol{x}_\tau)\mathbf{z}_n(\boldsymbol{x}_t) - \sum_{\tau=1}^T \alpha_{j,\tau}^*\kappa_j(\boldsymbol{x}_\tau,\boldsymbol{x}_t)\right|.\end{aligned} \tag{97}$$

Using the Cauchy-Schwartz inequality, we can write

$$\begin{aligned}&\sum_{t=1}^T L\left|\sum_{\tau=1}^T \alpha_{j,\tau}^*\mathbf{z}_j^\top(\boldsymbol{x}_\tau)\mathbf{z}_n(\boldsymbol{x}_t) - \sum_{\tau=1}^T \alpha_{j,\tau}^*\kappa_j(\boldsymbol{x}_\tau,\boldsymbol{x}_t)\right| \\ &\leq \sum_{t=1}^T L\sum_{\tau=1}^T |\alpha_{j,\tau}^*|\left|\mathbf{z}_j^\top(\boldsymbol{x}_\tau)\mathbf{z}_j(\boldsymbol{x}_t) - \kappa_j(\boldsymbol{x}_\tau,\boldsymbol{x}_t)\right|.\end{aligned} \tag{98}$$

Using (94) and (98), we can conclude that the inequality

$$\left|\sum_{t=1}^T \mathcal{L}(\hat{f}_j^*(\boldsymbol{x}_t),y_t) - \sum_{t=1}^T \mathcal{L}(f_j^*(\boldsymbol{x}_t),y_t)\right| \leq \epsilon LTC \tag{99}$$

holds with probability at least $1 - 2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$. ■

Combining Lemma 1 with Lemma 9 and (95), it can be concluded that the following bound holds with probability at least $1 - 2^8(\frac{\sigma_n}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$,

$$\begin{aligned}&\sum_{t=1}^T \mathbb{E}_t[\mathcal{L}(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)] - \sum_{t=1}^T \mathcal{L}(f_i^*(\boldsymbol{x}_t),y_t) \\ =&\sum_{t=1}^T \mathbb{E}_t[\mathcal{L}(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)] - \sum_{t=1}^T \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t),y_t) + \sum_{t=1}^T \mathcal{L}(\hat{f}_i^*(\boldsymbol{x}_t),y_t) - \sum_{t=1}^T \mathcal{L}(f_i^*(\boldsymbol{x}_t),y_t) \\ <&\frac{\ln N}{\eta} + \frac{\|\boldsymbol{\theta}_i^*\|^2}{2\eta} + \eta_e JT + \epsilon LTC + \frac{\eta NT}{2(1-\eta_e)} + \frac{\eta L^2 NJT}{2\eta_e^2}\end{aligned} \tag{100}$$

which completes the proof of Theorem 2.

Appendix D. Proof of Lemma 4

According to (2), we obtain

$$\frac{1}{\mathcal{U}_d}\int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x} = \frac{1}{\mathcal{U}_d}\int |\sum_{t=1}^{T}\alpha_{i,t}\kappa_i(\boldsymbol{x},\boldsymbol{x}_t) - \sum_{t=1}^{T}\alpha_{j,t}\kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2 d\boldsymbol{x}. \tag{101}$$

Applying Arithmetic Mean-Geometric Mean inequality on the right hand side of (101), we find

$$\begin{aligned}&\frac{1}{\mathcal{U}_d}\int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x}\\ \leq&\frac{2}{\mathcal{U}_d}\int\left(|\sum_{t=1}^{T}\alpha_{i,t}\left(\kappa_i(\boldsymbol{x},\boldsymbol{x}_t) - \kappa_j(\boldsymbol{x},\boldsymbol{x}_t)\right)|^2 + |\sum_{t=1}^{T}(\alpha_{j,t}-\alpha_{i,t})\kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2\right)d\boldsymbol{x}.\end{aligned} \tag{102}$$

Using Cauchy-Schwartz inequality, (102) can be further relaxed to

$$\begin{aligned}&\frac{1}{\mathcal{U}_d}\int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x}\\ \leq&\frac{2}{\mathcal{U}_d}\int(\sum_{t=1}^{T}|\alpha_{i,t}|^2)(\sum_{t=1}^{T}|\kappa_i(\boldsymbol{x},\boldsymbol{x}_t) - \kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2)d\boldsymbol{x}\\ &+\frac{2}{\mathcal{U}_d}\int(\sum_{t=1}^{T}|\alpha_{j,t}-\alpha_{i,t}|^2)(\sum_{t=1}^{T}|\kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2)d\boldsymbol{x}.\end{aligned} \tag{103}$$

Considering the fact that $C_m := \max_i \sum_{t=1}^{T}|\alpha_{i,t}|^2$, from (103) it can be written that

$$\begin{aligned}&\frac{1}{\mathcal{U}_d}\int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x}\\ \leq&\frac{2C_m}{\mathcal{U}_d}\sum_{t=1}^{T}\int|\kappa_i(\boldsymbol{x},\boldsymbol{x}_t) - \kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2 d\boldsymbol{x} + \frac{4C_m}{\mathcal{U}_d}\sum_{t=1}^{T}\int|\kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2 d\boldsymbol{x}.\end{aligned} \tag{104}$$

Furthermore, based on (104) and the fact that $|\kappa_j(\boldsymbol{x},\boldsymbol{x}_t)|^2 \leq 1$, we can infer that

$$\frac{1}{\mathcal{U}_d}\int |\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})|^2 d\boldsymbol{x} \leq \frac{2C_m}{\mathcal{U}_d}\sum_{t=1}^{T}\left(\Delta_S(\kappa_i,\kappa_j) + 2\mathcal{U}_d\right) \tag{105}$$

which proves Lemma 4.

Appendix E. Proof of Theorem 5

Furthermore, in order to prove Theorem 5, the following intermediate Lemma is also proved.

Lemma 10 *The following inequality holds under (as1) and (as2)*

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{RF}(\boldsymbol{x}_t),y_t)] - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{RF,i}(\boldsymbol{x}_t),y_t) \leq \frac{\ln N}{\eta} + \eta(1+\frac{N}{2}-\frac{\eta}{2})T \tag{106}$$

where $\mathcal{L}_i(\hat{f}_{RF}(\boldsymbol{x}_t),y_t)$ denote the loss of function approximation when v_i is drawn.

Proof For any t, we can write

$$\frac{U_{t+1}}{U_t} = \sum_{i=1}^{N} \frac{u_{i,t+1}}{U_t} = \sum_{i=1}^{N} \frac{u_{i,t}}{U_t} \exp(-\eta \hat{\ell}_{i,t}). \tag{107}$$

Based on (33), we have $\frac{u_{i,t}}{U_t} = \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi}$ and as a result (107) can be rewritten as

$$\frac{U_{t+1}}{U_t} = \sum_{i=1}^{N} \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi} \exp(-\eta \hat{\ell}_{i,t}). \tag{108}$$

Using the inequality $e^{-x} \leq 1 - x + \frac{1}{2}x^2, \forall x \geq 0$ and (108), it can be concluded that

$$\frac{U_{t+1}}{U_t} \leq \sum_{i=1}^{N} \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi} \Big(1 - \eta \hat{\ell}_{i,t} + \frac{1}{2}(\eta \hat{\ell}_{i,t})^2\Big). \tag{109}$$

Taking logarithm from both sides of inequality (109), and use the fact that $1 + x \leq e^x$, we arrive at

$$\ln \frac{U_{t+1}}{U_t} \leq \sum_{i=1}^{N} \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi} \Big(-\eta \hat{\ell}_{i,t} + \frac{1}{2}(\eta \hat{\ell}_{i,t})^2\Big). \tag{110}$$

Summing (110) over t leads to

$$\ln \frac{U_{T+1}}{U_1} \leq \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi} \Big(-\eta \hat{\ell}_{i,t} + \frac{1}{2}(\eta \hat{\ell}_{i,t})^2\Big). \tag{111}$$

Furthermore, $\ln \frac{U_{T+1}}{U_1}$ can be bounded from below as

$$\ln \frac{U_{T+1}}{U_1} \geq \ln \frac{u_{i,T+1}}{U_1} = -\sum_{t=1}^{T} \eta \hat{\ell}_{i,t} - \ln N \tag{112}$$

for any i such that $1 \leq i \leq N$. Combining (111) with (112), we obtain

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{i=1}^{N} \frac{p_{i,t}\eta}{1-\xi} \hat{\ell}_{i,t} - \sum_{t=1}^{T} \eta \hat{\ell}_{i,t} \\ \leq &\ln N + \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\eta \xi \mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{(1-\xi)|\mathbb{D}|} \hat{\ell}_{i,t} + \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{1-\xi} \Big(\frac{1}{2}(\eta \hat{\ell}_{i,t})^2\Big). \end{aligned} \tag{113}$$

Multiplying both sides by $\frac{1-\xi}{\eta}$ it can be concluded that

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{i=1}^{N} p_{i,t} \hat{\ell}_{i,t} - \sum_{t=1}^{T} \hat{\ell}_{i,t} \\ \leq &\frac{\ln N}{\eta} + \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\xi \mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{|\mathbb{D}|} \hat{\ell}_{i,t} + \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\eta (p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}})}{2} \hat{\ell}_{i,t}^2. \end{aligned} \tag{114}$$

In addition, taking the expectation of $\hat{\ell}_{i,t}$ and $\hat{\ell}_{i,t}^2$, we get

$$\mathbb{E}_t[\hat{\ell}_{i,t}] = p_{i,t}\frac{\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)}{p_{i,t}} = \mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \tag{115a}$$

$$\mathbb{E}_t[\hat{\ell}_{i,t}^2] = p_{i,t}\frac{\mathcal{L}^2(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)}{p_{i,t}^2} = \frac{\mathcal{L}^2(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)}{p_{i,t}} \leq \frac{1}{p_{i,t}} \tag{115b}$$

Thus, taking the expectation from both sides of (114) leads to

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{i=1}^{N} p_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)\\ \leq&\frac{\ln N}{\eta} + \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{\xi\mathbf{1}_{\mathrm{i}\in\mathbb{D}}}{|\mathbb{D}|}\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t) + \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{\eta(p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}})}{2p_{i,t}}.\end{aligned} \tag{116}$$

Taking into account that $\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t) \leq 1$ and based on (116) we can write

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{i=1}^{N} p_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t)\\ \leq&\frac{\ln N}{\eta} + \xi T + \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{\eta(p_{i,t} - \frac{\xi}{|\mathbb{D}|}\mathbf{1}_{\mathrm{i}\in\mathbb{D}})}{2p_{i,t}}.\end{aligned} \tag{117}$$

Moreover, using (117) and the fact that $p_{i,t} \leq 1$, it can be concluded that

$$\sum_{t=1}^{T}\sum_{i=1}^{N} p_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq \frac{\ln N}{\eta} + (\xi + \frac{\eta N}{2} - \frac{\eta\xi}{2})T. \tag{118}$$

Furthermore, the expected loss incurred by OMKL-SFG given observed losses in prior time instants can be expressed as

$$\begin{aligned}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] &= \sum_{i=1}^{N} p_{i,t}\mathcal{L}(\sum_{j\in\mathbb{N}_{i,t}^{\mathrm{out}}}\frac{w_{j,t}}{\sum_{k\in\mathbb{N}_{i,t}^{\mathrm{out}}} w_{k,t}}\hat{f}_{\mathrm{RF},j}(\boldsymbol{x}_t), y_t)\\ &= \sum_{i=1}^{N} p_{i,t}\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t).\end{aligned} \tag{119}$$

Therefore, from (118) and (119), it can be inferred that

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t), y_t)] - \sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\mathrm{RF},i}(\boldsymbol{x}_t), y_t) \leq \frac{\ln N}{\eta} + (\xi + \frac{\eta N}{2} - \frac{\eta\xi}{2})T \tag{120}$$

which establishes the Lemma 10. ■

Furthermore, to prove Theorem 5, we prove the following Lemma.

Lemma 11 *For any $v_i \in \mathcal{V}$ and any $j \in \mathbb{N}_i^{out}$, it can be written that*

$$\sum_{t=1}^{T} \mathcal{L}_i(\hat{f}_{RF}(\boldsymbol{x}_t), y_t) - \sum_{t=1}^{T} \mathcal{L}(\hat{f}_{RF,j}(\boldsymbol{x}_t), y_t) \leq \frac{\ln |\mathbb{N}_i^{out}|}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \frac{1}{\bar{q}_{i,t}} \tag{121}$$

where $\frac{1}{\bar{q}_{i,t}} = \sum_{j \in \mathbb{N}_i^{out}} \frac{w_{j,t}}{q_{j,t} W_{i,t}}$.

Proof Let $W_{i,t} = \sum_{j \in \mathbb{N}_i^{\text{out}}} w_{j,t}$. For $v_i \in \mathcal{V}$ we find

$$\frac{W_{i,t+1}}{W_{i,t}} = \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t+1}}{W_{i,t}} = \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \exp(-\eta \ell_{j,t}). \tag{122}$$

The following inequality can be obtained using the inequality $e^{-x} \leq 1 - x + \frac{1}{2}x^2, \forall x \geq 0$ as follows

$$\frac{W_{i,t+1}}{W_{i,t}} \leq \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \bigg(1 - \eta \ell_{j,t} + \frac{1}{2}(\eta \ell_{j,t})^2\bigg). \tag{123}$$

Taking the logarithm from both sides of (123) and using the inequality $1 + x \leq e^x$, we get

$$\ln \frac{W_{i,t+1}}{W_{i,t}} \leq \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \bigg(-\eta \ell_{j,t} + \frac{1}{2}(\eta \ell_{j,t})^2\bigg). \tag{124}$$

Summing (124) over time, we obtain

$$\ln \frac{W_{i,T+1}}{W_{i,1}} \leq \sum_{t=1}^{T} \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \bigg(-\eta \ell_{j,t} + \frac{1}{2}(\eta \ell_{j,t})^2\bigg). \tag{125}$$

Moreover, for any $j \in \mathbb{N}_i^{\text{out}}$, $\ln \frac{W_{i,T+1}}{W_{i,1}}$ can be bounded from below as

$$\ln \frac{W_{i,T+1}}{W_{i,1}} \geq \ln \frac{w_{j,T+1}}{W_{i,1}} = -\sum_{t=1}^{T} \eta \ell_{j,t} - \ln |\mathbb{N}_i^{\text{out}}| \tag{126}$$

Combining (125) with (126), it can concluded that

$$\sum_{t=1}^{T} \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \ell_{j,t} - \sum_{t=1}^{T} \ell_{j,t} \leq \frac{\ln |\mathbb{N}_i^{\text{out}}|}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{j \in \mathbb{N}_i^{\text{out}}} \frac{w_{j,t}}{W_{i,t}} \ell_{j,t}^2. \tag{127}$$

For the expected value of $\ell_{i,t}$ and $\ell_{i,t}^2$, we have

$$\mathbb{E}_t[\ell_{j,t}] = \sum_{k \in \mathbb{N}_i^{\text{out}}} p_{k,t} \frac{\mathcal{L}(\boldsymbol{\theta}_{j,t}^\top \mathbf{z}_j(\boldsymbol{x}_t), y_t)}{q_{j,t}} = \mathcal{L}(\boldsymbol{\theta}_{j,t}^\top \mathbf{z}_j(\boldsymbol{x}_t), y_t) \tag{128a}$$

$$\mathbb{E}_t[\ell_{j,t}^2] = \sum_{k \in \mathbb{N}_i^{\text{out}}} p_{k,t} \frac{\mathcal{L}^2(\boldsymbol{\theta}_{j,t}^\top \mathbf{z}_j(\boldsymbol{x}_t), y_t)}{q_{j,t}^2} = \frac{\mathcal{L}^2(\boldsymbol{\theta}_{j,t}^\top \mathbf{z}_j(\boldsymbol{x}_t), y_t)}{q_{j,t}} \leq \frac{1}{q_{j,t}} \tag{128b}$$

Taking the expectation from (127), we get

$$\sum_{t=1}^{T}\sum_{j\in\mathbb{N}_i^{\text{out}}}\frac{w_{j,t}}{W_{i,t}}\mathcal{L}(\boldsymbol{\theta}_{j,t}^{\top}\mathbf{z}_j(\boldsymbol{x}_t),y_t)-\sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}_{j,t}^{\top}\mathbf{z}_j(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{\ln|\mathbb{N}_i^{\text{out}}|}{\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\sum_{j\in\mathbb{N}_i^{\text{out}}}\frac{w_{j,t}}{q_{j,t}W_{i,t}}. \tag{129}$$

Let $\frac{1}{\bar{q}_{i,t}}=\sum_{j\in\mathbb{N}_i^{\text{out}}}\frac{w_{j,t}}{q_{j,t}W_{i,t}}$ which is the weighted sum of $\frac{1}{q_{j,t}}$ such that $j\in\mathbb{N}_i^{\text{out}}$. Furthermore, according to (34), the loss $\mathcal{L}_i(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)$ can be written as

$$\mathcal{L}_i(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)=\mathcal{L}(\sum_{j\in\mathbb{N}_i^{\text{out}}}\frac{w_{j,t}}{W_{i,t}}\hat{f}_{\text{RF},j}(\boldsymbol{x}_t),y_t). \tag{130}$$

Based on the Jensen's inequality $\mathcal{L}_i(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)$ can be bounded from above as

$$\mathcal{L}_i(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)\leq\sum_{j\in\mathbb{N}_i^{\text{out}}}\frac{w_{j,t}}{W_{i,t}}\mathcal{L}(\hat{f}_{\text{RF},j}(\boldsymbol{x}_t),y_t). \tag{131}$$

Using (129) and (131), we can conclude that

$$\sum_{t=1}^{T}\mathcal{L}_i(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)-\sum_{t=1}^{T}\mathcal{L}(\boldsymbol{\theta}_{j,t}^{\top}\mathbf{z}_j(\boldsymbol{x}_t),y_t)\leq\frac{\ln|\mathbb{N}_i^{\text{out}}|}{\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\frac{1}{\bar{q}_{i,t}} \tag{132}$$

which proves the Lemma 11. ■

Combining Lemma 10 with Lemma 11, for any $v_j\in\mathcal{V}$ and any $i\in\mathbb{N}_j^{\text{in}}$ we obtain

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)]-\sum_{t=1}^{T}\mathcal{L}(\hat{f}_{\text{RF},j}(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{\ln N|\mathbb{N}_i^{\text{out}}|}{\eta}+(\xi+\frac{\eta N}{2}-\frac{\eta\xi}{2})T+\frac{\eta}{2}\sum_{t=1}^{T}\frac{1}{\bar{q}_{i,t}}. \tag{133}$$

In addition, combining Lemma 7 with (133), for any $v_j\in\mathcal{V}$ and any $i\in\mathbb{N}_j^{\text{in}}$ we can write

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\text{RF}}(\boldsymbol{x}_t),y_t)]-\sum_{t=1}^{T}\mathcal{L}(\hat{f}_j^{*}(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{\ln N|\mathbb{N}_i^{\text{out}}|}{\eta}+\frac{\|\boldsymbol{\theta}_j^{*}\|^2}{2\eta}+(\xi+\frac{\eta N}{2}-\frac{\eta\xi}{2})T+\frac{\eta}{2}\sum_{t=1}^{T}(\frac{1}{\bar{q}_{i,t}}+\frac{L^2}{q_{j,t}}) \tag{134}$$

We use the above inequality as a step-stone to prove Theorem 5.

Therefore, combining (134) with (95) and (99), it can be inferred that the following inequality

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t),y_t)]-\sum_{t=1}^{T}\mathcal{L}(f_j^*(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{\ln N|\mathbb{N}_i^{\mathrm{out}}|}{\eta}+\frac{(1+\epsilon)C^2}{2\eta}+\epsilon LTC+(\xi+\frac{\eta N}{2}-\frac{\eta\xi}{2})T+\frac{\eta}{2}\sum_{t=1}^{T}(\frac{1}{\bar{q}_{i,t}}+\frac{L^2}{q_{j,t}}) \tag{135}$$

holds for any $v_j\in\mathcal{V}$ and any $i\in\mathbb{N}_j^{\mathrm{in}}$ with probability at least $1-2^8(\frac{\sigma_i}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$. This completes the proof of Theorem 5.

Appendix F. Proof of Theorem 6

According to Theorem 5, the following inequality

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t),y_t)]-\sum_{t=1}^{T}\mathcal{L}(f^*(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{\ln N|\mathbb{N}_i^{\mathrm{out}}|}{\eta}+\frac{(1+\epsilon)C^2}{2\eta}+\epsilon LTC+(\xi+\frac{\eta N}{2}-\frac{\eta\xi}{2})T+\frac{\eta}{2}\sum_{t=1}^{T}(\frac{1}{\bar{q}_{i,t}}+\frac{L^2}{q_{j^*,t}}) \tag{136}$$

holds with probability at least $1-2^8(\frac{\sigma_{j^*}}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$ for any $\epsilon>0$ and any $i\in\mathbb{N}_{j^*}^{\mathrm{in}}$. When $\mathcal{G}'_t$ is generated by Algorithm 6 as the feedback graph, $\mathbb{D}'_t$ is a dominating set of $\mathcal{G}'_t$. Furthermore, $k\in\mathbb{D}'_t$ if $p_{k,t}\geq\beta$. Based on (36), if $i\in\mathbb{N}_{k,t}^{\mathrm{in}}$ (i.e. in-neighborhood of v_k in $\mathcal{G}'_t$), $q_{k,t}\geq p_{i,t}$. Also, considering the condition $\beta\leq\frac{1}{N}$, it is ensured that $\mathbb{D}'_t$ is not an empty set. Moreover, each node v_k in $\mathcal{G}'_t$ is out-neighbor of at least one node in $\mathbb{D}'_t$. Thus, we can conclude that $q_{k,t}\geq\beta$, $\forall v_k\in\mathcal{V}$. Hence, it can be written that

$$\sum_{t=1}^{T}(\frac{1}{\bar{q}_{i,t}}+\frac{L^2}{q_{j^*,t}})\geq\sum_{t=1}^{T}(\sum_{j\in\mathbb{N}_i^{\mathrm{out}}}\frac{w_{j,t}}{W_{i,t}\beta}+\frac{L^2}{\beta})=\frac{(L^2+1)T}{\beta}. \tag{137}$$

Furthermore, since $|\mathbb{N}_i^{\mathrm{out}}|\leq N$, we have $N|\mathbb{N}_i^{\mathrm{out}}|\leq N^2$. Combining (136) with (137), it can be inferred that in this case the stochastic regret of OMKL-SFG-R satisfies

$$\sum_{t=1}^{T}\mathbb{E}_t[\mathcal{L}(\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_t),y_t)]-\sum_{t=1}^{T}\mathcal{L}(f^*(\boldsymbol{x}_t),y_t)$$
$$\leq\frac{2\ln N}{\eta}+\frac{(1+\epsilon)C^2}{2\eta}+\epsilon LTC+(\xi+\frac{\eta}{2}\frac{L^2+N\beta+1}{\beta}-\frac{\eta\xi}{2})T \tag{138}$$

with probability at least $1-2^8(\frac{\sigma_{j^*}}{\epsilon})^2\exp(-\frac{D\epsilon^2}{4d+8})$ under (as1)-(as3) for any $\epsilon>0$ and any $\beta\leq(1-\xi)\max_k\frac{u_{k,t}}{U_t}+\frac{\xi}{N}$. This completes the proof of Theorem 6.