Fourier Learning with Cyclical Data

Yingxiang Yang[*1] **Zhihan Xiong**[*2] **Tianyi Liu**[*1] **Taiqing Wang**[1] **Chong Wang**[1]

Abstract

Many machine learning models for online applications, such as recommender systems, are often trained on data with cyclical properties. These data sequentially arrive from a time-varying distribution that is periodic in time. Existing algorithms either use streaming learning to track a time-varying set of optimal model parameters, yielding a dynamic regret that scales linearly in time; or partition the data of each cycle into multiple segments and train a separate model for each—a pluralistic approach that is computationally and storage-wise expensive. In this paper, we have designed a novel approach to overcome the aforementioned shortcomings. Our method, named "Fourier learning", encodes the periodicity into the model representation using a partial Fourier sequence, and trains the coefficient functions modeled by neural networks. Particularly, we design a Fourier multi-layer perceptron (F-MLP) that can be trained on streaming data with stochastic gradient descent (streaming-SGD), and we derive its convergence guarantees. We demonstrate Fourier learning's better performance with extensive experiments on synthetic and public datasets, as well as on a large-scale recommender system that is updated in real-time, and trained with tens of millions of samples per day.

1. Introduction

Cyclical data are an important component in a wide range of machine learning applications. In large-scale recommender systems such as YouTube and TikTok, users usually log into the platform during a relatively fixed time window each day (e.g. before bed or after work), resulting in a strong cyclical pattern in the system's revenue. In federated learning (Kairouz et al., 2019), the training data are cyclical in nature as the availability of each device is usually fixed throughout the day (Eichner et al., 2019). In financial markets, asset prices rise and fall periodically on a yearly basis, a phenomenon commonly known to the investors as "seasonality" (Gultekin & Gultekin, 1983). In search engines, the number of hits for certain keywords can also display periodic patterns (Tracà et al., 2021). Figure 1 depicts a few of these applications. How to exploit the periodicity within the training data to learn a better prediction model is an important issue for these applications.

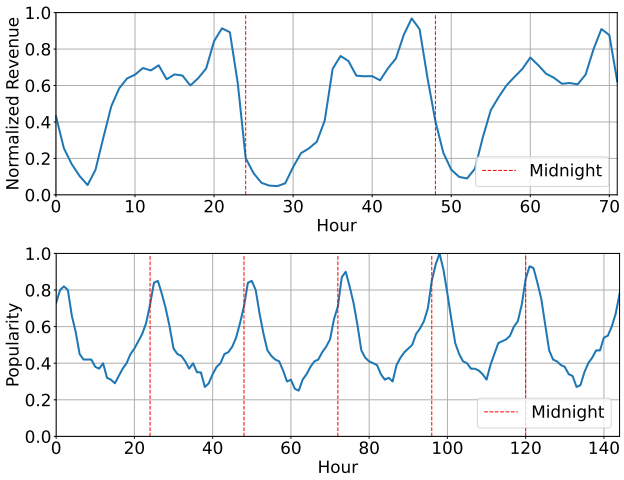

Figure 1. Top: Normalized revenue of a recommender system over 3 days. Bottom: Normalized trend of the word "dinner" over 6 days (data collected from `https://trends.google.com`).

Problem setup. Given samples denoted by (x, y, t), with $x \in \mathcal{X} \subset \mathbb{R}^d$ being the feature, $y \in \mathcal{Y} \subseteq \mathbb{R}$ being the label, and $t \in \mathbb{R}$ being the time at which the sample is generated, we wish to maintain a model f that can consistently and accurately predict y with x at any given t. In this paper, we focus on the scenario where the model is updated under an online learning (Hazan, 2019) or continual learning (Lopez-Paz & Ranzato, 2017) framework, while the data arrive in a streaming and cyclical fashion. More specifically, between two consecutive updates of the model at t and $t + \delta$, only samples arrived within the interval $[t, t + \delta)$ is available for training. In addition, if (x, y) is generated from a time-dependent distribution $\mathcal{D}_t$, then there exists a T such that $\mathcal{D}_t = \mathcal{D}_{t-T}$ for all t. Assuming that T is given, and that, for

[*]Equal contribution [1]ByteDance Inc [2]Paul G. Allen School of Computer Science & Engineering, University of Washington, WA. Correspondence to: Yingxiang Yang, Chong Wang <{yingxiang.yang, chong.wang}@bytedance.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

any (x, y, t), the triplet $(x, y, \text{mod}(t, T))$ is sampled from a joint distribution $p(\text{mod}(t, T))\mathcal{D}_{\text{mod}(t,T)}(x, y)$,[1] our goal is to solve the following set of optimization problems:

$$f_t^*(x) \in \underset{f \in \mathcal{L}_2(\mathcal{X})}{\text{argmin}}\, \mathbb{E}_{x,y\sim\mathcal{D}_t}[\ell(f(x), y)] \quad \forall t \in \mathbb{R}. \tag{1}$$

Here, we assume that $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact sets, and ℓ is strongly convex with respect to f for all $y \in \mathcal{Y}$. The optimization is conducted within $\mathcal{L}_2(\mathcal{X}) = \{f : \mathcal{X} \to \mathbb{R} | \int_{\mathcal{X}} f^2(x)dx < \infty\}$, a function space that contains all finite-energy functions defined over $\mathcal{X}$.

Motivation. The periodicity of the data distribution plays an important role in (1). Since $\mathcal{D}_t = \mathcal{D}_{t+nT}$ for all $n \in \mathbb{N}$, a solution $f_t^*(x)$ at t is also guaranteed to be a solution at $t + nT$. This immediately implies that the model learned at time t may offer useful information to improve the prediction accuracy at $t + nT$. This intuition motivates the design of a learning algorithm that can effectively exploit this information offered by the cyclical nature of the data.

Surprisingly, existing literature in optimization and machine learning offers little insight on how to exploit periodicity within the data distribution to solve (1) efficiently. This is particularly pronounced under a big-data setting, where industrial practices implement algorithms that simply underrate the periodicity within the training data. Below, we summarize existing approaches to the best of our knowledge from both academia and industry.

1.1. Related Works

Learning with a time feature. A straight-forward design to encode periodicity into the model structure is to include t as a model input and learn a model $f(x, t)$. Unfortunately, this approach is not guaranteed to learn a periodic function out-of-the-box, especially when $f(x, t)$ is approximated by a neural network (Ziyin et al., 2020). When $f(x, t)$ is assumed to belong to a non-parametric family, such as a reproducing kernel Hilbert space (RKHS) with a periodic kernel (Fukumizu et al., 2008; Wahba, 1990), the periodicity is usually encoded across all input dimensions, whereas in (1), $f(x, t)$ may be aperiodic in x.

An enhanced version of the above approach is to focus on a single period of $f(x, t)$ and learn a model $f(x, \text{mod}(t, T))$ instead. Although the pre-processing of t into $\text{mod}(t, T)$ guarantees periodicity during the inference stage, it still often requires laborious feature engineering, especially when x is high-dimensional and $f(x, t)$ has a complicated design, e.g., (Cheng et al., 2016).

The pluralistic approach. An alternative approach to (1) is to partition the time axis, and learn a model for every time interval (Eichner et al., 2019). When $\mathcal{D}_t$ is piecewise constant in time, e.g., $\mathcal{D}_t = \mathcal{D}_0\mathbb{1}[0 \le \text{mod}(t, T) < T/2] + \mathcal{D}_1\mathbb{1}[T/2 \le \text{mod}(t, T) < T]$, this approach allows each separate model to converge to its optimal as $t/T \to \infty$. On the other hand, however, the pluralistic approach suffers from an approximation error when the partition is crude (Eichner et al., 2019). More importantly, it is hard to scale as it requires storing multiple models. Although computationally efficient methods exist, e.g., partially sharing the network structure between the models, they typically compromise the theoretical guarantees as a trade-off.

Online learning. A standard industrial practice for training large-scale systems using sequential data is to follow the online learning protocol (Hazan, 2019). The performance of the learning algorithm is typically evaluated using the concept of dynamic regret (Mokhtari et al., 2016), which measures the model's capability to consistently and accurately predict the labels of the *latest* batch of arriving data. Crudely speaking, when t takes a set of discretized values, the dynamic regret measures $\sum_t \mathbb{E}_{\mathcal{D}_t}[\ell(f_t(x), y) - \ell(f_t^*(x), y)]$, the cumulative sum of the differences between the loss under the learned model and the optimal loss under $f_t^*(x)$ defined in (1). Although a plethora of optimization algorithms have been proposed to improve the dynamic regret analysis (Zhang et al., 2018b; 2016), none of them shed light on how to exploit periodicity within the training data. What is more, when $\mathcal{D}_t$ does not converge to a fixed distribution as t diverges, the dynamic regret scales *linearly* in t, implying that the gap between the learned model and the desired optimal does not vanish even when we know that the data is cyclical.

To summarize, learning with cyclical data remains largely an open problem for large-scale machine learning models that are trained with an online learning setup.

1.2. Our Contributions

In this paper, we address the challenge of learning with cyclical data by proposing a novel learning framework called "Fourier learning". Simply put, Fourier learning reduces (1) into a single optimization problem in a function space that naturally contains time-periodic functions, and learns the coefficients of a partial Fourier expansion for these functions using streaming-SGD. Theoretically, we support the Fourier learning framework from two different aspects: (i) from a modeling perspective, we derive Fourier learning naturally from a functional optimization problem that is equivalent to problem (1) under a strongly convex and a realizable setting; (ii) in terms of optimization, we show that the coefficient functions updated with streaming-SGD provably converge in the frequency domain. Practically, for large-scale learning systems with a neural network architecture, we introduce F-MLP, a deep learning component that can be incrementally

[1] For convenience, we denote $p(\text{mod}(t, T))$ and $\mathcal{D}_{\text{mod}(t,T)}$ as $p(t)$ and $\mathcal{D}_t$ interchangeably in the rest of the paper.

added to a wide range of existing model designs to increase capacity. Last but not least, we provide extensive numerical evidence on synthetic, public, and real-time production data of a recommender system to demonstrate the advantage of Fourier learning over the prior state-of-the-arts.

2. Learning Cyclical Data in Hilbert Spaces

A Hilbert space formulation. In this section, we lay the theoretical foundations for Fourier learning by deriving it as a natural solution to a function optimization problem. We start by reformulating the set of learning problems in (1) as one single learning problem in a Hilbert space. This allows us to learn a unified model that takes both x and t as inputs. Our learning objective takes the form of (2) below, where the expectation can be replaced by the empirical mean over datasets in practice:

$$\min_{f\in\mathcal{H}} L(f) := \mathbb{E}_{x,y,t\sim\mathcal{D}_t(x,y)p(t)}\Big[\ell(f(x,t),y)\Big]. \quad (2)$$

The unified objective (2) is related to (1) via the following lemma, the proof of which is given in Appendix A.

Lemma 1. *For $f_t^*(x)$ in (1), let $f_0(x,t) = f_t^*(x),\ \forall t \in \mathbb{R}$. If $f_0(x,t) \in \mathcal{H}$, then $f_0(x,t)$ minimizes $L(f)$ in $\mathcal{H}$.*

Lemma 1 implies that, if (2) has a unique minimizer, and if $f_t^*(x)$ belongs to $\mathcal{H}$ when treated as a function of both x and t, then the minimizer of (2) leads to the solution of (1). Hence, under a realizable setting, (2) serves as a proxy to solving (1). In the rest of the paper, we adopt this realizable learning setup and focus on solving (1) via (2).

Another critical element in (2) is the design of $\mathcal{H}$. Here, we focus particularly on functions that are continuous, periodic in time, and have finite energy in a single period. In addition, we require that the functions in $\mathcal{H}$ degenerate to $\mathcal{L}_2(\mathcal{X})$ as specified in (1) for every fixed t. We now introduce two important elements required for designing such an $\mathcal{H}$.

Functions on circles. Defining functions on circles is a common way to characterize periodic functions. These functions take the angular information of a point on a circle as input, and naturally have a period that is proportional to the circle's circumference. In our problem, we further define a Hilbert space structure over these functions by viewing a circle as a line segment with its end-points glued together: [2]

$$\mathcal{L}_2(S^T) = \Big\{ f : \mathbb{R}\to\mathbb{R} \;\Big|\; \int_0^T |f(t)|^2 \mathrm{d}t < \infty \quad \text{and} \quad f(t+T) = f(t)\ \forall t \in \mathbb{R} \Big\}. \quad (3)$$

[2] More strictly speaking, $S^T = \mathbb{R}/T\mathbb{Z}$ is a quotient space, and therefore could be treated more rigorously with group theory (Rudin, 2017).

As it turns out, if we define $\langle f, g\rangle \overset{\text{def}}{=} \int_0^T f(t)g(t)\mathrm{d}t$, then $(\mathcal{L}_2(S^T), \langle\cdot,\cdot\rangle)$ forms a Hilbert space.[3] This Hilbert space meets our needs in the special case when there is no input feature to the model, i.e., when $f(x,t)$ depends on t only.

Tensor product between Hilbert spaces. To further augment $\mathcal{L}_2(S^T)$ into a Hilbert space that contains functions dependent on both x and t, we turn to the concept of tensor product between Hilbert spaces, which is a direct metric space extension to the concept of Kronecker product between vectors in Euclidean spaces. Specifically, given two Hilbert spaces denoted by $(\mathcal{H}_1, \langle\cdot,\cdot\rangle_1)$ and $(\mathcal{H}_2, \langle\cdot,\cdot\rangle_2)$, respectively, the tensor product of $\mathcal{H}_1$ and $\mathcal{H}_2$ is a Hilbert space $(\mathcal{H} \triangleq \mathcal{H}_1 \otimes \mathcal{H}_2, \langle\cdot,\cdot\rangle)$ coupled by a bi-linear mapping $\phi : \mathcal{H}_1 \times \mathcal{H}_2 \to \mathcal{H}$. Together, $\mathcal{H}$ and ϕ must satisfy the following properties. (i) The set of vectors $\phi(u_1,u_2)$ with $u_1 \in \mathcal{H}_1$ and $u_2 \in \mathcal{H}_2$ must form a total subset of $\mathcal{H}$. That is, $\mathcal{H} = \text{Span}\{\phi(u_1,u_2) | u_1 \in \mathcal{H}_1, u_2 \in \mathcal{H}_2\}$. (ii) The inner product of $\mathcal{H}$, $\langle\cdot,\cdot\rangle$, must satisfy $\langle\phi(u_1,u_2), \phi(v_1,v_2)\rangle = \langle u_1, v_1\rangle_1 \langle u_2, v_2\rangle_2$ for any $u_1, v_1 \in \mathcal{H}_1$ and $u_2, v_2 \in \mathcal{H}_2$. If we adopt two orthonormal sets of basis functions, $\{e_{1i}\}_{i=1}^{\dim(\mathcal{H}_1)}$ and $\{e_{2j}\}_{j=1}^{\dim(\mathcal{H}_2)}$, for $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, these aforementioned properties would allow us to expand any element $\phi(u_1,u_2) \in \mathcal{H}$ into $\phi(u_1,u_2) = \sum_{i=1}^{\dim(\mathcal{H}_1)} \sum_{j=1}^{\dim(\mathcal{H}_2)} u_{1i}u_{2j}\phi(e_{1i},e_{2j})$, where $u_{1i} = \langle u_1, e_{1i}\rangle_1$ and $u_{2j} = \langle u_2, e_{2j}\rangle_2$, respectively. Furthermore, when $\mathcal{H}_1 = \mathcal{L}_2(\mathcal{X})$ and $\mathcal{H}_2(\mathcal{Y}) = \mathcal{L}_2(\mathcal{Y})$, as is the case for our problem, an isomorphism exists such that $\phi(e_{1i}, e_{2j}) \cong e_{1i}e_{2j}$. This implies that we can consider $\mathcal{H}$ as an isomorphism of $\mathcal{H}_1 \otimes \mathcal{H}_2$ containing functions that are linear combinations of $\{e_{1i}e_{2j}\}_{i=1,j=1}^{\infty}$, i.e, $\mathcal{H} = \mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(\mathcal{Y}) \cong \mathcal{L}_2(\mathcal{X}\times\mathcal{Y})$. We refer interested readers to (Reed, 2012) for more details on this topic.

2.1. A Tensor-Product-Based Design of $\mathcal{H}$

Augmenting $\mathcal{L}_2(S^T)$ by its tensor product with $\mathcal{L}_2(\mathcal{X})$, a natural choice of $\mathcal{H}$ is to set $\mathcal{H} \triangleq \mathcal{L}_2(\mathcal{X}\times S^T)$, where

$$\mathcal{L}_2(\mathcal{X}\times S^T) = \Big\{ f : \mathcal{X}\times\mathbb{R}\to\mathbb{R} \;\Big|\; \|f\|_{\mathcal{L}_2(\mathcal{X}\times[0,T])} < \infty \quad \text{and} \quad f(x,t) = f(x,t-T)\ \forall t \Big\}. \quad (4)$$

This $\mathcal{L}_2(\mathcal{X}\times S^T)$ expands $\mathcal{L}_2(\mathcal{X})$ with an additional dimension in t defined on a circle with a circumference T, which naturally restricts $f(x,t) \in \mathcal{H}$ to be a periodic function over t for any fixed $x \in \mathcal{X}$. The following lemma certifies that $\mathcal{H}$ is a Hilbert space and characterizes its basis functions using the isomorphism between $\mathcal{H}$ and $\mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(S^T)$.

Lemma 2. *Let $\mathcal{H}$ be defined in (4). For $f, g \in \mathcal{H}$, let $\langle f, g\rangle \overset{\text{def}}{=} \int_{\mathcal{X}} \int_0^T f(x,t)g(x,t)\mathrm{d}x\mathrm{d}t$, then $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ is a*

[3] For the simplicity of notations, we denote the inner product as $\langle\cdot,\cdot\rangle$ when its associated Hilbert space is clear by context.

Hilbert space. Furthermore, there exists an isomorphism between $\mathcal{H}$ and $\mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(S^T)$, i.e., if $\{\phi_i\}_{i=1}^{\infty}$ and $\{\psi_j\}_{j=1}^{\infty}$ are two orthonormal sets of bases for $\mathcal{L}_2(\mathcal{X})$ and $\mathcal{L}_2(S^T)$, respectively, then $\{\varphi_{ij}\}_{i,j=1}^{\infty}$, where $\varphi_{ij}(x,t) \stackrel{\text{def}}{=} \phi_i(x)\psi_j(t)$, is an orthonormal set of basis functions for $\mathcal{H}$.

The proof of Lemma 2 is given in Appendix B. This lemma paves the way to a framework that learns the model f via its basis expansion in $\mathcal{H}$, which we introduce in the following section. Meanwhile, we argue that $\mathcal{H}$ is general enough for our learning purpose in the sense that the function $f_0(x,t)$ defined pointwise by solutions in (1) belongs to $\mathcal{H}$ under mild assumptions. We start by introducing necessary definitions and assumptions as below.

Definition 3 (Continuity under total variation)**.** *Let $\mathcal{D}_t(x)$ be the conditional distribution of y given x under $\mathcal{D}_t$. We say $\mathcal{D}_t(x)$ is continuous in t under total variation distance if, for any fixed t and any $\epsilon > 0$, there exists $\delta > 0$ such that*

$$\|\mathcal{D}_{t'}(x) - \mathcal{D}_t(x)\|_{\text{TV}} \leq \epsilon \quad \textit{whenever} \quad |t' - t| \leq \delta.$$

Assumption 4. *Suppose: (i) $\mathcal{X}$ and $\mathcal{Y}$ are compact and convex sets; (ii) $\mathcal{D}_t(x)$ in Definition 3 is continuous under total variation for all $x \in \mathcal{X}$; (iii) the loss function $\ell(f(x), y)$ is σ-strongly-convex in its first argument for all $y \in \mathcal{Y}$; (iv) $f(x)$ in (1) is bounded and $\max_{x\in\mathcal{X}, y\in\mathcal{Y}} \ell(f(x), y) \leq K$ for some constant K.*

In practice, Assumption 4 can be easily satisfied by a wide range of machine learning systems. For instance, deep neural networks (DNNs) typically have bounded outputs when a clipping on the final output is enforced. The uniform strong convexity of the loss function also holds for a wide range of ℓ such as the the mean squared loss. With the above definition and assumption, we now state the following lemma.

Lemma 5. *Under Assumption 4, $f_t^*(x)$ is continuous in t for any given $x \in \mathcal{X}$. In addition, $f_0(x,t) \stackrel{\text{def}}{=} f_t^*(x) \in \mathcal{H}$.*

The proof of Lemma 5 is given in Appendix C. This lemma implies that, under Assumpiton 4, the optimal solution of (2), $f_0(x,t)$, belongs to $\mathcal{H}$. Combining Lemmas 1 and 5, we immediately see that the satisfaction of Assumption 4 allows us to acquire a set of desired solution of (1) by solving (2).

2.2. Fourier Learning with Cyclical Data

We now introduce Fourier learning, a learning framework that hard-wires the periodicity into the model's structure via a partial Fourier expansion and learns the model by learning its Fourier coefficient functions. From a modeling aspect, we invoke Lemma 2, and represent $f(x,t) \in \mathcal{H}$ by

$$f(x,t) = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} c_{i,j}\phi_i(x)\psi_j(t) = \sum_{j=1}^{\infty} c_j(x)\psi_j(t), \quad (5)$$

where $c_j(x) \in \mathcal{L}_2(\mathcal{X})$ is the sum of $c_{i,j}\phi_i(x)$ over i. Noticing that a set of bases for $\mathcal{L}_2(S^T)$ are the trigonometric functions with a base frequency $1/T$, we immediately have the following Theorem (the proof is given in Appendix D).

Theorem 6. *Any $f(x,t) \in \mathcal{H}$ can be represented by*

$$f(x,t) = \sum_{n=0}^{\infty}\left[a_n(x)\sin\left[\frac{2\pi nt}{T}\right] + b_n(x)\cos\left[\frac{2\pi nt}{T}\right]\right], \quad (6)$$

where $a_n(x), b_n(x) \in \mathcal{L}_2(\mathcal{X})$, and $a_0(x) \equiv 0$.

Theorem 6 provides an explicit way of designing periodic models and specifies how the time-feature could be exploited. Note that, it is entirely possible to construct $\mathcal{H}$ with a weighted $\mathcal{L}_2$ space defined on circles to guarantee periodicity, e.g., an RKHS with a periodic kernel. This allows us to deviate from the trigonometric functions and use potentially other periodic functions to encode periodicity, e.g., the periodic Gaussian kernel (MacKay et al., 1998).

Our goal now shifts towards learning the coefficient functions in the frequency domain. For tractable learning, we introduce a cutoff frequency N/T, and learn a truncated expansion of $f(x,t)$ instead:

$$f_N(x,t) = \sum_{n=0}^{N}\left[a_n(x)\sin\left[\frac{2\pi nt}{T}\right] + b_n(x)\cos\left[\frac{2\pi nt}{T}\right]\right]. \quad (7)$$

By (Carleson, 1966), the approximation error for all $f \in \mathcal{H}$,

$$E_N(f) = \left\|f(x,t) - f_N(x,t)\right\|_{\mathcal{H}}^2, \quad (8)$$

satisfies $\lim_{N\to\infty} E_N(f) = 0$. Hence, with a properly selected N, we can control the approximation error while limiting the amount of model parameters at the same time.

The coefficient functions, $a_n(x)$ and $b_n(x)$, can be learned under a variety of regimes. For example, they can be learned non-parametrically using function optimization algorithms (Yang et al., 2019) (see Section 4 for more details). In the remainder of the paper, we focus primarily on parameterizing $a_n(x)$ and $b_n(x)$ with neural networks, so as to apply Fourier learning to large-scale machine learning scenarios.

2.3. Discussions

Fourier analysis in machine learning. Fourier analysis, as a prominent branch of study in signal processing and digital communications (Oppenheim et al., 1983), has been widely adopted in machine learning and the designs of neural networks (Gallant & White, 1988; Uteuliyeva et al., 2020). In many applications, such as computer vision and time series analysis (Tancik et al., 2020; Zhang et al., 2018a), Fourier analysis has attracted substantial research interests as it yields more expressive features (Rahimi et al., 2007),

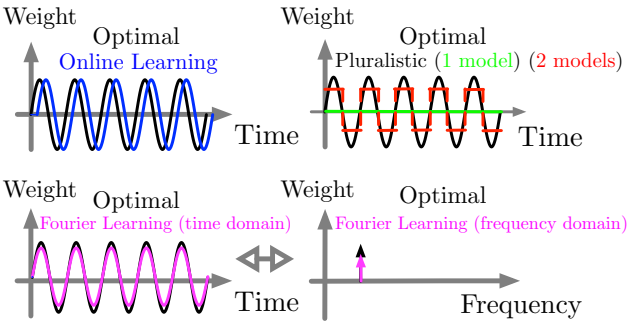

Figure 2. Illustration on how online learning, the pluralistic approach, and Fourier learning learn a sine function.

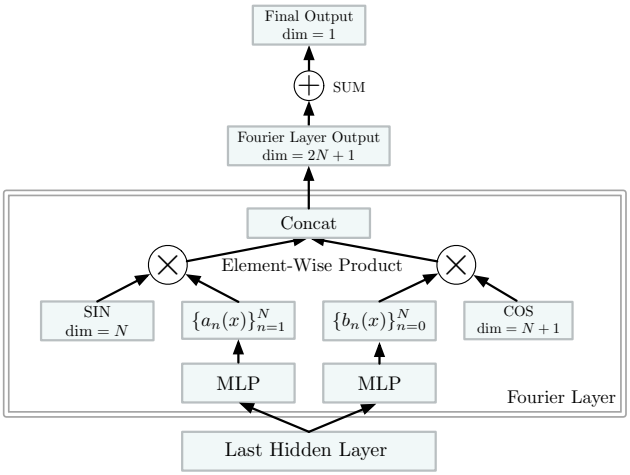

Figure 3. The design of a Fourier-MLP with a scalar output.

and improves the predictive power of the neural networks (Tancik et al., 2020; Sitzmann et al., 2020).

Unlike existing works, Fourier learning focuses on an online learning setup, and improves the model performance by exploiting periodicity within the data distribution. To further illustrate this difference, we compare Fourier learning, online learning, and the pluralistic approach in Example 1.

Example 1. *Consider learning* $f_t^*(x) = \sin(2\pi t/T)$ *given a stream of samples* $\{t_i, \sin(2\pi t_i/T)\}_{i=1}^{100}$, *with* $T = 10$ *and* $t_i = i$. *Under a mean squared error (MSE) loss, we sketched the output of online learning, the pluralistic approach, and Fourier learning in Figure 2. As illustrated, online learning learns a periodic function with a delay. This is because at time t, online learning can only learn with the samples collected prior to t. The pluralistic approach learns a fixed model for each interval. It suffers from an approximation error when the partition is crude. By comparison, Fourier learning learns the frequency representation of the model* $a_1(x)$, *and yields a near-optimal solution in time domain. By exploiting periodicity within the data distribution, Fourier learning mitigates the delay in online learning, and the approximation error in the pluralistic approach.*

In Example 1, Fourier learning is only tasked to learn a frequency component that is independent of x. In practice, $f_t(x)$ often highly depends on x, and is typically modeled by neural networks. This motivates us to design a deep learning solution to incorporate periodicity into the existing deep learning models, which we focus on in the next section.

3. Fourier Multi-Layer Perceptron (F-MLP)

To integrate Fourier learning in large-scale machine learning systems, we introduce F-MLP, a neural network structure that can be incrementally added to existing neural network designs to better learn a periodic model.

3.1. Architecture of F-MLP

F-MLP is designed by the following intuition: if we view x as the output of a neural network's last hidden layer, then (7) can be viewed as the network's output layer with an architecture shown in Figure 3. Specifically, F-MLP first transforms x into $a_n(x)$ and $b_n(x)$, and then element-wise multiplies them with basis vectors SIN and COS, yielding a $(2N+1)$-dimension layer. This layer is then added up, yielding a scalar output.[4] Notably, when $a_n(x) = b_n(x) = 0$ for all $n \geq 1$, the final output equals $b_0(x)$, which, by itself, can be interpreted as the original model's output. This implies that replacing the original model's output layer with an F-MLP increases its capacity, avoiding the need for laborious feature engineering.

3.2. Training F-MLP with Streaming-SGD

The training of F-MLP is performed jointly with the original model, following the procedure of streaming-SGD. This procedure is different from the standard SGD, which in practice would need sample data $(x, y, t) \sim \mathcal{D}_t(x, y)p(t)$. However, sampling from $p(t)$ is difficult for many online applications due to the real-time update requirement, where data arrive sequentially. Here we show that with streaming-SGD we can avoid this issue while still having good practical performances and convergence guarantees.

The training procedure is as follows. We parameterize $a_n(x)$ and $b_n(x)$ by $a_n(x;\theta_n)$ and $b_n(x;\rho_n)$, respectively, with θ_n and ρ_n being the neural network parameters. For cyclical data, we collect the τ-th mini-batch of data in the k-th cycle, and update the model with the following update rule:

$$\begin{aligned} \theta_n^{(k,\tau+1)} &= \theta_n^{(k,\tau)} - \eta_{k,\tau} g_{\theta_n}^{(k,\tau)}, \\ \rho_n^{(k,\tau+1)} &= \rho_n^{(k,\tau)} - \eta_{k,\tau} g_{\rho_n}^{(k,\tau)}. \end{aligned} \tag{9}$$

Here, $g_{\theta_n}^{(k,\tau)}$ and $g_{\rho_n}^{(k,\tau)}$ are gradients calculated using the

[4] We refer interested readers to Appendix H for the design of F-MLP with a vector output.

collected mini-batch of data:

$$g_{\theta_n}^{(k,\tau)} = \nabla_{\theta_n} \widehat{L}_{k,\tau}(f_N^{(k,\tau)}),$$
$$g_{\rho_n}^{(k,\tau)} = \nabla_{\rho_n} \widehat{L}_{k,\tau}(f_N^{(k,\tau)}), \tag{10}$$

where $f_N^{(k,\tau)}$ is computed with (7), while $\widehat{L}_{k,\tau}$ is the empirical version of the loss in (2) over this mini-batch of data. The overall training procedure is summarized in Algorithm 1, and we present the convergence analysis of it below.

3.3. Convergence Analysis

In this section, we discuss the convergence properties when training Fourier learning models with streaming-SGD. Recall that, using a truncated $f_N(x,t)$, problem (2) reduces into finding the optimal of

$$\min_{\{a_n\}_{n=1}^N, \{b_n\}_{n=0}^N} \mathbb{E}_{x,y,t\sim\mathcal{D}_t(x,y)p(t)}\Big[\ell(f_N(x,t),y)\Big] \tag{11}$$

in the frequency domain. We denote the optimal set of coefficient functions of (11) as $a_n^*(x)$ and $b_n^*(x)$, for which the corresponding model $f_N^*(x,t)$ can be expressed as

$$f_N^*(x,t) = \sum_{n=0}^{N} \left[a_n^*(x)\sin\left[\frac{2\pi nt}{T}\right] + b_n^*(x)\cos\left[\frac{2\pi nt}{T}\right]\right]. \tag{12}$$

In the following, we first show a gradient norm convergence result for streaming-SGD under a general non-convex setting, and then introduce a global convergence result under the assumption of strong convexity. Prior to that, we introduce some additional assumptions.

Assumption 7. *Suppose: (i) for all n,k,τ, the second moment of the update directions are bounded: $\max\{\mathbb{E}[\|g_{\rho_n}^{(k,\tau)}\|_2^2|\mathcal{F}^{(k)}], \mathbb{E}[\|g_{\theta_n}^{(k,\tau)}\|_2^2|\mathcal{F}^{(k)}]\} \leq G^2$ for some $G\in\mathbb{R}$, where $\mathcal{F}^{(k)}$ is the minimum σ-algebra generated by $a_n^{(\kappa,\tau)}$ and $b_n^{(\kappa,\tau)}$ for all n, $\kappa<k$ and $1\leq\tau\leq\Gamma$. In addition, we assume (ii) there exists $\Lambda>0$ such that $\|\nabla L(f_1)-\nabla L(f_2)\|\leq\Lambda\|f_1-f_2\|$ for all $f_1,f_2\in\mathcal{H}$.*

This assumption assumes bounded second moment of the update directions, and the Lipschitzness of the gradient, which are usually required in the convergence analysis of SGD-type algorithms. The following result shows that streaming-SGD with a proper learning rate achieves convergence under both non-convex and strongly convex settings.

Theorem 8 (Convergence of Streaming-SGD)**.** *Let (i), (ii) of Assumption 4 and Assumption 7 hold, and define $\|\nabla L(f_N^{(k,1)})\|$ as the gradient with respect to a joint parameter vector combining all $\theta_n^{(k,1)}$ and $\rho_n^{(k,1)}$.*

- *Let $\eta_{k,\tau}\triangleq\eta_k=\Theta(1/\sqrt{T_{\max}+1})$, then*

$$\min_{0\leq k\leq T_{\max}} \|\nabla L(f_N^{(k,1)})\|^2 = \mathcal{O}(1/\sqrt{T_{\max}+1}).$$

Algorithm 1 Streaming-SGD for F-MLP

Input: Macro-iteration number $T_{\max}$; loss L; step sizes $\{\eta_{k,\tau}\}_{k=0,\tau=1}^{T_{\max},\Gamma}$; period T; cutoff frequency N/T.
Initialize $\{a_n^{(0,1)}\}_{n=1}^N$ and $\{b_n^{(0,1)}\}_{n=0}^N$.
for $(k,\tau)=(0,1)$ to $(T_{\max},1)$ **do**
  Compute $f_N^{(k,\tau)}$ with $a_n^{(k,\tau)}$ and $b_n^{(k,\tau)}$ with (7).
  for $n\in\{0,\dots,N\}$ **do**
    Compute $g_{\theta_n}^{(k,\tau)}$ and $g_{\rho_n}^{(k,\tau)}$.
    Update $\theta_n^{(k,\tau)}$ and $\rho_n^{(k,\tau)}$.
  end for
end for
Output: $f_N^{(T_{\max},1)}$, $a_n^{(T_{\max},1)}$ and $b_n^{(T_{\max},1)}$.

- *Let $\eta_{k,\tau}\triangleq\eta_k=\Theta(1/\sqrt{k})$, then*

$$\min_{0\leq k\leq T_{\max}} \|\nabla L(f_N^{(k,1)})\|^2 = \mathcal{O}\left(\frac{\log(T_{\max}+1)}{\sqrt{T_{\max}+1}}\right).$$

Moreover, if L is σ-strongly convex with respect to θ_n and ρ_n, we can take $\eta_{k,\tau}=\psi/k$ with $\psi<(2\sigma^2)^{-1}$ and obtain

$$\mathbb{E}\|\theta_n^{(T_{\max},1)}-\theta_n^*\|_2^2 = \mathcal{O}(T_{\max}^{-1}),$$
$$\mathbb{E}\|\rho_n^{(T_{\max},1)}-\rho_n^*\|_2^2 = \mathcal{O}(T_{\max}^{-1}),$$

where we assume $a_n(x,\theta_n^*)=a_n^*(x)$, $b_n(x,\rho_n^*)=b_n^*(x)$.

The proof of Theorem 8 is given in Appendix F. Simply put, our learning framework offers a convergence rate of $\mathcal{O}(1/\sqrt{T_{\max}})$ under a general non-convex setting and $\mathcal{O}(1/T_{\max})$ under a strongly convex setting. If we further drive $N\to\infty$, the overall learning error of $f_0(x,t)$ can be driven to arbitrarily small. Compared to the online learning benchmark whose dynamic regret is affected by both the changing speed of the data-generating distribution and the variance of the stochastic gradients, Fourier learning yields a much smaller learning error and hence offers a potentially much better performance in many practical scenarios.

4. Extension to the Non-Parametric Regime

Apart from the parametric framework introduced in the prior sections, Fourier learning also fits into the non-parametric regime, where a_n and b_n are updated directly:

$$a_n^{(k,\tau+1)}(x) = a_n^{(k,\tau)}(x) - \eta_{k,\tau}\cdot g_{a_n}^{(k,\tau)}(x),$$
$$b_n^{(k,\tau+1)}(x) = b_n^{(k,\tau)}(x) - \eta_{k,\tau}\cdot g_{b_n}^{(k,\tau)}(x). \tag{13}$$

As the functional gradients in $\mathcal{L}_2$ often contain Dirac's δ-functions, causing discontinuous updates, we substitute the functional gradients with their kernel embeddings instead. Specifically, with $K(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ being a positive-definite kernel whose minimum eigen-value is bounded

away from 0, we let

$$g_{a_n}^{(k,\tau)}(x) = \langle \nabla_{a_n} \widehat{L}_{k,\tau}(f_N^{(k,\tau)})](\cdot), K(x,\cdot)\rangle,$$
$$g_{b_n}^{(k,\tau)}(x) = \langle [\nabla_{b_n} \widehat{L}_{k,\tau}(f_N^{(k,\tau)})](\cdot), K(x,\cdot)\rangle. \quad (14)$$

It is easy to verify that these kernel embeddings yield continuous updates of $a_n(x)$ and $b_n(x)$ at each iteration. At the same time, $g_{a_n}^{(k,\tau)}$ and $g_{b_n}^{(k,\tau)}$ are "close enough" to the exact gradients and retain the convergence guarantees (Yang et al., 2019). If we initialize $a_n^{(0,\tau)}$ and $b_n^{(0,\tau)}$ to be zeros, then $a_n^{(k,\tau)}$ and $b_n^{(k,\tau)}$ can be written as a linear combination of a finite set of kernels. An example of calculating the kernel embeddings of functional gradients is given in Appendix E. The convergence result for the non-parametric case is given in Theorem 9 below. The proof is given in Appendix G.

Theorem 9. *Let Assumption 4 and Assumption 7 hold. Let $g_{a_n}^{(k,\tau)}$ and $g_{b_n}^{(k,\tau)}$ be the kernel embeddings of functional gradient with $K(\cdot,\cdot)$ at iteration (k,τ), as defined in (14). Let $\eta_{k,\tau} = \sigma(k+0.5)\lambda^{-1}(k+1)^{-2}$. Then, $\mathbb{E}\|a_n^{(T_{\max},1)} - a_n^*\|_2^2 = \mathcal{O}(T_{\max}^{-1})$ and $\mathbb{E}\|b_n^{(T_{\max},1)} - b_n^*\|_2^2 = \mathcal{O}(T_{\max}^{-1})$, with a_n^* and b_n^* defined in (12).*

5. Numerical Simulations

In this section, we numerically demonstrate the superiority of Fourier learning over the prior state-of-the-arts on synthetic and public datasets. Our major benchmarks include: **(i)** Time-feature: a benchmark that encodes periodicity with $f(x, \text{mod}(t,T))$ using neural networks; **(ii)** Pluralistic (Eichner et al., 2019); and **(iii)**: Online learning (Hazan, 2019), as adopted by common industrial systems.

For each simulated method, we record the instantaneous loss at each iteration prior to updating the model. This allows us to evaluate the model's ability on tracking a constantly changing set of optimal model parameters and on maintaining a good prediction accuracy consistently. The source code and logs used to generate the reported experiments can be found at `https://github.com/Yangyx891121/Fourier-Learning`.

5.1. Synthetic Dataset: Linear Model

Experiment settings. We considered a toy experiment with $T = 1$ and $p(t) = 1$ for $t \in [0,T]$ in (1). For $t \in [0,T]$, let $x \sim \mathcal{N}(\sin(2\pi t), 0.1)$, and $y \sim \mathcal{N}(\sum_{n=1}^{6} \sin(2\pi n t)x, 0.01)$. We generated samples over $t \in [0, 50T]$, and fitted a linear time-varying system $y = \alpha(t)x$ using the mean squared error (MSE) loss. The optimal $\alpha(t)$ under this setting is $\alpha^*(t) = \sum_{n=1}^{6} \sin(2\pi n t)$, allowing us to evaluate the performance of each algorithm using the MSE between its estimated $\alpha(t)$ and $\alpha^*(t)$.

We implemented the aforementioned benchmarks under the following settings. For (i), we simulated two separate designs, one using $f(x,t) = \alpha(\text{mod}(t,T))x$ where $\alpha(\text{mod}(t,T))$ is a two-layer neural network, while the other models $f(x,t)$ with a two-layer neural network directly. For (ii), we used a vector $\boldsymbol{\alpha}$ to store m separate α's. The i-th value of $\boldsymbol{\alpha}$ is responsible for learning samples with $\text{mod}(t,T) \in [(i-1)\times \text{interval}, i \times \text{interval})$ where $\text{interval} = T/m$. The number of models, m, is set as a tuning parameter. For (iii), we directly optimized over α using online gradient descent. We grid-searched the learning rate, hidden layer sizes, and the number of Fourier bases.

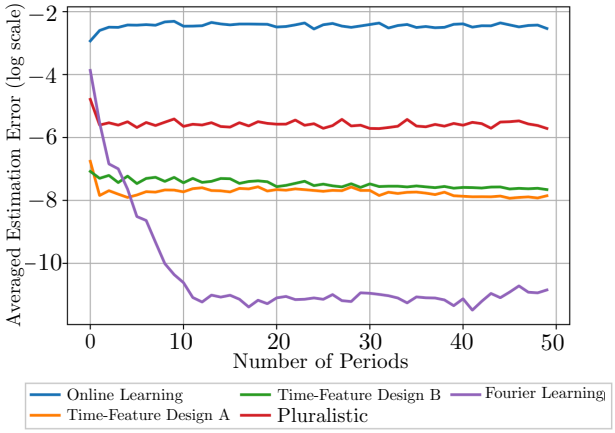

Figure 4. Aggregated MSE (in log scale) in each period between the estimated and ground truth $\alpha(t)$ over 50 periods.

Results. We plotted the MSE of the simulated algorithms in Figure 4, where we aggregated the MSE over each period so that the horizontal axis shows how an algorithm's performance improves as t/T increases. We see that Fourier learning has a much better performance than all the benchmarks. In addition, the online learning benchmark has almost no MSE reduction as t/T increases. This is because the online learning regime lacks exploitation of the cyclical nature of the data, which leaves behind a constant gap between the learned model and the desired optimal at each iteration.

5.2. Real Dataset: Sentiment140

Experiment settings. We classified the sentiment of tweets using a bag-of-words model over the Sentiment140 Twitter (Go et al., 2009) dataset. Following the experiment settings of Eichner et al. (2019), we manually created a cyclical data stream with $T = 1$ day $= 86400$ seconds by first dividing the samples into four blocks based on their associated timestamps, and then down-sampled the positive (or negative) samples of each block based on the sign and value of a randomly-generated ratio within a range of $[-0.7, 0.7]$.

For the bag-of-words model, we used a three-layers neural network, with a 4096-dimensional input layer, a 64-dimensional hidden layer, and a two-dimensional output

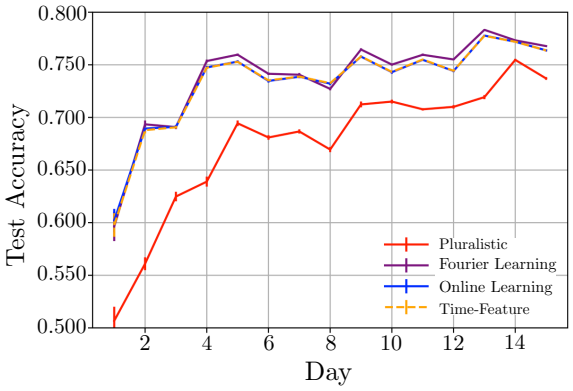

Figure 5. Testing error of Fourier learning and benchmarks, averaged over 10 trials.

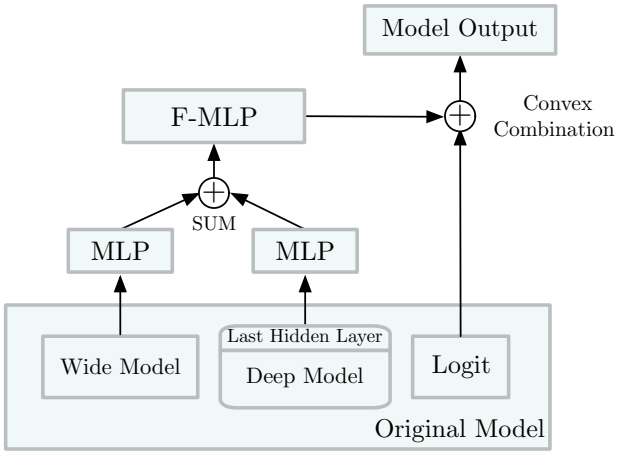

Figure 6. Deploying F-MLP to a wide and deep neural network.

layer, largely following the settings of (Eichner et al., 2019). The time feature was added as a one-dimensional input to the input layer. For the pluralistic approach, we reduced the hidden layer size to keep the number of parameters in line with the other methods. For Fourier learning, a 64-dimensional Fourier layer is added at the output end.

Results. We conducted the experiment over 15 days of data, and plotted the results for lr $= 0.1$ in Figure 5 (see Appendix I for more results). We see that Fourier learning has a better performance than all the benchmarks. We also see that adding a time feature and online learning have very similar performances. This suggests that directly adding a time-feature may not bring immediate performance improvement, which warrants the need for further feature engineering. Furthermore, the gap between Fourier learning and the aforementioned benchmarks cannot be easily closed. In fact, even upon trippling the network size, the performance of online learning is still inferior than Fourier learning (see Appendix I). Lastly, despite the competitiveness of the pluralistic approach (Eichner et al., 2019), its performance lags as the storage constraint forces a reduction to its model size.

6. Fourier Learning in Recommender Systems

In this section, we report the performance of Fourier learning implemented on a conversion rate (CVR) prediction model in an industrial recommender system. As demonstrated in Figure 1, the system's revenue displays a periodic pattern due to the periodic patterns in the lifestyles of its users. The model architecture follows the design of a wide-and-deep network (Cheng et al., 2016), with thousands of features, and several sub-networks that are either wide or deep. The model output at (k, τ)-th iteration has the form $f_{\text{oco}}(x; (k, \tau)) = \sum_{q=1}^{Q} f_q(x; (k, \tau))$. Here, $f_q(x; (k, \tau))$ is the output of the q-th (out of a total of Q) sub-network at iteration (k, τ).

Design. We added Fourier learning to this model by mixing $f_{\text{oco}}(x; (k, \tau))$ with $f_{\text{FL}}(x, (k, \tau))$, the output of Fourier learning which uses t as an input. This output of Fourier learning is obtained by first aligning the dimensions of the last hidden layers of all the sub-networks, and then passing their sum through a single Fourier layer. Exploiting the linearity of the Fourier transform, this design avoids the need of learning a Fourier layer for each sub-network. The final output has the form $f(x, (k, \tau)) = \xi f_{\text{oco}}(x; (k, \tau)) + (1 - \xi) f_{\text{FL}}(x, (k, \tau))$, where $\xi \in [0, 1]$ is a hyperparameter. We visualized this design in Figure 6.

Intuitively, the mixture of logits has two advantages. First, it circumvents drastic alterations to the original model architecture. In fact, setting $\xi = 1$ retains the original model: $f(x, (k, \tau)) = f_{\text{oco}}(x; (k, \tau))$. Secondly, it handles the practical situation that the data distribution may not strictly follow a periodic pattern. When the periodicity pattern is relatively strong, we expect Fourier learning to improve the performance of the original model; when the periodicity pattern is relatively weak, the presence of $f_{\text{oco}}(x; (k, \tau))$ limits the model mismatch introduced by Fourier learning. It turns out that this strategy is highly effective.

Learning setting. We used production-level data spanned over three months, denoted by months F, G, and H, respectively. Each day has thirty to sixty million training samples. Under the aforementioned design, we set the base frequency of the Fourier learning model to 2.4192MHz, or $(28 \text{ days})^{-1}$, and set $N = 28 \times 4$ in (7), providing a frequency band of up to $(6 \text{ hours})^{-1}$. The coefficient ξ is set to 0.5. The rest of the model parameters are the same as baseline. We trained the model on a distributed machine learning platform with 3,600 CPU cores. The training took one day. A total of five benchmarks are implemented, with the hyperparameters carefully tuned via grid search.

- **Online learning:** baseline model, the current production model that serves hundreds of millions of users every single day. It does not use time features.
- **Time-feature:** learning $f(x, \text{mod}(t, T))$ with a time of day feature $\text{mod}(t, T)$. This feature is inserted into the model the same way as all other features.
- **Positional encoding:** learning $f(x, g(\text{mod}(t, T)))$ with the time of day feature $(\text{mod}(t, T))$ embedded in a positional encoder $g(\cdot)$ (Vaswani et al., 2017). This encoded feature is inserted into the model as an input to the deep sub-network.
- **Pluralistic (Eichner et al., 2019):** a total of six models are trained but the models only differ in the last few layers. Since a single model in the production system costs a tremendous amount of CPU cores and memory to train and serve, the pluralistic approach is simply impossible to implement in its original form.
- **Online learning (large):** the online learning approach with a larger network that has the same number of parameters as Fourier learning. We use this benchmark to show that Fourier learning's performance gain is not due to a simple increment in the model size.

Results. We reported the experiment results in Figure 7, measured by the Area under the Curve (AuC) aggregated on a monthly basis. The variance of the readings due to the distributed training environment is typically $< 0.02\%$. Table 4 shows that Fourier learning has a clear and consistent advantage of $> 0.1\%$ over all benchmarks, none of which possesses a significant ($> 0.02\%$) AuC improvement over the baseline model. Furthermore, this improvement is not due to the increase in the model size, as online learning (large), which has the same number of model parameters as Fourier learning, does not show a consistent AuC gain over the baseline. This is significant for such a large-scale system, since a 0.1% improvement in AuC may bring in millions of dollars of revenue growth on a monthly basis.

Apart from AuC, we also examined the calibration of the trained models, aggregated montly. The calibration is defined as Calibration $=$ pCVR/CVR, where pCVR and CVR are the predicted and empirical conversion ratio, averaged over all samples. Note that a well calibrated Ads CVR prediction model should have a calibration reading close to one. As is shown in Table 1, the Fourier learning model's calibration is very close to one and is comparable to others.

7. Conclusion and Discussions

In this paper, we introduced a novel framework, "Fourier learning", to efficiently train large-scale machine learning models with cyclical data. Using Fourier analysis, we transformed the learning problem into the frequency domain, and proposed a theoretically guaranteed optimization algorithm to learn the coefficient functions. We introduced F-MLP in the context of deep learning which strictly increases the original model's capacity. We also demonstrated the superiority of Fourier learning over the state-of-the-arts on synthetic, public, and production-level data.

This work opens up several research directions. Among them, an important one is to measure the periodicity by the degree to which the data distribution oscillates. This is necessary because, under the current definition provided in this paper, even i.i.d. data can be deemed periodic as T can be arbitrary. However, as explained in Section 1.1 and illustrated in Example 1, Fourier learning may have an advantage over online learning only when the data distribution oscillates. In light of this, how the degree of oscillation relates to the advantage of Fourier learning needs to be answered.

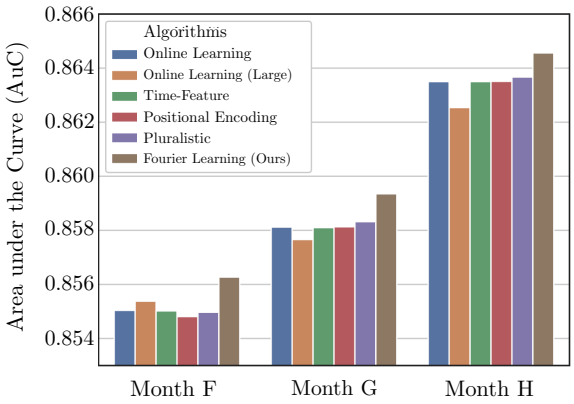

Figure 7. AuC for the implemented methods, aggregated monthly.

Algorithm	Month F	Month G	Month H
Online learning	0.9927	0.9974	0.9977
Online learning (L)	0.9918	0.9974	0.9979
Time-feature	0.9926	0.9974	0.9976
Positional encoding	0.9928	0.9975	0.9977
Pluralistic	0.9929	0.9976	0.9978
Fourier learning (O)	0.9922	0.9973	0.9975

Table 1. Calibration of the algorithms, aggregated monthly. Here, "L" and "O" are abbreviations for "Large" and "Ours", respectively.

Acknowledgements We sincerely thank Maryam Fazel, Kevin Jamieson, Junchi Yang, and Pengkun Yang for useful discussions. We also thank anonymous reviewers for their comments and suggestions during the review process, and pointers to prior literature.

References

Carleson, L. On convergence and growth of partial sums of Fourier series. *Acta Mathematica*, 116(1):135–157, 1966.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. Wide & deep learning for recommender systems. In *Workshop on Deep Learning for Recommender Systems*, pp. 7–10, 2016.

Eichner, H., Koren, T., McMahan, B., Srebro, N., and Talwar, K. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1764–1773. PMLR, 2019.

Fukumizu, K., Sriperumbudur, B. K., Gretton, A., and Schölkopf, B. Characteristic kernels on groups and semigroups. In *Advances in Neural Information Processing Systems*, pp. 473–480, 2008.

Gallant, A. R. and White, H. There exists a neural network that does not make avoidable mistakes. In *International Conference on Neural Networks*, pp. 657–664, 1988.

Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Gultekin, M. N. and Gultekin, N. B. Stock market seasonality: International evidence. *Journal of Financial Economics*, 12(4):469–481, 1983.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6467–6476, 2017.

MacKay, D. J. et al. Introduction to Gaussian processes. *NATO ASI series F computer and systems sciences*, 168: 133–166, 1998.

Mokhtari, A., Shahrampour, S., Jadbabaie, A., and Ribeiro, A. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Conference on Decision and Control*, pp. 7195–7201. IEEE, 2016.

Oppenheim, A., Willsky, A., Young, I., and Nawad, H. *Signals and Systems*. Prentice-Hall signal processing series. Prentice-Hall, 1983. ISBN 9780138097318.

Rahimi, A., Recht, B., et al. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 3, pp. 5. Citeseer, 2007.

Reed, M. *Methods of modern mathematical physics: Functional analysis*. Elsevier, 2012.

Rudin, W. *Fourier analysis on groups*. Courier Dover Publications, 2017.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

Tracà, S., Rudin, C., and Yan, W. Regulating greed over time in multi-armed bandits. *Journal of Machine Learning Research*, 22:3–1, 2021.

Uteuliyeva, M., Zhumekenov, A., Takhanov, R., Assylbekov, Z., Castro, A. J., and Kabdolov, O. Fourier neural networks: A comparative study. *Intelligent Data Analysis*, 24(5):1107–1120, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wahba, G. *Spline models for observational data*. SIAM, 1990.

Yang, Y., Wang, H., Kiyavash, N., and He, N. Learning positive functions with pseudo mirror descent. In *Advances in Neural Information Processing Systems*, volume 32, pp. 14144–14154, 2019.

Zhang, J., Lin, Y., Song, Z., and Dhillon, I. Learning long term dependencies via Fourier recurrent units. In *International Conference on Machine Learning*, pp. 5815–5823. PMLR, 2018a.

Zhang, L., Yang, T., Yi, J., Jin, R., and Zhou, Z.-H. Improved dynamic regret for non-degenerate functions. *arXiv preprint arXiv:1608.03933*, 2016.

Zhang, L., Yang, T., Zhou, Z.-H., et al. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pp. 5882–5891. PMLR, 2018b.

Ziyin, L., Hartwig, T., and Ueda, M. Neural networks fail to learn periodic functions and how to fix it. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Appendix

A. Proof of Lemma 1

Starting from (2), we have, for any $f(x,t) \in \mathcal{H}$,

$$\begin{aligned} L(f_0(x,t)) &= \mathbb{E}_{x,y,t\sim\mathcal{D}_{\text{mod}(t,T)}(x,y)p(\text{mod}(t,T))}\Big[\ell(f_0(x,t),y)\Big] \\ &= \mathbb{E}_{p(\text{mod}(t,T))}\Big\{\mathbb{E}_{x,y\sim\mathcal{D}_{\text{mod}(t,T)}(x,y)}\Big[\ell(f_0(x,t),y)\Big]\Big\} \\ &\leq \mathbb{E}_{p(\text{mod}(t,T))}\Big\{\mathbb{E}_{x,y\sim\mathcal{D}_{\text{mod}(t,T)}(x,y)}\Big[\ell(f(x,t),y)\Big]\Big\} \\ &= L(f(x,t)), \end{aligned} \tag{15}$$

where the inequality follows from the assumption that

$$\mathbb{E}_{x,y\sim\mathcal{D}_{\text{mod}(t,T)}(x,y)}\Big[\ell(f_0(x,t),y)\Big] \leq \mathbb{E}_{x,y\sim\mathcal{D}_{\text{mod}(t,T)}(x,y)}\Big[\ell(f(x,t),y)\Big] \tag{16}$$

for any $f(x,t) \in \mathcal{H}$. Hence, $f_0(x,t)$ is a minimizer of (2).

B. Proof of Lemma 2

We prove the lemma in two parts.

- First, we show that $\mathcal{H}$ is a Hilbert space.
- Then, we show the existence of an isomorphism between $\mathcal{H}$ and $\mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(S^T)$.

B.1. Proving $\mathcal{H}$ is a Hilbert space.

$\mathcal{H}$ is a pre-Hilbert space. It is relatively straight-forward that $\mathcal{H}$ is a linear vector space, where the zero-vector is $f(x) \equiv 0$ almost everywhere under the uniform measure over $\mathcal{X} \times [0,T]$. We now show that $\langle\cdot,\cdot\rangle$ is an inner product.

First, $\langle\cdot,\cdot\rangle$ is symmetric:

$$\langle f,g\rangle = \int_{\mathcal{X}}\int_0^T f(x,t)g(x,t)\mathrm{d}x\mathrm{d}t \int_{\mathcal{X}}\int_0^T g(x,t)f(x,t)\mathrm{d}x\mathrm{d}t = \langle g,f\rangle.$$

It is bi-linear:

$$\begin{aligned} \langle f_1+f_2,g\rangle &= \int_{\mathcal{X}}\int_0^T (f_1(x,t)+f_2(x,t))g(x,t)\mathrm{d}x\mathrm{d}t \\ &= \int_{\mathcal{X}}\int_0^T f_1(x,t)g(x,t)\mathrm{d}x\mathrm{d}t + \int_{\mathcal{X}}\int_0^T f_2(x,t)g(x,t)\mathrm{d}x\mathrm{d}t \\ &= \langle f_1,g\rangle + \langle f_2,g\rangle; \end{aligned}$$

and

$$\langle \lambda f,g\rangle = \int_{\mathcal{X}}\int_0^T \lambda f(x,t)g(x,t)\mathrm{d}x\mathrm{d}t = \lambda\int_{\mathcal{X}}\int_0^T f(x,t)g(x,t)\mathrm{d}x\mathrm{d}t = \lambda\langle f,g\rangle.$$

It is positive-definite:

$$\langle f,f\rangle = \int_{\mathcal{X}}\int_0^T f^2(x,t)\mathrm{d}x\mathrm{d}t \geq 0,$$

where the equality holds if and only if $f(x,t) \equiv 0$ almost everywhere on $\mathcal{X} \times [0,T]$ under the uniform measure.

Therefore, $\langle \cdot, \cdot \rangle$ is an inner product, and $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is a pre-Hilbert space.

$\mathcal{H}$ is a Hilbert space. A pre-Hilbert space $\mathcal{H}$ is a Hilbert space if it is also complete, i.e., every Cauchy sequence $\{f_i\}_{i=1}^{\infty}$ in $\mathcal{H}$ has a limit in $\mathcal{H}$. To prove this, it suffices to find a convergent subsequence of $\{f_i\}_{i=1}^{\infty}$.

Let n_k be such that $\|f_{n_k} - f_j\|^2 \leq 2^{-k}$, for all $j \geq n_k$. Let $B_j = \{(x,t) : |f_{n_{j+1}}(x,t) - f_{n_j}(x,t)| \geq 2^{-j/3}\}$, and $B = \cap_{k=1}^{\infty} \cup_{j \geq k} B_j$. Let μ be the uniform probability measure over $\mathcal{X} \times [0,T]$, then by Chebyshev's inequality, we have (subject to a multiplicative volume constant)

$$\mu(B_j) \leq 2^{2j/3} \|f_{n_{j+1}} - f_{n_j}\|^2 \leq 2^{-j/3}. \tag{17}$$

Noticing that $\sum_j 2^{-j/3}$ is convergent, we can apply the Borel-Cantelli lemma to reach the conclusion that $\mu(B) = 0$. This means that the set of (x,t) that belongs to infinitely many B_j's has 0 measure, which further implies that $\{f_{n_j}\}_{j=1}^{\infty}$ converges point-wise almost everywhere. Define the point-wise limit as f, and as a standard result in real analysis, we know that f is measurable. Then, it only remains to show that $f \in \mathcal{H}$. Using Fatou's lemma and the definition of the subsequence $\{f_{n_j}\}_{j=1}^{\infty}$, we have

$$\begin{aligned}
\|f_{n_j} - f\|^2 &= \int_{\mathcal{X}} \int_0^T \lim_{k \to \infty} |f_{n_j}(x,t) - f_k(x,t)|^2 \mathrm{d}x\mathrm{d}t \\
&\leq \liminf_{k \to \infty} \int_{\mathcal{X}} \int_0^T |f_{n_j}(x,t) - f_k(x,t)|^2 \mathrm{d}x\mathrm{d}t \\
&= \liminf_{k \to \infty} \|f_{n_j} - f_k\|^2 \\
&\leq 2^{-j}.
\end{aligned}$$

Hence, we have both $\|f\|^2 \leq \infty$ by triangle inequality of the norm, and the convergence of f_{n_j} to f in $\mathcal{H}$ norm. Therefore, $\mathcal{H}$ is a Hilbert space.

B.2. Proving the existence of an isohorphism.

We define the following mapping U such that

$$U(\phi_i \otimes \psi_j)(x,t) = \phi_i(x)\psi_j(t)$$

for $x \in \mathcal{X}, t \in [0,T]$. Then, U is a unitary mapping of $\mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(S^T)$ onto $\mathcal{H}$. For any $f \in \mathcal{L}_2(\mathcal{X})$ and $g \in \mathcal{L}_2(S^T)$, assume

$$f(x) = \sum_i a_i \phi_i(x), \qquad \text{and} \qquad g(t) = \sum_j b_i \psi_j(t),$$

then, by linearity,

$$\begin{aligned}
U(f \otimes g)(x,t) &= U\left(\left(\sum_i a_i \phi_i\right) \otimes \left(\sum_j b_i \psi_j\right)\right)(x,t) \\
&= U\left(\sum_{i,j} a_i b_j (\phi_i \otimes \psi_j)\right)(x,t) \\
&= \sum_{i,j} a_i b_j \phi_i(x) \psi_j(t) \\
&= f(x)g(t).
\end{aligned}$$

This shows that there exists an isomorphism between $\mathcal{L}_2(\mathcal{X}) \otimes \mathcal{L}_2(S^T)$ and $\mathcal{H}$.

C. Proof of Lemma 5

We prove the lemma in two steps:

- First, we show that $f_t^*(x)$ is continuous in t for any $x \in \mathcal{X}$.
- Exploiting the continuity of $f_t^*(x)$, we show $f_0(x,t) \stackrel{\text{def}}{=} f_t^*(x)$ for every t is in $\mathcal{H}$.

In addition, we need the following lemma.

Lemma 10. *Let $\mathcal{X}$ be some metric space and $\widetilde{h} : \mathcal{X} \mapsto \mathbb{R}$ be some function such that $\|\widetilde{h}\|_\infty \stackrel{\text{def}}{=} \max_{x\in\mathcal{X}} \widetilde{h}(x) \le K$. Let p, q be two distributions over $\mathcal{X}$. Then*

$$\mathbb{E}_{X\sim p}[\widetilde{h}(X)] - \mathbb{E}_{X\sim q}[\widetilde{h}(X)] \le 2K\|p-q\|_{\text{TV}}.$$

Proof. The proof follows from the following reasoning:

$$\begin{aligned}
\mathbb{E}_{X\sim p}[\widetilde{h}(X)] - \mathbb{E}_{X\sim q}[\widetilde{h}(X)] &\le \sup\left\{ |\mathbb{E}_{X\sim p}[h(X)] - \mathbb{E}_{X\sim q}[h(X)]| \;\middle|\; \|h\|_\infty \le K \right\} \\
&= K \sup\left\{ \left|\mathbb{E}_{X\sim p}\left[\frac{h(X)}{K}\right] - \mathbb{E}_{X\sim q}\left[\frac{h(X)}{K}\right]\right| \;\middle|\; \|h\|_\infty \le K \right\} \\
&= K \sup\left\{ |\mathbb{E}_{X\sim p}[h(X)] - \mathbb{E}_{X\sim q}[h(X)]| \;\middle|\; \|h\|_\infty \le 1 \right\} \\
&= 2K\|p-q\|_{\text{TV}}. \qquad \text{(By the dual representation of total variation distance)}
\end{aligned}$$

□

C.1. Showing $f_t^*(x)$ is continuous in t for any $x \in \mathcal{X}$.

Without loss of generality, consider a fixed $x \in \mathcal{X}$ and a fixed $\epsilon > 0$. Under a realizable setting, we can define

$$u_x(t) \stackrel{\text{def}}{=} f_t^*(x) \in \operatorname*{argmin}_{u\in\mathbb{R}} \mathbb{E}_{y\sim\mathcal{D}_t(x)}[\ell(u,y)], \qquad \forall t \in [0,T].$$

Here, $u(t)$ is a well-defined function because $\ell(u,y)$ is strongly convex in u for each $y \in \mathcal{Y}$. Since $\mathcal{D}_t(x)$ is continuous in t under total variation distance, we can find some $\delta > 0$ such that whenever $|t-t'| \le \delta$, we have $\|\mathcal{D}_t(x) - \mathcal{D}_{t'}(x)\|_{\text{TV}} \le \epsilon$.

Now, for arbitrary t' such that $|t-t'| \le \delta$, the function $g_{t'}(u) \stackrel{\text{def}}{=} \mathbb{E}_{y\sim\mathcal{D}_{t'}(x)}[\ell(u,y)]$ is σ-strongly convex in u. By definition, we can see that $u_x(t')$ minimizes $g_{t'}(u)$. Thus, the property of strongly convex function yields the following:

$$\begin{aligned}
|u_x(t) - u_x(t')| \cdot \frac{\sigma}{2} &\le g_{t'}(u_x(t)) - g_{t'}(u_x(t')) = g_{t'}(u_x(t)) - g_t(u_x(t)) + g_t(u_x(t)) - g_{t'}(u_x(t')) \\
&\le g_{t'}(u_x(t)) - g_t(u_x(t)) + g_t(u_x(t')) - g_{t'}(u_x(t')) \qquad \text{(Since } u(t) \text{ minimizes } g_t(u)) \\
&= \mathbb{E}_{y\sim\mathcal{D}_{t'}(x)}[\ell(u_x(t),y)] - \mathbb{E}_{y\sim\mathcal{D}_t(x)}[\ell(u_x(t),y)] + \mathbb{E}_{y\sim\mathcal{D}_t(x)}[\ell(u_x(t'),y)] - \mathbb{E}_{y\sim\mathcal{D}_{t'}(x)}[\ell(u_x(t'),y)] \\
&\le 4K\|\mathcal{D}_t(x) - \mathcal{D}_{t'}(x)\|_{\text{TV}} \le 4K\epsilon. \quad \text{(By Lemma 10 and boundedness assumption of loss function)}
\end{aligned}$$

Since $\sigma > 0$, we have $|u(t) - u(t')| = |f_t^*(x) - f_{t'}^*(x)| \le 8K\sigma^{-1}\epsilon$ for all $|t-t'| < \delta$. Hence, driving ϵ and δ towards 0 leads to the continuity of $f_t^*(x)$ as a function of t for any $x \in \mathcal{X}$.

C.2. Showing $f_t^*(x) \in \mathcal{H}$.

Since $f_t^*(x)$ is continuous in t for any $x \in \mathcal{X}$, $h(t) \stackrel{\text{def}}{=} \int_{\mathcal{X}} |f_t^*(x)|^2 \mathrm{d}x$ is also continuous in t. Since $f_t^* \in \mathcal{L}_2(\mathcal{X})$, $h(t)$ is finite for any t. Therefore, $h(t)$ is bounded over $[0,T]$.

Let $M^2 = \max_{t\in[0,T]} h(t)$. We have

$$\|f_0(x,t)\|_{\mathcal{H}}^2 = \int_{\mathcal{X}} \int_0^T f_0^2(x,t)\mathrm{d}x\mathrm{d}t \le \int_0^T \|f_t^*(x)\|_{\mathcal{L}_2(\mathcal{X})}^2 \mathrm{d}t \le M^2 T. \tag{18}$$

This implies that $f_0(x,t)$ has finite $\mathcal{H}$-norm, and therefore we have $f_0(x,t) \in \mathcal{H}$.

D. Proof of Theorem 6

The proof follows directly upon noticing that $\{\exp(2\pi jnt/T)\}_{n=-\infty}^{\infty}$, where j is the imaginary number, is a complete orthonormal set of basis for $\mathcal{L}_2(S^T)$ and applying Eq. (5).

E. Computing the Kernel Embeddings of a Functional Gradient

In this section, we illustrate the computation of kernel embeddings for functional gradients through an example. Adopting a simplified setting, we consider the following loss function:

$$\ell(f(x,t),y) = \frac{1}{2}(f(x,t)-y)^2.$$

Denoting this functional as $L(a_n, b_n)$ with a_n and b_n defined in (7), we can derive the (stochastic) functional gradient for a_n as follows given samples $(x_m, y_m, t_m)_{m=1}^M$ that arrive between the (k,τ)-th and the $(k,\tau+1)$-th iterations:

$$[\nabla_{a_n}\widehat{L}_{k,\tau}(a_n,b_n)](x) = \sum_{m=1}^{M}(f(x_m,t_m)-y_m)\times\sin(2\pi nt/T)\delta(x-x_m).$$

Let $K(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be a positive-definite kernel, we have the kernel embedding of

$$\begin{aligned} g_{a_n}(x) &= \int_{\mathcal{X}}[\nabla_{a_n}\widehat{L}_{k,\tau}(a_n,b_n)](y)K(x,y)\mathrm{d}y \\ &= \sum_{m=1}^{M}(f(x_m,t_m)-y_m)\times\sin(2\pi nt/T)K(x_m,x), \end{aligned}$$

where the second equality follows from the property of the Dirac's δ-function.

F. Proof of Theorem 8

In this section, we prove the major Theorems related to the convergence of streaming-SGD for Fourier learning. In particular, we first prove 8 under both strongly convex and non-convex setting. Then we prove 9, which extends the result to non-parametric case.

F.1. Strongly Convex Case

For the sake of simplicity, denote the concatenation of all $\theta_n^{(k,\tau)}$ for $n\in\{0,\dots,N\}$ and $\rho_n^{(k,\tau)}$ for $n\in\{1,\dots,N\}$ as $U^{(k,\tau)}$, and denote its optimal as U^*. By the moment assumption, we immediately have

$$\max_{k,\tau}\mathbb{E}\|\nabla\widehat{L}_{k,\tau}(U^{(k,\tau)})\|^2 \le (2N+1)G^2, \tag{19}$$

and

$$\max_{k,\tau}\mathbb{E}\|\nabla\widehat{L}_{k,\tau}(U^{(k,\tau)})\| \le \sqrt{(2N+1)G^2}. \tag{20}$$

In addition, denoting $\eta_k = \eta_{k,\tau}$ for $\tau\in\{1,\dots,\Gamma\}$, we also have

$$U^{(k+1,1)} = U^{(k,1)} - \eta_k\sum_{\tau=1}^{\Gamma}\nabla\widehat{L}_{k,\tau}(U^{(k,\tau)}), \tag{21}$$

where $\nabla\widehat{L}_{k,\tau}$ is the incremental gradient of L with respect to U calculated using the data that arrived between (k,τ)-th and $(k,\tau-1)$-th iterations.[5] Since $\nabla\widehat{L}_{k,\tau}(U^{(k,\tau)}$ is computed using only the data between the (k,τ)-th and the $(k,\tau+1)$-th iterations, we can alternatively write (21) as

$$U^{(k+1,1)} = U^{(k,1)} - \eta_k(\nabla\widehat{L}_k(U^{(k,1)}) - e^{(k)}), \tag{22}$$

[5]Note that for a batch with size n, we have chosen to compute the empirical average of $\widehat{L}_{k,\tau}$ using coefficient $\frac{1}{n\Gamma}$ instead of $\frac{1}{n}$. This design ensures $\mathbb{E}[\widehat{L}_k] \overset{\text{def}}{=} \mathbb{E}\left[\sum_{\tau=1}^{\Gamma}\widehat{L}_{k,\tau}\right] = L$.

where $\nabla \widehat{L}_k(U^{(k,1)})$ is now the stochastic gradient of L evaluated at $U^{(k,1)}$, calculated using the data that arrived in macro-iteration k:

$$\nabla \widehat{L}_k(U) = \sum_{\tau=1}^{\Gamma} \nabla \widehat{L}_{k,\tau}(U). \tag{23}$$

Meanwhile, the error term $e^{(k)}$ has the following expression:

$$e^{(k)} = \sum_{\tau=1}^{\Gamma} \left[\nabla \widehat{L}_{k,\tau}(U^{(k,1)}) - \nabla \widehat{L}_{k,\tau}(U^{(k,\tau)}) \right]. \tag{24}$$

Taking norm on both sides and using triangle inequality, we have

$$\|e^{(k)}\| \leq \sum_{\tau=1}^{\Gamma} \|\nabla \widehat{L}_{k,\tau}(U^{(k,1)}) - \nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})\|. \tag{25}$$

Further invoking the Lipschitzness of the gradient gives us

$$\|e^{(k)}\| \leq \sum_{\tau=1}^{\Gamma} \Lambda \|U^{(k,1)} - U^{(k,\tau)}\|. \tag{26}$$

Since $\|U^{(k,1)} - U^{(k,\tau)}\| \leq \eta_k(\tau-1) \cdot \max_{\kappa,\nu} \|\nabla \widehat{L}_{\kappa,\nu}(U^{(\kappa,\nu)})\| \leq \eta_k(\tau-1)G\sqrt{2N+1}$, we can invoke (20) and get

$$\mathbb{E}\|e^{(k)}\| \leq \sum_{\tau=1}^{\Gamma} \Lambda(\tau-1)\eta_k G\sqrt{2N+1} = \frac{1}{2}\Lambda\Gamma(\Gamma-1)\eta_k G\sqrt{2N+1}, \tag{27}$$

and

$$\mathbb{E}\|e^{(k)}\|^2 \leq \frac{1}{6}\eta_k^2\Lambda^2\Gamma^2(\Gamma-1)(2\Gamma-1)G^2(2N+1) \tag{28}$$

upon taking expectations on both sides. This bound implies that the expected error of the micro-iterations over τ vanishes asymptotically if we choose a set of diminishing step sizes η_k.

In the meantime, using (22), we have

$$\begin{aligned}
\|U^{(k+1,1)} - U^*\|^2 &= \|U^{(k,1)} - U^* - \eta_k(\nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)})\|^2 &(29)\\
&= \|U^{(k,1)} - U^*\|^2 + \eta_k^2\|\nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)}\|^2 - 2\eta_k\langle U^{(k,1)} - U^*, \nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)}\rangle. &(30)
\end{aligned}$$

By the rule of total expectation and the assumption on strong convexity of L, we have

$$\begin{aligned}
\mathbb{E}\left[2\eta_k\langle U^{(k,1)} - U^*, \nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)}\rangle\right] &= \mathbb{E}\left[2\eta_k\langle U^{(k,1)} - U^*, \nabla \widehat{L}_k(U^{(k,1)})\rangle\right] - \mathbb{E}\left[2\eta_k\langle U^{(k,1)} - U^*, e^{(k)}\rangle\right] &(31)\\
&= \mathbb{E}\left[2\eta_k\langle U^{(k,1)} - U^*, \nabla L(U^{(k,1)})\rangle\right] - \mathbb{E}\left[2\eta_k\langle U^{(k,1)} - U^*, e^{(k)}\rangle\right] &(32)\\
&\geq 2\eta_k\sigma\mathbb{E}\|U^{(k,1)} - U^*\|^2 - 2\eta_k\mathbb{E}\|U^{(k,1)} - U^*\| \cdot \mathbb{E}\|e^{(k)}\| &(33)\\
&\geq 2\eta_k\sigma\mathbb{E}\|U^{(k,1)} - U^*\|^2 - 2\eta_k^2 G\Gamma\sqrt{2N+1} \cdot \mathbb{E}\|U^{(k,1)} - U^*\| &(34)
\end{aligned}$$

Combining (29) and (31), we have

$$\begin{aligned}
\mathbb{E}\|U^{(k+1,1)} - U^*\|^2 \leq \mathbb{E}\|U^{(k,1)} - U^*\|^2 + 2\eta_k^2(\mathbb{E}\|\nabla \widehat{L}_k(U^{(k,1)})\|^2 + \mathbb{E}\|e^{(k)}\|^2) - 2\eta_k\sigma\mathbb{E}\|U^{(k,1)} - U^*\|^2 + \\
+ 2\eta_k^2 G\Gamma\sqrt{2N+1} \cdot \mathbb{E}\|U^{(k,1)} - U^*\|. \qquad (35)
\end{aligned}$$

Rearranging and assuming D to be the diameter of the space containing $U^{(k,\tau)}$, we can further upper bound the right-hand side of (35) by

$$\mathbb{E}\|U^{(k+1,1)} - U^*\|^2 \leq (1-2\eta_k\sigma)\mathbb{E}\|U^{(k,1)} - U^*\|^2 + 2\eta_k^2(\mathbb{E}\|\nabla\widehat{L}_k(U^{(k,1)})\|^2 + \mathbb{E}\|e^{(k)}\|^2) + 2\eta_k^2 G\Gamma D\sqrt{2N+1} \quad (36)$$

$$= (1-2\eta_k\sigma)\mathbb{E}\|U^{(k,1)} - U^*\|^2 + 2\eta_k^2(2N+1)G^2 + 2\eta_k^2 G\Gamma D\sqrt{2N+1} + \\ + \frac{1}{3}\eta_k^4\Lambda^2\Gamma^2(\Gamma-1)(2\Gamma-1)G^2(2N+1) \quad (37)$$

Denoting

$$C_1 = (2N+1)G^2 + G\Gamma D\sqrt{2N+1} + \frac{1}{6}\eta_k^2\Gamma^2\Lambda^2(\Gamma-1)(2\Gamma-1)G^2(2N+1), \quad (38)$$

we have

$$\mathbb{E}\|U^{(k+1,1)} - U^*\|^2 \leq (1-2\eta_k\sigma)\mathbb{E}\|U^{(k,1)} - U^*\|^2 + 2\eta_k^2 C_1. \quad (39)$$

Finally, selecting $\eta_k = \mathcal{O}(1/k)$ and using mathematical induction yields $\mathbb{E}\|U^{(k+1,1)} - U^*\|^2 = \mathcal{O}(1/k)$.

F.2. Non-Convex Case

In this section, we show that gradient norm vanishes for streaming-SGD under a constant or a cycle-wise diminishing learning rate.

Following the notations defined in Theorem 8, we invoke the smoothness assumption, and obtain

$$L(U^{(k+1,1)}) \leq L(U^{(k,1)}) - \eta_k\langle\nabla L(U^{(k,1)}), V^{(k)}\rangle + \frac{\Lambda^2\eta_k^2}{2}\|V^{(k)}\|^2, \quad (40)$$

where

$$V^{(k)} = \nabla\widehat{L}_k(U^{(k,1)}) - e^{(k)} \quad (41)$$

with $e^{(k)}$ defined in (24). Taking the unconditional expectations on both sides, we have

$$\mathbb{E}[L(U^{(k+1,1)})] \leq \mathbb{E}[L(U^{(k,1)})] - \eta_k\mathbb{E}\|\nabla L(U^{(k,1)})\|^2 + \eta_k\mathbb{E}\langle\nabla L(U^{(k,1)}), e^{(k)}\rangle + \frac{\Lambda^2\eta_k^2}{2}\mathbb{E}\|V^{(k)}\|^2 \quad (42)$$

$$\leq \mathbb{E}[L(U^{(k,1)})] - \eta_k\mathbb{E}\|\nabla L(U^{(k,1)})\|^2 + \eta_k\mathbb{E}\|\nabla L(U^{(k,1)})\|\mathbb{E}\|e^{(k)}\| + \frac{\Lambda^2\eta_k^2}{2}\mathbb{E}\|V^{(k)}\|^2, \quad (43)$$

where we have invoked the Cauchy-Schwartz inequality in the second step. Once again, invoking (20) and (27), we obtain the following bounds:

$$\mathbb{E}\|\nabla L(U^{(k,1)})\| = \mathbb{E}\|\mathbb{E}[\nabla\widehat{L}_k(U^{(k,1)})|\mathcal{F}^{(k)}]\| \leq \mathbb{E}\|\Gamma\cdot\max_{k,\tau}\mathbb{E}[\nabla\widehat{L}_{k,\tau}(U^{(k,\tau)})|\mathcal{F}^{(k)}]\| \leq \Gamma\sqrt{(2N+1)G^2}, \quad (44)$$

and

$$\mathbb{E}[\|e^{(k)}\||\mathcal{F}^{(k)}] \leq \frac{1}{2}\Gamma(\Gamma-1)\eta_k G\sqrt{2N+1}. \quad (45)$$

Therefore, there exists a constant C_7 such that

$$\mathbb{E}\|\nabla L(U^{(k,1)})\|\mathbb{E}\|e^{(k)}\| \leq \eta_k C_7. \quad (46)$$

In the meantime, we can upper bound $\mathbb{E}\|V^{(k)}\|^2$ by

$$\mathbb{E}\|V^{(k)}\|^2 \leq 2\mathbb{E}\|\nabla\widehat{L}_k(U^{(k,1)})\|^2 + 2\mathbb{E}\|e^{(k)}\|^2, \quad (47)$$

invoking (20) and (28), we see that there exists a constant C_8 such that

$$\mathbb{E}\|V^{(k)}\|^2 \leq C_8. \quad (48)$$

Plugging (48) and (46) into (43), we see that there exists a constant C_9 such that

$$\mathbb{E}[L(U^{(k+1,1)})] \leq \mathbb{E}[L(U^{(k,1)})] - \eta_k \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 + \eta_k^2 C_9. \tag{49}$$

Hence,

$$\sum_{k=0}^{T_{\max}} \eta_k \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \leq C_9 \sum_{k=0}^{T_{\max}} \eta_k^2 + \mathbb{E}[L(U^{(0,1)})] - \mathbb{E}[L(U^{(T_{\max}+1,1)})] \tag{50}$$

$$\leq C_9 \sum_{k=0}^{T_{\max}} \eta_k^2 + \mathbb{E}[L(U^{(0,1)})] - L(U^*). \tag{51}$$

Constant learning rate. Let $\eta_k = 1/\sqrt{T_{\max}+1}$. Then

$$\sum_{k=0}^{T_{\max}} \frac{1}{\sqrt{T_{\max}+1}} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \leq C_9 + \max(\mathbb{E}[L(U^{(0,1)})] - L(U^*)). \tag{52}$$

Assuming $\max(\mathbb{E}[L(U^{(0,1)})] - L(U^*)) \leq C_{10}$, we have

$$\min_{0\leq k\leq T_{\max}} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \leq \frac{C_9 + C_{10}}{\sqrt{T_{\max}+1}}. \tag{53}$$

Diminishing learning rate. Let $\eta_k = 1/\sqrt{(k+1)}$. Then, (50) can be re-written as

$$\sum_{k=0}^{T_{\max}} \frac{\eta_k}{\sum_{\kappa=0}^{T_{\max}} \eta_\kappa} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \leq \frac{C_9 \sum_{k=0}^{T_{\max}} \frac{1}{k+1} + C_{10}}{\sum_{\kappa=0}^{T_{\max}} \eta_\kappa}. \tag{54}$$

Here, the left-hand side is lower bounded by

$$\sum_{k=0}^{T_{\max}} \frac{\eta_k}{\sum_{\kappa=0}^{T_{\max}} \eta_\kappa} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \geq \min_{1\leq k\leq T_{\max}} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2. \tag{55}$$

In the meantime, the numerator of the right-hand side's numerator is upper bounded by $\mathcal{O}(\log T_{\max})$, whereas the denominator is bounded by $\Theta(\sqrt{T_{\max}})$. Hence, we see that

$$\min_{1\leq k\leq T_{\max}} \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 = \mathcal{O}(\log T_{\max}/\sqrt{T_{\max}}). \tag{56}$$

G. Proof of Theorem 9

The non-parametric case is more challenging as the update rule involves the notion of pseudo-gradients. Unlike the parametric case, where the update is performed in a Euclidean space, the updates for the non-parametric case are performed in $\mathcal{H}$. Nevertheless, the majority of the proof for the parametric case can be retained as follows.

Auxiliary results and lemmas. Similar to the parametric case, we combine all a_n and b_n into a vector

$$U^{(k,\tau)}(x) = [a_0^{(k,\tau)}(x), \ldots, a_N^{(k,\tau)}(x), b_1^{(k,\tau)}(x), \ldots, b_N^{(k,\tau)}(x)] \in \mathcal{H}^{2N+1}. \tag{57}$$

In addition, we define a composite norm for such vectors, denoted by $\|\cdot\|$. Denoting the i-th component of $U^{(k,\tau)}$ as $U_i^{(k,\tau)}$, we define, with a slight abuse of notations,

$$\|U^{(k,\tau)}\| \stackrel{\text{def}}{=} \left\| \left[\|U_1^{(k,\tau)}\|_{\mathcal{H}}, \ldots, \|U_{2N+1}^{(k,\tau)}\|_{\mathcal{H}} \right] \right\|. \tag{58}$$

This composite norm first computes the $\mathcal{H}$-norm of each element of $U^{(k,\tau)}$, and then compute the vector norm of the resulting vector. With this composite norm, we can easily check that (19) and (20) hold for the non-parametric case as well.

Next, we proceed to write the composite update for the non-parametric case. Since we have chosen the pseudo-gradient to be the kernel embedding of the (incremental) functional gradient, we have

$$U^{(k+1,1)}(x) = U^{(k,1)}(x) - \eta_k \sum_{\tau=1}^{\Gamma} \langle [\nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})](\cdot), \mathbf{K}(x,\cdot)\rangle \tag{59}$$

$$= U^{(k,1)}(x) - \eta_k \langle [\nabla \widehat{L}_{k}(U^{(k,1)}) - e^{(k)}](\cdot), \mathbf{K}(x,\cdot)\rangle, \tag{60}$$

where, once again, $\nabla \widehat{L}_k(U^{(k,1)})$ is the empirical gradient function calculated using the samples that arrive during the macro-iteration k, whereas

$$\nabla \widehat{L}_{k,\tau}(U^{(k,\tau)}) = [\nabla_{a_0} \widehat{L}_{k,\tau}(U^{(k,\tau)}), \ldots, \nabla_{a_N} \widehat{L}_{k,\tau}(U^{(k,\tau)}), \nabla_{b_1} \widehat{L}_{k,\tau}(U^{(k,\tau)}), \ldots, \nabla_{b_n} \widehat{L}_{k,\tau}(U^{(k,\tau)})]^\top. \tag{61}$$

Here, $\mathbf{K}(\cdot,\cdot)$ is a $(2N+1)$-dimensional vector stacked up by the kernel $K(\cdot,\cdot)$:

$$\mathbf{K}(\cdot,\cdot) = [K(\cdot,\cdot), \ldots, K(\cdot,\cdot)]^\top. \tag{62}$$

Similar to the definition of the composite norm, the inner product in (60) should be interpreted as a composite inner product, which first takes the inner product of $\mathcal{H}$ over each element of $\nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})$ and $\mathbf{K}(\cdot,\cdot)$, and then takes the vector dot product over the resulting vectors in the $(2N+1)$-dimensional Euclidean space.

Lastly, in (60), $e^{(k)}$ is the accumulated error function:

$$e^{(k)} = \sum_{\tau=1}^{\Gamma} \left[\nabla \widehat{L}_{k,\tau}(U^{(k,1)}) - \nabla \widehat{L}_{k,\tau}(U^{(k,\tau)}) \right]. \tag{63}$$

Note that $e^{(k)}$ is a function of x, as are $\nabla \widehat{L}_{k,\tau}(U^{(k,1)})$ and $\nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})$. Taking the composite norm on both sides, and invoking the Lipschitzness of the gradient for the $\mathcal{H}$-norm, we have

$$\|e^{(k)}\| \leq \sum_{\tau=1}^{\Gamma} \|\nabla \widehat{L}_{k,\tau}(U^{(k,1)}) - \nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})\| \leq \sum_{\tau=1}^{\Gamma} \Lambda \|U^{(k,1)} - U^{(k,\tau)}\|. \tag{64}$$

Since $\|U^{(k,1)} - U^{(k,\tau)}\| \leq \eta_k(\tau-1) \max_{\kappa,\nu} \|\nabla \widehat{L}_{\kappa,\nu}(U^{(\kappa,\nu)})\| \leq \eta_k(\tau-1)G\sqrt{2N+1}$, we can invoke (20) and get

$$\mathbb{E}[\|e^{(k)}\| | \mathcal{F}^{(k)}] \leq \frac{1}{2}\Gamma(\Gamma-1)\eta_k G\sqrt{2N+1}, \tag{65}$$

as well as

$$\mathbb{E}[\|e^{(k)}\|^2 | \mathcal{F}^{(k)}] \leq \frac{1}{6}\eta_k^2 \Lambda^2 \Gamma^2 (\Gamma-1)(2\Gamma-1)G^2(2N+1), \tag{66}$$

upon taking expectations on both sides. Similar to the parametric case, this bound again implies that the approximation error of $U^{(k,\tau)}$ using $U^{(k,1)}$ vanishes if we choose a set of diminishing step sizes η_k.

The remaining part of the proof deviates from the parametric case as the updates use pseudo-gradients. Before introducing the main proof, we first need the following lemma.

Lemma 11 (Lemma 11 of (Yang et al., 2019)). *Letting $\|\cdot\|_\sharp$ be a norm and $\|\cdot\|_{\sharp,*}$ be its dual norm. In addition, suppose $f \in \mathcal{H}$ is continuously differentiable, and satisfies $\|\nabla f(x+y) - \nabla f(y)\|_\sharp \leq L\|y\|_{\sharp,*}$. Then*

$$f(x+y) - f(x) - \langle \nabla f(x), y\rangle \leq \frac{L}{2}\|y\|_{\sharp,*}. \tag{67}$$

Main proof. Letting $\|\cdot\|_{\sharp,*}$ be the composite norm, and denoting

$$V^{(k)}(x) = \langle [\nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)}](\cdot), \mathbf{K}(x,\cdot)\rangle \tag{68}$$

as the pseudo-gradient with error during the macro-iteration k, we can invoke Lemma 11 and (60) on the loss function L, and obtain

$$L(U^{(k+1,1)}) \leq L(U^{(k,1)}) - \eta_k \langle \nabla L(U^{(k,1)}), V^{(k)} \rangle + \frac{\Lambda^2 \eta_k^2}{2} \|V^{(k)}\|^2. \tag{69}$$

Taking unconditional expectations on both sides, we have

$$\mathbb{E}\Big[L(U^{(k+1,1)}) - L(U^*)\Big] \leq \mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] - \eta_k \mathbb{E}\Big[\langle \nabla L(U^{(k,1)}), \mathbb{E}[V^{(k)}|\mathcal{F}^{(k)}] \rangle\Big] + \frac{\Lambda^2 \eta_k^2}{2} \mathbb{E}\|V^{(k)}\|^2 \tag{70}$$

$$\leq \mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] - \eta_k \lambda \|\nabla L(U^{(k,1)})\|^2 + \frac{\Lambda^2 \eta_k^2}{2} \mathbb{E}\|V^{(k)}\|^2 + \\ + \eta_k \mathbb{E}\Big[\langle [\nabla L(U^{(k,1)})](\star), \langle \mathbb{E}[e^{(k)}|\mathcal{F}^{(k)}](\cdot), \mathbf{K}(\star, \cdot) \rangle \rangle\Big] \tag{71}$$

where $\mathcal{F}^{(k)}$ is the minimum σ-algebra generated by $U^{(\kappa,\tau)}$ for all $\kappa \leq k$ and $1 \leq \tau \leq \Gamma$. In addition, the second inequality is obtained by invoking the assumption that the minimum eigenvalue of $K(\cdot, \cdot)$ is $\lambda > 0$. The last term on the right-hand side has two inner products: the outer one is taken with respect to "$\star$", while the inner is taken with respect to "$\cdot$". Note that, both inner products are composite.

By Cauchy's inequality, and the upper bound on $\mathbb{E}[\|e^{(k)}\||\mathcal{F}^{(k)}]$, we get

$$\mathbb{E}\Big[\langle [\nabla L(U^{(k,1)})](\star), \langle \mathbb{E}[e^{(k)}|\mathcal{F}^{(k)}](\cdot), \mathbf{K}(\star, \cdot) \rangle \rangle\Big] \leq \mathbb{E}\|\nabla L(U^{(k,1)})\| \mathbb{E}\|\mathbf{K}(\star, \cdot)\| \cdot \frac{1}{2}\Gamma(\Gamma - 1)\eta_k G\sqrt{2N+1}. \tag{72}$$

Letting

$$C_2 = \mathbb{E}\|\mathbf{K}(\star, \cdot)\| \cdot \frac{1}{2}\Gamma(\Gamma - 1) G\sqrt{2N+1}, \tag{73}$$

we get

$$\mathbb{E}\Big[\langle [\nabla L(U^{(k,1)})](\star), \langle \mathbb{E}[e^{(k)}|\mathcal{F}^{(k)}](\cdot), \mathbf{K}(\star, \cdot) \rangle \rangle\Big] \leq C_2 \eta_k \mathbb{E}\|\nabla L(U^{(k,1)})\|. \tag{74}$$

Note that, in (74), we can further upper bound $\mathbb{E}\|\nabla L(U^{(k,1)})\|$ by the convexity of the composite norm:

$$\mathbb{E}\|\nabla L(U^{(k,1)})\| = \mathbb{E}\|\mathbb{E}[\nabla \widehat{L}_k(U^{(k,1)})|\mathcal{F}^{(k)}]\| \leq \mathbb{E}\|\Gamma \cdot \max_{k,\tau} \mathbb{E}[\nabla \widehat{L}_{k,\tau}(U^{(k,\tau)})|\mathcal{F}^{(k)}]\| \leq \Gamma\sqrt{(2N+1)G^2}. \tag{75}$$

Therefore, letting $C_3 = C_2 \Gamma \sqrt{(2N+1)G^2}$ gives us

$$\mathbb{E}\Big[\langle [\nabla L(U^{(k,1)})](\star), \langle \mathbb{E}[e^{(k)}|\mathcal{F}^{(k)}](\cdot), \mathbf{K}(\star, \cdot) \rangle \rangle\Big] \leq \eta_k C_3. \tag{76}$$

In the meantime, we can invoke the Cauchy's inequality again, and, according to the expression of $V^{(k)}$ given in (68), we have

$$V^{(k)}(x) \leq \|\mathbf{K}(x, \cdot)\| \|\nabla \widehat{L}_k(U^{(k,1)}) - e^{(k)}\| \leq \max_{x \in \mathcal{X}} \|\mathbf{K}(x, \cdot)\| (\|\nabla \widehat{L}_k(U^{(k,1)})\| + \|e^{(k)}\|). \tag{77}$$

Since $\mathcal{X}$ is compact, there exists a constant C_4 such that

$$\|V^{(k)}\|^2 \leq C_4 (\|\nabla \widehat{L}_k(U^{(k,1)})\| + \|e^{(k)}\|)^2 \leq 2C_4 (\|\nabla \widehat{L}_k(U^{(k,1)})\|^2 + \|e^{(k)}\|^2). \tag{78}$$

Taking the unconditional expectations on both sides, and invoking the upper bounds in (19) and (66), we get

$$\mathbb{E}\|V^{(k)}\|^2 \leq 2C_4 \Big[(2N+1)G^2 + \frac{1}{6}\eta_k^2 \Lambda^2 \Gamma^2 (\Gamma - 1)(2\Gamma - 1) G^2 (2N+1)\Big]. \tag{79}$$

For simplicity, we denote the right-hand side of (79) as C_5, which gives the following succinct relationship:

$$\mathbb{E}\|V^{(k)}\|^2 \leq C_5. \tag{80}$$

Combining (80), (76) and (71), we get

$$\mathbb{E}\Big[L(U^{(k+1,1)}) - L(U^*)\Big] \leq \mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] - \eta_k \lambda \mathbb{E}\|\nabla L(U^{(k,1)})\|^2 + \eta_k^2 C_6, \tag{81}$$

where

$$C_6 = \frac{\Lambda^2 C_5}{2} + C_3. \tag{82}$$

Since L is σ-strongly-convex, we have

$$\mathbb{E}\|\nabla L(U^{(k,1)})\|^2 \geq 2\sigma^{-1}\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big]. \tag{83}$$

Hence,

$$\mathbb{E}\Big[L(U^{(k+1,1)}) - L(U^*)\Big] \leq (1 - 2\sigma^{-1}\lambda\eta_k)\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] + \eta_k^2 C_6. \tag{84}$$

Choosing

$$\eta_k = \frac{\sigma(2k+1)}{2\lambda(k+1)^2}, \tag{85}$$

we get

$$\mathbb{E}\Big[L(U^{(k+1,1)}) - L(U^*)\Big] \leq \frac{k^2}{(k+1)^2}\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] + C_6 \cdot \frac{(2k+1)^2}{(k+1)^4}, \tag{86}$$

or equivalently,

$$(k+1)^2\mathbb{E}\Big[L(U^{(k+1,1)}) - L(U^*)\Big] \leq k^2\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] + C_6\frac{(2k+1)^2}{(k+1)^2} \tag{87}$$

$$\leq k^2\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big] + 4C_6. \tag{88}$$

Denoting

$$\delta(k) = k^2\mathbb{E}\Big[L(U^{(k,1)}) - L(U^*)\Big], \tag{89}$$

we immediately have

$$\delta(k+1) \leq \delta(k) + 4C_6. \tag{90}$$

Noticing that $\delta(0) = 0$, we can sum both sides over 0 to $T_{\max}$, and get

$$\delta(T_{\max}) \leq 4T_{\max}C_6, \tag{91}$$

implying

$$\mathbb{E}\Big[L(U^{(T_{\max},1)}) - L(U^*)\Big] = \mathcal{O}(T_{\max}^{-1}). \tag{92}$$

The bound on the norm difference follows from the strong convexity assumption of L.

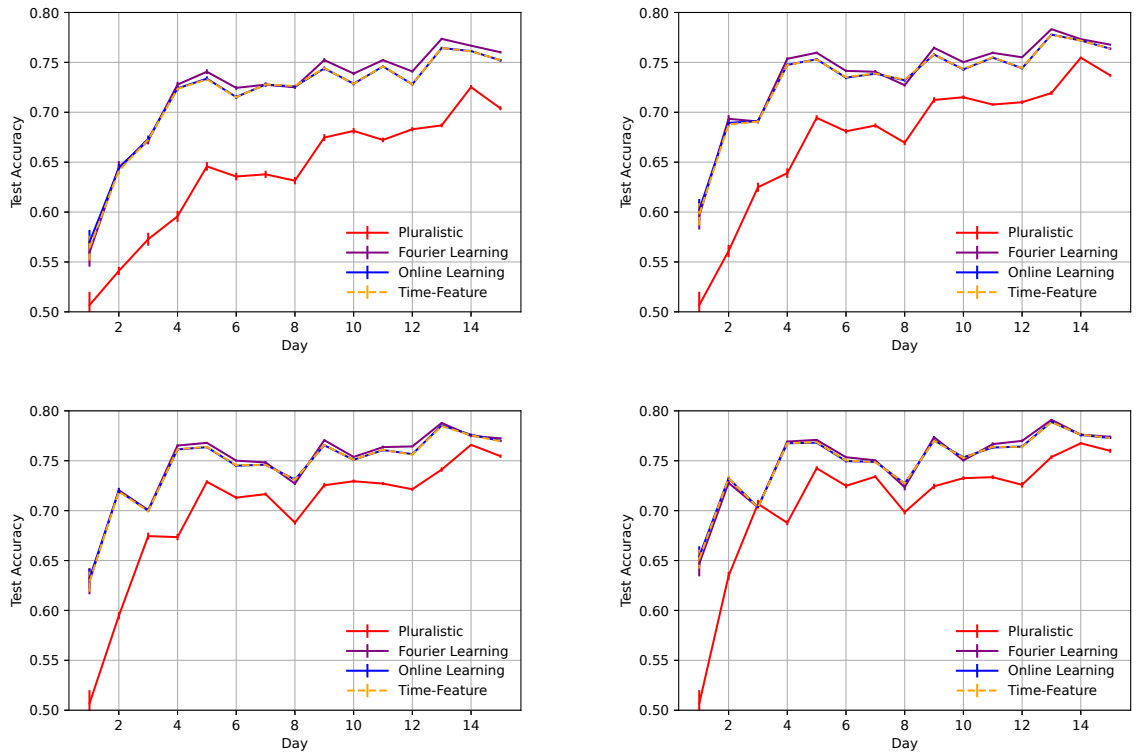

Figure 8. Fourier learning over Sentiment140 Twitter dataset with learning rates 0.05 (top left), 0.1 (top right), 0.2 (bottom left), 0.4 (bottom right), respectively.

H. Multi-Dimensional Fourier-MLP

If the Fourier layer is to replace a hidden layer with a multi-dimensional output, we can substitute the summation of the Fourier layer in Figure 3 with another MLP, so that each output neuron has a unique time interpolation. Using $\text{F-MLP}^{[N]}_{d_1 \to d_2}(x) \in \mathbb{R}^{d_2 \times 1}$ as a general expression for an F-MLP with input dimension d_1 and output dimension d_2, we have:

$$\text{F-MLP}^{[N]}_{d_1 \to d_2}(x) = (W_2 \odot \text{COS}) \cdot \text{MLP}_{d_1 \to (N+1)}(x) + (W_1 \odot \text{SIN}) \cdot \text{MLP}_{d_1 \to N}(x), \tag{93}$$

where $x \in \mathbb{R}^{d_1 \times 1}$ is the input to the Fourier layer; $\text{MLP}_{d_1 \to N}(x) \in \mathbb{R}^{N \times 1}$ is a regular MLP that maps x into a vector of dimension N, having no activations; $W_1 \in \mathbb{R}^{d_2 \times N}$ and $W_2 \in \mathbb{R}^{d_2 \times (N+1)}$; while SIN and COS are matrices stacked up by row vectors $[\sin(2\pi t/T), \ldots, \sin(2\pi N t/T)]$ and $[1, \cos(2\pi t/T), \ldots, \cos(2\pi N t/T)]$ a total of d_2 times. The operator $\odot$ is the Hadamard product. When $d_2 = 1$, W_1 and W_2 can be merged into $\text{MLP}_{d_1 \to N}$ and $\text{MLP}_{d_1 \to (N+1)}$, which serve the role of $a_n(x)$ and $b_n(x)$ in (7), respectively.

I. Additional Experiment Results

I.1. Sentiment140

We report additional results for Fourier learning over the Sentiment140 Twitter dataset in Figure 8. Fourier learning has a better performance across the board. In Figure 9, online learning is still inferior than Fourier learning even after we trippled its network size. This suggests that the gap in Figures 5 and 8 cannot be easily mitigated by tweaking the benchmark.

I.2. Fourier Learning in Recommender Systems

We report the raw data used to plot Figure 7 in Table 2.

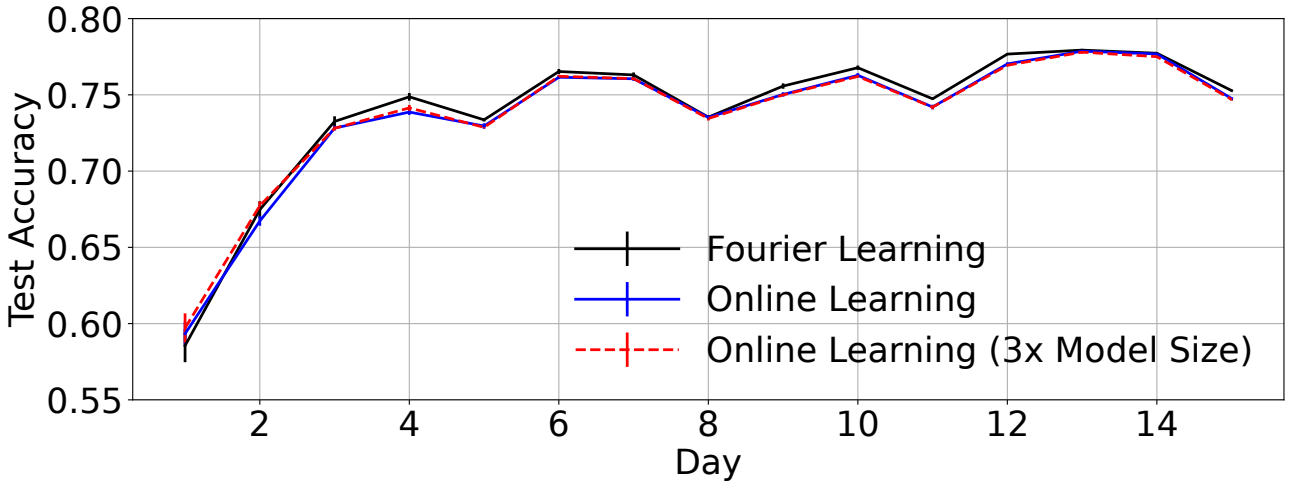

Figure 9. F-MLP against a large online learning model.

Algorithm	Month F	Month G	Month H
Online learning	0.85504	0.85812	0.86350
Online learning (large)	0.85538	0.85766	0.86254
Time-feature	0.85502	0.85810	0.86350
Positional encoding	0.85481	0.85813	0.86351
Pluralistic	0.85497	0.85832	0.86367
Fourier learning (ours)	**0.85627**	**0.85935**	**0.86456**

Table 2. AuC for the implemented algorithms, aggregated monthly.