

Nonstationary Nonparametric Online Learning: Balancing Dynamic Regret and Model Parsimony

Amrit Singh Bedi, Alec Koppel, Ketan Rajawat, and Brian M. Sadler

Abstract—An open challenge in supervised learning is *conceptual drift*: a data point begins as classified according to one label, but over time the notion of that label changes. Beyond linear autoregressive models, transfer and meta learning address drift, but require data that is representative of disparate domains at the outset of training. To relax this requirement, we propose a memory-efficient *online* universal function approximator based on compressed kernel methods. Our approach hinges upon viewing non-stationary learning as online convex optimization with dynamic comparators, for which performance is quantified by dynamic regret.

Prior works control dynamic regret growth only for linear models. In contrast, we hypothesize actions belong to reproducing kernel Hilbert spaces (RKHS). We propose a functional variant of online gradient descent (OGD) operating in tandem with greedy subspace projections. Projections are necessary to surmount the fact that RKHS functions have complexity proportional to time.

For this scheme, we establish sublinear dynamic regret growth in terms of both loss variation and functional path length, and that the memory of the function sequence remains moderate. Experiments demonstrate the usefulness of the proposed technique for online nonlinear regression and classification problems with non-stationary data.

I. INTRODUCTION

A well-known challenge in supervised learning is *conceptual drift*: a data point begins as classified according to one label, but over time the notion of that label changes. For example, an autonomous agent classifies the terrain it traverses as grass, but as the sun sets, the grass darkens. The class label has not changed, but the data distribution has. Mathematically, this situation may be encapsulated by supervised learning with time-series data. Classical approaches assume the current estimate depends linearly on its past values, as in autoregressive models [2], for which parameter tuning is not difficult [3]. While successful in simple settings, these approaches do not apply to classification, alternate quantifiers of model fitness, or universal statistical models such as deep networks [4] or kernel methods [5]. Such modern tools are essential to learning unknown dynamics when assumptions of linear additive Gaussian noise in system identification are invalid, for instance [6], [7].

In the presence of non-stationarity, efforts to train models beyond linear have focused on recurrent networks [8], but such approaches inherently require the temporal patterns of the past and future to be similar. In contrast, transfer learning seeks to

adapt a statistical model trained on one domain to another [9], but requires (1) data to be available in advance of training, and (2) a priori knowledge of when domain shifts happen, typically based on hand-crafted features. Meta-learning overcomes the need for hand-crafted statistics of domain shift by collecting experience over disparate domains and discerning decisions that are good with respect to several environments' training objectives [10]. Combining such approaches with deep networks have yielded compelling results recently [11], [12], although they still require (1) offline training. Hence, in domains where a priori data collection is difficult, due to, e.g., lack of cloud access or rapid changes in the environment, transfer and meta-learning do not apply. In these instances, *online training* is required.

For online training, there are two possible approaches to define learning in the presence of non-stationarity: expected risk minimization [13], [14], and online convex optimization (OCO) [15]. The former approach, due to the fact the data distribution is time-varying distribution, requires the development of stochastic algorithms whose convergence is attuned to temporal aspects of the distribution such as mixing rates [16], [17]. Although mixing rates are difficult to obtain, they substantially impact performance [18]. To mitigate these difficulties, we operate within online convex optimization.

Online convex optimization OCO formulates supervised learning in a distribution-free manner [15]. At each time, a learner selects action f_t after which an arbitrary convex cost $\ell_t : \mathcal{H} \times \mathbb{R}^p \rightarrow \mathbb{R}$ is evaluated as well as parameters $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^p$ of the cost ℓ_t , i.e., the learner suffers cost $\ell(f_t(\mathbf{x}_t))$. Typically, actions f_t are defined by a parameter vector. In contrast, we hypothesize actions $f_t \in \mathcal{H}$ belong to a *function space* \mathcal{H} motivated by nonparametric regression whose details will be deferred to later sections [19]. In classic OCO, one compares cost with a single best action in hindsight; however, with non-stationarity, the quintessential quantifier of performance is instead *dynamic regret*, defined as the cost accumulation as compared with a best action at each time:

$$\mathbf{Reg}_T^D = \sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t))$$

where $f_t^* = \operatorname{argmin}_{f \in \mathcal{H}} L_t(f(\mathbf{S}_t))$. (1)

OCO concerns the design of methods such that \mathbf{Reg}_T^D grows sublinearly in horizon T for a given sequence f_t , i.e., the average regret goes to null with T (referred to as no-regret [20]). Observe that \mathbf{Reg}_T^D , in general, decouples the problem into T time-invariant optimization problems since the minimizer is inside the sum. However, in practice, temporal dependence is intrinsic, as in wireless communications [21],

A.S. Bedi and A. Koppel contributed equally to this work. They both are with the U.S. Army Research Laboratory, Adelphi, MD, USA (e-mail: amrit0714@gmail.com, alec.e.koppel.civ@mail.mil). K. Rajawat is with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: ketan@iitk.ac.in). B. M. Sadler is a senior scientist with the U.S. Army Research Laboratory, Adelphi, MD, USA (email:brian.m.sadler6.civ@mail.mil). A part of this work is submitted to American Control Conference (ACC), Denver, CO, USA, 2020 [1].

autonomous path planning [22], [23], or obstacle detection [24]. Thus, we define (1) in terms of an augmented cost-data pair (L_t, \mathbf{S}_t) which arises from several times, either due to new or previously observed pairs (ℓ_t, \mathbf{x}_t) . Specifications of L_t to time-windowing or batching are discussed in Sec. II.

A. Related Work and Contributions

OCO seeks to develop algorithms whose regret grows sublinearly in time horizon T . In the static case, the simplest approach is online gradient descent (OGD), which selects the next action to descend along the gradient of the loss at the current time. OGD attains static regret growth $\mathcal{O}(T^{1/2})$ when losses are convex [20] and $\mathcal{O}(\log T)$ strongly convex [31], respectively. See Table I for a summary of related works.

The plot thickens when we shift focus to dynamic regret: in particular, [26] establishes the impossibility of attaining sublinear dynamic regret, meaning that one *cannot* track an optimizer varying arbitrarily across time, a fact discerned from an optimization perspective in [32]. Moreover, [26] shows that dynamic regret to be an irreducible function of quantifiers of the problem dynamics called the cost function variation V_T and variable variation W_T (definitions in Sec. II). Thus, several works establish sublinear growth of dynamic regret up to factors depending on V_T and W_T , i.e., $\mathcal{O}(T^{1/2}(1 + W_T))$ for OGD or mirror descent with convex losses [20], [25], more complicated expressions that depend on D_T , the variation of instantaneous gradients [27], and $\mathcal{O}(1 + W_T)$ for strongly convex losses [28].

The aforementioned works entirely focus on the case where decisions define a linear model $\mathbf{w}_t \in \mathcal{W} \subset \mathbb{R}^p$, which, by the estimation-approximation error tradeoff [14], yield small dynamic regret at the cost of large approximation error. Hypothetically, one would like actions to be chosen from a universal function class such as a deep neural network (DNN) [33], [34] or RKHS [35] while attaining no-regret. It's well-understood that no-regret algorithms often prescribe convexity of the loss with respect to actions as a prerequisite [15], thus precluding the majority of DNN parameterizations. While exceptions to this statement exist [36], instead we focus on parameterizations defined in nonparametric statistics [19], namely, RKHS [5], due to the fact they yield universality *and* convexity. Doing so allows us to attain methods that are *both* no-regret and universal in the non-stationary setting. We note that [30] considers a similar setting based on random features [37], but its design cannot be tuned to the learning dynamics; and yields faster regret growth.

Contributions We propose a variant of OGD adapted to RKHS. A challenge for this setting is that the function parameterization stores all observations from the past [38], via the Representer Theorem [39]. To surmount this hurdle, we greedily project the functional OGD iterates onto subspaces constructed from subsets of points observed thus far which are ϵ -close in RKHS norm (Algorithm 1), as in [40], [41], which allows us to explicitly tune the sub-optimality caused by function approximation, in contrast to random feature expansions [37]. Doing so allows us to establish sublinear dynamic regret in terms of both the loss function variation

(Theorem 1) and function space path length (Theorem 2). Moreover, the learned functions yield finite memory (Lemma 1). In short, we derive a tunable tradeoff between memory and dynamic regret, establishing for the first time global convergence for a universal function class in the non-stationary regime (up to metrics of non-stationarity [26]). These results translate into experiments in which one may gracefully address online nonlinear regression and classification problems with non-stationary data, contrasting alternative kernel methods and other state of the art online learning methods.

II. NON-STATIONARY LEARNING

In this section, we clarify details of the loss, metrics of non-stationarity, and RKHS representations that give rise to the derivation of our algorithms in Sec. III. To begin, we assume Tikhonov regularization, i.e., $\ell_t(f(\mathbf{x})) := \check{\ell}_t(f(\mathbf{x})) + (\lambda/2)\|f\|_{\mathcal{H}}^2$ for some convex function $\check{\ell}_t: \mathcal{H} \times \mathcal{X} \rightarrow \mathbb{R}$, which links these methods to follow the *regularized leader* in [15].

Time-Windowing and Mini-Batching To address when the solutions f_t^* are correlated across time or allow for multiple samples per time slot, we define several augmentations of loss data-pairs (ℓ_t, \mathbf{x}_t) .

(i) Classical loss: $L_t = \ell_t$ and $\mathbf{S}_t = \mathbf{x}_t$, and the minimization may be performed over a single datum. In other words, the action taken depends only on the present, as in fading wireless communication channel estimation.

$$(ii) \text{ H-Window : } L_t(f(\mathbf{S}_t)) = \sum_{\tau=t-H+1}^t \ell_{\tau}(f(\mathbf{x}_{\tau})),$$

$$(iii) \text{ Mini-batch : } L_t(f(\mathbf{S}_t)) = \sum_{i=1}^B \ell_t(f(\{\mathbf{x}_t^i\}_{i=1}^B)). \quad (2)$$

The first cost $L_t(f(\mathbf{S}_t))$ in (2)(ii) for each time index t consists $H-1$ previous cost-data pairs $\{\ell_{\tau}, \mathbf{x}_{\tau}\}_{\tau=t-P+1}^{t-1}$ and new cost-data pair (ℓ_t, \mathbf{x}_t) , where we denote samples $\{\mathbf{x}_{\tau}\}$ in this time window as \mathbf{S}_t . $H=1$ simplifies to dynamic regret as in [30]. (2) is useful for, e.g., obstacle avoidance, where obstacle is correlated with time. Typically, we distinguish between the sampling rate of a system and the rate at which model updates occur. If one takes B samples per update, then mini-batching is appropriate, as in (2)(iii). In this work, we focus windowing in (2)(ii), i.e., $H > 1$. Further, instead of one point at t given by \mathbf{x}_t , one may allow B points $\{\mathbf{x}_t^i\}_{i=1}^B$, yielding a hybrid of (2)(ii) - (iii). Our approach naturally extends to mini-batching. For simplicity, we focus on $B=1$. We denote \check{L}_t as the component of (2) without regularization.

Metrics of Non-Stationarity With the loss specified, we shift focus to illuminating the challenges of non-stationarity. As mentioned in Sec. I, [26] establishes that designing no-regret [cf. (1)] algorithms against dynamic comparators when cost functions change arbitrarily is impossible. Moreover, dynamic regret is shown to be an irreducible function of fundamental quantifiers of the problem dynamics called cost function variation and variable variation, which we now define.

Reference	Regret Notion	Loss	Function Class	Regret Bound
[20], [25]	$\sum_{t=1}^T \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}_t^*)$	Convex	Parametric	$\mathcal{O}(\sqrt{T}(1+W_T))$
[26]	$\sum_{t=1}^T \mathbb{E}[\ell_t(\mathbf{w}_t)] - \ell_t(\mathbf{w}_t^*)$	Convex	Parametric	$\mathcal{O}(T^{2/3}(1+W_T)^{1/3})$
[26]	$\sum_{t=1}^T \mathbb{E}[\ell_t(\mathbf{w}_t)] - \ell_t(\mathbf{w}_t^*)$	Strongly convex	Parametric	$\mathcal{O}(\sqrt{T(1+W_T)})$
[27]	$\sum_{t=1}^T \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}_t^*)$	Convex	Parametric	$\mathcal{O}(\sqrt{D_T+1} + \min\{\sqrt{(D_T+1)V_T}, [(D_T+1)W_T T]^{1/3}\})$
[28], [29]	$\sum_{t=1}^T \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}_t^*)$	Strongly convex	Parametric	$\mathcal{O}(1+W_T)$
[30]	$\sum_{t=1}^T \ell_t(f_t(\mathbf{w}_t)) - \sum_{t=1}^T \ell_t(f_t^*(\mathbf{w}_t))$	Convex	Nonparametric	$\mathcal{O}(T^{2/3}V_T^{1/3})$
This Work	$\sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t))$	Convex	Nonparametric	$\mathcal{O}(T^{2/3}V_T^{1/3} + \epsilon T^{2/3}V_T^{-1/3})$
This Work	$\sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t))$	Convex	Nonparametric	$\mathcal{O}(1 + T\sqrt{\epsilon} + W_T)$
This Work	$\sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t))$	Strongly convex	Nonparametric	$o(1 + T\sqrt{\epsilon} + W_T)$

TABLE I: Summary of related works on dynamic online learning. In this work, we have derived the dynamic regret both in terms of V_T and W_T with an additional compression parameter ϵ to control complexity of nonparametric functions, which permits sublinear regret growth for dynamic regret in terms of W_T under selection $\epsilon = \mathcal{O}(T^{-\alpha})$ with $\alpha \in (0, \frac{1}{p}]$, where p is the parameter dimension. Note that for the strongly convex case with $\epsilon = 0$, we obtain $o(1 + W_T)$ which is better than its parametric counterpart obtained in [28]. In particular, we just need the compression budget to be $\epsilon < \mathcal{O}\left(\left(\frac{W_T}{T}\right)^2\right)$ to achieve $\mathcal{O}(1 + W_T)$ dynamic regret.

Specifically, the cost function variation $\text{Var}(L_1, L_2, \dots, L_T)$ tracks the largest loss drift across time:

$$\text{Var}(L_1, L_2, \dots, L_T) := \sum_{t=2}^T |L_t - L_{t-1}|, \quad \mathcal{V} := \left\{ \{L_t\}_{t=1}^T ; \sum_{t=2}^T |L_t - L_{t-1}| \leq V_T \right\}, \quad (3)$$

where $|L_t - L_{t-1}| := \sup_{f \in \mathcal{H}} |L_t(f(\mathbf{S})) - L_{t-1}(f(\mathbf{S}))|$ for all $\mathbf{S} \in \mathcal{X}$ and denote \mathcal{V} as the class of convex losses bounded by V_T for any set of points $\mathbf{S} \in \mathcal{X}$. Further define the variable variation W_T as

$$W_T := \sum_{t=1}^T \|f_{t+1}^* - f_t^*\|_{\mathcal{H}} \quad (4)$$

which quantifies the drift of the optimal function f_t^* over time t . One may interpret (3) and (4) as the distribution-free analogue of mixing conditions in stochastic approximation with dependent noise in [42] and reinforcement learning [43]. Then, our goal is to design algorithms whose growth in dynamic regret (1) is sub-linear, up to constant factors depending on the fundamental quantities (3)-(4).

III. ALGORITHM DEFINITION

Reproducing Kernel Hilbert Space With the metrics and motivation clear, we detail the function class \mathcal{H} that defines how decisions f_t are made. As mentioned in Sec. I, we would like one that satisfies universal approximation theorems [35], i.e., the hypothesis class containing the Bayes optimal [14], while also permitting the derivation of no-regret algorithms through links to convex analysis. RKHSs [5] meet these specifications, and hence we shift to explaining their properties. A RKHS is a Hilbert space equipped with an inner product-like map called a kernel $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ which satisfies

$$(i) \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} = f(\mathbf{x}), \quad (ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad (5)$$

for all $\mathbf{x} \in \mathcal{X}$. Common choices κ include the polynomial kernel and the radial basis kernel, i.e., $\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + b)^c$ and $\kappa(\mathbf{x}, \mathbf{x}') = e^{-(\|\mathbf{x} - \mathbf{x}'\|_2^2)/2c^2}$, respectively, where $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. For such spaces, the function $f^*(\mathbf{x})$ that minimizes the sum, $R(f; \{\mathbf{x}_t\}_{t=1}^T) = \frac{1}{T} \sum_{t=1}^T \ell_t(f; (\mathbf{x}_t))$, over T losses satisfies the Representer Theorem [44], [39]. Specifically, the optimal f may be written as a weighted sum of kernels evaluated *only* at training examples as $f(\mathbf{x}) = \sum_{t=1}^T w_t \kappa(\mathbf{x}_t, \mathbf{x})$, where $\mathbf{w} = [w_1, \dots, w_T]^T \in \mathbb{R}^T$ denotes a set of weights. We define the upper index T as the *model order*.

One may substitute this expression into the minimization of $R(f)$ to glean two observations from the use of RKHS in online learning: the latest action is a weighted combination of kernel evaluations at previous points, e.g., a mixture of Gaussians or polynomials centered at previous data $\{\mathbf{x}_u\}_{u \leq T}$; and that the function's complexity becomes unwieldy as time progresses, since its evaluation involves all past points. Hence, in the sequel, we must control both the growth of regret *and* function complexity.

Functional Online Gradient Descent Begin with functional online gradient method, akin to [38]:

$$\begin{aligned} f_{t+1} &= (1 - \eta H \lambda') f_t - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \\ &= (1 - \eta H \lambda') f_t - \eta \sum_{\tau=t-P+1}^t \check{\ell}'_{\tau}(f_t(\mathbf{x}_{\tau})) \kappa(\mathbf{x}_{\tau}, \cdot), \end{aligned} \quad (6)$$

where the later equality makes use of the definition of $L_t(f_t(\mathbf{S}_t))$ [cf. (2)], the chain rule, and the reproducing property of the kernel (5) – see [38]. We define $\lambda = \lambda' H$. Step-size $\eta > 0$ is chosen as a small constant – see Section IV. We require that, given $\lambda > 0$, the step-size satisfies $\eta < 1/\lambda$ and initialization $f_0 = 0 \in \mathcal{H}$. Given this initialization, one may apply induction and Representer Theorem [39] to write the function f_t at time t as a weighted kernel expansion over past data \mathbf{x}_t as

$$f_t(\mathbf{x}) = \sum_{u=1}^{t-1} w_u \kappa(\mathbf{x}_u, \mathbf{x}) = \mathbf{w}_t^T \boldsymbol{\kappa}_{\mathbf{x}_t}(\mathbf{x}). \quad (7)$$

Algorithm 1 Dynamic Parsimonious Online Learning with Kernels (DynaPOLK)

Require: $\{\mathbf{x}_t, \eta, \epsilon\}_{t=0,1,2,\dots}$

initialize $f_0(\cdot) = 0, \mathbf{D}_0 = [], \mathbf{w}_0 = [],$ i.e. initial dictionary, coefficient vectors are empty

for $t = 0, 1, 2, \dots$ **do**

Obtain independent data realization (\mathbf{x}_t) and loss $\ell_t(\cdot)$

Compute unconstrained functional online gradient step

$$\tilde{f}_{t+1}(\cdot) = (1 - \eta\lambda)f_t - \eta\nabla_f \check{L}_t(f_t(\mathbf{S}_t))$$

Revise dict. $\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t, \mathbf{x}_t]$, weights \mathbf{w}_{t+1} via (11)-(12)

Compress function via KOMP [45] with budget ϵ

$$(f_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) = \mathbf{KOMP}(\tilde{f}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon)$$

end for

On the right-hand side of (7) we have introduced the notation $\mathbf{X}_t = [\mathbf{x}_1, \dots, \mathbf{x}_{t-1}] \in \mathbb{R}^{p \times (t-1)}$, $\kappa_{\mathbf{X}_t}(\cdot) = [\kappa(\mathbf{x}_1, \cdot), \dots, \kappa(\mathbf{x}_{t-1}, \cdot)]^T$, and $\mathbf{w}_t = [w_1; \dots; w_{t-1}]$. We may glean from (7), that the functional update (6) amounts to updates on the data matrix \mathbf{X} and coefficient w_{t+1} :

$$\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_t], \quad w_{t+1} = -\eta\check{\ell}'_t(f_t(\mathbf{x}_t)), \quad (8)$$

In addition, we need to update the last $H - 1$ weights over range $\tau = t - H + 1$ to $t - 1$:

$$w_\tau = \begin{cases} (1 - \eta\lambda)w_\tau - \eta\check{\ell}'_\tau(f_t(\mathbf{x}_\tau)) & \text{for } \tau \in \{t - H + 1, \dots, t - 1\} \\ (1 - \eta\lambda)w_\tau & \text{for } \tau < t - H + 1. \end{cases} \quad (9)$$

Observe that (8) causes \mathbf{X}_{t+1} to have one more column than \mathbf{X}_t . Define the *model order* as number of points (columns) M_t in the data matrix at time t . $M_t = t - 1$ for OGD, growing unbounded.

Model Order Control via Subspace Projection To overcome the aforementioned bottleneck, we propose projecting the OGD sequence (6) onto subspaces $\mathcal{H}_{\mathbf{D}} \subseteq \mathcal{H}$ defined by some dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_M] \in \mathbb{R}^{p \times M}$, i.e., $\mathcal{H}_{\mathbf{D}} = \{f : f(\cdot) = \sum_{i=1}^M w_i \kappa(\mathbf{d}_i, \cdot) = \mathbf{w}^T \kappa_{\mathbf{D}}(\cdot)\} = \text{span}\{\kappa(\mathbf{d}_i, \cdot)\}_{i=1}^M$, inspired by [40]. For convenience we have defined $[\kappa_{\mathbf{D}}(\cdot) = \kappa(\mathbf{d}_1, \cdot) \dots \kappa(\mathbf{d}_M, \cdot)]$, and $\mathbf{K}_{\mathbf{D}, \mathbf{D}}$ as the resulting kernel matrix from this dictionary. We ensure parsimony by ensuring $M_t \ll t$.

Rather than allowing model order of f to grow in perpetuity [cf. (8)], we project f onto subspaces defined by dictionaries $\mathbf{D} = \mathbf{D}_{t+1}$ extracted from past data. Deferring the selection of \mathbf{D}_{t+1} for now, we note it has dimension $p \times M_{t+1}$, with $M_{t+1} \ll t$. Begin by considering function f_{t+1} is parameterized by dictionary \mathbf{D}_{t+1} and weight vector \mathbf{w}_{t+1} . Moreover, we denote columns of \mathbf{D}_{t+1} as \mathbf{d}_t for $t = 1, \dots, M_{t+1}$. We propose a projected variant of OGD:

$$\begin{aligned} f_{t+1} &= \underset{f \in \mathcal{H}_{\mathbf{D}_{t+1}}}{\operatorname{argmin}} \left\| f - \left((1 - \eta\lambda)f_t - \eta\nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right) \right\|_{\mathcal{H}}^2 \\ &:= \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \left[(1 - \eta\lambda)f_t - \eta\nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right] \end{aligned} \quad (10)$$

where we define the projection operator \mathcal{P} onto subspace $\mathcal{H}_{\mathbf{D}_{t+1}} \subset \mathcal{H}$ by the update (10).

Coefficient update The update (10), for a fixed dictionary $\mathbf{D}_{t+1} \in \mathbb{R}^{p \times M_{t+1}}$, implies an update only on coefficients. To illustrate this point, define the online gradient update without projection, given function f_t parameterized by dictionary \mathbf{D}_t and coefficients \mathbf{w}_t , as $\tilde{f}_{t+1} = (1 - \eta H \lambda) f_t - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t))$. This update may be represented using dictionary and weight vector as

$$\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t, \mathbf{x}_t], \quad w_{t+1} = -\eta\check{\ell}'_t(f_t(\mathbf{x}_t)). \quad (11)$$

and revising last $H - 1$ weights with $\tau = t - H + 1$ to $t - 1$, yielding the update for coefficients as

$$w_\tau = \begin{cases} (1 - \eta\lambda)w_\tau - \eta\check{\ell}'_\tau(f(\mathbf{x}_\tau)) & \text{for } \tau = t - H + 1, \dots, t - 1 \\ (1 - \eta\lambda)w_\tau & \text{for } \tau < t - H + 1. \end{cases} \quad (12)$$

For fixed dictionary \mathbf{D}_{t+1} , the projection (10) is a least-squares problem on coefficients \mathbf{w}_{t+1} [46]:

$$\mathbf{w}_{t+1} = \mathbf{K}_{\mathbf{D}_{t+1}, \mathbf{D}_{t+1}}^{-1} \mathbf{K}_{\mathbf{D}_{t+1}, \tilde{\mathbf{D}}_{t+1}} \tilde{\mathbf{w}}_{t+1}. \quad (13)$$

Given that projection of \tilde{f}_{t+1} onto subspace $\mathcal{H}_{\mathbf{D}_{t+1}}$ for a fixed dictionary \mathbf{D}_{t+1} is a simple least-squares multiplication, we turn to explaining the selection of the kernel dictionary \mathbf{D}_{t+1} from past data $\{\mathbf{x}_u\}_{u \leq t}$.

Dictionary Update One way to obtain the dictionary \mathbf{D}_{t+1} from $\tilde{\mathbf{D}}_{t+1}$, as well as the coefficient \mathbf{w}_{t+1} , is to apply a destructive variant of *kernel orthogonal matching pursuit* (KOMP) with pre-fitting [45][Sec. 2.3] as in [40]. KOMP operates by beginning with full dictionary $\tilde{\mathbf{D}}_{t+1}$ and sequentially removing columns while the condition $\|\tilde{f}_{t+1} - f_{t+1}\|_{\mathcal{H}} \leq \epsilon$ holds. The projected FOGD is defined as:

$$(f_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) = \mathbf{KOMP}(\tilde{f}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon), \quad (14)$$

where ϵ is the compression budget which dictates how many model points are thrown away during model order reduction. By design, we have $\|\tilde{f}_{t+1} - f_{t+1}\|_{\mathcal{H}} \leq \epsilon$, which allows us tune ϵ to only keep dictionary elements critical for online descent directions. These details allow one to implement Dynamic Parsimonious Online Learning with Kernels (DynaPOLK) (Algorithm 1) efficiently. Subsequently, we discuss its theoretical and experimental performance.

IV. BALANCING REGRET AND MODEL PARSIMONY

In this section, we establish the sublinear growth of dynamic regret of Algorithm 1 up to factors depending on (4) and the compression budget parameter that parameterizes the algorithm. To do so, some conditions on the loss, its gradient, and the data domain are required which we subsequently state.

Assumption 1. *The feature space $\mathcal{X} \subset \mathbb{R}^p$ is compact, and the reproducing kernel is bounded:*

$$\sup_{\mathbf{x} \in \mathcal{X}} \sqrt{\kappa(\mathbf{x}, \mathbf{x})} = X < \infty. \quad (15)$$

Assumption 2. *The loss $\check{\ell}_t : \mathcal{H} \times \mathcal{X} \rightarrow \mathbb{R}$ is uniformly C -Lipschitz continuous for all $z \in \mathbb{R}$:*

$$|\check{\ell}_t(z) - \check{\ell}_t(z')| \leq C|z - z'|. \quad (16)$$

Assumption 3. The loss $\tilde{\ell}_t(f(\mathbf{x}))$ is convex and differentiable w.r.t. $f(\mathbf{x})$ on \mathbb{R} for all $\mathbf{x} \in \mathcal{X}$.

Assumption 4. The gradient of the loss $\nabla \ell_t(f(\mathbf{x}))$ is Lipschitz continuous with parameter $\tilde{L} > 0$:

$$\|\nabla_f \ell_t(f(\mathbf{S}_t)) - \nabla_g \ell_t(g(\mathbf{S}_t))\|_{\mathcal{H}} \leq \tilde{L} \|f - g\|_{\mathcal{H}} \quad (17)$$

for all t and $f, g \in \mathcal{H}$.

Assumption 1 and Assumption 3 are standard [38], [47]. Assumptions 2 and 4 ensures the instantaneous loss $\tilde{\ell}_t(\cdot)$ and its derivative are smooth, which is usual for gradient-based optimization [48], and holds, for instance, for the square, squared-hinge, or logistic losses. Because we are operating under the windowing framework over last P losses (2), we define the Lipschitz constant of $L_t(\cdot)$ as CP and that of its gradient as $L = H\tilde{L}$. Doing so is valid, as the sum of Lipschitz functions is Lipschitz [49].

Before analyzing the regret of Alg. 1, we discern the influence of the learning rate, compression budget, and problem parameters on the model complexity of the function. In particular, we provide a minimax characterization of the number of points in the kernel dictionary in the following lemma, which determines the required complexity for sublinear dynamic regret growth in different contexts.

Lemma 1. Let f_t be the function sequence of Algorithm 1 with step-size $\eta < \min\{1/\lambda, 1/L\}$ and compression ϵ . Denote M_t as the model order (no. of columns in dictionary \mathbf{D}_t) of f_t . For a Lipschitz Mercer kernel κ on compact set $\mathcal{X} \subseteq \mathbb{R}^p$, there exists a constant Y s.t. for data $\{\mathbf{x}_t\}_{t=1}^{\infty}$, M_t satisfies

$$H \leq M_t \leq Y(CH)^p \left(\frac{\eta}{\epsilon}\right)^p. \quad (18)$$

Lemma 1 (proof in Appendix A) establishes that the model order of the learned function is lower bounded by the time-horizon H and its upper bound depends on the ratio of the step-size to the compression budget, as well as the Lipschitz constant [cf. (16)]. Next, we shift to characterizing the dynamic regret of Algorithm 1. Our first result establishes that the dynamic regret, under appropriate step-size and compression budget selection, grows sublinearly up to a factor that depends on a batch parameter and the cost function variation (3), and that the model complexity also remains moderate. This result extends [26][Proposition 2] to nonparametric settings.

Theorem 1. Denote as $\{f_t\}$ the sequence generated by Algorithm 1 run for T total iterations partitioned into $m = \lceil \frac{T}{\Delta_T} \rceil$ mini-horizons of length Δ_T . Over mini-horizons, Algorithm 1 is run for Δ_T steps. Under Assumptions 1-4, the dynamic regret (1) grows with horizon T and loss variation (3) as:

$$\mathbf{Reg}_T^D = \lceil \frac{T}{\Delta_T} \rceil \mathcal{O} \left(\frac{1 + (\epsilon + \eta^2)\Delta_T}{\eta} \right) + 2\Delta_T V_T, \quad (19)$$

which is sublinear for $\eta = \mathcal{O}(\Delta_T^{-a})$ and $\epsilon = \mathcal{O}(\Delta_T^{-b})$ with mini-horizon $\Delta_T = o(T)$, provided $p(a-b) \in (0, 1)$. That is, with $\eta = \Delta_T^{-1/2}$ and $\epsilon = \Delta_T^{-(p-1)/2p}$, (19) grows sublinearly in T and V_T .

Proof. Consider the expression for the dynamic regret is given by

$$\mathbf{Reg}_T^D = \sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t)). \quad (20)$$

Add subtract the term $\sum_{t=1}^T \ell_t(f^*(\mathbf{x}_t))$ to the right hand side of (20), we obtain

$$\begin{aligned} \mathbf{Reg}_T^D &= \sum_{t=1}^T L_t(f_t(\mathbf{S}_t)) - \sum_{t=1}^T L_t(f^*(\mathbf{x}_t)) \\ &\quad + \sum_{t=1}^T L_t(f^*(\mathbf{x}_t)) - \sum_{t=1}^T L_t(f_t^*(\mathbf{S}_t)) \\ &= \mathbf{Reg}_T^S + \sum_{t=1}^T [L_t(f^*(\mathbf{x}_t)) - L_t(f_t^*(\mathbf{S}_t))]. \end{aligned} \quad (21)$$

We have utilized the definition of static regret in (97) to obtain (21). Note that the behavior in terms of static regret of Algorithm 1 is characterized in Theorem 3. To analyze the dynamic regret in terms of V_T , we need to study the difference between the static optimal and dynamic optimal given by the second term on the right hand side of (21). The difference between the two benchmarks (static and dynamic) is determined by the size of T and fundamental quantifiers of non-stationarity defined in Section II. To connect (21) with the loss function variation, following [26], we split the interval T into equal size m batches with each of size Δ_T except the last batch given by $\mathcal{T}_j = \{t; (j-1)\Delta_T + 1 \leq t \leq \min\{j\Delta_T, T\}\}$ for $j = 1, \dots, m$ where $m = \lceil \frac{T}{\Delta_T} \rceil$. We can rewrite the expression in (21) as follows

$$\begin{aligned} \mathbf{Reg}_T^D &= \sum_{s=1}^{\lceil \frac{T}{\Delta_T} \rceil} \sum_{t \in \mathcal{T}_s} [L_t(f_t(\mathbf{S}_t)) - L_t(f_s^*(\mathbf{x}_t))] \\ &\quad + \sum_{s=1}^{\lceil \frac{T}{\Delta_T} \rceil} \sum_{t \in \mathcal{T}_s} [L_t(f_s^*(\mathbf{x}_t)) - L_t(f_t^*(\mathbf{S}_t))] \end{aligned} \quad (22)$$

where we define $f_s^* = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{t \in \mathcal{T}_s} L_t(f(\mathbf{S}_t))$ for all $s = 1, 2, \dots, m$, and note that the outer sum over s indexes the batch number, whereas inner one indexes elements of a particular batch \mathcal{T}_s . The expression for the dynamic regret in (22) is decomposed into two sums. Note that the first sum represents the sum of the regrets against a single batch action for each batch \mathcal{T}_s . The second term in (22) quantifies the non-stationarity of the optimizer: it is a sum over differences between the best action over batch s and corresponding dynamic optimal actions. Next, we bound the each term on the right hand side of (22) separately. From the static regret in (111), it holds that

$$\sum_{t \in \mathcal{T}_s} [L_t(f_t(\mathbf{S}_t)) - L_t(f_s^*(\mathbf{x}_t))] = \mathcal{O} \left(\frac{1 + (\epsilon + \epsilon^2)\Delta_T}{\eta} + \eta\Delta_T \right) \quad (23)$$

for all $s = 1, 2, \dots, m$. To upper bound the term in (22) associated with non-stationarity, i.e., the second term on the right-hand side, by definition of the minimum, we have

$$\sum_{t \in \mathcal{T}_s} [L_t(f_s^*(\mathbf{x}_t)) - L_t(f_t^*(\mathbf{S}_t))] \leq \sum_{t \in \mathcal{T}_s} [L_t(f_k^*(\mathbf{x}_t)) - L_t(f_t^*(\mathbf{S}_t))] \quad (24)$$

where k denotes the first epoch of batch \mathcal{T}_s and the inequality in (24) holds from the optimality of $f^*(\mathbf{S}_t)$. Further taking maximum over batch, we obtain the upper bound for (24) as

$$\Delta_T \max_{t \in \mathcal{T}_s} [L_t(f_k^*(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))]. \quad (25)$$

Next, we need to upper bound the right hand side of (25) in terms of the loss function variation budget V_T . To do that, let us first define the loss function variation over each batch \mathcal{T}_s as follows

$$V_s := \sum_{t \in \mathcal{T}_s} |L_t - L_{t-1}| \quad (26)$$

and note that $V_T = \sum_{s=1}^m V_s$. With this definition, we now show that

$$\max_{t \in \mathcal{T}_s} [L_t(f_k^*(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \leq 2V_s \quad (27)$$

by contradiction. Let us assume that the inequality in (27) is not true which means that there is at least one epoch, say $m \in \mathcal{T}_s$, for which the following property is valid:

$$L_m(f_k^*(\mathbf{S}_m)) - L_m(f_m^*(\mathbf{S}_m)) > 2V_s. \quad (28)$$

Since V_j is the maximal variation for batch \mathcal{T}_s , it holds that

$$L_t(f_m^*(\mathbf{S}_t)) \leq L_m(f_m^*(\mathbf{S}_m)) + V_s. \quad (29)$$

Substituting the upper bound for $L_m(f_m^*(\mathbf{S}_m))$ from (28) into (29), we get

$$\begin{aligned} L_t(f_m^*(\mathbf{S}_t)) &< L_m(f_k^*(\mathbf{S}_m)) - V_s \\ &\leq L_m(f_k^*(\mathbf{S}_m)). \end{aligned} \quad (30)$$

for all $t \in \mathcal{T}_s$. The second inequality in (30) holds by dropping the negative terms. We note that the inequality in (30) is a contradiction for $t = m$, since a positive number cannot be less than itself. Therefore, the hypothesis in (28) is invalid, which implies that (27) holds true. Next, we utilize the upper bound in (27) to the right hand side of (25), we get

$$\sum_{t \in \mathcal{T}_s} [L_t(f^*(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \leq 2\Delta_T V_s. \quad (31)$$

Now, we return to the aggregation of static regret and the drift of the costs over time in (21), applying (23) and (31) into (22) to obtain final expression for the dynamic regret as

$$\mathbf{Reg}_T^D \leq \lceil \frac{T}{\Delta_T} \rceil \mathcal{O} \left(\frac{1 + (\epsilon + \epsilon^2)\Delta_T}{\eta} + \eta\Delta_T \right) + 2\Delta_T V_T. \quad (32)$$

Suppose we make the parameter selections

$$\eta = \mathcal{O}(\Delta_T^{-a}) \quad \text{and} \quad \epsilon = \mathcal{O}(\Delta_T^{-b}) \quad (33)$$

with $\Delta_T < \mathcal{O}(T)$. Then the right-hand side of (34) takes the form

$$\begin{aligned} \mathbf{Reg}_T^D &\leq \lceil \frac{T}{\Delta_T} \rceil \mathcal{O} \left(\Delta_T^a + (\Delta_T^{-b} + \Delta_T^{-2b})\Delta_T^{(1+a)} + \Delta_T^{1-a} \right) \\ &\quad + 2\Delta_T V_T. \end{aligned} \quad (34)$$

with model order $M = \mathcal{O}(\Delta_T^{p(a-b)})$ by substituting (33) into the result of Lemma 1. For the dynamic regret to be sublinear, we need $b \in (0, 1)$ and $a \in (b, b + \frac{1}{p})$. As long as the dimension p is not too large, we always have a range for a . This implies that $p(a-b) \in (0, 1)$ and hence M is sublinear. One specification of that satisfies this range is $a = 1/2$ and $b = (p-1)/2p$, as stated in Theorem 1. We obtain the result presented in Table I for the selection $\eta = \mathcal{O}(1/\sqrt{\Delta_T})$ and $\Delta_T = (T/V_T)^{2/3}$. \square

The batch parameter Δ_T tunes static versus non-stationary performance: for large Δ_T , then the algorithm attains smaller regret with respect to the static oracle, i.e., the first terms on the right-hand side of (19), but worse in terms of the non-stationarity as quantified by function variation V_T , the last term. On the other hand, if the batch size is smaller, we do worse in terms of static regret terms but better in terms of non-stationarity. This contrasts with the parametric setting as well [26]: the $\mathcal{O}(\epsilon)$ term appears due to the compression-induced error.

Up to now, we quantified algorithm performance by loss variation (3); however, this is only a surrogate for the performance of the sequence of *time-varying* optimizers (4), which is fundamental in time-varying optimization [32], [50], and may be traced to functions of bounded variation in real analysis [49]. Thus, we shift focus to analyzing Algorithm 1 in terms of this fundamental performance metric.

First, we note that the path length (4) is unique when losses are strongly convex. On the other hand, when costs are non-strongly convex, then (4) defines a set of optimizers. Thus, these cases must be treated separately. First, we introduce an assumption used in the second part of the upcoming theorem.

Assumption 5. *The instantaneous loss $L_t : \mathcal{H} \times \mathcal{X} \rightarrow \mathbb{R}$ is strongly convex with parameter μ :*

$$L_t(f) - L_t(\tilde{f}) \geq \mu \|f - \tilde{f}\|_{\mathcal{H}}^2 \quad (35)$$

for all t and any functions $f, \tilde{f} \in \mathcal{H}$.

With the technical setting clarified, we may now present the main theorem regarding dynamic regret in terms of path length (4).

Theorem 2. *Denote $\{f_t\}$ as the function sequence generated by Algorithm 1 run for T iterations. Under Assumptions 1-4, with regularization $\lambda > 0$ the following dynamic regret bounds hold in terms of path length (4) and compression budget ϵ :*

(i) *when costs ℓ_t are convex, regret is sublinear with $\eta < \min\{\frac{1}{\lambda}, \frac{1}{L}\}$ and for any $\epsilon = \mathcal{O}(T^{-\alpha})$ with $\alpha \in (0, \frac{1}{p}]$, we have*

$$\begin{aligned} \mathbf{Reg}_T^D &= \mathcal{O} \left(\frac{1 + T\sqrt{\epsilon} + W_T}{\eta} \right) \\ &= \mathcal{O} \left(1 + T\sqrt{\epsilon} + W_T \right). \end{aligned} \quad (36)$$

(ii) Alternatively, if the cost functions ℓ_t are strongly convex, i.e., Assumption 5 holds, with $\eta < \min\{\frac{1}{\lambda}, \frac{\mu}{L^2}\}$ and for any $\epsilon \in \mathcal{O}(T^{-\alpha})$ with $\alpha \in (0, \frac{1}{p}]$, we have

$$\begin{aligned} \mathbf{Reg}_T^D &= \mathcal{O}\left(\frac{1 + T\sqrt{\epsilon} + W_T}{1 - \rho}\right) \\ &= o(1 + T\sqrt{\epsilon} + W_T), \end{aligned} \quad (37)$$

where $\rho := \sqrt{(1 - 2\eta(\mu - \eta L^2))} \in (0, 1)$ is a contraction constant for a given η .

Proof of Theorem 2(i) Begin by noting that the descent relation in Lemma 3 also holds for time-varying optimizers f_t^* , which allows us to write

$$\begin{aligned} \|f_{t+1} - f_t^*\|_{\mathcal{H}}^2 &\leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \\ &\quad + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (38)$$

From the inequality in (103), we have

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \leq \frac{2\epsilon^2}{\eta^2} + 2\|\nabla_f L_t(f_t(\mathbf{S}_t))\|^2. \quad (39)$$

For a Lipschitz continuous gradient function [Assumption 4] with $\nabla_f L_t(f_t^*(\mathbf{S}_t)) = 0$, we have

$$\|\nabla_f L_t(f_t(\mathbf{S}_t))\|^2 \leq 2L[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))], \quad (40)$$

which implies that

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \leq \frac{2\epsilon^2}{\eta^2} + 2L[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))]. \quad (41)$$

Next, substitute the upper bound in (41) for the last term on the right hand side of (38), we obtain

$$\begin{aligned} \|f_{t+1} - f_t^*\|_{\mathcal{H}}^2 & \quad (42) \\ &\leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} \\ &\quad + 2\epsilon^2 + 2\eta^2 L[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \\ &= \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta(1 - \eta L)[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \\ &\quad + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} + 2\epsilon^2 \\ &\leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta(1 - \eta L)[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \\ &\quad + \frac{4\epsilon CX}{\lambda} + 2\epsilon^2. \end{aligned}$$

The second inequality in (42) is obtained by using the upper bound derived in Proposition 1. To proceed further, we will use the following inequality. For positive scalars u , v , and w that satisfy $2u^2 > v$, by simple manipulation of the quadratic formula over the positive reals, it holds that

$$\begin{aligned} \sqrt{u^2 - v + w^2} &\leq \sqrt{u^2 - v + v^2/4u^2 + w^2} \quad (43) \\ &= \sqrt{u^2 \left(1 - \frac{v}{2u^2}\right)^2 + w^2} \leq u \left(1 - \frac{v}{2u^2}\right) + w \\ &= u - \frac{v}{2u} + w. \end{aligned}$$

The first inequality in (43) holds since we add a positive quantity $\frac{v^2}{4u^2}$ inside the square root. After rearranging the terms, we get the second equality of (43). With the condition $2u^2 > v$ in hand, we used the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any non-negative a and b . Again rearranging the terms, we obtain the final equality in (43).

We can use (43) to upper-estimate the right-hand side of (42) with the following identifications: $u = \|f_t - f_t^*\|_{\mathcal{H}}$, $v = 2\eta(1 - \eta L)[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))]$, and $w = \sqrt{\frac{4\epsilon CX}{\lambda} + 2\epsilon^2}$, such that

$$\begin{aligned} \|f_{t+1} - f_t^*\|_{\mathcal{H}} &\leq \|f_t - f_t^*\|_{\mathcal{H}} - \eta(1 - \eta L) \frac{L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))}{\|f_t - f_t^*\|_{\mathcal{H}}} \\ &\quad + \sqrt{\frac{4\epsilon CX}{\lambda} + 2\epsilon^2}. \end{aligned} \quad (44)$$

The inequality in (44) holds since for a Lipschitz gradient convex loss function (c.f. Assumption 4), we have

$$L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t)) \leq \frac{L}{2}\|f_t - f_t^*\|_{\mathcal{H}}^2. \quad (45)$$

Note to satisfy the condition $u^2 > \frac{v}{2}$, it is sufficient to show that $u^2 > v$ holds. Note that from (45), it holds that

$$u^2 \geq \frac{\nu}{\eta L(1 - \eta L)} \quad (46)$$

from the definitions of ν and u . The required condition of $u^2 > v$ holds if we select $\eta < \frac{1}{L}$. Next, in order to derive the dynamic regret, from triangle's inequality, it holds that

$$\begin{aligned} \|f_{t+1} - f_{t+1}^*\|_{\mathcal{H}} &= \|f_{t+1} - f_t^* + f_t^* - f_{t+1}^*\|_{\mathcal{H}} \\ &\leq \|f_{t+1} - f_t^*\|_{\mathcal{H}} + \|f_{t+1}^* - f_t^*\|_{\mathcal{H}}. \end{aligned} \quad (47)$$

Substitute the upper bound in (44) for the first term on the right hand side of (47), we get

$$\begin{aligned} \|f_{t+1} - f_{t+1}^*\|_{\mathcal{H}} &\leq \|f_t - f_t^*\|_{\mathcal{H}} - \eta(1 - \eta L) \frac{L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))}{\|f_t - f_t^*\|_{\mathcal{H}}} \\ &\quad + \sqrt{\frac{4\epsilon CX}{\lambda} + 2\epsilon^2} + \|f_{t+1}^* - f_t^*\|_{\mathcal{H}}. \end{aligned} \quad (48)$$

Next, rearranging the terms in (48), and utilizing the upper bound in Proposition 1, it holds that $\frac{1}{\|f_t - f_t^*\|_{\mathcal{H}}} > \frac{\lambda}{2CX}$ and we obtain

$$\begin{aligned} &\frac{\eta\lambda(1 - \eta L)}{2CX} L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t)) \\ &\leq \|f_t - f_t^*\|_{\mathcal{H}} - \|f_{t+1} - f_{t+1}^*\|_{\mathcal{H}} \\ &\quad + \sqrt{\frac{4\epsilon CX}{\lambda} + 2\epsilon^2} + \|f_{t+1}^* - f_t^*\|_{\mathcal{H}}. \end{aligned} \quad (49)$$

Take the summation from $t = 1$ to T , we get

$$\begin{aligned} &\frac{\eta\lambda(1 - \eta L)}{2CX} \sum_{t=1}^T [L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \quad (50) \\ &\leq \|f_1 - f_1^*\|_{\mathcal{H}} + T\sqrt{\frac{4\epsilon CX}{\lambda} + 2\epsilon^2} \\ &\quad + \sum_{t=1}^T \|f_{t+1}^* - f_t^*\|_{\mathcal{H}}. \end{aligned}$$

We have dropped the negative terms on right hand side of (50). Next, multiplying both sides by $\frac{2CX}{\eta\lambda(1 - \eta L)}$ and utilizing the definition of path length from (4), we get

$$\begin{aligned} \mathbf{Reg}_T^D &\leq \frac{2CX\|f_1 - f_1^*\|_{\mathcal{H}}}{\eta\lambda(1 - \eta L)} + \frac{2CX}{\eta\lambda(1 - \eta L)} \left(\sqrt{\frac{4T^2\epsilon CX}{\lambda} + 2\epsilon^2 T^2 + W_T} \right) \\ &\leq \mathcal{O}\left(\frac{1 + T\sqrt{\epsilon} + W_T}{\eta}\right) \end{aligned} \quad (51)$$

which is sublinear in T up to factors depending on path length W_T for $\epsilon = \mathcal{O}(T^{-\alpha})$ with $\alpha \in (0, \frac{1}{p}]$ as stated in Theorem 2(i). \square

Proof of Theorem 2(ii) Again, we begin with the descent related stated in Lemma 3 for time-varying optimizer f_t^* :

$$\|f_{t+1} - f_t^*\|_{\mathcal{H}}^2 \leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \quad (52)$$

Consider the last term in (52) as follows

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \quad (53) \\ = \|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t)) + \nabla_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t^*(\mathbf{S}_t))\|_{\mathcal{H}}$$

where we add and subtract the term $\nabla_f L_t(f_t(\mathbf{S}_t))$ and utilize the optimality condition that $\nabla_f L_t(f_t^*(\mathbf{S}_t)) = 0$. Using Cauchy-Schwartz inequality and $(a+b)^2 \leq (2a^2 + 2b^2)$ in (53), we get

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \leq \left(\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} + \|\nabla_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t^*(\mathbf{S}_t))\|_{\mathcal{H}} \right)^2 \\ \leq 2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 + 2\|\nabla_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t^*(\mathbf{S}_t))\|_{\mathcal{H}}^2. \quad (54)$$

Next, utilizing the result of Proposition 2 and Assumption 4 into (54), we obtain

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \leq \frac{2\epsilon^2}{\eta^2} + 2\|\nabla_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t^*(\mathbf{S}_t))\|_{\mathcal{H}}^2 \\ \leq \frac{2\epsilon^2}{\eta^2} + 2L^2\|f_t - f_t^*\|_{\mathcal{H}}^2. \quad (55)$$

The last inequality in (54) holds from Assumption 4. Next, substitute the upper bound in (54) into (52), we obtain

$$\|f_{t+1} - f_t^*\|_{\mathcal{H}}^2 \leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} + 2\eta^2 L^2\|f_t - f_t^*\|_{\mathcal{H}}^2. \quad (56)$$

From the strong convexity of the objective function (Assumption 5), we have (35), which we may substitute in for the second term on right hand side of (56) to obtain

$$\|f_{t+1} - f_t^*\|_{\mathcal{H}}^2 \leq \|f_t - f_t^*\|_{\mathcal{H}}^2 - 2\eta\mu\|f_t - f_t^*\|_{\mathcal{H}}^2 + 2\epsilon\|f_t - f_t^*\|_{\mathcal{H}} + 2\eta^2 L^2\|f_t - f_t^*\|_{\mathcal{H}}^2 \\ \leq (1 - 2\eta\mu + 2\eta^2 L^2)\|f_t - f_t^*\|_{\mathcal{H}}^2 + \epsilon\frac{8CX}{\lambda} + 2\epsilon^2 \quad (57)$$

where for the second inequality we have used the statement of Proposition (1) for the third term on the right-hand side of the first inequality. Take square root on both sides of (57), we get

$$\|f_{t+1} - f_t^*\|_{\mathcal{H}} \leq \rho\|f_t - f_t^*\|_{\mathcal{H}} + \sqrt{\epsilon\frac{8CX}{\lambda} + 2\epsilon^2}, \quad (58)$$

where $\rho := \sqrt{(1 - 2\eta(\mu - \eta L^2))}$. The value of $\rho \in (0, 1)$ defines a contraction mapping provided η satisfies $0 < \eta < \frac{\mu}{L^2}$. With the help of triangle inequality, we can write the difference $\|f_{t+1} - f_{t+1}^*\|_{\mathcal{H}}$ as

$$\|f_{t+1} - f_{t+1}^*\|_{\mathcal{H}} \leq \|f_{t+1} - f_t^*\|_{\mathcal{H}} + \|f_{t+1}^* - f_t^*\|_{\mathcal{H}}. \quad (59)$$

Utilize the upper bound in (58) into (59), and taking the summation over t on the both sides, we get

$$\sum_{t=1}^T \|f_t - f_t^*\|_{\mathcal{H}} \leq \|f_1 - f_1^*\|_{\mathcal{H}} + \rho \sum_{t=1}^T \|f_t - f_t^*\|_{\mathcal{H}} + \sqrt{\epsilon T^2 \frac{8CX}{\lambda} + 2\epsilon^2 T^2} + \sum_{t=1}^T \|f_t^* - f_{t-1}^*\|_{\mathcal{H}}. \quad (60)$$

After rearranging and dividing the both sides by $1 - \rho$, we get

$$\sum_{t=1}^T \|f_t - f_t^*\|_{\mathcal{H}} \leq \frac{\|f_1 - f_1^*\|_{\mathcal{H}}}{1 - \rho} + \frac{1}{1 - \rho} \left(\sqrt{\epsilon T^2 \frac{8CX}{\lambda} + 2\epsilon^2 T^2} + W_T \right) \\ \leq \mathcal{O} \left(\frac{1 + T\sqrt{\epsilon} + W_T}{1 - \rho} \right) \quad (61)$$

where we have used the definition of path length W_T (4) on the right-hand side of (60). From the first order convexity condition, we can write

$$\sum_{t=1}^T [L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \leq \sum_{t=1}^T \langle \nabla_f L_t(f_t(\mathbf{S}_t)), f_t - f_t^* \rangle_{\mathcal{H}} \\ \leq \sum_{t=1}^T \|\nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \|f_t - f_t^*\|_{\mathcal{H}} \quad (62)$$

where the second inequality in (62) holds due to Cauchy-Schwartz inequality. Next, since the space \mathcal{X} is compact, the gradient norm $\|\nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}$ evaluated for any \mathbf{S}_t will be upper bounded by some finite constant G , which implies that $\|\nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \leq G$. Using the gradient upper bound on the right hand side of (62), we obtain

$$\sum_{t=1}^T [L_t(f_t(\mathbf{S}_t)) - L_t(f_t^*(\mathbf{S}_t))] \leq G \sum_{t=1}^T \|f_t - f_t^*\|_{\mathcal{H}}. \quad (63)$$

Next, utilizing the upper bound in (61) into the right hand side of (63), we obtain the final regret result as

$$\mathbf{Reg}_T^D = \mathcal{O} \left(\frac{1 + T\sqrt{\epsilon} + W_T}{1 - \rho} \right). \quad (64)$$

Observe that (64) is sublinear in T up to terms depending on the path length for any step-size η and for compression constant $\epsilon = \mathcal{O}(T^{-\alpha})$ with $\alpha \in (0, \infty]$. The expression in (61) is similar to the one on (36) except for the term $(1 - \rho)$ in the denominator. If we choose η such that $(1 - \rho) > \eta$, the results for strongly convex functions is improved. Rearrange this expression to obtain

$$(1 - \eta)^2 > \rho^2 = 1 - 2\eta(\mu - \eta L^2)$$

which, upon solving for a condition on η , simplifies to

$$\eta < \frac{2(\mu - 1)}{2L^2 - 1}. \quad \square$$

Theorem 2 generalizes existing dynamic regret bounds of [20], [25], [26], [28] to the case where decisions are defined by functions f_t belonging to RKHS \mathcal{H} . To facilitate this generalization, gradient projections are employed to control function complexity, which appears as an additional term depending on compression budget ϵ in the dynamic regret

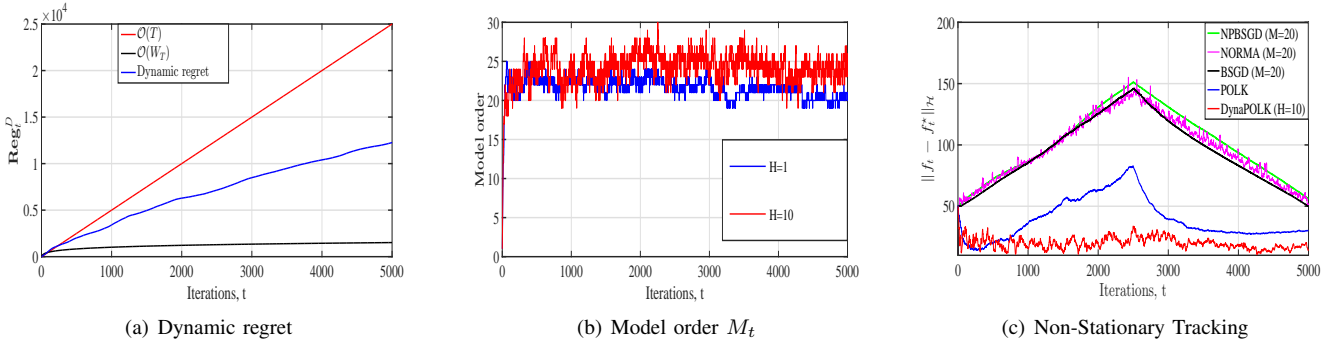


Fig. 1: Experiments with non-stationary nonlinear regression common to phase retrieval: scalar targets are $y_t = a_t \sin(b_t \mathbf{x}_t + c_t) + \eta_t$, which one would like to predict via sequentially observed \mathbf{x}_t , where η_t is additive Gaussian noise. DynaPOLK attains sublinear regret, and is able to track a shifting nonlinearity with low model complexity. In contrast, alternatives are unable to adapt to drift.

bounds, in particular, the product $T\sqrt{\epsilon}$ in the expressions (36) and (37). For smaller ϵ , the regret is smaller, but the model complexity increases, and vice versa. Overall, this compression induced error in the gradient is a version of inexact functional gradient descent algorithm with a tunable tradeoff between convergence accuracy and memory. Note that for $\epsilon = 0$, these results becomes of the order of $\mathcal{O}(1 + W_T)$ which matches [28] and improves upon existing results [20], [25], [26]. Even for the strongly convex case with $\epsilon = 0$, we obtain $o(1 + W_T)$ which is better than its parametric counterpart obtained in [28].

Regarding the complexity reduction technique for kernel methods, we note that dynamic regret bounds for random feature approximations in the looser sense of (34) have been recently established [30]. These results hinge upon tuning the random feature incurred error to gradient bias. However, in practice, the number of random features required to ensure a specific directional bias is unknown, which experimentally dictates one using a large enough number of random features to hope the bias is small. However, this error is in the *function representation* itself, not the gradient direction. This issue could be mitigated through double kernel sampling [47], a technique whose use in non-stationary settings remains a direction for future research.

Parameter Selection For step-size $\eta < \min\{\frac{1}{\lambda}, \frac{1}{L}\}$ and compression budget $\epsilon = \mathcal{O}(T^{-\alpha})$, substituted into Lemma 1 yields model complexity $M = \mathcal{O}(T^{\alpha p})$. To obtain sublinear regret (up to factors depending on W_T) and model complexity in the non-strongly convex case, we require $\alpha \in (0, \frac{1}{p}]$ and $\alpha p \in (0, 1)$, which holds, for instance, if $\epsilon = T^{-1/(p+1)}$. Note that the dynamic regret result in (36) and the model order, using Lemma 1, becomes

$$\mathbf{Reg}_T^D = \mathcal{O}\left(1 + T^{(1-\frac{\alpha}{2})} + W_T\right), \quad M = \mathcal{O}(T^{\alpha p}). \quad (65)$$

For the regret to be sublinear, we need $\alpha \in (0, \frac{1}{p}]$. As long as the dimension p is not too large, we always have a range for α . This implies that $\alpha p \in (0, 1)$ and hence M is sublinear.

Observe that the rate for the strongly convex case (37) is strictly better the non-strongly convex counterpart (37) whenever η satisfies $(1 - \rho) > \eta$. This holds, provided

α	Regret	M	Comments
$\alpha = 0$	$\mathcal{O}(T) + W_T$	$\mathcal{O}(1)$	Linear regret
$\alpha = \frac{1}{p}$	$\mathcal{O}\left(T^{\frac{(2p-1)}{2p}} + W_T\right)$	$\mathcal{O}(T)$	Linear M
$\alpha = \frac{1}{p+1}$	$\mathcal{O}\left(T^{\frac{2p+1}{2p+2}} + W_T\right)$	$\mathcal{O}(T^{p/(1+p)})$	Sublinear M

TABLE II: Summary of dynamic regret rates for convex loss function. Note that the same rates are obtained for the strongly convex loss function but \mathcal{O} is replaced by small o .

$\eta < (2(\mu - 1))/(2L^2 - 1)$. Taken together, Theorems 1 - 2 establish that Algorithm 1 is effective for non-stationary learning problems. In the next section, we experimentally benchmark these results on representative tasks.

V. EXPERIMENTS

In this section, we evaluate the ability of Algorithm 1 to address online regression and classification in non-stationary regimes and compare it with some alternatives.

Online Regression We first consider a simple online regression to illustrate performance: target variables are of the form $y_t = a_t \sin(b_t \mathbf{x}_t + c_t) + \eta_t$, which one would like to predict upon the basis of sequentially observed values of \mathbf{x}_t . Here $\eta \sim \mathcal{N}(\mu_t, \sigma^2)$ is Gaussian noise. Such models arise in phase retrieval, as in medical imaging, acoustics, or communications. Non-stationarity comes from parameters (a_t, b_t, c_t) changing with t : a_t and c_t increase from 0 to 3 and then decrease to 1, both linearly, while b_t is increased from 0 to 1 linearly. We consider a square loss function given by $\ell_t(f(\mathbf{x})) = (f(\mathbf{x}) - y_t)^2$ and run the simulations for $T = 5000$ iterations. For experiments, we select Gaussian kernels of bandwidth $\sigma = 0.252$, step-size $\eta = T^{-0.4}$, and compression parameter $\epsilon = T^{-0.1}$. The dynamic regret for $H = 1$ is shown in Fig. 1(a) – observe that it grows sublinearly with time. Path length W_T is shown for reference. Fig. 1(b) shows the model order relative to time for window lengths $H = 1$ and $H = 10$, which remains moderate. Observe that Algorithm 1 is able to track shifting data more gracefully with larger H as clear from Fig. 3(a). This figure shows the true

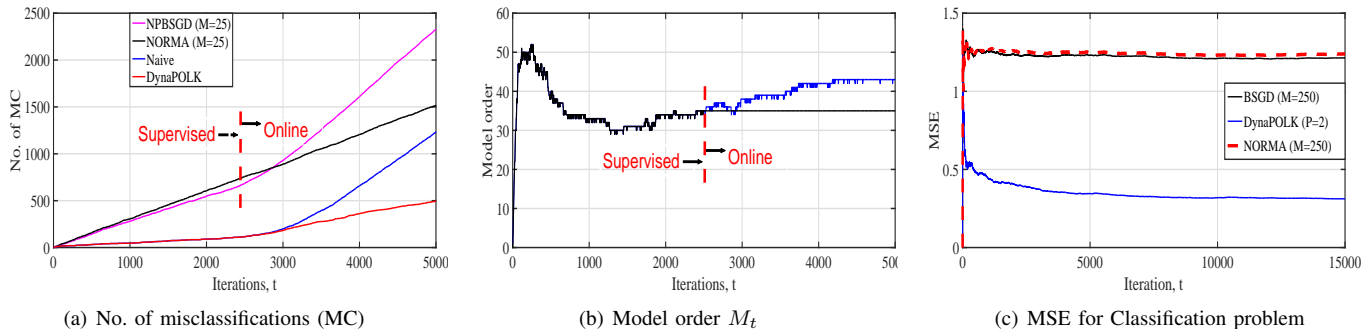


Fig. 2: Comparison of DynaPOLK to other kernel methods (left) for an online non-stationary classification on Gaussian Mixtures data [51] with dynamic class means. Alternative methods experience nearly linear regret, and their mean-square error on the time-series classification problem defined in [52] is relatively uncontrolled (right).

Algorithms/Dataset	Twitter	Tom	Energy	Air
AdaRaker	2.6	1.9	13.8	1.3
DyanPOLK	0.06	0.68	0.0052	0.14
Model order (DyanPOLK)	50	24	31	33

TABLE III: MSE (10^{-3}) performance of the different algorithms with $B = D = 50$ (as in [30]).

function at the first and last time, i.e., $f_1(x)$ at iteration 1 to $f_T(x)$ at iteration T . The red curve shows the learned function via DynaPOLK, which better adheres to the target for $H = 10$. An animation of online nonlinear regression in the presence of non-stationarity is appended to this submission, and the supplementary regression video. We further compare DynaPOLK against the alternative methods, namely, NPBSGD [53], NORMA [38], BSGD [54], and POLK [40]. We plot the distance from the optimal $\|f_t - f_t^*\|_{\mathcal{H}}$ in Fig. 1(c). Fig. 1(c) we observe DynaPOLK with $H = 10$ is able to track the time-varying nonlinearity, whereas the others experience nearly linear regret during the non-stationary phase. We remark that a recent algorithm AdaRaker is proposed in [30] to solve the nonparametric online learning problems. The authors in [30] shows that AdaRaker performs better than all the other available techniques in the literature. Hence, in this work, we compare the proposed DynaPOLK algorithm mainly with the algorithms of [30] and show the improvement as provided in Table III (see [30] for the datasets description).

Online Classification Consider the multi-class classification in non-stationary environments, a salient problem in terrain adaption of autonomous systems [55]. Motivated by this setting, we experiment on multi-class problems with label drift. Specifically, data is stationary for the first 2500 iterations during which it reduces to standard supervised learning. After the first 2500 iterations, the data drifts and we require learning the classifier online. We fix the loss as the multi-class hinge (SVM) loss $\ell_t(f(\mathbf{x}))$ as in [56], and generate Gaussian Mixtures data akin to [51]. The synthetic Gaussian Mixtures dataset for classification is generated in a manner similar to [51]. It consists of $N = 5000$ feature-label pairs out of which last 2500 are generated with the drift. For the first 2500 points, we generate $\mathbf{x}_n \in \mathbb{R}^p$ as $\mathbf{x} | y \sim (1/3) \sum_{j=1}^3 \mathcal{N}(\boldsymbol{\mu}_{y,j}, \sigma_{y,j}^2 \mathbf{I})$ where $\sigma_{y,j}^2 = 0.2$ for all values of y and j , where also depends

upon the class as $\boldsymbol{\mu}_{y,j} \sim \mathcal{N}(\boldsymbol{\theta}_y, \sigma_y^2 \mathbf{I})$. The class mean value $\{\boldsymbol{\theta}_i\}_{i=1}^C$ is placed around unit circle. We fix $\sigma_y^2 = 1.0$ and $C = 5$. To add drift, after first 2500 points, we shift each point to the right by 0.1 at each instant which is clear from the video attached with the submission. Moreover, we focus on SVM for ease of interpretation. Its definition the multi-class context is given as

$$\ell_t(f, \mathbf{x}_t, y_t) = \max(0, 1 + f_r(\mathbf{x}_t) - f_{y_t}(\mathbf{x}_t)) + \lambda \sum_{c'=1}^C \|f_{c'}\|_{\mathcal{H}}^2,$$

where $r = \arg \max_{c' \neq y_t} f_{c'}(\mathbf{x})$. This definition is taken exactly from [56]. With dynamic class means during the drift phase: each mean shifts rightward by 0.1 per step. The results are presented in Fig.2: misclassifications over time is shown in Fig.2(a). DynaPOLK yields fewer mistakes in the non-stationary regime. Model complexity (Fig.2(b)) increases when the data is non-stationary, suggesting that it may be effective for change point detection. Fig. 3(b) displays the learned decision surface on stationary data, and Fig. 3(c) shows evolution to rightward-drifted data. Black dots denote dictionary elements and black lines are decision boundaries – the supplementary classification video visualizes the classifier evolution. As all the class means shift rightward, DynaPOLK is able to stably and accurately adapt its model.

Further, we did an additional experiments on time-series classification [52]. This dataset consists of 60000 examples with 3 features and 3 classes. Features take values between 0 and 10, and the data is broken up into four blocks, where values of the features shift across the different blocks. See [52][Table 1] for more specific details. We report the results of comparing DynaPOLK to the alternatives mentioned in Sec. V in Figure 2(c). Specifically, we display the mean-square error, i.e., for each time, we compute misclassification square error and average it to the previous one. Note that DynaPOLK attains favorable performance.

VI. CONCLUSION

In this work, we focused on non-stationary learning, for which we proposed an online universal function approximator based on compressed kernel methods. We characterized its

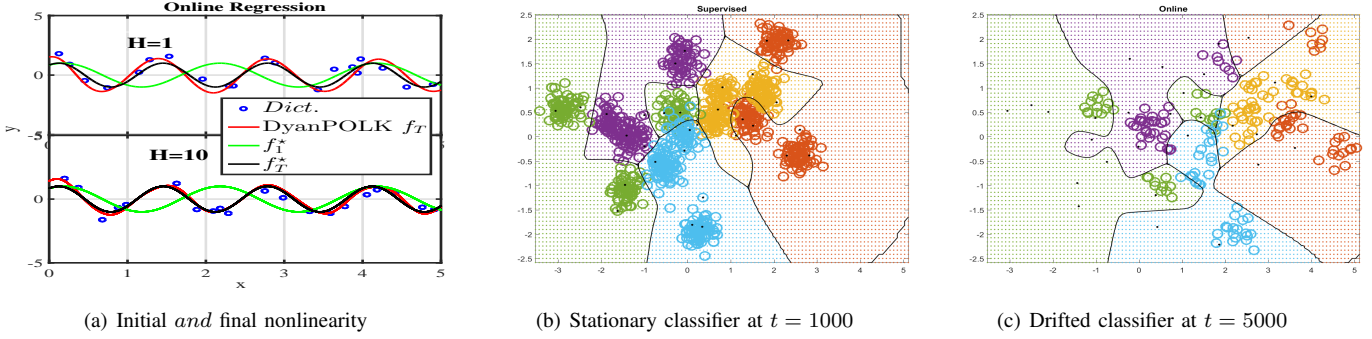


Fig. 3: Left: regression with initial & final target denoted as f_1^* & f_T^* . DynaPOLK tracks nonlinearity drifting with (a_t, b_t, c_t) . Windowing ($H = 10$) improves performance. Center: decision surface of DynaPOLK on stationary Gaussian Mixtures [51]. Right: classifier adapting to data drift.

dynamic regret as well as its model efficiency, and experimentally observed it yields a favorable tradeoffs for learning in the presence of non-stationarity. Future questions involve the development of model order as use for change point detection, improving the learning rates through second-derivative information, variance reduction, or strong convexity, and coupling it to the design of learning control systems.

APPENDIX A PROOF OF LEMMA 1

Before proving Lemma 1, we present a lemma which allows us to relate the stopping criterion of our sparsification procedure to a Hilbert subspace distance.

Lemma 2. Define the distance of an arbitrary feature vector \mathbf{x} evaluated by the feature transformation $\phi(\mathbf{x}) = \kappa(\mathbf{x}, \cdot)$ to $\mathcal{H}_{\mathbf{D}} = \text{span}\{\kappa(\mathbf{d}_t, \cdot)\}_{t=1}^M$, the subspace of the Hilbert space spanned by a dictionary \mathbf{D} of size M , as

$$\text{dist}(\kappa(\mathbf{x}, \cdot), \mathcal{H}_{\mathbf{D}}) = \min_{f \in \mathcal{H}_{\mathbf{D}}} \|\kappa(\mathbf{x}, \cdot) - \mathbf{v}^T \kappa_{\mathbf{D}}(\cdot)\|_{\mathcal{H}}. \quad (66)$$

This set distance simplifies to following least-squares projection when $\mathbf{D} \in \mathbb{R}^{p \times M}$ is fixed

$$\text{dist}(\kappa(\mathbf{x}, \cdot), \mathcal{H}_{\mathbf{D}}) = \left\| \kappa(\mathbf{x}, \cdot) - [\mathbf{K}_{\mathbf{D}, \mathbf{D}}^{-1} \kappa_{\mathbf{D}}(\mathbf{x})]^T \kappa_{\mathbf{D}}(\cdot) \right\|_{\mathcal{H}}. \quad (67)$$

Proof. The distance to the subspace $\mathcal{H}_{\mathbf{D}}$ is defined as

$$\begin{aligned} \text{dist}(\kappa(\mathbf{x}, \cdot), \mathcal{H}_{\mathbf{D}_t}) &= \min_{f \in \mathcal{H}_{\mathbf{D}}} \|\kappa(\mathbf{x}, \cdot) - \mathbf{v}^T \kappa_{\mathbf{D}}(\cdot)\|_{\mathcal{H}} \\ &= \min_{\mathbf{v} \in \mathbb{R}^M} \|\kappa(\mathbf{x}, \cdot) - \mathbf{v}^T \kappa_{\mathbf{D}}(\cdot)\|_{\mathcal{H}}, \end{aligned} \quad (68)$$

where the first equality comes from the fact that the dictionary \mathbf{D} is fixed, so $\mathbf{v} \in \mathbb{R}^M$ is the only free parameter. Now plug in the minimizing weight vector $\tilde{\mathbf{v}}^* = \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \kappa_{\mathbf{D}_t}(\mathbf{x}_t)$ into (68) which is obtained in an analogous manner to the logic which yields (13). Doing so simplifies (68) to the following

$$\text{dist}(\kappa(\mathbf{x}_t, \cdot), \mathcal{H}_{\mathbf{D}_t}) = \left\| \kappa(\mathbf{x}_t, \cdot) - [\mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \kappa_{\mathbf{D}_t}(\mathbf{x}_t)]^T \kappa_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}}. \quad (69)$$

□ The minimal error is achieved by considering the square of

A. Proof of Lemma 1

The proof is similar to that of [40, Theorem 3] and provided here in detail for completeness. Consider the model order of the function iterates f_t and f_{t+1} generated by Algorithm 1 denoted by M_t and M_{t+1} , respectively, at two arbitrary subsequent times t and $t+1$. The number of elements in \mathbf{D}_t are $M_t = (t-1)$. After performing the algorithm update at t , we add a new data points to the dictionary and increase the model order by one, hence $M_{t+1} = M_t + 1$.

Begin by assuming function f_{t+1} is parameterized by dictionary \mathbf{D}_{t+1} and weight vector \mathbf{w}_{t+1} . Moreover, we denote columns of \mathbf{D}_{t+1} as \mathbf{d}_t for $t = 1, \dots, M_{t+1}$. Suppose the model order of the function f_{t+1} is less than or equal to that of f_t , i.e. $M_{t+1} \leq M_t$. This relation holds when the stopping criterion of KOMP, stated as $\min_{\{j=1, \dots, M_{t+1}\}} \gamma_j > \epsilon$, is not satisfied for the kernel dictionary matrix with the newest data point \mathbf{x}_t appended: $\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t; \mathbf{x}_t]$ [cf. (11)], which is of size $M_t + 1$. Thus, the negation of the termination condition of KOMP holds for this case, stated as

$$\min_{\{j=1, \dots, M_{t+1}\}} \gamma_j \leq \epsilon. \quad (70)$$

Observe that the left-hand side of (70) lower bounds the approximation error $\gamma_{M_{t+1}}$ of removing the recent batch of the feature vectors \mathbf{S}_t due to the minimization over j , that is, $\min_{\{j=1, \dots, M_{t+1}\}} \gamma_j \leq \gamma_{M_{t+1}}$. Consequently, if $\gamma_{M_{t+1}} \leq \epsilon$, then (70) holds and the model order does not grow. Thus it suffices to consider $\gamma_{M_{t+1}}$.

The definition of $\gamma_{M_{t+1}}$ with the substitution of \tilde{f}_{t+1} defined by (11) allows us to write

$$\begin{aligned} \gamma_{M_{t+1}} &= \min_{\mathbf{u} \in \mathbb{R}^{M_t}} \left\| (1-\eta\lambda)f_t - \eta \nabla_f \tilde{L}_t(f_t(\mathbf{S}_t)) - \sum_{k \in \mathcal{I} \setminus \{M_{t+1}\}} u_k \kappa(\mathbf{d}_k, \cdot) \right\|_{\mathcal{H}} \\ &= \min_{\mathbf{u} \in \mathbb{R}^{M_t}} \left\| (1-\eta\lambda) \sum_{k \in \mathcal{I} \setminus \{M_{t+1}\}} w_k \kappa(\mathbf{d}_k, \cdot) - \eta \nabla_f \tilde{L}_t(f_t(\mathbf{S}_t)) \right. \\ &\quad \left. - \sum_{k \in \mathcal{I} \setminus \{M_{t+1}\}} u_k \kappa(\mathbf{d}_k, \cdot) \right\|_{\mathcal{H}}. \end{aligned} \quad (71)$$

the expression inside the minimization and expand to get

$$\begin{aligned} & \left\| (1-\eta\lambda) \sum_{k \in \mathcal{I} \setminus \{M_t+1\}} w_k \kappa(\mathbf{d}_k, \cdot) - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) - \sum_{k \in \mathcal{I} \setminus \{M_t+1\}} u_k \kappa(\mathbf{d}_k, \cdot) \right\|_{\mathcal{H}}^2 \\ &= (1-\eta\lambda)^2 \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t} \mathbf{w} + \eta^2 (\nabla_f \check{L}_t(f_t(\mathbf{S}_t)))^2 + \mathbf{u}^T \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t} \mathbf{u} \\ & \quad - 2(1-\eta\lambda) \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau) \\ & \quad + 2\eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \mathbf{u}^T \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau) - 2(1-\eta\lambda) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t} \mathbf{u}. \end{aligned} \quad (72)$$

To obtain the minimum, we compute the stationary solution of (72) with respect to $\mathbf{u} \in \mathbb{R}^{M_t}$ and solve for the minimizing $\tilde{\mathbf{u}}^*$, which in a manner similar to the logic in (13), is given as

$$\tilde{\mathbf{u}}^* = (1-\eta\lambda) \mathbf{w} - \eta \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau). \quad (73)$$

Plug $\tilde{\mathbf{u}}^*$ in (73) into the expression in (71) and using the short-hand notation $f_t(\cdot) = \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\cdot)$ and $\sum_k u_k \kappa(\mathbf{d}_k, \cdot) = \mathbf{u}^T \mathbf{K}_{\mathbf{D}_t}(\cdot)$. Doing so simplifies (71) to

$$\begin{aligned} & \left\| (1-\eta\lambda) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\cdot) - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) - \mathbf{u}^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}} \quad (74) \\ &= \left\| (1-\eta B \lambda) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\cdot) - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \kappa(\mathbf{x}_\tau, \cdot) \right. \\ & \quad \left. - \left[(1-\eta\lambda) \mathbf{w} - \eta \mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau) \right]^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}}. \end{aligned}$$

The above expression may be simplified by canceling like terms $(1-\eta\lambda) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\cdot)$ and collecting the like terms, we get

$$\begin{aligned} & \left\| (1-\eta\lambda) \mathbf{w}^T \mathbf{K}_{\mathbf{D}_t}(\cdot) - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) - \mathbf{u}^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}} \quad (75) \\ &= \eta \left\| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \left[\kappa(\mathbf{x}_\tau, \cdot) - \eta [\mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau)]^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right] \right\|_{\mathcal{H}} \\ &\leq \eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right| \cdot \left\| \kappa(\mathbf{x}_\tau, \cdot) - \eta [\mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau)]^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}}. \end{aligned}$$

The second inequality in (75) is achieved by the use of triangle and Cauchy Schwartz inequality. Notice that the right-hand side of (75) may be identified as the distance to the subspace $\mathcal{H}_{\mathbf{D}_t}$ in (69) defined in Lemma 2 scaled by at most a factor of P times $\eta |\check{\ell}'_\tau(f_\tau(\mathbf{x}_\tau))|$. We may write the right hand side of (75) as

$$\begin{aligned} & \eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right| \cdot \left\| \kappa(\mathbf{x}_\tau, \cdot) - \eta [\mathbf{K}_{\mathbf{D}_t, \mathbf{D}_t}^{-1} \mathbf{K}_{\mathbf{D}_t}(\mathbf{x}_\tau)]^T \mathbf{K}_{\mathbf{D}_t}(\cdot) \right\|_{\mathcal{H}} \\ &= \eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right| \text{dist}(\kappa(\mathbf{x}_\tau, \cdot), \mathcal{H}_{\mathbf{D}_t}) \end{aligned} \quad (76)$$

where we have applied (67) regarding the definition of the subspace distance on the right-hand side of (76) to replace the Hilbert-norm term. Now, when the KOMP stopping criterion is violated, i.e., (70) holds, which implies $\gamma_{M_t+1} \leq \epsilon$. Therefore, the right-hand side of (76) is upper-bounded by ϵ , we can write

$$\eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right| \text{dist}(\kappa(\mathbf{x}_t, \cdot), \mathcal{H}_{\mathbf{D}_t}) \leq \epsilon. \quad (77)$$

After rearranging the terms in (77), we write

$$\text{dist}(\kappa(\mathbf{x}_t, \cdot), \mathcal{H}_{\mathbf{D}_t}) \leq \frac{\epsilon}{\eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right|}, \quad (78)$$

where we have divided both sides by $\eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right|$. Observe that if (78) holds, then $\gamma_{M_t+1} \leq \epsilon$ holds, but since $\gamma_{M_t+1} \geq \min_j \gamma_j$, we may conclude that (70) is satisfied. Consequently the model order at the subsequent step does not grow $M_{t+1} \leq M_t$ whenever (78) is valid.

Now, let's take the contrapositive of the preceding expressions to observe that growth in the model order ($M_{t+1} = M_t + 1$) implies that the condition

$$\text{dist}(\kappa(\mathbf{x}_t, \cdot), \mathcal{H}_{\mathbf{D}_t}) > \frac{\epsilon}{\eta \left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right|} \quad (79)$$

holds. Therefore, each time a new point is added to the model, the corresponding kernel function is guaranteed to be at least a distance of $\frac{\epsilon}{\eta \left| \check{\ell}'_t(f_t(\mathbf{x}_t)) \right|}$ from every other kernel function in the current model.

By the C -Lipschitz continuity of the instantaneous loss (Assumption 2): specifically $1/\left| \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right| \geq 1/HC$, we can lower-bound the threshold condition in (79) as

$$\frac{\epsilon}{\eta \left| \check{\ell}'_t(f_t(\mathbf{x}_t)) \right|} \geq \frac{\epsilon}{\eta CH} \quad (80)$$

We have

$$\text{dist}(\kappa(\mathbf{x}_t, \cdot), \mathcal{H}_{\mathbf{D}_t}) > \frac{\epsilon}{\eta CH} \quad (81)$$

Therefore, For a fixed compression budget ϵ and step size η , the KOMP stopping criterion is violated for the newest point whenever distinct dictionary points \mathbf{d}_k and \mathbf{d}_j for $j, k \in \{1, \dots, M_t\}$, satisfy the condition $\|\phi(\mathbf{d}_j) - \phi(\mathbf{d}_k)\|_{\mathcal{H}} > \frac{\epsilon}{\eta CH}$. Next, we follow the similar argument as provided in the proof of Theorem 3.1 in [57]. Since \mathcal{X} is compact and κ is continuous, the range $\phi(\mathcal{X})$ (where $\phi(\mathbf{x}) = \kappa(\mathbf{x}, \cdot)$ for $\mathbf{x} \in \mathcal{X}$) of the kernel transformation of feature space \mathcal{X} is compact. Therefore, the number minimum of balls (covering number) of radius δ (here, $\delta = \frac{\epsilon}{\eta CH}$) needed to cover the set $\phi(\mathcal{X})$ is finite (see, e.g., [58]) for a fixed compression budget ϵ and step-size η .

To arrive at the characterization (18), we note that [57, Proposition 2.2] states that for a Lipschitz continuous Mercer kernel κ on compact set $\mathcal{X} \subseteq \mathbb{R}^p$, there exists a constant Y such that for any training set $\{\mathbf{x}_t\}_{t=1}^\infty$ and any $\nu > 0$, and it holds for the number of elements in dictionary that

$$M \leq Y \left(\frac{1}{\nu} \right)^p. \quad (82)$$

where Y is a constant depends upon \mathcal{X} and the kernel function. By (81), we have that $\nu = \frac{\epsilon}{\eta CH}$, which we may substitute into (82) to obtain

$$M \leq Y (CH)^p \left(\frac{\eta}{\epsilon} \right)^p. \quad (83)$$

as stated in (18). The lower bound H in (18) comes from the fact that to represent the instantaneous gradient of the windowed loss of length H , a minimum of H points are required. \square

REFERENCES

- [1] A. S. Bedi, A. Koppel, K. Rajawat, and B. M. Sadler, "Nonstationary nonparametric online learning," *IEEE American Control Conference*, 2020.
- [2] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, no. 1, pp. 243–247, 1969.
- [3] D. R. Brillinger, *Time series: data analysis and theory*. Siam, 1981, vol. 36.
- [4] S. Haykin, "Neural networks: A comprehensive foundation," *Macmillan College Publishing Company*, 1994.

- [5] A. Berline and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.
- [6] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.
- [7] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee, and P. C. Wei, "Adaptive tracking of linear time-variant systems by extended rls algorithms," *IEEE Transactions on signal processing*, vol. 45, no. 5, pp. 1118–1128, 1997.
- [8] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*, vol. 5.
- [9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [10] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.
- [11] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 3981–3989.
- [12] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," *arXiv preprint arXiv:1710.11622*, 2017.
- [13] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [14] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.
- [15] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.
- [16] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009, vol. 48.
- [17] M. Mohri and A. Rostamizadeh, "Stability bounds for stationary φ -mixing and β -mixing processes," *J. Mach. Learn. Res.*, vol. 11, no. Feb, pp. 789–814, 2010.
- [18] A. Nagabandi, C. Finn, and S. Levine, "Deep online learning via meta-learning: Continual adaptation for model-based rl," *arXiv preprint arXiv:1812.07671*, 2018.
- [19] L. Wasserman, *All of nonparametric statistics*. Springer Science & Business Media, 2006.
- [20] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th ICML*, vol. 20, no. 2, Washington DC, USA, Aug. 21–24 2003, pp. 928–936.
- [21] R. W. Heath and A. Paulraj, "A simple scheme for transmit diversity using partial channel feedback," in *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284)*, vol. 2. IEEE, 1998, pp. 1073–1078.
- [22] P. Vernaza, B. Taskar, and D. D. Lee, "Online, self-supervised terrain classification via discriminatively trained submodular markov random fields," in *IEEE ICRA*. IEEE, 2008, pp. 2750–2757.
- [23] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite markov decision processes with gaussian processes," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 4312–4320.
- [24] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of IEEE ICRA*, vol. 2, 2010.
- [25] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, 2015.
- [26] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Op. Res.*, vol. 63, no. 5, pp. 1227–1244, 2015.
- [27] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.
- [28] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *IEEE 55th CDC*, 2016, pp. 7195–7201.
- [29] A. S. Bedi, P. Sarma, and K. Rajawat, "Tracking moving agents via inexact online gradient descent algorithm," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 202–217, Feb 2018.
- [30] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 773–808, 2019.
- [31] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.
- [32] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591.
- [33] V. Tikhomirov, "On the representation of continuous functions of several variables as superpositions of continuous functions of one variable and addition," in *Selected Works of AN Kolmogorov*. Springer, 1991, pp. 383–387.
- [34] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural networks*, vol. 11, no. 1, pp. 15–37, 1998.
- [35] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.
- [36] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *Proceedings of the 34th International Conference on Machine Learning—Volume 70*. JMLR. org, 2017, pp. 146–155.
- [37] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Adv. Neural Inf. Process. Syst.*, 2008, pp. 1177–1184.
- [38] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online Learning with Kernels," *IEEE Trans. Signal Process.*, vol. 52, pp. 2165–2176, August 2004.
- [39] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," *Subseries of Lecture Notes in Computer Science Edited by JG Carbonell and J. Siekmann*, p. 416, 2001.
- [40] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *J. Mach. Learn. Res.*, vol. 20, no. 3, pp. 1–44, 2019.
- [41] A. Koppel, "Consistent online gaussian process regression without the sample complexity bottleneck," in *IEEE ACC*. IEEE, 2019.
- [42] V. S. Borkar, "Stochastic approximation with 'controlled markov' noise," *Systems & control letters*, vol. 55, no. 2, pp. 139–145, 2006.
- [43] P. Karmakar and S. Bhatnagar, "Two time-scale stochastic approximation with controlled markov noise and off-policy temporal-difference learning," *Mathematics of Operations Research*, vol. 43, no. 1, pp. 130–151, 2017.
- [44] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.
- [45] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, 2002.
- [46] C. K. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Adv. Neural Inf. Process. Syst.*, 2001, pp. 682–688.
- [47] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. F. Balcan, and L. Song, "Scalable kernel methods via doubly stochastic gradients," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 3041–3049.
- [48] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.
- [49] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.
- [50] A. Simonetto, "Time-varying convex optimization via time-varying averaged operators," *arXiv preprint arXiv:1704.07338*, 2017.
- [51] J. Zhu and T. Hastie, "Kernel Logistic Regression and the Import Vector Machine," *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.
- [52] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proc. Seventh ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. ACM, 2001, pp. 377–382.
- [53] T. Le, V. Nguyen, T. D. Nguyen, and D. Phung, "Nonparametric budgeted stochastic gradient descent," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 654–662.
- [54] Z. Wang, K. Crammer, and S. Vucetic, "Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3103–3131, 2012.
- [55] J. Sun, J. L. Moore, A. Bobick, and J. M. Rehg, "Learning visual object categories for robot affordance prediction," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 174–197, 2010.
- [56] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.
- [57] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug 2004.
- [58] M. Anthony and P. L. Bartlett, *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

SUPPLEMENTARY MATERIAL FOR
 “NONSTATIONARY NONPARAMETRIC ONLINE LEARNING:
 BALANCING DYNAMIC REGRET AND MODEL PARSIMONY”

APPENDIX B
 PRELIMINARY TECHNICAL RESULTS

Next, we establish some technical conditions in terms of Proposition 1, Proposition 2, and Lemma 3, which are essential to the ensuing proofs of Theorem 1 and Theorem 2. For instance, the result of Proposition 1 is utilized in (42) and (49), the result of Proposition 2 is used in (55), and the statement of Lemma 3 is utilized in 38.

Proposition 1. *Let Assumptions 1-3 hold and denote $\{f_t\}$ as the sequence generated by Algorithm 1 with $f_0 = 0$. Further, denote f^* as the optimum defined by (97). Both quantities are bounded by the constant $K := CX/\lambda$ in Hilbert norm for all t as*

$$\|f_t\|_{\mathcal{H}} \leq \frac{CX}{\lambda}, \quad \|f^*\|_{\mathcal{H}} \leq \frac{CX}{\lambda} \quad (84)$$

The proof of Proposition 1 is similar to in [40, Proposition 7] but adapted to the distribution-free non-stationary case considered here.

Proof. Since we repeatedly use the Cauchy-Schwartz inequality together with the reproducing kernel property in the following analysis, we here note that for all $g \in \mathcal{H}$, $|g(\mathbf{x}_t)| \leq |\langle g, \kappa(\mathbf{x}_t, \cdot) \rangle_{\mathcal{H}}| \leq X\|g\|_{\mathcal{H}}$. Now, consider the magnitude of f_1 in the Hilbert norm, given $f_0 = 0$

$$\begin{aligned} \|f_1\|_{\mathcal{H}} &= \left\| \mathcal{P}_{\mathcal{H}_{\mathcal{D}_1}} \left[\eta_0 \nabla_f \check{\ell}(0) \right] \right\|_{\mathcal{H}} \\ &\leq \eta_0 \|\nabla_f \check{\ell}(0)\|_{\mathcal{H}} \leq \eta_0 |\check{\ell}'(0)| \|\kappa(\mathbf{x}_0, \cdot)\|_{\mathcal{H}} \\ &\leq \eta_0 CX < \frac{CX}{\lambda}. \end{aligned} \quad (85)$$

The first equality comes from substituting in $f_0 = 0$ and the second inequality comes from the definition of optimality condition of the projection operator and the homogeneity of the Hilbert norm, and the chain rule applied to definition of the functional stochastic gradient in with the Cauchy-Schwartz inequality. Lastly, we make use of Assumptions 1 and 2 to bound the scalar derivative $\check{\ell}'$ using the Lipschitz constant, and the boundedness of the kernel map [cf. (15)]. The final strict inequality in (85) comes from applying the step-size condition $\eta_0 < 1/\lambda$.

Now we consider the induction step. Given the induction hypothesis $\|f_t\|_{\mathcal{H}} \leq CX/\lambda$, consider the magnitude of the iterate at the time $t + 1$ as

$$\begin{aligned} \|f_{t+1}\|_{\mathcal{H}} &= \left\| \mathcal{P}_{\mathcal{H}_{\mathcal{D}_{t+1}}} \left[(1 - \eta H \lambda) f_t - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) \right] \right\|_{\mathcal{H}} \\ &\leq \|(1 - \eta H \lambda) f_t - \eta \nabla_f \check{L}_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \\ &\leq (1 - \eta H \lambda) \|f_t\|_{\mathcal{H}} + \eta \|\nabla_f \check{L}_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}, \end{aligned} \quad (86)$$

where we have applied the non-expansion property of the projection operator for the first inequality on the right-hand side of (86), and the triangle inequality for the second. Now, apply the induction hypothesis $\|f_t\|_{\mathcal{H}} \leq CX/\lambda$ to the first term on the right-hand side of (86), and the chain rule together with the triangle inequality to the second to obtain

$$\begin{aligned} \|f_{t+1}\|_{\mathcal{H}} &\leq (1 - \eta H \lambda) \frac{CX}{\lambda} + \eta \sum_{\tau=t-H+1}^t |\check{\ell}'_t(f_t(\mathbf{x}_t))| \|\kappa(\mathbf{x}_t, \cdot)\|_{\mathcal{H}} \\ &\leq \left(\frac{1}{\lambda} - \eta H\right) CX + \eta H C X = \frac{CX}{\lambda} \end{aligned} \quad (87)$$

where we have made use of Assumptions 1 and 2 to bound the scalar derivative $\check{\ell}'$ using the Lipschitz constant, and the boundedness of the kernel map [cf. (15)] as in the base case for f_1 , as well as the fact that $\eta < 1/(H\lambda)$. The same bound holds for f^* by applying [38][Section V-B] with $m \rightarrow \infty$. \square

Next we introduce a proposition which quantifies the error due to subspace projections in terms of the ratio of the compression budget to the learning rate.

Proposition 2. *Fix an independent realization \mathbf{x}_t that parameterizes the loss L_t at time t . Then the difference between the projected online functional gradient and the un-projected online functional gradient of the regularized loss by (99) and (98), respectively, is bounded for all t as*

$$\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \leq \frac{\epsilon}{\eta} \quad (88)$$

where $\eta > 0$ denotes the algorithm step-size and $\epsilon > 0$ is the compression parameter of Algorithm 1.

Proof. Consider the square-Hilbert-norm difference of $\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))$ and $\nabla_f L_t(f_t(\mathbf{S}_t))$ defined in (98) and (99), respectively,

$$\begin{aligned} & \|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \\ &= \left\| \left(f_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \left[f_t - \eta \nabla_f L_t(f_t(\mathbf{S}_t)) \right] \right) / \eta - \nabla_f L_t(f_t(\mathbf{S}_t)) \right\|_{\mathcal{H}}^2 \end{aligned} \quad (89)$$

Multiply and divide $\nabla_f L_t(f_t(\mathbf{S}_t))$, the last term, by η , and reorder terms to write

$$\begin{aligned} & \left\| \left(f_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \left[f_t - \eta \nabla_f L_t(f_t(\mathbf{S}_t)) \right] \right) / \eta - \nabla_f L_t(f_t(\mathbf{S}_t)) \right\|_{\mathcal{H}}^2 \\ &= \left\| \frac{1}{\eta} (f_t - \eta \nabla_f L_t(f_t(\mathbf{S}_t))) - \frac{1}{\eta} \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \left[f_t - \eta \nabla_f L_t(f_t(\mathbf{S}_t)) \right] \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{\eta^2} \|\tilde{f}_{t+1} - f_{t+1}\|_{\mathcal{H}}^2 \end{aligned} \quad (90)$$

where we have substituted the definition of \tilde{f}_{t+1} and f_{t+1} (10), and pulled the nonnegative scalar η outside the norm. Now, note that the KOMP stopping criterion in Algorithm 1 is $\|\tilde{f}_{t+1} - f_{t+1}\|_{\mathcal{H}} \leq \epsilon$, which we apply to the last term on the right-hand side of (90) to conclude (88). \square

Next we establish that Algorithm 1 yields a sequence that satisfies a standard descent relation in the statement of Lemma 3, via convexity and smoothness of the cost functions.

Lemma 3. Consider the sequence generated $\{f_t\}$ by Algorithm 1 with $f_0 = 0$. Under Assumptions 1-4, the following online descent relation holds for a static comparator f^* as defined in (97).

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &\leq \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t))] \\ &\quad + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (91)$$

With Propositions 1 - 2 and Lemma 3 stated, we may now establish some basic results that form the backbone of our dynamic regret analysis to come.

Proof. Begin by considering the square of the Hilbert-norm difference between f_{t+1} and f^* defined by (97), and expand the square to write

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &= \|f_t - \eta \tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - f^*\|_{\mathcal{H}}^2 \\ &= \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta\langle f_t - f^*, \tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) \rangle_{\mathcal{H}} \\ &\quad + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (92)$$

Add and subtract the gradient of the regularized instantaneous risk $\nabla_f L_t(f_t(\mathbf{S}_t))$ defined in (98) to the second term on the right-hand side of (92) to obtain

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &= \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta\langle f_t - f^*, \nabla_f L_t(f_t(\mathbf{S}_t)) \rangle_{\mathcal{H}} \\ &\quad - 2\eta\langle f_t - f^*, \tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t)) \rangle_{\mathcal{H}} \\ &\quad + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (93)$$

We deal with the third term on the right-hand side of (93), which represents the directional error associated with the sparse stochastic projections, by applying the Cauchy-Schwartz inequality together with Proposition 2 to obtain

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &\leq \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta\langle f_t - f^*, \nabla_f L_t(f_t(\mathbf{S}_t)) \rangle_{\mathcal{H}} \\ &\quad + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (94)$$

From the convexity of loss function at each t , it holds that

$$L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t)) \leq \langle f_t - f^*, \nabla_f L_t(f_t(\mathbf{S}_t)) \rangle_{\mathcal{H}}, \quad (95)$$

which we substitute into the second term on the right-hand side of the relation given in (94) to obtain

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &\leq \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t))] \\ &\quad + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + \eta^2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2. \end{aligned} \quad (96)$$

as stated in Lemma 3. \square

Here, to establish some baseline results that are helpful in the dynamic setting, we characterize the static regret of Algorithm 1. We are interested in deriving the dynamic regret bounds for DynaPOLK in terms of the function variations \mathcal{V}_T . To achieve that, the following static regret analysis is precursor and presented here in detail for the completeness.

A. Static Regret

The classical performance metric for an action sequence $\{f_t\}_{t=1}^T$ is its cost accumulation as compared with a best single action in hindsight f^* , defined as the static regret:

$$\mathbf{Reg}_T^S = \sum_{t=1}^T \ell_t(f_t(\mathbf{x}_t)) - \sum_{t=1}^T \ell_t(f^*(\mathbf{x}_t)), \quad (97)$$

where $f^* = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{t=1}^T \ell_t(f(\mathbf{x}_t))$. We begin by defining some key quantities to simplify the analysis and clarify the technical setting for which our regret bounds are valid. To be specific, define the regularized online gradient as

$$\nabla_f L_t(f_t(\mathbf{S}_t)) = \nabla_f \check{L}_t(f_t(\mathbf{S}_t)) + \lambda f_t \quad (98)$$

and its projected variant associated with the step defined in (10):

$$\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) = \left(f_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \left[f_t - \eta_t \nabla_f L_t(f_t(\mathbf{S}_t)) \right] \right) / \eta \quad (99)$$

such that the Hilbert space update of Algorithm 1 [cf. (10)] may be expressed as an online projected gradient step

$$f_{t+1} = f_t - \eta \tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)). \quad (100)$$

The definitions (99) - (98) will be used to analyze the convergence behavior of the algorithm.

We first establish a foundational result for the subsequent analysis of the non-stationary setting, which is conditions under which Algorithm 1 is asymptotically no-regret. This result is stated next.

Theorem 3. *Suppose $\{f_t\} \subset \mathcal{H}$ is the function sequence generated by Algorithm 1 for T iterations. Then for regularization parameter $\lambda > 0$, with step-size $\eta < \min(1/(H\lambda), 1/L)$, under Assumptions 1-3, we have the following regret bound:*

$$\begin{aligned} \mathbf{Reg}_T^S &\leq \frac{\|f_1 - f^*\|_{\mathcal{H}}^2}{2\eta} + \frac{2\epsilon TCX}{\eta\lambda} + \frac{\epsilon^2 T}{\eta} + \frac{\eta Z^2 T}{2} \\ &= \mathcal{O}\left(\frac{1 + (\epsilon + \epsilon^2)T}{\eta} + \eta T\right). \end{aligned} \quad (101)$$

where $Z := CH(1 + X)$. Then, the static regret grows sublinearly $\mathbf{Reg}_T^S \leq \mathcal{O}(\sqrt{T})$ in T for step-size selection $\eta = \mathcal{O}(T^{-1/2})$ and compression budget $\epsilon > \mathcal{O}(T^{-1/2})$.

Proof. Begin by noting that

$$\begin{aligned} &\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \\ &= \|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t)) + \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \end{aligned} \quad (102)$$

where we add subtract the term $\nabla_f L_t(f_t(\mathbf{S}_t))$. Using Cauchy-Schwartz inequality and the result of Proposition 2, we get

$$\begin{aligned} &\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \\ &\leq \left(\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} + \|\nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}} \right)^2 \\ &\leq 2\|\tilde{\nabla}_f L_t(f_t(\mathbf{S}_t)) - \nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 + 2\|\nabla_f L_t(f_t(\mathbf{S}_t))\|_{\mathcal{H}}^2 \\ &\leq \frac{2\epsilon^2}{\eta^2} + 2\|\nabla_f L_t(f_t(\mathbf{S}_t))\|^2. \end{aligned} \quad (103)$$

Substitute the upper bound obtained in (103) into the descent property stated in Lemma3 [c.f. (96)] to obtain

$$\begin{aligned} \|f_{t+1} - f^*\|_{\mathcal{H}}^2 &\leq \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t))] \\ &\quad + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + 2\epsilon^2 + 2\eta^2\|\nabla_f L_t(f_t(\mathbf{S}_t))\|^2. \end{aligned} \quad (104)$$

From the definition of loss function $\ell_t(\cdot)$, we have

$$\begin{aligned} \|\nabla_f L_t(f_t(\mathbf{S}_t))\| &= \|\nabla_f \check{L}_t(f_t(\mathbf{S}_t)) + \lambda H f_t(\mathbf{S}_t)\| \\ &= \|\nabla_f \check{L}_t(f_t(\mathbf{S}_t))\| + \lambda H \|f_t(\mathbf{S}_t)\|. \end{aligned} \quad (105)$$

The Assumption 2 implies that $\|\nabla_f \check{L}_t(f_t(\mathbf{S}_t))\| \leq CH$ and from the result of Proposition 1, we get

$$\|\nabla_f L_t(f_t(\mathbf{S}_t))\| \leq CH(1 + X). \quad (106)$$

(a, b)	Regret	Model Order M	Comments
$a = b$	$\mathcal{O}(T)$	$\mathcal{O}(1)$	Linear regret
$a - b = 1/p$	$\mathcal{O}(T^{(p-1)/p})$	$\mathcal{O}(T)$	Linear Model Order
$a - b = 1/(1+p)$	$\mathcal{O}(T^{p/(1+p)})$	$\mathcal{O}(T^{p/(1+p)})$	Sublinear Order Model <i>and</i> Regret

TABLE IV: Summary of convergence rates for different parameter selections.

Let us define $Z = CH(1 + X)$ and utilize the upper bound in (106) into (104), we get

$$\|f_{t+1} - f^*\|_{\mathcal{H}}^2 \leq \|f_t - f^*\|_{\mathcal{H}}^2 - 2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t))] + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + 2\epsilon^2 + 2\eta^2 Z^2. \quad (107)$$

After rearranging the terms in (107), we get

$$2\eta[L_t(f_t(\mathbf{S}_t)) - L_t(f^*(\mathbf{S}_t))] \leq \|f_t - f^*\|_{\mathcal{H}}^2 - \|f_{t+1} - f^*\|_{\mathcal{H}}^2 + 2\epsilon\|f_t - f^*\|_{\mathcal{H}} + 2\epsilon^2 + \eta^2 Z^2. \quad (108)$$

Next, divide both sides of (108) by 2η , we get

$$\begin{aligned} & [L_t(f_t(\mathbf{x}_t)) - L_t(f^*(\mathbf{x}_t))] \\ & \leq \frac{\|f_t - f^*\|_{\mathcal{H}}^2}{2\eta} - \frac{\|f_{t+1} - f^*\|_{\mathcal{H}}^2}{2\eta} + \frac{\epsilon}{\eta}\|f_t - f^*\|_{\mathcal{H}} + \frac{\epsilon^2}{\eta} + \frac{\eta Z^2}{2} \\ & \leq \frac{\|f_t - f^*\|_{\mathcal{H}}^2}{2\eta} - \frac{\|f_{t+1} - f^*\|_{\mathcal{H}}^2}{2\eta} + \frac{\epsilon}{\eta} \cdot \frac{2CX}{\lambda} + \frac{\epsilon^2}{\eta} + \frac{\eta Z^2}{2}. \end{aligned} \quad (109)$$

We apply the upper bound obtained in Proposition 1 to get the second inequality in (109). Taking the sum from $t = 1$ to T in (109) and dropping the negative terms from the right hand side, we get

$$\begin{aligned} \sum_{t=1}^T [L_t(f_t(\mathbf{x}_t)) - L_t(f^*(\mathbf{x}_t))] & \leq \frac{\|f_1 - f^*\|_{\mathcal{H}}^2}{2\eta} + \frac{2\epsilon TCX}{\eta\lambda} \\ & \quad + \frac{\epsilon^2 T}{\eta} + \frac{\eta Z^2 T}{2}. \end{aligned} \quad (110)$$

For a fixed step size η such that $\eta < \frac{1}{\lambda}$, we obtain the static regret as

$$\mathbf{Reg}_T^S = \mathcal{O}\left(\frac{1 + (\epsilon + \epsilon^2)T}{\eta} + \eta T\right) \quad (111)$$

which grows sublinearly in T for $\eta = \mathcal{O}(T^{-1/2})$ and $\epsilon \in [0, T^{-1/2})$. For instance, $\mathbf{Reg}_T^S \leq \mathcal{O}(\sqrt{T})$ if we select $\eta = \frac{1}{\sqrt{T}}$ and $\epsilon = \frac{1}{T}$. However, if instead we select η and ϵ as follows

$$\eta = \mathcal{O}(T^{-a}) \quad \text{and} \quad \epsilon = \mathcal{O}(T^{-b}). \quad (112)$$

then the right-hand side of (111) becomes

$$\mathbf{Reg}_T^S = \mathcal{O}\left(T^b + T^{1-(a-b)} + T^{1-b}\right). \quad (113)$$

When we substitute (112) into Lemma 1, we obtain

$$M = \mathcal{O}(T^{p(a-b)}). \quad (114)$$

For the static regret to be sublinear, we need $b \in (0, 1)$ and $a \in (b, b + \frac{1}{p})$. As long as the dimension p is not too large, we always have a range for a . This implies that $p(a - b) \in (0, 1)$ to ensure sublinear model order M as well. \square

Theorem 3 establishes that Algorithm 1 exhibits the asymptotic no-regret property under appropriate step-size and compression budget selections, and that there is a tunable tradeoff between regret and memory: for tighter regret, one should set $\epsilon \in [0, T^{-1/2})$ to be closer to null. However, making ϵ too small causes the model order to grow as the time horizon T . This result exactly matches regret growth for parametric settings with $\epsilon = 0$. The largest compression budget allowable that still yields asymptotic no-regret is $\epsilon = T^{-1/2} - \gamma$ for $\gamma > 0$. Observe that by setting $\epsilon = \mathcal{O}(T^{-a})$ and $\eta = \mathcal{O}(T^{-b})$, we have that sublinear regret for $b \in (0, 1)$ and $a \in (b, b + \frac{1}{p})$. To have the model complexity under control, via (114), this imposes constraints $p(a - b) \in (0, 1)$. These observations are summarized in Table IV.

APPENDIX C
SELECTION OF THE STEP SIZE η AND ϵ FOR THE DYNAMIC CASE

In this section, we provide the detailed analysis of the parameters (η, ϵ) selection and the relative effect on the regret performance and model order. First of all, we collect all the conditions for η and ϵ imposed by the analysis in the paper. We have

$$\eta < \frac{1}{\lambda} \quad \text{and} \quad \eta < \frac{1}{L}. \quad (115)$$

Next, we combine the conditions and get

$$\eta < \min \left\{ \frac{1}{\lambda}, \frac{1}{L} \right\} \quad (116)$$

which implies that there is an upper bound on the value of the step size. Similarly, for the strongly convex case, we have $\eta < \min\{\frac{1}{\lambda}, \frac{\mu}{L^2}\}$.

Before proceeding with the analysis, let us make a common selection for ϵ as follows

$$\epsilon = \mathcal{O}(T^{-\alpha}). \quad (117)$$

The values of the positive constant α will be decided later. We discuss the ϵ selection for each of the case separately and corresponding upper bound on the number of elements in the dictionary.

(1) Dynamic regret convex case in terms of V_T

$$\mathbf{Reg}_T^D \leq \lceil \frac{T}{\Delta_T} \rceil \mathcal{O} \left(\frac{1 + \epsilon \Delta_T}{\eta} + \eta \Delta_T \right) + 2\Delta_T V_T. \quad (118)$$

We can select it in a similar way to Case I with $\epsilon = \mathcal{O}(\Delta_T^{-\alpha})$ and $\eta = \mathcal{O}(\Delta_T^{-\beta})$.

(2) Dynamic regret, convex case, in terms of W_T : The expression for dynamic regret in terms of W_T and convex loss function is given by

$$\mathbf{Reg}_T^D \leq (1 + T\sqrt{\epsilon} + W_T). \quad (119)$$

Using the ϵ selection in (117), we can write

$$\mathbf{Reg}_T^D = \mathcal{O} \left(1 + T^{(1-\frac{\alpha}{2})} + W_T \right). \quad (120)$$

and

$$M = \mathcal{O}(T^{\alpha p}). \quad (121)$$

For the regret and the model order to be sublinear up to the variations W_T , we need $\alpha \in (0, \frac{1}{p}]$. As long as the dimension p is not too large, we always have a range for a . This implies that $\alpha p \in (0, 1)$ and hence M is sublinear.

α	Regret	M	Comments
$\alpha = 0$	$\mathcal{O}(T) + W_T$	$\mathcal{O}(1)$	Linear regret
$\alpha = \frac{1}{p}$	$\mathcal{O} \left(T^{\frac{(2p-1)}{2p}} + W_T \right)$	$\mathcal{O}(T)$	Linear M
$\alpha = \frac{1}{p+1}$	$\mathcal{O} \left(T^{\frac{2p+1}{2p+2}} + W_T \right)$	$\mathcal{O}(T^{p/(1+p)})$	Sublinear M

TABLE V: Summary of dynamic regret rates for convex loss function.

(4) Dynamic regret, strongly convex case, in terms of W_T : The expression for dynamic regret in terms of W_T and strongly convex loss function is given by

$$\begin{aligned} \mathbf{Reg}_T^D &\leq \mathcal{O} \left(\frac{1 + T\sqrt{\epsilon} + W_T}{1 - \rho} \right) \\ &\leq o(1 + T\sqrt{\epsilon} + W_T). \end{aligned} \quad (122)$$

where $\rho = \sqrt{(1 - 2\eta(\mu - \eta L^2))}$. The expression in (122) is similar to the one on (119) except for the term $(1 - \rho)$ in the denominator. If we choose η such that $(1 - \rho) > \eta$, the results for strongly convex functions is improved. Rearrange this expression to obtain

$$(1 - \eta)^2 > \rho^2 = 1 - 2\eta(\mu - \eta L^2)$$

which, upon solving for a condition on η , simplifies to

$$\eta < \frac{2(\mu - 1)}{2L^2 - 1}.$$

Next, we summarize the dynamic regret rates achieved for a constant η and $\epsilon = \mathcal{O}(T^{-\alpha})$ with different α in Table VI.

α	Regret	M	Comments
$\alpha = 0$	$o(T) + W_T$	$o(1)$	Linear regret
$\alpha = \frac{1}{p}$	$o\left(T^{\frac{(2p-1)}{2p}} + W_T\right)$	$o(T)$	Linear M
$\alpha = \frac{1}{p+1}$	$o\left(T^{\frac{2p+1}{2p+2}} + W_T\right)$	$o(T^{p/(1+p)})$	Sublinear M

TABLE VI: Summary of dynamic regret rates for strongly convex loss function.