From Fairness to Infinity: Outcome-Indistinguishable (Omni)Prediction in Evolving Graphs

Cynthia Dwork DWORK@SEAS.HARVARD.EDU
Harvard

Chris Hays JHAYS@MIT.EDU
MIT

Nicole Immorlica NICIMM@GMAIL.COM
Microsoft Research

Juan C. Perdomo JCPERDOMO@G.HARVARD.EDU
Harvard

Pranay Tankala PRANAY_TANKALA@G.HARVARD.EDU
Harvard

Editors: Nika Haghtalab and Ankur Moitra

Abstract

Professional networks provide invaluable entree to opportunity through referrals and introductions. A rich literature shows they also serve to entrench and even exacerbate a status quo of privilege and disadvantage. Hiring platforms, equipped with the ability to nudge link formation, provide a tantalizing opening for beneficial structural change. We anticipate that key to this prospect will be the ability to estimate the likelihood of edge formation in an evolving graph.

Outcome-indistinguishable prediction algorithms ensure that the modeled world is indistinguishable from the real world by a family of statistical tests. Omnipredictors ensure that predictions can be post-processed to yield loss minimization competitive with respect to a benchmark class of predictors for many losses simultaneously, with appropriate post-processing. We begin by observing that, by combining a slightly modified form of the online K29* algorithm of Vovk (2007) with basic facts from the theory of reproducing kernel Hilbert spaces, one can derive simple and efficient online algorithms satisfying outcome indistinguishability and omniprediction, with guarantees that improve upon, or are complementary to, those currently known. This is of independent interest; for example, we obtain efficient outcome indistinguishability for some interesting infinite collections of tests, as well as for any bounded function — including those computable by deep (graph) neural networks.

We apply these techniques to evolving graphs by designing efficient kernel functions that capture socially meaningful features of nodes and their neighborhoods. We obtain online outcome-indistinguishable omnipredictors for rich — possibly infinite — sets of distinguishers yielding, *inter alia*, multicalibrated predictions of edge formation with respect to pairs of demographic groups, and the ability to simultaneously optimize loss as measured by a variety of social welfare functions.

Keywords: link prediction, computational indistinguishability, multicalibration, kernels

1. Introduction

Professional networks provide invaluable entree to opportunity through referrals and introductions. A rich literature shows they may also serve to entrench and even exacerbate a status quo of privilege and disadvantage. For example, in a network with two disjoint groups with equal ability distribution, homophily can, through job referrals, result in the draining of opportunity from the smaller group to the larger (Bolte et al., 2020; Calvo-Armengol and Jackson, 2004; Okafor, 2020). Remedies are few. Hiring platforms, equipped with the ability to nudge link formation, provide a tantalizing opening for beneficial structural change.

Key to this prospect is the ability to estimate edge formation in an evolving network. This is a prediction problem for the universe of pairs of network nodes (individuals) (i, j), suggesting that standard prediction methods can be applied. While this intuition is correct, the situation is complicated by the fact that edge formation need not be a property of the endpoints alone, but can also depend on the topology and other features of the neighborhoods of the principals i and j. For example, the probability that the edge (i, j) forms may be depend in part on the number of contacts that i and j have in common. Let us informally call this the problem of complex domains. To complicate matters even further, these features change over time as individuals grow their networks, switch jobs, *etc*. We treat edge prediction in a social network as an online, distribution-free problem and aim to make predictions that are valid and useful, *regardless* of the underlying edge formation process.

Outcome indistinguishability (OI) (Dwork et al., 2021) frames learning not as loss minimization — the dominant paradigm in supervised machine learning – but instead as satisfaction of a collection of "indistinguishability" constraints. Outcome indistinguishability considers two alternate worlds of individual-outcome pairs: in the natural world, individuals' outcomes are generated by Real Life's true distribution; in the simulated world, individuals' outcomes are sampled according to a predictive model. Outcome indistinguishability requires the learner to produce a predictor in which the two worlds are computationally indistinguishable. This is captured by specifying a class of distinguishers to be fooled by the predictor. For example, in our network setting, we could consider a class of distinguishers that includes the fraction of links formed between two women and the fraction of links formed between MIT graduates. Outcome indistinguishability then guarantees the predictions of the learner match the Real Life fractions of intragroup connections for both women and MIT graduates simultaneously. Simplifying for ease of exposition, one may define a class of distinguishers corresponding to a (possibly infinite) collection of (arbitrarily intersecting) demographic groups and prediction values, in which case outcome indistinguishability ensures that the predictor is calibrated simultaneously on each group when viewed in isolation. This is *multicalibration*, defined in the seminal work of Hébert-Johnson, Kim, Reingold, and Rothblum (Hébert Johnson et al., 2018)[1]; the view of simultaneous calibration in different demographic groups as a potential *fairness* goal was introduced by Kleinberg, Mullainathan, and Raghavan (Kleinberg et al., 2017)[2].

(Online and batch) *omnipredictors* (Gopalan et al., 2022; Garg et al., 2024) produce predictions that can be used to ensure loss minimization for a wide, even infinite, collection of loss functions, with respect to a benchmark class of predictors. For example, in the batch case one might train a predictor to optimize squared loss, but later one might wish to deploy the predictor in a way that minimizes 0-1 loss *with no*

1. (Dwork et al., 2021) defines a hierarchy of outcome indistinguishability results, according to the degree of access to the predictor that is given to the distinguishers. When not otherwise specified, we are referring to *sample-access* OI. The term *multicalibration* has become more general than its usage here, referring also to a class of real-valued functions (see, *e.g.*, (Gopalan et al., 2022)). For equivalences, see (Dwork et al., 2021; Gopalan et al., 2022).
2. Thus, what began as a requirement for fairness for a small collection of demographic groups has transformed to a guarantee for potentially infinitely many functions.

further training. In the online case, a company running a professional networking platform may offer a variety of products, such as mentoring circles and professional training programs. The different product groups may take different kinds actions based on the shared predictions. The mentoring circle product might seek individuals with high probability of link formation from different levels of seniority whereas the professional training program might seek individuals with high probability of link formation with different types of expertise. Differing goals can lead to different loss functions for the different product groups. Omnipredictors address this problem. Omniprediction, too, can be expressed in the language of outcome indistinguishability (Gopalan et al., 2023)[3].

A full treatment of fairness in networks requires understanding which kinds of links will advance social and/or individual welfare and which nudges are likely to be most beneficial. In this vein (Dwork et al., 2025), in an effort emerging from this work, show that inequality-aware platforms may reduce inequality by subsidizing connections, through link recommendations that reduce costs, between privileged and underprivileged individuals. Indeed, they develop a model in which mixed-privilege connections can be shown to be not only inequality improving but also welfare improving, *over all possible equilibria*, compared to not recommending links or recommending some smaller fraction of cross-group links.

1.1. Our contributions and related work.

We initiate the study of online outcome indistinguishability and omniprediction for link prediction in a dynamic graph. Our technical starting point is a novel, randomized variant of Vovk's $K29^*$ online prediction algorithm (Vovk, 2007), which we extend to handle kernels that are not continuous. Our algorithm, which we call the Any Kernel algorithm, achieves kernel outcome indistiguishability, that is, indistinguishability with respect to any infinite collection of real-valued functions in a reproducing kernel Hilbert space.[4] To our knowledge, our work is the first in the multigroup fairness literature to use kernel methods for online prediction; in independent work, (Gopalan et al., 2024a) observed their usefulness for the batch case (as a natural consequence of the equivalence of auditing and agnostic learning in this setting (Hébert Johnson et al., 2018; Kearns et al., 2018)). See (Pérez-Suay et al., 2017; Tan et al., 2020; Perez-Suay et al., 2023; Kumar et al., 2018; Błasiok et al., 2023) for other applications to fairness.

Our algorithm is quite general. By designing bespoke efficient kernel functions for use in the Any Kernel algorithm, we obtain efficient outcome indistinguishability with respect to distinguishers that capture features such as edge connections between pairs of demographic groups, number of mutual connections between pairs of nodes, isomorphism classes of the local neighborhoods, and any bounded function – including those computable by graph neural networks. A core advantage of our approach is that, by designing computationally efficient kernels, we can efficiently guarantee indistinguishability with respect to rich classes of functions (e.g. all isomorphism invariant functions of the neighborhood) without having to solve a complex search problem over the entire class (e.g. weak agnostic learning).

Link predictions may be used for a variety of downstream decisions; for example loss functions may be used to measure predictive accuracy or desirability of outcomes. Moreover the precise loss function may not be known to the predicting algorithm, as the predictions may be used by many different parties. We show how to address these problems by using the Any Kernel algorithm to achieve computationally efficient low-regret omniprediction with respect to potentially infinite and continuous-valued comparison classes; again, it is precisely the connection to kernel functions that makes this possible. Our algorithms do not depend on access to a regression oracle (cf., (Garg et al., 2024)).

3. Sample access OI (equivalently, multicalibration) implies omniprediction (Gopalan et al., 2022), but weaker outcome indistinguishability conditions also suffice (Gopalan et al., 2023; Dwork et al., 2023).
4. Informally for now, reproducing kernel Hilbert spaces are potentially very rich classes of non-parametric functions.

We also (1) extend our results to quantile regression and high-dimensional regression, which are of general interest in forecasting; (2) examine the relationship of *offline* kernel methods with previous results in batch outcome indistinguishability, and (3) initiate the study of the *distance to multicalibration*.

From Abstraction to Application. As it is infeasible to make predictions for all non-edges and a random nudge may likely be useless, platform-assisted fair networking will require policies for focusing the platform's attention. However, when link predictions are used to recommend connections, multiple recommendations are likely made (somewhat) simultaneously; for example, the platform may recommend, to a single individual, a batch of 5 potential connections, violating the pure online model. Moreover, the platform is likely to make such small batches of recommendations to a large number of users independently, without waiting for any single user to act, again violating the online model. These are a rich veins for future work.

Relation to the graph prediction literature. A great deal of research addresses link formation, typically in the batch setting, in which a subset of edges are presented as training data; see, for example, the book (Hamilton, 2020). A few papers have also considered prediction on *evolving* graphs (Kumar et al., 2019; Trivedi et al., 2019; Ma et al., 2020; Rossi et al., 2020; Yu et al., 2023). Graph machine learning is a very active area of research with many research directions left unexplored (Morris et al., 2024). These approaches tend to focus on specific representations of graphs, which may be tailored to the semantics of nodes and edges. Our approach differs in two main respects: first, we consider the online case in which the graph is evolving over time; at any given time step the algorithm may be given a pair of vertices (i, j) and the goal is to predict whether an edge will form between them at the given time. Secondly, inspired by the observation that online calibrated forecasting can be achieved by *backcasting* (Foster and Hart, 2021), we take a more formal approach, *ignoring* the semantics of the nodes and edges. The semantics are introduced via the class of distinguishers.

Comparison with previous work in algorithmic fairness. We postpone detailed comparison to previous work in multicalibration, outcome indistinguishability and omniprediction to Appendices B and C respectively. Connections between outcome-indistinguishable simple edge prediction and forms of graph regularity were investigated in (Dwork et al., 2023). Our algorithm is the first online $\mathcal{O}(\sqrt{T})$ omnipredictor that can compete with infinite or real-valued comparison classes $\mathcal{H}$. Our results are non-asymptotic (*i.e.*, hold for all T), and the constants hidden in the big-$\mathcal{O}$ are usually small. Unlike previous online algorithms, we require neither a regression or weak agnostic learning oracle for omniprediction (Garg et al., 2024; Okoroafor et al., 2025) nor explicit enumeration over all distinguishers for outcome indistinguishability (Gupta et al., 2022). Unlike our work, (Garg et al., 2024) offers the stronger guarantee of *swap* omniprediction (see Appendix C). Finally, our bound for outcome indistinguishability error may deteriorate by a factor of m for RKHSs that contain m *arbitrary* Boolean-valued functions, such as (pairs of) arbitrary demographic group memberships; for the other real-valued function classes mentioned above and in Appendix A, we pay no such price.

Paper organization. The remainder of this paper is organized as follows. Section 2 gives an overview of our technical results. Appendix A gives a full formulation of the fair link prediction problem. Appendix B introduces our main algorithm and results for online outcome indistinguishability. Our results on omniprediction appear in Appendix C. Additional miscellaneous results are derived in Appendix D.

2. Overview of technical results.

Our work has two main sets of technical results. The first set concerns online outcome indistinguishability and the second set concerns efficient, $\sqrt{T}$, online omniprediction. In both cases, we focus on developing

machinery for online prediction that we later specialize to link prediction. As a byproduct of these investigations, we also arrived at new results for online quantile and vector regression, as well as kernel batch algorithms and notions of distance to multicalibration that are of independent interest.

Online outcome indistinguishability (Dwork et al., 2021). The technical starting point of our paper is a result by Vovk (Vovk, 2007) which guarantees online outcome indistinguishability with respect to specific classes of functions $\mathcal{F}$ that form an RKHS, or reproducing kernel Hilbert space. We review both of these concepts below.

An algorithm guarantees online outcome indistinguishability with respect to a class $\mathcal{F} \subseteq \{\mathcal{X} \times [0,1] \to \mathbb{R}\}$ of *distinguishers* if it is guaranteed to generate a sequence of predictions p_t satisfying the following guarantee:

$$\left| \sum_{t=1}^{T} \mathbb{E}(y_t - p_t) f(x_t, p_t) \right| \leqslant o(T) \text{ for all } f \in \mathcal{F}.$$

Here, (x_t, y_t) are an arbitrary sequence of (feature, outcome) pairs in $\in \mathcal{X} \times \{0,1\}$, which can be chosen adversarially and adaptively, and the expectation is taken over the internal randomness of the algorithm. Notably, y_t can be chosen with knowledge of the entire history $\{(x_{t'}, p_{t'}, y_{t'})\}_{t'=1}^{t-1}$, and may depend on x_t and in some cases p_t (see Appendix A for details).

In other words, a sequence of predictions is outcome-indistinguishable if no distinguisher in $\mathcal{F}$ can reliably (with constant advantage) tell the difference between outcomes drawn according to the learner's predictions p_t, and the true outcomes y_t (see Appendix A.1.1 for further discussion).

RKHSs, the K29* algorithm, and the Any Kernel algorithm. A reproducing kernel Hilbert space (RKHS) $\mathcal{F} \subset \{\mathcal{X} \to \mathbb{R}\}$ is a class of functions that can be defined over arbitrary domains (*e.g.*, graphs). Functions in an RKHS have the property that they can be *implicitly* represented by a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Indeed, each kernel k represents a unique RKHS $\mathcal{F}_k$.[5]

The kernel representation enables one to design computationally efficient learning algorithms with guarantees that hold over all functions in the RKHS $\mathcal{F}$, without necessarily having to explicitly solve a search problem over $f \in \mathcal{F}$ (*e.g.*, weak agnostic learning). The efficiency of learning over $\mathcal{F}$ reduces to efficient evaluation of the kernel k. This enables for instance, the design of online algorithms that run in time $\mathcal{O}(t \cdot n \cdot d)$ yet guarantee indistinguishability with respect to $\mathcal{O}(n^d)$ degree d, n-variate polynomials. In addition to their computational benefits, RKHSs can be very expressive. By carefully designing the kernel function k, one can guarantee that the corresponding RKHS of functions $\mathcal{F}_k$ contains specific classes of distinguishers of interest.[6]

Building on the work of Vovk (Vovk, 2007) and insights from (Foster and Hart, 2021), we introduce the Any Kernel algorithm, which guarantees online indistinguishability with respect to any RKHS $\mathcal{F}$. The algorithm is hyperparameter free, and runs in polynomial time whenever the kernel k is bounded and efficiently computable. We summarize its main guarantees below.

Theorem 1 (Informal) *Let k be any kernel function and let $\mathcal{F}$ be its associated RKHS. Then, the Any Kernel algorithm generates a sequence of predictions $p_t \sim \Delta_t$ such that for any $f \in \mathcal{F}$:*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t}(y_t - p_t) f(x_t, p_t) \right| \leqslant \|f\|_{\mathcal{F}} \sqrt{1 + \mathop{\mathbb{E}}_{p_t} \sum_{t=1}^{T} p_t(1-p_t) k((x_t, p_t), (x_t, p_t))} \leqslant B \cdot \|f\|_{\mathcal{F}} \sqrt{T}$$

5. Common classes of functions like linear functions or polynomials are an RKHS, but we will see many others.
6. See Appendix B for a overview of RKHS and formal definition of norms in these spaces. Briefly, an RKHS is a Hilbert space and hence has an inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}} \to \mathbb{R}$. This inner product defines a norm $\|f\|_{\mathcal{F}}^2 = \langle f, f \rangle_{\mathcal{F}}$ which serves a complexity measure for functions f in the space $\mathcal{F}$.

The second inequality holds if $k((x_t, p_t), (x_t, p_t)) \leqslant B^2$ for all t. Here, $\|f\|_{\mathcal{F}}$ is the norm of the function f in RKHS $\mathcal{F}$ and the expectations are taken over the distributions Δ_t produced by the algorithm.

The proof of the theorem above, which we defer to Appendix B, draws heavily on the ideas from the literature on game-theoretic statistics (Shafer and Vovk, 2005), defensive forecasting (Vovk et al., 2005), and forecast hedging (Foster and Hart, 2021). The Any Kernel algorithm extends Vovk's K29* algorithm (Vovk, 2007) so as to work for *any* kernel k and correspondingly any RKHS $\mathcal{F}$. In particular, K29* requires the kernel k to be continuous in the prediction p and hence can only guarantee indistinguishability with respect to functions $f : \mathcal{X} \times [0,1] \to \mathbb{R}$ that are continuous in p.[7] Removing this restriction enables us to consider binary distinguishers or tests that are not continuous in p. These were the central focus of the initial work on outcome indistinguishability (Dwork et al., 2021) and multicalibration (Hébert Johnson et al., 2018).

The key insight behind our extension is that we show that regardless of the choice of kernel, the learner can always find a distribution over two forecasts such that predicting according to this distribution guarantees that their calibration error grows sublinearly no matter the outcome. This simple, yet very powerful idea was previously illustrated by (Foster and Hart, 2021) for the pure calibration setting where there are no x_t. Here, we extend it to the contextual case to develop an algorithm that achieves the optimal $\sqrt{T}$ regret for *any* RKHS.[8]

To operationalize this theorem and guarantee indistinguishability with respect to a pre-specified collection of functions $\mathcal{F}'$, there are two main sets of technical challenges. First, we need to understand how the choice of kernel k relates to its corresponding RKHS $\mathcal{F}_k$ so that we can guarantee that $\mathcal{F}' \subseteq \mathcal{F}_k$. Second, we need to pay special attention to ensure that the kernel can be computed efficiently, has bounded values $k((x,p),(x,p)) \leqslant \mathcal{O}(1)$, and that the functions $f' \in \mathcal{F}'$ have bounded norm in the RKHS $\mathcal{F}_k$ ($\|f'\|_{\mathcal{F}_k}$ is bounded). See Appendix B for specific instantiations of this general recipe for important function classes like low-degree polynomials, or all isomorphism invariant functions a subgraph.

Our results on online outcome indistinguishability directly address these core issues. Building on the rich literature on RKHS, we specialize our results to the link prediction problem and design efficient, bounded kernels whose RKHS contain interesting distinguishers f on graphs and where each f has small RKHS norm. These in particular include powerful predictors such as deep (graph) neural networks.

Proposition 2 (Informal) *Consider the link prediction problem where x_t consists of a pairs of individuals (i_t, j_t) and a graph G_t. For each of the following classes of functions $\mathcal{F}'$, there exists a computationally efficient and bounded kernel whose corresponding RKHS $\mathcal{F}_k$ contains $\mathcal{F}'$:*

1. *All pairs of demographic groups. $\mathcal{F}'$ consists of distinguishers which examine whether the pair (i,j) belong to any pair of demographic groups from a finite list.*
2. *Number of connections and isomorphism classes. $\mathcal{F}'$ consists of tests that examine the number of mutual connections between the pair (i_t, j_t), or the isomorphism class of their local neighborhoods.*
3. *An arbitrary pre-specified set of bounded functions. $\mathcal{F}'$ is a finite benchmark class of deep learning based link predictors (e.g., graph neural networks), or any other bounded function.*

Furthermore, the norms of $f' \in \mathcal{F}'$ in the corresponding RKHS $\mathcal{F}_k$ are all $\mathcal{O}(1)$ in each setting. Therefore, the Any Kernel algorithm instantiated with these kernels guarantees online indistinguishability with respect to any of the $\mathcal{F}'$ above with indistinguishability error bounded by $\mathcal{O}(\sqrt{T})$.[9]

7. In our analysis, it helps to distinguish between the set of features $\mathcal{X}$ and the predictions $p \in [0,1]$.
8. See (Vovk, 2007) for a lower bound.
9. The functions f' in these constructions can additionally depend on the prediction p. For instance, by letting f' examine whether predictions belong to a particular bin $[a,b] \subseteq [0,1]$.

The Any Kernel algorithm

Input: A kernel $k : (\mathcal{X} \times [0,1])^2 \to \mathbb{R}$

For $t = 1, 2, \ldots$

1. Given history $\{(x_i, p_i, y_i)\}_{i=1}^{t-1}$ and current features x_t define $S_t : [0,1] \to \mathbb{R}$ as

$$S_t(p) \stackrel{\text{def}}{=} \sum_{i=1}^{t-1} k((x_t,p),(x_i,p_i))(y_i - p_i) + \frac{1}{2}k((x_t,p),(x_t,p))(1-2p).$$

2. If $\operatorname{sign} S_t(0) = \operatorname{sign} S_t(1) \neq 0$, **return** $\Delta_t = p_t = \frac{1}{2}(1 + \operatorname{sign} S_t(0))$.

3. Define $\varepsilon_t = 1/(10t^3 B_t)$ for $B_t = \max_{t' \leqslant t} k((x_{t'}, p_{t'}), (x_{t'}, p_{t'}))$. If k is continuous in p:

   - Run binary search to find $p_t \in [0,1]$ such that $|S_t(p_t)| \leqslant \varepsilon_t$; **return** $\Delta_t = p_t$ w.p. 1

4. Else, if k is not continuous in p:

   - Run binary search to find $q, q' \in [0,1]$ with $0 \leqslant |q'_t - q_t| \leqslant \varepsilon_t$ and $\operatorname{sign} q_t = \operatorname{sign} S_t(0)$ and $\operatorname{sign} q'_t = \operatorname{sign} S_t(1)$.
   - **return**

$$\Delta_t = \begin{cases} q_t & \text{with probability } \tau \\ q'_t & \text{with probability } 1-\tau. \end{cases} \qquad \text{for } \tau = \frac{|S_t(q'_t)|}{|S_t(q_t)| + |S_t(q'_t)|} \in [0,1]$$

Figure 1: Pseudocode for the Any Kernel algorithm. Steps 1-3 are as in (Vovk, 2007). Step 4 is inspired by (Foster and Hart, 2021). In each iteration, solve the binary search problems in steps 3 or 4 using at most $\log(1/\varepsilon_t)$ oracle evaluations of S_t. Each evaluation of S_t requires t evaluations of the kernel k, hence the runtime at round t is $\widetilde{\mathcal{O}}(t \cdot \mathsf{time}_k)$. If k is forecast-continuous Δ_t is just a point mass at p_t. Otherwise, Δ_t is near deterministic: it is supported on just 2 points q_t, q'_t which are very close together, $|q - q'| \leqslant \mathcal{O}(t^{-3})$. See Theorem 8 for formal guarantees.

While developed for the link prediction problem, the guarantees of the Any Kernel algorithm hold for general domains and can also be used to generate indistiguishability with respect to other interesting classes of functions such as low degree polynomials over the Boolean hypercube (see Theorem 9). Furthermore, by leveraging composition properties of kernels, we can also guarantee predictions which are indistiguishable with respect to sums or products of tests in different RKHSs. This in particular implies indistinguishability with respect to practically important predictors like random forests or gradient boosted decision trees.

Online omniprediction results. While the first set of results focused on algorithms that guaranteed valid *predictions* p_t, our second set of results pertain to the design of algorithms that lead to useful *decisions* $\hat{y}_t$.[10] Assuming that the learner's utility over data $(x_t, \hat{y}_t, y_t)$ is captured by a loss function ℓ, we aim to

10. Note that $\hat{y}_t$ need not be of the same type as y_t; for example, the first might be any value in $[0,1]$ while the second might be Boolean.

achieve lower average loss than functions in a benchmark class $\mathcal{H}$:[11]

$$\frac{1}{T}\sum_{t=1}^{T}\ell(x_t,\hat{y}_t,y_t) \leqslant \inf_{h\in\mathcal{H}}\frac{1}{T}\sum_{t=1}^{T}\ell(x_t,h(x_t),y_t)+o(1). \tag{1}$$

In the link prediction context, predictions have the added advantage that they are likely *performative* (Perdomo et al., 2020). By informing downstream decisions, such as the link recommendations made to a user, predictions don't just forecast the future: they actively shape the likelihood of edge formation. This means that platforms are likely to experiment with the choice of loss function ℓ. They may choose losses favoring predictions to match outcomes, *e.g.*, squared loss $(\hat{y}-y)^2$, or "loss" functions that favor specific outcomes over others, like link formation $1-y$.

Given the diversity of plausible goals, we design online algorithms that generate predictions which can be post-processed to produce good decisions for a wide variety of losses. Importantly, each individual loss may correspond to a different high level objective (forecasting vs. steering). In particular, we generate algorithms which satisfy the following omniprediction definition.

Let $\mathcal{H}$ be a benchmark class of functions and $\mathcal{L}$ be a class of losses. An algorithm $\mathcal{A}$ is an $(\mathcal{L},\mathcal{H},\mathcal{R}_{\mathcal{A}}(T))$-online omnipredictor if it generates predictions p_t such that for all losses $\ell\in\mathcal{L}$,

$$\sum_{t=1}^{T}\ell(x_t,\pi_\ell(x_t,p_t),y_t) \leqslant \inf_{h\in\mathcal{H}}\sum_{t=1}^{T}\ell(x_t,h(x_t),y_t)+\mathcal{R}_{\mathcal{A}}(T). \tag{2}$$

Here, $\pi_\ell(x,p)\in\arg\min_{\hat{y}} p\cdot\ell(x,\hat{y},1)+(1-p)\cdot\ell(x,\hat{y},0)$ (the $\arg\min$ may not be unique) and $\mathcal{R}_{\mathcal{A}}:\mathbb{N}\to\mathbb{R}_{\geqslant 0}$ is $o(T)$. We refer to $\mathcal{R}_{\mathcal{A}}$ as the regret bound for the algorithm $\mathcal{A}$. Since it is sublinear in T, if we divide through by T, an online omnipredictor is guaranteed to achieve Equation (1) not just for a specific loss, but for any loss $\ell\in\mathcal{L}$.

Conceptually, our technical approach for online omniprediction is most closely related to the work by (Gopalan et al., 2023) which illustrates a connection between outcome indistinguishability and omniprediction in the batch setting. They show how given a set of losses $\mathcal{L}$ and a function class $\mathcal{H}$, one can construct a class of distinguishers $\mathcal{F}$ (that depends on $\mathcal{L}$ and $\mathcal{H}$) such that any predictor that is indistinguishable with respect to $\mathcal{F}$ is also a $(\mathcal{L},\mathcal{H})$-omnipredictor. Therefore, omniprediction reduces to outcome indistinguishability.

We prove a similar reduction in the online setting. Moreover, we illustrate how one can leverage the Any Kernel algorithm and RKHS machinery we developed previously in order to provably achieve the necessary indistinguishability guarantees in a computationally efficient manner. Taken together, we achieve unconditionally efficient (vanilla) online omnipredictors with $\sqrt{T}$ regret for common losses $\mathcal{L}$ and rich (infinite, real-valued) comparator classes $\mathcal{H}$. An easy consequence of this result is that by applying an online to batch conversion, we get offline omnipredictors with the optimal $\mathcal{O}(1/\varepsilon^2)$ sample complexity. This significantly improves upon the previous $\mathcal{O}(1/\varepsilon^{10})$ bounds from (Gopalan et al., 2022).[12] We now give a brief overview of the main ingredients that go into the proof of this result.

First, as in (Gopalan et al., 2023) and (Kim and Perdomo, 2023), we show that algorithms which satisfy certain decision and hypothesis outcome indistinguishability conditions (OI) are also omnipredictors.

11. Unlike previous work on omniprediction, we allow losses to depend on x. See Appendix A.1.2 for detailed discussion of this point.

12. Okoroafor et al. (2025) get similar $\mathcal{O}(1/\varepsilon^2)$ sample complexity bounds. However, their algorithms are oracle efficient and require access to a online weak agnostic learner. Ours rely on the ability efficiently evaluate a kernel. We focus on establishing end-to-end results which are unconditionally efficient for common classes of losses $\mathcal{L}$ and $\mathcal{H}$. Hu et al. (2024) also get $\mathcal{O}(1/\varepsilon^2)$ sample complexity for restricted classes classes of GLMs and Lipschitz losses.

Given a comparator class $\mathcal{H}$ and set of losses $\mathcal{L}$, we say that an algorithm $\mathcal{A}$ satisfies *online hypothesis OI* if it generates a sequence of predictions that are outcome indistinguishable with respect to the following class of functions,

$$\mathcal{F}_{HOI}(\mathcal{L},\mathcal{H}) = \{\partial\ell(x,h(x_t)) : \ell \in \mathcal{L}, h \in \mathcal{H}\} \text{ where } \partial\ell(x,\hat{y}) = \ell(x,\hat{y},1) - \ell(x,\hat{y},0). \tag{3}$$

Similarly, we say that a online algorithm satisfies online decision OI if it is outcome indistinguishable with respect to the following class of tests:

$$\mathcal{F}_{DOI}(\mathcal{L}) = \{\partial\ell(x,\pi_\ell(x)) : \ell \in \mathcal{L}\} \text{ where } \pi_\ell(x,p) = \arg\min_{\hat{y}} \mathop{\mathbb{E}}_{\tilde{y}\sim\text{Ber}(p)} \ell(x,\hat{y},\tilde{y}). \tag{4}$$

Using these definitions, we prove the following lemma.

Lemma 3 (Informal) *Let $\mathcal{L}$ be a class of loss functions and $\mathcal{H}$ be a comparator class. If $\mathcal{A}$ is online outcome indistinguishable with respect to the union of $\mathcal{F}_{DOI}(\mathcal{L})$ and $\mathcal{F}_{HOI}(\mathcal{L},\mathcal{H})$ with indistinguishability error bounded by $\mathcal{R}_\mathcal{A}$, then $\mathcal{A}$ is an online omnipredictor with regret rate $\mathcal{O}(\mathcal{R}_\mathcal{A})$.*

While it is interesting that this relationship, first identified in (Gopalan et al., 2023), carries over to the online setting, it is not quite useful without also knowing that the necessary indistinguishability requirements are also efficiently achievable. The main technical contributions of our work towards establishing online omniprediction is the design of efficiently computable kernel functions whose corresponding RKHSs contain the requisite distinguishers for hypothesis and decision OI.

We defer a detailed presentation of these constructions to Appendix C. However, the main technical ideas behind these results rely heavily on the theory behind reproducing kernel Hilbert space and the fact that it is relatively simple to compose kernel functions together. This ease of composition also allows one to characterize their corresponding (composed) function spaces. Being able to reason about composition is fundamental to these constructions since decision and hypothesis OI are both defined in terms of composition of functions (*i.e.*, $\partial\ell(x,\pi_\ell(p))$, $\partial\ell(x,h(x))$). A technical challenge of our work is showing how certain RKHS remain closed under post-processing. In particular, as a stepping stone to proving the necessary decision OI guarantees, we identify natural conditions on RKHSs $\mathcal{F}$ which guarantee that if $\ell(x,p,y)$ is in $\mathcal{F}$ then so is $\ell(x,\pi_\ell(p),y)$.

Our results can be used to guarantee regret that grows only as $\sqrt{T}$ for online omniprediction with respect to various different kinds of comparator classes $\mathcal{H}$ and losses $\mathcal{L}$ (Theorem 17 below clarifies the impact of the particular choice of $\mathcal{H}$). However, in the following theorem we instantiate this general recipe to provide an end to end guarantee for classes $\mathcal{H}$ and $\mathcal{L}$ that are commonly considered in the literature. We refer the reader to Appendix C for further examples.

Theorem 4 (Informal statement of Theorem 17) *There exist an efficient kernel k, such that the Any Kernel algorithm instantiated with kernel k is a $(\mathcal{H},\mathcal{L},\mathcal{O}(\sqrt{T}))$-online omnipredictor for the following settings,*

- *The comparator class $\mathcal{H}$ contains all low-depth regression trees taking values in $[-1,1]$ and all functions h' in a pre-specified finite set $\mathcal{H}'$.*
- *The set of losses $\mathcal{L}$ is any smooth, proper scoring rule*[13]*, loss function that is strongly convex in $\hat{y}$, or an arbitrary bounded loss ℓ' in a pre-specified finite collection $\mathcal{L}'$.*

13. Proper scoring rules ℓ are those which are optimized by reporting the true likelihood of outcome. That is, if $y \sim \text{Ber}(p)$, then p is a minimizer of this expectation, $\mathbb{E}_{y\sim\text{Ber}(p)}\ell(x,\hat{y},y)$.

In the link prediction context, one can in particular choose losses mapping onto the utility of a range of different decisions, including predictive performance (e.g., $\ell(x, \hat{y}, y) = (\hat{y} - y)^2$) and desirability of outcomes (e.g., $\ell(x, \hat{y}, y) = 1 - y$ if the goal is link formation)[14].

Loss functions may also be feature-dependent, like losses that more heavily weight decisions that affect a pair of individuals from different demographic groups or for which the induced subgraph on a pair of individuals has a certain structure (like having $c \in \mathbb{N}$ neighbors in common).

This result pushes the boundary of what is achievable in terms of online omniprediction in several ways. First, to the best of our knowledge, it is the first $\sqrt{T}$ online omniprediction guarantee which holds for comparison classes $\mathcal{H}$ that are real-valued, or of infinite size (there are infinitely many low-depth regression trees). Second, the statements are unconditional. The computational efficiency of our algorithm does not rely on the existence of an online regression oracle for the class $\mathcal{H}$.

Furthermore, we can include any function $h' : \mathcal{X} \to [-1, 1]$ in the class $\mathcal{H}$. In the context of link prediction, this implies that the algorithm can compete with any bespoke comparison function that a platform may already be using (*e.g.*, deep network). Furthermore, as we mentioned previously, these results hold even for the performative case where the outcomes y_t depend the near-deterministic distribution Δ_t from which the predictions are sampled from. For the reader familiar with the performative prediction literature, this guarantee is best understood as a novel form of online performative *stability*. It does not quite imply performative *optimality* or performative omniprediction as in (Kim and Perdomo, 2023). See Appendix C.7 for more details.

Other results. As a serendipitous consequence of our investigation into kernel methods for online indistinguishability and omniprediction, we obtain algorithms for other online prediction problems. These are not directly related to the link prediction problem which is our main focus, but are of independent interest.

We design a new algorithm for online multicalibrated quantile regression. In quantile regression, outcomes y are real-valued instead of binary. Given a quantile $q \in [0, 1]$, the goal is output a prediction p such that $y \in \mathbb{R}$ is less than $p \in \mathbb{R}$ exactly a q fraction of the time. In the batch setting where $(x, y) \sim \mathcal{D}$, one aims to find a predictor h that minimizes the error:

$$|\Pr_{(x,y)\sim\mathcal{D}}[y \leqslant h(x)] - q|.$$

Quantile regression is a common problem in domains like weather forecasting or financial prediction, where one is interested in deriving confidence intervals or predicting the likely range of outcomes, rather than the average outcome. In Appendix D.1, we introduce a new online algorithm, the Quantile Any Kernel algorithm, which satisfies the following guarantee for the online setting where "Real Life" draws (real-valued) outcomes $y_t \sim o_t$ from a different distribution o_t at every time step:

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t, y_t\sim o_t} [(1\{y_t \leqslant p_t\} - q) f(x_t, p_t)] \leqslant \|f\|\sqrt{T} \text{ for all } f \in \mathcal{F}$$

Like the Any Kernel algorithm, the Quantile Any Kernel algorithm works for any RKHS $\mathcal{F}$ and runs in polynomial time whenever the associated kernel k is efficiently computable. Furthermore, using our previous results relating kernels k to their corresponding RKHSs $\mathcal{F}_k$, one can instantiate the algorithm to guarantee online quantile multicalibration with respect to common real-valued functions $\mathcal{F}$. These results complement those in (Garg et al., 2024) and (Roth, 2022) since the functions f can now be real-valued,

14. Losses like $1 - y$ make sense in settings where the learner's predictions $\hat{y}$ actively change the likelihood of the outcome y (for instance, by influencing the platforms recommendation decisions).

the set $\mathcal{F}$ can be of infinite size, and the algorithm does not depend on enumeration over $\mathcal{F}$ or access to a computational oracle.

In addition to quantiles, one can also extend the algorithm to high dimensional regression, where y is now a vector in a compact set $\mathcal{Y} \subseteq \mathbb{R}^d$ instead of a scalar in $\mathbb{R}$. In Appendix D.2, drawing on the theory of matrix valued kernels (Álvarez et al., 2012; Micchelli and Pontil, 2005), we introduce the Vector Any Kernel algorithm which satisfies the following guarantee for any vector valued RKHS $\mathcal{F} \subseteq \{\mathcal{X} \times \mathcal{Y} \to \mathcal{Y}\}$,

$$\sum_{t=1}^{T} (y_t - p_t)^\top f(x_t, p_t) \leqslant \|f\|_{\mathcal{F}} \sqrt{T}.$$

The computational efficiency of the Vector Any Kernel algorithm relies on the ability to solve a *variational inequality*. These have been the subject of intense study within the optimization literature and efficient algorithms exist for various common choices of matrix valued kernels.

Beyond these contributions, and inspired by the recent works by (Qiao and Zheng, 2024; Błasiok et al., 2023) we also initiate the study of distance to *multi*calibration in Appendix D.3 (previous work addresses distance to simple calibration) and analyze how straightforward instantiations of the Any Kernel algorithm can be used to generate predictions that satisfy small distance to multicalibration in the online setting.

Lastly, we observe in Appendix D.4 that any function class that is an RKHS with an efficient kernel also admits a weak agnostic learner (WAL), matching a result of (Gopalan et al., 2024a) through a different argument. This connection implies that any multicalibration algorithm that relied on an oracle WAL for a class $\mathcal{F}$ is unconditionally efficient for the case where $\mathcal{F}$ is an RKHS.

Acknowledgments

We would like to thank Aaron Roth for helpful comments and discussion on online algorithms and Tina Eliassi-Rad for pointers to the networking literature. This work was supported in part by Simons Foundation Grant 733782 and Cooperative Agreement CB20ADR0160001 with the United States Census Bureau. JCP was in part supported by the Harvard Center for Research of Computation and Society.

References

Rediet Abebe, Nicole Immorlica, Jon Kleinberg, Brendan Lucier, and Ali Shirali. On the effect of triadic closure on network segregation. In *ACM Conference on Economics and Computation*, 2022.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 2012.

Sinan Aral and Marshall Van Alstyne. The diversity-bandwidth trade-off. *American Journal of Sociology*, 2011.

Eshwar Ram Arunachaleswaran, Natalie Collina, Aaron Roth, and Mirah Shi. An elementary predictor obtaining 2sqrt(t) distance to calibration. *Symposium on Discrete Algorithmms*, 2025.

Aili Asikainen, Gerardo Iñiguez, Javier Ureña-Carrión, Kimmo Kaski, and Mikko Kivelä. Cumulative effects of triadic closure and homophily in social networks. *Science Advances*, 2020.

Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 2001.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1727–1740, 2023.

Lukas Bolte, Nicole Immorlica, and Matthew O Jackson. The role of referrals in immobility, inequality, and inefficiency in labor markets. *arXiv preprint arXiv:2012.15753*, 2020.

Stephen P Borgatti and Pacey C Foster. The network paradigm in organizational research: A review and typology. *Journal of Management*, 2003.

Regina S Burachik and Alfredo N Iusem. A generalized proximal point algorithm for the variational inequality problem in a hilbert space. *SIAM journal on Optimization*, 1998.

Ronald S Burt. *Toward a structural theory of action*. 1982.

Ronald S Burt. Structural holes and good ideas. *American Journal of Sociology*, 2004.

Antoni Calvo-Armengol and Matthew O Jackson. The effects of social networks on employment and inequality. *American Economic Review*, 2004.

Yair Censor, Aviv Gibali, and Simeon Reich. Extensions of korpelevich's extragradient method for the variational inequality problem in euclidean space. *Optimization*, 2012.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Symposium on Theory of Computing*, 2021.

Cynthia Dwork, Daniel Lee, Huijia Lin, and Pranay Tankala. From pseudorandomness to multi-group fairness and back. In *Conference on Learning Theory*, 2023.

Cynthia Dwork, Chris Hays, Lunjia Hu, Nicole Immorlica, and Juan C. Perdomo. Integration through recommendations (working title). 2025. Manuscript submitted for publication.

Nathan Eagle, Michael Macy, and Rob Claxton. Network diversity and economic development. *Science*, 2010.

David Easley, Jon Kleinberg, et al. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

LawrenceCraig Evans. *Measure theory and fine properties of functions*. Routledge, 2018.

Gaetano Fichera. Sul problema elastostatico di signorini con ambigue condizioni al contorno. *Atti Accad. Naz. Lincei, VIII. Ser., Rend., Cl. Sci. Fis. Mat. Nat*, 1963.

Dean P Foster and Sergiu Hart. Forecast hedging and calibration. *Journal of Political Economy*, 2021.

Dean P Foster and Sham M Kakade. Calibration via regression. In *IEEE Information Theory Workshop*, 2006.

Dean P Foster and Rakesh V Vohra. Asymptotic calibration. *Biometrika*, 1998.

Dylan Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, 2020.

Noah E Friedkin. Structural bases of interpersonal influence in groups: A longitudinal case study. *American Sociological Review*, 1993.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Subgroup robustness grows on trees: An empirical baseline investigation. *Advances in Neural Information Processing Systems*, 2022.

Sumegha Garg, Christopher Jung, Omer Reingold, and Aaron Roth. Oracle efficient online multicalibration and omniprediction. In *Symposium on Discrete Algorithms*, 2024.

Matthew Gentzkow and Jesse M. Shapiro. Ideological Segregation Online and Offline *. *The Quarterly Journal of Economics*, 2011.

Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *Innovations in Theoretical Computer Science Conference*, 2022.

Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss Minimization Through the Lens Of Outcome Indistinguishability. In *Innovations in Theoretical Computer Science Conference*, 2023.

Parikshit Gopalan, Lunjia Hu, and Guy N. Rothblum. On computationally efficient multi-class calibration. In *Conference on Learning Theory*, 2024a.

Parikshit Gopalan, Michael Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. *Advances in Neural Information Processing Systems*, 2024b.

Mark Granovetter. Economic action and social structure: The problem of embeddedness. *American journal of sociology*, 91(3):481–510, 1985.

Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 1973.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 2022.

Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M. Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. In *Innovations in Theoretical Computer Science Conference*, 2022.

William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

Moritz Hardt and Celestine Mendler-Dünner. Performative prediction: Past and future. *arXiv preprint arXiv:2310.16608*, 2023.

David Haussler et al. Convolution kernels on discrete structures. Technical report, Citeseer, 1999.

Chris Hays, Zachary Schutzman, Manish Raghavan, Erin Walk, and Philipp Zimmer. Simplistic collection and labeling practices limit the utility of benchmark datasets for twitter bot detection. In *ACM Web Conference*, 2023.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 2007.

Ursula Hébert Johnson, Michael P Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, 2018.

Lunjia Hu, Kevin Tian, and Chutong Yang. Omnipredicting single-index models with multi-index models. 2024.

Matthew O Jackson and Brian W Rogers. Meeting strangers and friends of friends: How random are social networks? *American Economic Review*, 2007.

Eaman Jahani, Samuel P. Fraiberger, Michael Bailey, and Dean Eckles. Long ties, disruptive life events, and economic prosperity. *Proceedings of the National Academy of Sciences*, 2023.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, 2018.

Michael P Kim and Juan C Perdomo. Making decisions under outcome performativity. In *Innovations in Theoretical Computer Science*, 2023.

Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

David Kinderlehrer and Guido Stampacchia. *An introduction to variational inequalities and their applications*. SIAM, 2000.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science*, 2017.

Andrei Nikolaevich Kolmogorov and Guido Castelnuovo. Sur la loi des grands nombres. *G. Bardi, tip. della R. Accad. dei Lincei*, 1929.

Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 2006.

Gueorgi Kossinets and Duncan J Watts. Origins of homophily in an evolving social network. *American Journal of Sociology*, 115(2):405–450, 2009.

Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 2020.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR, 2018.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *International Conference on Knowledge Discovery & Data Mining*, 2019.

Daniel Lee, Georgy Noarov, Mallesh M. Pai, and Aaron Roth. Online minimax multiobjective optimization: Multicalibeating and other applications. In *Neural Information Processing Systems*, 2021.

David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *International Conference on Information and Knowledge Management*, pages 556–559, 2003.

Eugene M Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 1982.

Yao Ma, Ziyi Guo, Zhaocun Ren, Jiliang Tang, and Dawei Yin. Streaming graph neural networks. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Comput. Surv.*, 2016.

Andreas Maurer and Massimiliano Pontil. Concentration inequalities under sub-gaussian and sub-exponential conditions. *Advances in Neural Information Processing Systems*, 2021.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.

Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 2005.

John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, 2021.

Ha Quang Minh. Operator-valued bochner theorem, fourier feature maps for operator-valued kernels, and vector-valued learning. *ArXiv*, 2016.

Christopher Morris, Fabrizio Frasca, Nadav Dym, Haggai Maron, Ismail Ilkan Ceylan, Ron Levie, Derek Lim, Michael M. Bronstein, Martin Grohe, and Stefanie Jegelka. Position: Future directions in the theory of graph machine learning. In *Forty-first International Conference on Machine Learning*, 2024.

Assaf Naor. On the banach-space-valued azuma inequality and small-set isoperimetry of alon–roichman graphs. *Combinatorics, Probability and Computing*, 2012.

Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. *arXiv preprint arXiv:2310.17651*, 2023.

Muhammad Aslam Noor. General variational inequalities. *Applied Mathematics Letters*, 1(2):119–122, 1988.

Ryan O'Donnell. Analysis of boolean functions. *arXiv preprint arXiv:2105.10386*, 2021.

Chika O Okafor. Social networks as a mechanism for discrimination. *arXiv preprint arXiv:2006.15988*, 2020.

Princewill Okoroafor, Robert Kleinberg, and Michael P. Kim. Near-optimal algorithms for omniprediction. 2025. Available at https://pokoroafor.github.io/assets/pdf/nearoptimalomniprediction.pdf.

Vern I Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*. Cambridge University Press.

Juan Perdomo, Tijana Zrnic, Celestine Mendler Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, 2020.

Juan Carlos Perdomo Silva. *Performative Prediction: Theory and Practice*. PhD thesis, UC Berkeley, 2023.

Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. Fair kernel learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017.

Adrian Perez-Suay, Paula Gordaliza, Jean-Michel Loubes, Dino Sejdinovic, and Gustau Camps-Valls. Fair kernel regression through cross-covariance operators. *Transactions on Machine Learning Research*, 2023.

Mingda Qiao and Letian Zheng. On the distance from calibration in sequential prediction. *arXiv preprint arXiv:2402.07458*, 2024.

Karthik Rajkumar, Guillaume Saint-Jacques, Iavor Bojinov, Erik Brynjolfsson, and Sinan Aral. A causal test of the strength of weak ties. *Science*, 2022.

Ray Reagans and Bill McEvily. Network structure and knowledge transfer: The effects of cohesion and range. *Administrative Science Quarterly*, 2003.

Francisco Aparecido Rodrigues. Network centrality: an introduction. *A mathematical modeling approach from nonlinear dynamics to complex systems*, 2019.

M. Puck Rombach, Mason A. Porter, James H. Fowler, and Peter J. Mucha. Core-periphery structure in networks. *SIAM Journal on Applied Mathematics*, 2014.

Emanuele Rossi, Benjamin Paul Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *ArXiv*, 2020.

Aaron Roth. Uncertain: Modern topics in uncertainty estimation. *Unpublished Lecture Notes*, 2022.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019.

Glenn Shafer and Vladimir Vovk. *Probability and finance: it's only a game!*, volume 491. John Wiley & Sons, 2005.

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

Georg Simmel. *Soziologie*. Duncker & Humblot Leipzig, 1908.

Ingo Steinwart. *Support Vector Machines*. Springer, 2008.

Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *World Wide Web Conference*, 2018.

Zilong Tan, Samuel Yeom, Matt Fredrikson, and Ameet Talwalkar. Learning fair representations for kernel models. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.

Johan Ugander, Lars Backstrom, Cameron Marlow, and Jon Kleinberg. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 2012.

Lois M Verbrugge. The structure of adult friendship choices. *Social Forces*, 1977.

Vladimir Vovk. Non-asymptotic calibration and resolution. *Theoretical Computer Science*, 2007.

Vladimir Vovk, Ilia Nouretdinov, Akimichi Takemura, and Glenn Shafer. Defensive forecasting for linear protocols. In *Conference on Algorithmic Learning Theory*, 2005.

Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 2001.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Conference on Neural Information Processing Systems*, 2023.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in Neural Information Processing Systems*, 2019.

Dan Zeltzer. Gender homophily in referral networks: Consequences for the medicare physician earnings gap. *American Economic Journal: Applied Economics*, 2020.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Revisiting graph neural networks for link prediction. 2020.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 2020.

Appendix A. The Link Prediction Problem

Data. We represent a professional network as a graph G_t consisting of nodes (people) and edges (connections between people) that evolve over time. Each node i is associated with a features $z_{i,t}$ containing information that pertains specifically to i, such as their employment and demographic information. This can vary over time. In addition to this node-level information, the graph G_t is defined by a set of undirected edges detailing which individuals are connected at time t. Edges can be added to or removed from the graph arbitrarily at every time step and need not follow any predefined dynamic or process such as triadic closure (Simmel, 1908). The underlying set of nodes can also change. The only restriction we will make is that the platform has the ability to *observe* the entire graph G_t as it evolves over time.[15]

Prediction protocol. At every time step t, the platform is presented with a pair of individuals $a_t = (i, j)$ and generates a prediction p_t regarding the likelihood that i and j will be connected at the next time step (i and j may or may not be connected at time t). After producing the prediction, the platform then observes a binary outcome y_t, which is 1 if i and j are connected at time $t+1$ and 0 otherwise. As per our earlier observability comment, the platform observes the outcome y_t before having to make a prediction at time $t+1$. Variants of this prediction problem were proposed as early as 2003 (Liben-Nowell and Kleinberg, 2003).

In our setting, we allow the outcome y_t to also depend on the distribution Δ_t where p_t is drawn from.[16] That is, predictions can be *performative* (Perdomo et al., 2020) and influence the likelihood of the outcome. This dynamic naturally occurs whenever the platform uses predictions to inform recommendations. For instance, a platform such as LinkedIn may opt to recommend that a pair of individuals connect via the "People You May Know" panel if p_t is above some threshold. Forecasts in this setting are hence likely to be self-fulfilling (although our results hold for any dynamic).

Notation. We denote by $\mathcal{Z}$ the set of possible node-level features of an individual, at any point in time. We define the graph G_t to be a set $\{(v, z_{v,t}, \Gamma_t(v))\}_{v \in V_t}$, where $v \in \mathbb{N}$ is the id of a node, $z_{v,t} \in \mathcal{Z}$ are the node-level features of v at time t, and $\Gamma_t(v) \subseteq V_t$ is the set of nodes containing v and its immediate neighbors at time t. Here, $V_t \subseteq \mathbb{N}$ is the set of nodes present in the graph at time t. We will use $\Gamma_G^{(r)}(v)$ to denote the set of nodes that are at distance at most r from v in G. If the sequence of graphs $\{G_t\}_{t=1}^T$ is clear from context, we will write $\Gamma_t(v) = \Gamma_{G_t}(v)$, and adopt the shorthands $\Gamma_G^{(1)}(v) = \Gamma_t(v)$ for v's immediate neighborhood.

Furthermore, we will (exclusively) use $\mathcal{U} = (\mathbb{N} \times \mathbb{N}) \times \mathcal{G}$ to refer to the universe of possible elements $u = (a, G)$ consisting of pairs of individuals $a = (u, v)$ and graphs $G \in \mathcal{G}$. We will use $\mathcal{X}$ to refer to a general set.

A.1. Formal desiderata.

The dynamics underlying professional networking are complex. In this paper, we address the challenge of efficiently generating forecasts that are guaranteed to be *a) valid* and *b) useful*, without imposing any modeling assumptions regarding how networks evolve.

15. While the platform has the ability to examine all of G_t, algorithms need not read the entire input G_t. They only examine the subset of G_t relevant to the distinguishers.
16. The difference between y_t depending on the distribution Δ_t versus the draw $p_t \sim \Delta_t$ is relatively neglible since in all our algorithms, Δ_t is only ever supported on 2 points which are very close together. For intuition, one can essentially assume that Nature chooses y_t while knowing p_t up to some small rounding error.

A.1.1. VALIDITY AND OUTCOME INDISTINGUISHABILITY.

Defining what it means for a forecast of arbitrary, non-repeatable events to be valid is in and of itself a challenging task. However, one common perspective within the sciences is that a theory, or prediction, is valid if it withstands efforts to falsify it. This viewpoint was recently formalized in the computer science literature by (Dwork et al., 2021) who introduced the notion of *outcome indistinguishability* (OI). Briefly, a predictor is outcome indistinguishable if no analyst can refute the validity of the predictor on the basis of a particular set of computational tests.

This idea of the analyst is operationalized via a class $\mathcal{F}_A$ of *distinguishers* that take in a set of observation information x, a prediction p, a binary outcome y, and return a score (think True/False).[17] A sequence of predictions p_t is outcome indistinguishable with respect to $\mathcal{F}_A$ if, when averaged over the sequence, all distinguishers $A \in \mathcal{F}_A$ give (approximately) the same output in the case where they are given (a) the synthetic outcome $\tilde{y}_t \sim \text{Ber}(p_t)$ sampled according to the learner's prediction p_t and (b) the true outcome y_t revealed by "Real Life". That is,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\tilde{y}\sim\text{Ber}(p_t)} A(x_t, p_t, \tilde{y}_t) \approx \frac{1}{T}\sum_{t=1}^{T} A(x_t, p_t, y_t). \tag{5}$$

In their initial work, (Dwork et al., 2021) focused on the batch, or distributional setting, where features are sampled from a fixed, *static* distribution $x \sim \mathcal{D}$, and outcomes y are sampled from some conditional distribution, $y \sim \text{Ber}(p^*(x))$. As discussed previously, networking dynamics are complex and the likelihood of a link forming between any pair of individuals changes as networks evolve. Assuming any kind of static, or slowly moving distribution over (x, y) is a non-starter for the link prediction problem.

Instead of generating predictions that are indistinguishable under a specific choice of static distribution, we tackle the challenge of (efficiently) producing predictions that are outcome indistinguishable against arbitrary sequences $\{(x_t, p_t, y_t)\}_{t=1}^{T}$. That is, "Real Life"' can choose outcomes $y_t \in \{0, 1\}$ arbitrarily, and the choice of y_t may even depend on the learners predictions. Formally, we aim to generate link predictions that satisfy the following online outcome indistinguishability guarantee:

Definition 5 *An algorithm $\mathcal{A}$ is $(\mathcal{F}, \mathcal{R}_{\mathcal{A}})$-online outcome indistinguishable if it generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^{T}$ such that for all distinguishers $f \in \mathcal{F}$*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} (y_t - p_t) f(x_t, p_t) \right| \leqslant \mathcal{R}_{\mathcal{A}}(T, f) \tag{6}$$

where the indistinguishability error rate $\mathcal{R}_{\mathcal{A}} : \mathbb{N} \times \mathcal{F} \to \mathbb{R}_{\geqslant 0}$ is $o(T)$ for every f.

Although stated differently, the condition above is essentially equivalent to that presented in Equation (5) since,

$$A(x_t, p_t, y_t) - \mathop{\mathbb{E}}_{\tilde{y}_t\sim\text{Ber}(p_t)} A(x_t, p_t, \tilde{y}_t) = (y_t - p_t)(A(x_t, p_t, 1) - A(x_t, p_t, 0)) = (y_t - p_t) f_A(x_t, p_t),$$

for $f_A(x, p) = A(x, p, 1) - A(x, p, 0)$. Therefore,

$$\left| \lim_{T\to\infty} \frac{1}{T}\sum_{t=1} \mathop{\mathbb{E}}_{\tilde{y}\sim\text{Ber}(p_t)} A(x_t, p_t, \tilde{y}_t) - A(x_t, p_t, y_t) \right| = 0 \iff \left| \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} (y_t - p_t) f_A(x_t, p_t) \right| = 0.$$

17. This corresponds sample-access OI, the second level in the OI hierarchy presented in (Dwork et al., 2021). For ease of presentation, we assume that all distinguishers A are deterministic.

Although initially defined with respects functions f that are binary valued — where f was the characteristic function of a set or demographic group (Hébert Johnson et al., 2018) — the distinction between binary and real-valued functions has since been blurred in the multicalibration literature. In this work, we keep to earlier conventions and refer to the above guarantee (Equation (6)) as indistinguishability since we focus mostly on real-valued f and because we work with a formulation of omniprediction that is expressed in terms of outcome indistinguishability (Gopalan et al., 2023). However, we do so with the understanding that both terms are very tightly linked.

Returning to the intuition that predictions will be regarded as valid (for now!) if they cannot be falsified, we note that predictions satisfying Equation (6) with $\mathcal{R}_{\mathcal{A}}(T, f) = \mathcal{O}(\sqrt{T})$ cannot be refuted on the basis of a common class of tests based on the theory of martingales. To see this, assume that the outcomes y_t are the realizations of a stochastic process $(Y_t)_{t=1}^T$ where the binary random variables Y_t are not necessarily independent nor identically distributed, but satisfy $\mathbb{E}\, Y_t = p_t^*$. Then, it's not hard to check that $Z_t = \sum_{t=1}^T Y_t - p_t^*$ is a martingale with bounded differences. By Azuma-Hoeffding, the best one can guarantee on the deviations $|\sum_{t=1}^T y_t - p_t^*|$ is that they scale at $\mathcal{O}(\sqrt{T})$ rates. Therefore, a sequence of predictions $(p_t)_{t=1}^T$ that are OI with respect to the constant function $f = 1$ and satisfy $|\sum_{t=1}^T Y_t - p_t| \leqslant \mathcal{O}(\sqrt{T})$ behave *as if* they were the true sequence $(p_t^*)_{t=1}^T$ that generate the data. We cannot refute them on the basis of these martingale tests.

The above online OI guarantee is stronger, it holds not just on average over the sequence but even with respect to distinguishers that also examine information present in x_t and the prediction p_t itself. We will develop link prediction algorithms that fool distinguishers which examine a wide variety of information about the pair of individuals including their node-level features, their mutual connections, and the features of people to whom they are connected.

A.1.2. UTILITY AND OMNIPREDICTION.

In addition to the notion of empirical validity above, we aim to generate predictions that are *useful* for decision-making. We will thus move beyond analysis of predictions p_t and consider *decisions* $\hat{y}_t$ made on the basis of a prediction p_t and the relevant context x_t.

We will also assume that decision-makers' utilities can be specified by a (class of) *loss function*(s). For example, decision-makers may want to forecast outcomes, so that predictions closely match outcomes, or steer them, so that desirable outcomes occur more often. In such cases, a loss function will encode some notion of distance between predictions and outcomes. Or, it might simply produce higher outputs when outcomes are undesirable and lower outputs when they they are desirable. As we noted previously, our "platform" setting allows for performativity, meaning that outcomes y can depend on decisions $\hat{y}$ — this is the power of the platform that we wish to exploit and what gives us hope that the latter goal of steering subjects towards desirable outcomes may be attainable.

We will focus on minimizing loss with respect to the best fixed action in retrospect: An algorithm $\mathcal{A}$ generating a transcript of (feature, decision, outcomes) tuples $\{x_t, \hat{y}_t, y_t\}_{t=1}^T$ achieves $\mathcal{R}_{\mathcal{A}}(T)$ regret with respect to a comparison, or benchmark, class of functions $\mathcal{H}$ and loss ℓ if

$$\sum_{t=1}^{T} \ell(x_t, \hat{y}_t, y_t) \leqslant \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(x_t, h(x_t), y_t) + \mathcal{R}_{\mathcal{A}}(T).$$

In the equation above, we note that loss functions can depend on *features* x_t as well as predicted and realized outcomes. This is because many loss minimization settings in complex domains depend on the object we are making predictions about as well as on the prediction and realized outcome. For example, one may wish to more heavily weight decisions that affect disadvantaged demographic groups, in which

case the loss function will depend on the features of individuals. However, one can always drop the x argument to ℓ for losses that do not depend on features (as in in prior work on omniprediction (Gopalan et al., 2023; Garg et al., 2024)).

In link prediction, a platform may want to determine which links are likely to form or make recommendations that nudge certain links towards forming. The utility of a decision in an evolving network may also depend on characteristics of the decision subjects, such as the demographic group membership of the pair of individuals across a potential connection. We allow for loss functions that take into account characteristics of pairs of individuals (and also their neighborhoods and neighbors' features).

Finally, we will focus on creating predictors that can be efficiently post-processed so as to minimize loss, with respect to a given comparator class, for any in *large classes of loss functions*. These are called omnipredictors (Gopalan et al., 2022; Gupta et al., 2022). Online omnipredictors can be defined formally as follows.

Definition 6 *An algorithm $\mathcal{A}$ is an $(\mathcal{L}, \mathcal{H}, \mathcal{R}_{\mathcal{A}})$-online omnipredictor if it generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^{T}$ such that for all $\ell \in \mathcal{L}$ there exists a $\pi_\ell : \mathcal{X} \times [0,1] \to [0,1]$ such that*

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} \ell(x_t, \pi_\ell(x_t, p_t), y_t) \leqslant \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(x_t, h(x_t), y_t) + \mathcal{R}_{\mathcal{A}}(T). \tag{7}$$

where $\mathcal{R}_{\mathcal{A}} : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ is $o(T)$.

In particular, we will take π_ℓ to be

$$\begin{aligned} \pi_\ell(x, p) &\in \operatorname*{arg\,min}_{\hat{y} \in [0,1]} \mathop{\mathbb{E}}_{y \sim \mathrm{Ber}(p)} [\ell(x, \hat{y}, y)], \\ &= \operatorname*{arg\,min}_{\hat{y} \in [0,1]} p \cdot \ell(x, \hat{y}, 1) + (1-p) \cdot \ell(x, \hat{y}, 0), \end{aligned}$$

which is a simple optimization problem over the unit interval that can be efficiently solved. (We will assume argmin returns the set of values achiving a minimum, and that π_ℓ is an arbitrary member of this set.) Finally if ℓ is invariant to x, the x argument to π_ℓ can also be dropped.

We focus on omnipredictors for two reasons. First, link predictions may be used for a variety of downstream decisions on a platform. As mentioned previously, a class of loss functions can simultaneously be used to measure *predictive quality* (*e.g.*, squared loss: $\ell(x, \hat{y}, y) = (y - \hat{y})^2$) or *desirability of outcomes* (*e.g.*, link formation: $\ell(x, \hat{y}, y) = 1 - y$, which is minimized when an edge forms). Additionally, platforms may use link predictions within different "People You May Know" recommendations serving different goals (*e.g.*, different types of connections), and they may hope to tailor other on-platform experiences on the basis of the predicted evolution of the network. Second, the loss function may not be known at prediction time: for example, a predictive system may need to be fixed in advance of A/B tests determining which loss function in a certain class gives the best proxy for some long-term objective.

In Appendix C, we discuss learning algorithms which are omnipredictors with respect to large classes of losses (*e.g.*, all bounded differentiable loss functions) and with expressive comparator classes, like deep neural nets.

Appendix B. Online Outcome Indistinguishability and Applications to Link Prediction

In this section, we consider the first task detailed in Appendix A.1 of generating link predictions for an evolving network that satisfy the following outcome indistinguishability guarantee:

$$\sum_{t=1}^{T}(p_t - y_t)f(x_t, p_t) \leqslant o(T) \text{ for all } f \in \mathcal{F}.$$

We are specifically interested in designing online algorithms that are (a) computationally-efficient, (b) indistinguishable with respect to rich classes of functions $\mathcal{F}$ defined on complex, graph-based domains $\mathcal{U}$, and (c) achieve the optimal $\mathcal{O}(\sqrt{T})$ outcome indistinguishability error, henceforth *OI error*.

We present a more detailed comparison to prior work later on. However, briefly, previous online algorithms for this problem which achieved the optimal $\sqrt{T}$ OI error bound were either computationally inefficient for super polynomially sized sets $\mathcal{F}$ (Foster and Kakade, 2006; Gupta et al., 2022), could only achieve the above guarantee for restricted classes of functions f that were continuous in the forecast p (Vovk, 2007), or which where binary valued (Gupta et al., 2022). Our algorithm overcomes these issues and achieves all three of the above desiderata. This will enable new possibilities for *omniprediction* as we detail in Appendix C, accomplished by appropriate choice of the kernel function, folding the benchmark functions into the corresponding RKHS $\mathcal{F}$.

Technical approach. We develop new, general-purpose algorithms guaranteeing online outcome indistinguishability and then specialize them to the link prediction setting. In particular, we focus on developing algorithms which guarantee calibration with respect to sets $\mathcal{F}$ that form a *reproducing kernel Hilbert space* (RKHS). Intuitively, an RKHS is a set of functions $\mathcal{F} \subseteq \{\mathcal{X} \to \mathbb{R}\}$ that are implicitly represented by a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, for a universe $\mathcal{X}$.

This kernel based viewpoint is useful for our link prediction problem because it provides a computationally efficient way to guarantee calibration with respect to rich classes of functions defined on graphs. Building on the theory of RKHSs, we design computationally efficient kernels that guarantee indistinguishability with respect to classes of distinguishers that take into account graph topology (*e.g.*, number of mutual connections, isomorphism class of the local neighborhoods), or functions computable by arbitrary finite sets of pre-specified functions, like graph neural network link predictors.

Our technical approach is directly builds on a result by Vovk (Vovk, 2007) that is in turn inspired by the breakthrough work of (Foster and Vohra, 1998). In his paper, which predates the definition of multicalibration by (Hébert Johnson et al., 2018) or OI (Dwork et al., 2021), Vovk introduces an algorithm that guarantees indistinguishability with respect to any RKHS of functions $f(u, p)$ that are continuous in p. Drawing on ideas from (Foster and Hart, 2021), we introduce the Any Kernel algorithm, which guarantees indistiguishability with respect to *any* RKHS $\mathcal{F}$, not just those that are continuous in p.

B.1. The algorithm.

We now formally present our online Any Kernel algorithm, which forms the backbone of our later results. The algorithm builds on the earlier K29* algorithm from (Vovk, 2007) that is in turn inspired by Kolmogorov's 1929 proof of the weak law of large numbers (Kolmogorov and Castelnuovo, 1929). The reader familiar with reproducing kernel Hilbert spaces can skip the brief background highlights outlined below.

Background on reproducing kernel Hilbert spaces. Our guarantees are stated in terms of a kernel k and its associated reproducing kernel Hilbert space $\mathcal{F}_k$. We drop the subscript when it is clear from context. We briefly review the basic facts behind RKHSs here and provide a self-contained formal review

of the facts we need. In Appendix E, we list out various kernels and RKHS that we then use to instantiate the algorithm. We refer the reader to texts such as (Paulsen and Raghupathi; Steinwart, 2008) for further background on this material.

Definition 7 *Let $\mathcal{X}$ be an arbitrary set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a* kernel *on $\mathcal{X}$ if it satisfies*

1. *Symmetry: $k(x, x') = k(x', x)$ for all $x, x' \in \mathcal{X}$.*
2. *Positive Definiteness: $\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j k(x_i, x_j) \geqslant 0$ for all $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathcal{X}$ and $\lambda \in \mathbb{R}^n$.*

Every kernel k is associated with a unique Hilbert space $\mathcal{F} \subseteq \{\mathcal{X} \to \mathbb{R}\}$ of real-valued functions. By virtue of being a Hilbert space, $\mathcal{F}$ is equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ that defines a norm on the elements $f \in \mathcal{F}$, $\|f\|_{\mathcal{F}}^2 = \langle f, f \rangle_{\mathcal{F}}$. The set is called a *reproducing* kernel Hilbert space since for every element $x \in \mathcal{X}$, there exists an element $\Phi(x) \in \mathcal{F}$ such that

$$f(x) = \langle f, \Phi(x) \rangle_{\mathcal{F}} \text{ for all } f \in \mathcal{F},$$

where $\langle \cdot, \Phi(x) \rangle_{\mathcal{F}}$ is continuous. The function $\Phi : \mathcal{X} \to \mathcal{F}$ is called the reproducing kernel or feature map. It also satisfies the property that for all $x, x' \in \mathcal{X}$,

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}.$$

Given any kernel k, or equivalently a feature map Φ, the Moore-Aronszajn theorem provides an explicit characterization of the set of functions $\mathcal{F}$. In particular,

$$\mathcal{F} = \overline{\mathsf{span}}\{\Phi(x) : x \in \mathcal{X}\},$$

where,

$$\mathsf{span}\{\Phi(x) : x \in \mathcal{X}\} = \left\{ f \ : \ f = \sum_{i=1}^{n} \lambda_i \Phi(x_i) \text{ for all } n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{U} \text{ and } \lambda \in \mathbb{R}^n \right\},$$

and the overline denotes the completion of the set. That is, $\mathcal{F}$ is the set of all finite linear combinations of feature maps Φ augmented with the limits of any Cauchy sequences of such linear combinations.

Throughout our work we will use the fact that kernels *compose*. That is, if k_1 and k_2 are kernels for RKHSs $\mathcal{F}_1 \subseteq \{\mathcal{X}_1 \to \mathbb{R}\}$ and $\mathcal{F}_2 \subseteq \{\mathcal{X}_2 \to \mathbb{R}\}$. Then $k_1 + k_2$ is a kernel for $\mathcal{F}_1 + \mathcal{F}_2$ and $k_1 \cdot k_2$ is a kernel for $\mathcal{F}_1 \cdot \mathcal{F}_2$ where,

$$\begin{aligned} \mathcal{F}_1 + \mathcal{F}_2 &\subseteq \{f_1(x_1) + f_2(x_2) \ : \ x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}, \quad \text{and} \\ \mathcal{F}_1 \cdot \mathcal{F}_2 &\subseteq \{f_1(x_1) f_2(x_2) \ : \ x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}. \end{aligned}$$

A direct implication of the first line is that two different RKHSs on the same domain can be combined to make a new one, where the set of functions in the RKHS contains the union of functions in each of the RKHSs. Further details are deferred to Theorem 56 and Theorem 57. However, the key point is that these composition properties make it easy to "mix and match" various indistinguishability guarantees.

Description of algorithm. The algorithm is at a high-level very simple. It only takes as input a kernel function k,

$$k : (\mathcal{X} \times [0,1]) \times (\mathcal{X} \times [0,1]) \to \mathbb{R}.$$

At every round t, it constructs a function $S_t : [0,1] \to \mathbb{R}$ defined from the history $\{(x_i, p_i, y_i)\}_{j=1}^{t-1}$. If the kernel is continuous, it chooses a prediction p_t that is a zero of S_t, $S_t(p_t) \approx 0$. If the kernel k is discontinuous in p, it instead finds two points q_1 and q_2 which are very close together (i.e., $|q_1 - q_2| \approx 0$) and outputs a distribution Δ_t supported on q_1, q_2 such that the expectation of S_t over Δ_t is approximately 0. Both of these search problems are efficiently solved via binary search. The algorithm in which the kernel k is continuous is the same as in Vovk's K29* algorithm, while the discontinuous case is new. In particular, the procedure in the discontinuous case draws on ideas from (Foster and Hart, 2021) and their results on near deterministic calibration.

Guarantees of algorithm. With these preliminaries out of the way, we now state the main guarantees of the theorem.

Theorem 8 *Let k be a kernel with associated RKHS $\mathcal{F}$. Then, the Any Kernel algorithm (Figure 1) instantiated with kernel k generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^{T}$ such that for any $f \in \mathcal{F}$:*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} f(x_t, p_t)(y_t - p_t) \right| \leqslant \|f\|_{\mathcal{F}} \sqrt{1 + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} p_t(1-p_t) k((x_t, p_t), (x_t, p_t))}.$$

If k is forecast-continuous, then the guarantee is deterministic since Δ_t is a point mass. Otherwise, it is near-deterministic. The distribution Δ_t is supported on points that are $\mathcal{O}(t^{-3})$ apart.[18] *If the kernel is bounded by B,*

$$\sup_{(x,p) \in \mathcal{X} \times [0,1]} k((x,p),(x,p)) \leqslant B,$$

then the per round runtime of the algorithm is bounded by $\mathcal{O}(t \cdot \log(tB) \cdot \mathsf{time}(k))$, where $\mathsf{time}(k)$ is a uniform upper bound on the runtime of computing the kernel function k.

Proof If $\operatorname{sign} S_t(0) = \operatorname{sign} S_t(1) \neq 0$ in round t, selecting $p_t = (1 + \operatorname{sign} S_t(0))/2$ guarantees that,

$$S_t(p_t)(y_t - p_t) \leqslant 0,$$

regardless of whether y_t is 1 or 0. Otherwise, $p_t \sim \Delta_t$ where Δ_t places probability τ on q_t and $1 - \tau$ on q_t'. In this case, letting $\tau' = 1 - \tau$, we can write:

$$\begin{aligned} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} [S_t(p_t)(y_t - p_t)] &= \tau S_t(q_t)(y_t - q_t) + (1-\tau) S_t(q_t')(y_t - q_t') \\ &= [\tau S_t(q_t) + \tau' S_t(q_t')](y_t - q_t') + \tau S(q_t)(q_t' - q_t) \end{aligned}$$

By choice of $\tau = |S_t(q_t')|/(|S_t(q_t)| + |S_t(q_t')|)$, and the fact that $S_t(q_t')$ and $S_t(q_t)$ have opposite signs, the term inside the brackets is equal to 0 (this is the forecast hedging idea from (Foster and Hart, 2021)). Summarizing, we have that:

$$\mathop{\mathbb{E}}_{p_t \sim \Delta_t} [S_t(p_t)(y_t - p_t)] = \tau S_t(q_t)(q_t' - q_t) \leqslant |S_t(q_t)||q_t - q_t'| \leqslant |q_t - q_t'| \cdot t \cdot \max_{t' \leqslant t} k((x_t, p_t), (x_t, p_t)).$$

18. One could change this from $\mathcal{O}(t^{-3})$ to $\mathcal{O}(t^{-\alpha})$ for any $\alpha > 3$ without changing the asymptotic runtime.

Since $|q_t - q'_t| \leqslant \varepsilon_t = 1/(10B_t t^3)$ where $B_t = \max_{t' \leqslant t} k((x_t, p_t), (x_t, p_t))$, we conclude that regardless of whether y_t is 0 or 1,

$$\mathop{\mathbb{E}}_{p_t \sim \Delta_t}[S_t(p_t)(y_t - p_t)] \leqslant \frac{1}{10t^2}. \tag{8}$$

We now seek an upper bound on the expected value of

$$\left\| \sum_{t=1}^{T} (y_t - p_t)\Phi(x_t, p_t) \right\|_{\mathcal{F}}^2 = \sum_{t=1}^{T} \sum_{s=1}^{T} (y_t - p_t)(y_s - p_s)\langle \Phi(x_t, p_t), \Phi(x_s, p_s) \rangle_{\mathcal{F}}.$$

To this end, first observe the symmetry of the summands in (s, t), so the right side simplifies to

$$\sum_{t=1}^{T} (y_t - p_t)^2 \|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 + 2 \sum_{t=1}^{T} (y_t - p_t) \left(\sum_{s=1}^{t-1} k((x_t, p_t), (x_s, p_s))(y_s - p_s) \right).$$

Next, we apply the identity $(y_t - p_t)^2 = p_t(1 - p_t) + (1 - 2p_t)(y_t - p_t)$, which holds for all $y_t \in \{0, 1\}$ and $p_t \in [0, 1]$ and rewrite the above expression as:

$$\sum_{t=1}^{T} p_t(1 - p_t)\|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 + 2 \sum_{t=1}^{T} (y_t - p_t) \left(\sum_{s=1}^{t-1} k((x_t, p_t), (x_s, p_s))(y_s - p_s) + \|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 (1 - 2p_t) \right).$$

Since the rightmost parenthesized term is, by definition, precisely $S_t(p_t)$, we have shown that

$$\mathbb{E} \left\| \sum_{t=1}^{T} (y_t - p_t)\Phi(x_t, p_t) \right\|_{\mathcal{F}}^2 = \mathbb{E} \left[\sum_{t=1}^{T} p_t(1 - p_t)\|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 \right] + 2 \sum_{t=1}^{T} \mathbb{E}[S_t(p_t)(y_t - p_t)].$$

Now, using our earlier result (Eq. (8)), we conclude that:

$$\begin{aligned} \mathbb{E} \left\| \sum_{t=1}^{T} (y_t - p_t)\Phi(x_t, p_t) \right\|_{\mathcal{F}}^2 &\leqslant \mathbb{E} \left[\sum_{t=1}^{T} p_t(1 - p_t)\|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 \right] + 2 \sum_{t=1}^{T} \frac{1}{10t^2} \\ &\leqslant \mathbb{E} \left[\sum_{t=1}^{T} p_t(1 - p_t)\|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 \right] + \frac{2}{10} \cdot \frac{\pi^2}{6} \end{aligned}$$

where we used the fact that $\sum_{t=1}^{\infty} t^{-2} = \pi^2/6$. Noting that

$$p_t(1 - p_t)\|\Phi(x_t, p_t)\|_{\mathcal{F}}^2 = p_t(1 - p_t)k((x_t, p_t), (x_t, p_t)),$$

and applying Jensen's inequality, the above equation implies that:

$$\mathbb{E} \left\| \sum_{t=1}^{T} (y_t - p_t)\Phi(x_t, p_t) \right\|_{\mathcal{F}} \leqslant \sqrt{1 + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} p_t(1 - p_t)k((x_t, p_t), (x_t, p_t))}. \tag{9}$$

To conclude the proof, we use the reproducing property $f(x,p) = \langle f, \Phi(x,p)\rangle_{\mathcal{F}}$, which, along with Cauchy-Schwarz, relates the indistinguishability error to the above expression as follows:

$$\begin{aligned}\left|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t}(y_t - p_t)f(x_t,p_t)\right| &= \left|\mathop{\mathbb{E}}_{p_t\sim\Delta_t}\left[\langle f, \sum_{t=1}^{T}(y_t-p_t)\Phi(p_t,x_t)\rangle_{\mathcal{F}}\right]\right| \\ &\leqslant \|f\|_{\mathcal{F}}\,\mathbb{E}\left\|\sum_{t=1}^{T}(y_t-p_t)\Phi(x_t,p_t)\right\|_{\mathcal{F}}.\end{aligned}$$

■

Discussion. The bound guarantees non-asymptotic OI error of at most $\sqrt{T}$ for all functions f that lie in the RKHS $\mathcal{F}$ induced by a pre-specified kernel k. [19] While the bound holds for all functions in the RKHS, it is *adaptive*. For each f, it depends on the norm $\|f\|_{\mathcal{F}}$ but not on the number of functions $|\mathcal{F}|$ (which is in fact infinite for every choice of kernel k). The norm of a function in an RKHS can often be interpreted as an instance-specific notion of complexity. Consequently, the OI error bound satisfies the intuitive property that it is smaller for simple functions, and larger for more complicated functions.

The guarantees are also adaptive since they depend on norms of the features in the sequence, $k((x_t,p_t),(x_t,p_t)) = \|\Phi(x_t,p_t)\|^2_{\mathcal{F}}$, and the variance of the predictions $p_t(1-p_t)$. Adapting to the variance is particularly useful in the link prediction setting since we expect most edges in professional networks to be unlikely to form, meaning that the OI error bound is smaller.

We also note that neither the run-time of the algorithm nor the associated regret bounds have any explicit dependence on number of functions $|\mathcal{F}|$. Both of these properties are determined by the kernel function k.

In the following propositions, we instantiate the theorem above with specific choices of kernel functions k, illustrating how it can be used to guarantee indistinguishability with respect to interesting classes of functions $\mathcal{F}$. We then compare our results to previous work.

We will use multi-index notation to denote $x_S = \prod_{i\in S} x_i$ for $S \subseteq [n]$. Informally, Theorem 9 states that the algorithm guarantees outcome indistinguishability at $\sqrt{T}$ rates with respect to tests that are the product of a low-degree function on $\mathcal{X} \subseteq \{0,1\}^n$ and either binned functions or functions satifying mild smoothness conditions of the prediction p.

Corollary 9 (Low-degree functions on $\{0,1\}^n$) *Let $\mathcal{F}_{\text{LowDeg}} \subseteq \{\{-1,1\}^n \to [-1,1]\}$ be a set of Boolean functions whose Fourier spectrum is supported on monomials of degree at most d (e.g., decision trees of depth d, or polynomials).*[20]

$$\mathcal{F}_{\text{LowDeg}} = \left\{ f \;:\; \exists\ \alpha \text{ such that } \|\alpha\|_\infty \leqslant 1, f(x) = \sum_{S\subset[n],|S|\leqslant d} \alpha_S x_S, \forall x \in \{0,1\}^n \right\}.$$

Furthermore, let $\mathcal{F}_{\text{Cts}} \subseteq \{[0,1] \to [-1,1]\}$ be the class of continuous, differentiable functions with derivative uniformly bounded in $[-1,1]$ and $\mathcal{F}_{\text{Grid}}$ to be the set of functions

$$f_r(p) = 1\left\{\frac{r-1}{N} \leqslant p < \frac{r}{N}\right\}$$

19. In particular, the bound holds for all values of T.

20. Recall that Boolean functions over $\{-1,1\}^n$ can always be written as polynomials, and that the Fourier spectrum of functions on $\{-1,1\}^n$ are simply the coefficients of monomials in the polynomial. See Example 3 for more discussion of functions on the Boolean hypercube.

parametrized by some positive integer N and $r \in \{1, \ldots, N-1\}$. We also define $f_N(p) = 1\{(N-1)/N \leqslant p \leqslant 1\}$ so the grid covers the whole interval. Then, the Any Kernel algorithm run on the kernel

$$k((x,p),(x',p')) \stackrel{\text{def}}{=} \left(\frac{(e^{\min\{p,p'\}} + e^{-\min\{p,p'\}})(e^{1-\max\{p,p'\}} + e^{\max\{p,p'\}-1})}{2(e-e^{-1})} + 1\{\exists\, r \in [N] \,:\, f_r(p) = f_r(p') = 1\} \right) \sum_{S \subset [n], |S| \leqslant d} x_S x'_S,$$

generates a sequence of predictions such that for all $f_x \in \mathcal{F}_{\text{LowDeg}}$ and $f_p \in \mathcal{F}_{\text{Cts}} \cup \mathcal{F}_{\text{Grid}}$:

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} f_x(x_t) f_p(p_t)(y_t - p_t) \right| \leqslant 6\sqrt{n^d T}.$$

Proof From Example 6, we have that $\mathcal{F}_{\text{LowDeg}}$ is the RKHS induced by the kernel

$$\begin{aligned} k_{\text{LowDeg}}(x,x') &= \sum_{S \subset [n], |S| \leqslant d} x_S x'_S \\ &= \sum_{k=1}^{d} \binom{n}{k} < d\left(\frac{ne}{d}\right)^d < 4n^d, \end{aligned}$$

since $x_S^2 \leqslant 1$. Also, from the example, for $f \in \mathcal{F}_{\text{LowDeg}}$, the norm of f is the ℓ^2 norm of the coefficients α, which is bounded by 1 by assumption: $\|f\|_{\mathcal{F}_{\text{LowDeg}}} \leqslant 1$.

Next, from Example 5 (Berlinet and Thomas-Agnan, 2011), note that $\mathcal{F}_{\text{Cts}}$ is in the Sobolev space $W^{1,2}([0,1])$ associated with the kernel,

$$k_{\text{Cts}}(p,p') = \frac{(e^{\min\{p,p'\}} + e^{-\min\{p,p'\}})(e^{1-\max\{p,p'\}} + e^{\max\{p,p'\}-1})}{2(e-e^{-1})}.$$

and with associated function norm:

$$\|f\|^2_{\mathcal{F}_{\text{Cts}}} = \int_0^1 f(p)^2\, dp + \int_0^1 f'(p)^2\, dp.$$

Intuitively, functions in the Sobolev space $W^{1,2}([0,1])$ are differentiable, have bounded L^2 norm and have derivative with bounded L^2 norm. See Example 5 for a definition and discussion of the Sobolev space $W^{1,2}([0,1])$. Now, by assumption, for all $f \in \mathcal{F}_{\text{Cts}}$, it holds $\sup_p f(p)^2 \leqslant 1$ and $\sup_p f'(p)^2 \leqslant 1$. Hence, $\|f\|_{\mathcal{F}_{\text{Cts}}} \leqslant \sqrt{2}$. Also, $k_{\text{Cts}}(p,p) \leqslant 2$.

Next, we can apply Theorem 59, to show that $\mathcal{F}_{\text{Grid}}$ is in the RKHS induced by

$$\begin{aligned} k_{\text{Grid}}(p,p') &= \sum_{r=1}^{N} f_r(p) f_r(p') \\ &= 1\left\{\exists\, r \in [N] \,:\, \frac{r}{N} \leqslant p, p' \leqslant \frac{r+1}{N}\right\}. \end{aligned}$$

From the lemma, $\|f\|_{\text{Grid}} \leqslant 1$ and $k_{\text{Grid}}(p,p) \leqslant 1$. Defining,

$$k \stackrel{\text{def}}{=} (k_{\text{Cts}} + k_{\text{Grid}}) \cdot k_{\text{LowDeg}},$$

from the calculations above we have that for all $x, p \in \mathcal{X} \times [0,1]$,

$$k((x,p),(x,p)) \leqslant 12n^d.$$

And, by Theorem 56 and Theorem 57, $f_x \cdot f_p \in \mathcal{F}$ for $\mathcal{F}$ the RKHS associated with k and for all $f_p \in \mathcal{F}_{\text{Cts}} \cup \mathcal{F}_{\text{Grid}}$ and $f_x \in \mathcal{F}_{\text{LowDeg}}$.

Applying the triangle and Cauchy-Schwarz inequalities, we have, for all $f_p \in \mathcal{F}_{\text{Cts}} \cup \mathcal{F}_{\text{Grid}}$ and $f_x \in \mathcal{F}_{\text{LowDeg}}$, $\|f_p\|_{\mathcal{F}_{\text{Cts}}+\mathcal{F}_{\text{Grid}}} \leqslant \sqrt{2}+1$ so

$$\|f_p \cdot f_x\|_{(\mathcal{F}_{\text{Cts}}+\mathcal{F}_{\text{Grid}})\cdot\mathcal{F}_{\text{Grid}}} \leqslant (\sqrt{2}+1)\cdot 1.$$

Finally, applying Theorem 8 with the function and feature norms above, we have the desired bound:

$$\begin{aligned}\left|\sum_{t=1}^{T} f_x(x_t) f_p(p_t)(y_t - p_t)\right| &\leqslant (\sqrt{2}+1)\sqrt{1+\sum_{t=1}^{T} 12n^d/4} \\ &\leqslant 3\sqrt{1+3n^d T} \leqslant 6\sqrt{n^d T}.\end{aligned}$$

■

We note that there is a great deal of flexibility when deciding how the distinguishers above depend on the prediction p. Here, we chose a the union of a specific class of indicator functions with the set of continuous, differentiable functions with bounded domain and first derivative. However, we could equivalently have chosen a different class of functions satisfying mild smoothness conditions or a different (possibly infinite) partition of $[0,1]$. Alternately, if p is always in a finite set $\mathcal{P}$, $|\mathcal{P}| < \infty$, distinguishers could be chosen to be $1\{p = \bar{p}\}$ for all $\bar{p} \in \mathcal{P}$.

Before we move on, we state two importance

Remark 10 (Boundedness of functions) *Throughout this work, we will often impose requirements that various functions or their derivative be bounded on [-1,1]. However, functions can be trivially re-scaled to hold for constants other than 1.*

Remark 11 (Non-asymptotic results) *The rates we achieve in this paper are non-asymptotic. Throughout, we take care to derive the constant so that dependencies on auxiliary parameters (in the case of Theorem 9, n and d) so their dependence is clear. We opt for simpler rather than tighter constants throughout for clarity.*

Our next corollary gives a similar guarantee to the previous for any finite set of bounded functions.

Corollary 12 (Any set of real-valued functions whose L^2 counting measure is bounded uniformly over x, p) *Let $\mathcal{X}$ be any set, let $\mathcal{I}$ be any index set and let m be a constant. Also, let $\mathcal{F} = \{f_i\}_{i\in I}$ be a collection of functions $f_i : \mathcal{X} \times [0,1] \to \mathbb{R}$ indexed by $\mathcal{I}$. Suppose that for each $x \in \mathcal{X}, p \in [0,1]$, we have*

$$\sum_{i\in\mathcal{I}} f_i(x,p)^2 \leqslant m, \tag{10}$$

Then, the Any Kernel algorithm run on the kernel

$$k((x,p),(x',p')) \stackrel{\text{def}}{=} \sum_{i\in I} f_i(x,p) f_i(x',p'), \tag{11}$$

(where we assume the sum can be evaluated in polynomial time in T) is guaranteed to generate a sequence of predictions such that for all $f \in \mathcal{F}$,

$$\Big|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} f(x_t, p_t)(y_t - p_t)\Big| \leqslant \sqrt{mT + 1}.$$

Proof The result follows as a direct consequence of Theorem 59 and Theorem 8. The feature norm is uniformly bounded by m and for all $f \in \mathcal{F}$, $\|f\|_{\mathcal{F}} \leqslant 1$. ■

A sufficient (but not necessary) condition for Equation (10) to hold is that $\mathcal{F}$ is finite, in which case $\mathcal{F}$ might contain arbitrary pre-existing predictors with which we would like the Any Kernel algorithm to guarantee outcome indistinguishability with respect to. In other cases, $\mathcal{I}$ need not be countable, in which case, the sum appearing in Equation (10) should be interpreted as an integral with respect to the counting measure on $\mathcal{I}$. In this case, a necessary (but not sufficient) condition for Eq. (10) to hold is that for each $x \in \mathcal{X}$, there are at most countably many $i \in \mathcal{I}$ such that $f_i(x) \neq 0$.

Comparison to prior work. As per our earlier discussion, the closest work to ours is (Vovk, 2007). The K29* algorithm presented therein achieves a similar guarantee, but requires that the kernel $k(x, p)$ is continuous in p. This restriction rules out indistinguishability with respect to binary functions (or any other discontinuous f). Distinguishers of this form were the main focus of (Hébert Johnson et al., 2018; Dwork et al., 2021). Our algorithm works for *any* kernel, and in particular can be used to guarantee indistinguishability with respect to binary functions as in first example above. The computation complexity of our algorithm and Vovk's are essentially identical.

Also closely related to our work, the algorithm in (Gupta et al., 2022) guarantees online indistinguishability with respect to a finite set of binary valued functions $\mathcal{F}$. Furthermore, while their OI error bound scales as $\sqrt{\log|\mathcal{F}|}$, the per round computational complexity scales linearly with $|\mathcal{F}|$. In comparison, our algorithm can be used to guarantee indistinguishability with respect to both real- and Boolean-valued functions. Achieving indistinguishability with respect to real-valued functions is crucial for our later results on omniprediction.

Furthermore, as stated previously, the computational complexity and OI error of the Any Kernel algorithm have no explicit dependence on the size of $\mathcal{F}$. Both of these are determined by the kernel k. As seen in Theorem 9, certain infinite classes of functions can be efficiently represented by kernels that can be computed in constant time. For certain worst-case classes $\mathcal{F}$, we can still guarantee indistinguishability (as in the second part of Theorem 9). However, the kernel in this construction requires enumerating over $\mathcal{F}$ and both the runtime and OI error scale polynomially with $|\mathcal{F}|$. Therefore, for the specific case where one aims to be indistinguishable with respect to a finite set of *Boolean* functions not known to be efficiently represented by a kernel, the algorithm in (Gupta et al., 2022) is preferable. In that setting, both our procedure and the one in (Gupta et al., 2022) have run times linear in $|\mathcal{F}|$, but their OI error is significantly smaller (polylogarithmic vs polynomial).

The principal strength of Theorem 12 is that we can guarantee indistinguishability with regards to any real-valued function f that is efficiently computable. This in particular includes any neural network or prediction baseline one might consider. We return to this point in the next section.

Additive models and boosting. As a final remark before the proof of the proposition, we note that the previous result also guarantees outcome indistinguishability with respect models like random forests or gradient boosted decision trees. These learning algorithms are the gold standard in certain data modalities (Gardner et al., 2022; Grinsztajn et al., 2022).

In particular, let $\mathcal{F}_{DTd} \subseteq \{\{\pm 1\}^n \to [-1,1]\}$ be the class of regression trees of depth d. Random forests and gradient-boosted trees are additive ensembles of the form:

$$f(x) = \sum_i \lambda_i f_i(x) \tag{12}$$

where λ_i are real-valued coefficients and $f_i \in \mathcal{F}_{DTd}$ Since, $\mathcal{F}_{DT} \subseteq \mathcal{F}_{\text{LowDeg}}$ (see e.g (O'Donnell, 2021)), then the Any Kernel algorithm instantiated with the kernel from Theorem 9 guarantees indistinguishability with respect to any $f \in \mathcal{F}_{DTd}$. Since indistinguishability is closed under addition, then the same algorithm also guarantees indistinguishability with error $\mathcal{O}(\gamma\sqrt{n^d T})$ with respect to additive ensembles as in Equation (12) as long as $\sum_i |\lambda_i|$ is $\mathcal{O}(\gamma)$.

B.2. Specializing the Any Kernel algorithm to the link prediction problem

Having introduced this technical machinery, we now specialize it to the link prediction problem, turning our attention to designing specific kernels whose corresponding function spaces contain interesting classes of distinguishers that operate on graphs. The tests we consider fall into two broad categories: those capturing socially salient information and those for which passing these tests likely implies good predictive performance. Socially salient tests might include whether a pair of individuals belong, respectively, to a specific pair of demographic groups (*i.e.*, multicalibration). On the other hand, predictive performance tests aim to capture correlations between features, predictions, and outcomes.

In this section, we change notation from $f(x,p)$ to $f(u,p)$ reflect the fact that distinguishers f operate over the universe $\mathcal{U}$ consisting of pairs of nodes $a = (i,j)$ and a graph G. We will also make liberal use the set of grid indicator functions $\mathcal{F}_{\text{Grid}} = \{f_r\}_{r=1}^N$ for a positive integer N where $f_r = 1\{(r-1)/N \leqslant p < r/N\}$ for $r = 1,\ldots,N-1$ and $f_N = 1\{(N-1)/N \leqslant p \leqslant 1\}$. As in Theorem 9, this choice is somewhat arbitrary: we could equivalently use the sets of functions satisfying mild smoothness conditions or arbitrary partitions of the unit interval. We will assume N is a universal constant throughout.

Group membership tests. A simple starting point for socially salient tests are those which given a pair of individuals (i,j) outputs 1 if i belongs to a demographic group g and j belongs to group g'. Groups may be defined by, for example, race, ethnicity, gender, age, religion, education, occupation and/or political or organizational affiliation. We will let g be a binary function $\mathcal{Z} \to \{0,1\}$ which takes in node-level features $z_{i,t}$ and returns 0 or 1. These tests are analogous to multiaccuracy (Hébert Johnson et al., 2018; Kim et al., 2019) (if they do not depend on predictions p) and multicalibration (Hébert Johnson et al., 2018) (if they do), adapted to the link prediction setting, and allowing for arbitrary pairs of demographic groups. Indeed, cross-group ties are the focus of significant study in the networks literature (Abebe et al., 2022; Calvo-Armengol and Jackson, 2004; Zeltzer, 2020; Stoica et al., 2018; Okafor, 2020), and platforms may wish to ensure predictions are calibrated with respect to them.

Proposition 13 (Pairs of demographic groups) *Let $\mathcal{G} \subseteq \{\mathcal{Z} \to \{0,1\}\}$ be a (not necessarily disjoint or finite) collection of demographic group indicator functions on $\mathcal{Z}$ such that each individual i at any time t belongs to at most m groups for some positive integer m:*

$$\max_{t\in[T], i\in V_t} \left| \sum_{g\in\mathcal{G}} g(z_{i,t}) \right| \leqslant m.$$

For a positive integer N and given $u = (i,j,G)$ and $u' = (i',j',G')$, define the kernel k to be

$$k((u,p),(u',p')) = 1\{\exists\ r \in [N]\ :\ f_r(p) = f_r(p') = 1\} \sum_{g,g'\in\mathcal{F}_G} g(z_i)g'(z_j)g(z_{i'})g'(z_{j'})$$

where (z_i, z_j) *are the node-level features of the pair* (i, j) *in* G *and* $(z_{i'}, z_{j'})$ *are the node level features of* $(i', j') \in G'$*. Then, the Any Kernel algorithm with kernel* k *generates a sequence of predictions satisfying,*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (y_t - p_t) 1\left\{ g(z_{i,t}) = 1, g'(z_{j,t}) = 1, f_r(p_t) = 1 \right\} \right| \leqslant \sqrt{mT + 1}.$$

for all $g, g' \in G$ *and* $r \in 1, \ldots, N$ *where* $u_t = (i_t, j_t, G_t)$*.*

Assuming that checking whether a pair of predictions p, p' fall in the same grid cell and evaluating the indicator functions $g \in \mathcal{G}$ takes constant time, then the kernel can be naively computed in time $\mathcal{O}(1)$. Therefore, following Theorem 8, at time t, the algorithm generates a prediction p_t in time $\widetilde{\mathcal{O}}(tm)$.

Proof The result is a direct implication of Theorem 12. Let $\mathcal{F}$ in Theorem 12 be the cross product of group membership indicators and grid indicators $\mathcal{G} \times \mathcal{F}_{\text{Grid}}$ and notice

$$\begin{aligned} k((u,p),(u',p')) &= \sum_{r=1}^{N} f_r(p) f_r(p') \sum_{g,g' \in \mathcal{F}_G} g(z_i) g'(z_j) g(z_i') g'(z_j') \\ &= 1\{\exists\, r \in [N] \,:\, f_r(p) = f_r(p') = 1\} \sum_{g,g' \in \mathcal{F}_G} g(z_i) g'(z_j) g(z_i') g'(z_j') \end{aligned}$$

is the associated kernel as defined in Theorem 12. Notice that Equation (10) is satisfied with the m in the statement of the result, since x cannot be in more than m groups and p cannot be in more than one grid cell. Thus, we have verified the assumptions in the corollary and the bound holds. ■

Closely related to group membership is the idea of homophily (McPherson et al., 2001). Informally, homophily is the tendency of individuals to connect those who are similar to themselves. Homophily may be defined by membership in a demographic group as well as geographic proximity (Verbrugge, 1977), social capital (Borgatti and Foster, 2003), and political/social attitudes/beliefs (Gentzkow and Shapiro, 2011). All of these measures of homophily are *scalar valued* functions of node-level features. In these cases, the proposition above can be straightforwardly extended so that the algorithm generates predictions with are outcome indistinguishable with respect to (functions of) these measures.

An alternate formulation of the link prediction problem would also consider edge-level features such the frequency or intensity of interaction between individuals. For example, the influential notion of *weak ties*, originally characterized qualitatively as a "combination of the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie" (Granovetter, 1973), are usually defined quantitatively in terms of interaction intensity (see, *e.g.*, (Rajkumar et al., 2022)). Our results could be trivially extended to solve this formulation of link prediction where distinguisher may also consider edge level features. However, for simplicity of presentation, we omit including edge-level features.

Network topology tests. We now consider tests that depend on the structure of the graph. A particularly simple set of such tests is based on embeddedness, or the number of mutual connections between two individuals (i, j) on a graph G. The sociological notion of embeddedness, as discussed in (Granovetter, 1985), concerns the degree to which individuals' activities are *embedded* within in social relations, *i.e.*, networks. Formally for $u = (i, j, G)$, we quantify the structural embeddedness of u (following the definition in (Easley et al., 2010)) as

$$\mathsf{Em}(u) \stackrel{\text{def}}{=} |\Gamma_G(i) \cap \Gamma_G(j)|. \tag{13}$$

Note that the pair of individuals themselves need not be connected. For example, a rich literature studies *long ties* or *local bridges*, which are ties with embeddedness zero (see, *e.g.*, (Granovetter, 1973; Burt, 2004; Jahani et al., 2023; Easley et al., 2010)). Embeddedness is measured and carefully analyzed by digital platforms like LinkedIn in practice (Rajkumar et al., 2022). It also underlies classical theories of network evolution through triadic closure (Kossinets and Watts, 2006; Jackson and Rogers, 2007; Asikainen et al., 2020; Abebe et al., 2022). Here in our next result, we show one can construct an efficient kernel k that guarantees online outcome indistinguishability with respect to embeddedness tests.

Proposition 14 (Embeddedness) *For $u = (i, j, G)$ and $u' = (i', j', G')$ define the kernel*

$$k((u,p),(u',p')) \stackrel{\text{def}}{=} 1\{\mathsf{Em}_t(u) = \mathsf{Em}_t(u'), \exists r \in [N] \,:\, f_r(u) = f_r(u') = 1\}.$$

Then, the Any Kernel algorithm run with kernel k generates a sequence of predictions satisfying,

$$\Big|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (y_t - p_t) 1\{\mathsf{Em}_t(u_t) = c, f_r(p) = 1\}\Big| \leqslant \sqrt{\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} p_t(1-p_t) + 1} \leqslant 2\sqrt{T}.$$

for all $c \in \mathbb{N}$ and $r \in [N]$.

Since the kernel only checks whether two different pairs of individuals have the predictions that fall in the same grid cell and have an identical number of mutual friends, the kernel can be computed in the time it takes to compute neighborhood intersections.

An advantage of the class $1\{\mathsf{Em}_G(u_t) = c, f_r(p) = 1\}_{c \in \mathbb{N}, r \in [N]}$ is that neither the run time nor OI error depends on the maximum degree of nodes in the graph. We also note that the above formulation could be straightforwardly modified to include indicator functions for having embeddedness more or less than c, as long as it is efficient to compute embeddedness. Lastly, we note that the construction can be generalized to include distinguishers of the form i and j have c distance r neighbors in common by simply changing Γ to $\Gamma^{(r)}$ in the definitions above.

We can generalize the embeddedness tests above even further to guarantee outcome indistinguishability with respect to all tests that depend on the isomorphism class of the subgraph induced by the neighborhoods $\Gamma(i), \Gamma(j)$.

A function f from graphs G to the real-line is isomorphism-invariant if for any two graphs G and G such that G and G' are isomorphic, it holds that $f(G) = f(G')$. Abusing notation, we can write isormorphism-invariant functions f as those defined on isomorphism (equivalence) classes $\bar{G}$ where $\bar{G}$ is a set of graphs that are all isomorphic to each other.

Several interesting classes of functions f are isomorphism-invariant. For instance, any function f that just depends on the number of nodes or edges in the graph, the degree distribution, or the spectrum of the graph Laplacian is isomorphism-invariant. Several classes of isomophism-invariant functions have been studied extensively in the networks literature, like various notions of structural cohesion (which might, *e.g.*, measure the edge density of the induced subgraph in an individual's neighborhood (Friedkin, 1993)).

In the following proposition, we will use the following notation: given a set of nodes S and a graph G, let $G[S]$ denote the induced subgraph of S on G. Also, we will use $\Gamma(i), \Gamma'(i')$ to refer to the neighborhoods $\Gamma_G(i), \Gamma_{G'}(j)$ for graphs G, G' respectively. We will write $G \simeq G'$ to denote that G and G' are isomorphic.

Proposition 15 *Let $\mathcal{F}_{\text{iso}} \subseteq \{\mathcal{G} \to \mathbb{R}\}$ denote the set of all isomorphism invariant functions and $\mathcal{F}_{\text{Grid}} = \{f_1, \ldots, f_N\}$ be the grid indicator functions on the unit interval as above. Furthermore, for $u = (i, j, G)$ and $u' = (i', j', G')$ define the function k to be*

$$k((u,p),(u',p)) = 1\{G[\Gamma(i) \cup \Gamma(j)] \simeq G'[\Gamma'(i) \cup \Gamma'(j)],\ \exists r \in [N] \,:\, f_r(p) = f_r(p') = 1\}.$$

Suppose all graphs in the sequence $\{G_t\}_{t=1}^T$ degree bounded by a constant. Then k can be computed in polynomial time and the Any Kernel algorithm instantiated with the kernel k is guaranteed to generate a sequence of predictions satisfying:

$$\Big|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t}(y_t - p_t)f(u_t)1\{p=\bar{p}\}\Big| \leqslant \|f\|_{\mathcal{F}}\sqrt{\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} p_t(1-p_t)+1} \leqslant 2\|f\|_{\mathcal{F}}\sqrt{T}.$$

for any $f \in \mathcal{F}_{\text{iso}} \subseteq \mathcal{F}$. For the special case of functions $f_{\bar{G}}(i,j,G) = 1\{G \in \bar{G}\}$ for some isomorphism class $\bar{G}$, the dependence on $\|f_{\bar{G}}\|_{\mathcal{F}}$ can be removed since $\|f\|_{\mathcal{F}} \leqslant 1$ for every $\bar{G}$.

Proof Let $\bar{G}_1, \bar{G}_2, \ldots$ be the sequence of graph isomorphism classes in some ordering (perhaps lexicographic, where all isomorphism classes for graphs of size n come before those of size $n+1$ for all $n \in \mathbb{N}$). Let $\Phi(G)$ be the feature map defined as,

$$\Phi(G) = (1\{G \in \bar{G}_1\},\ 1\{G \in \bar{G}_2\},\ \ldots). \tag{14}$$

For $u = (i,j,G)$ and $u' = (i',j',G')$,

$$k_{\text{iso}}(u,u') \stackrel{\text{def}}{=} \langle \Phi(G[\Gamma(i)\cup\Gamma(j)]), \Phi(G'[\Gamma'(i)\cup\Gamma'(j)])\rangle$$

where the inner product $\langle\cdot,\cdot\rangle$ is the standard inner product in ℓ^2, the Hilbert space of square summable sequences ($\langle x,y\rangle = \sum_{i=1}^{\infty} x_i y_i$). Since G can only be in one of the $\bar{G}_i$, $\Phi(G)$ is a square-summable sequence (only one element is 1, all the others are 0). So k_{iso} is a valid kernel and $k_{\text{iso}}(u,u') \leqslant 1$ for all $u,u' \in \mathcal{U}$. Since all nodes in G_t are assumed to have bounded degree, there are only a constant number of isomorphism classes for the subgraph $G[\Gamma(i)\cup\Gamma(j)]$. Thus, k can be computed efficiently via brute force search.[21]

The fact that $\mathcal{F}_{\text{iso}} \subseteq \mathcal{F}_{k_{\text{iso}}}$ for $\mathcal{F}_{k_{\text{iso}}}$, the RKHS associated with kernel k_{iso}, follows from the Moore-Aronszajn Theorem (Theorem 54) which states that the corresponding RKHS of the kernel $\mathcal{F}$ is equal to

$$\mathsf{span}\{\Phi(G) : G \text{ is a graph}\}.$$

Given any isomorphism invariant function f, we can write it as,

$$f(G) = \langle \Phi(G), f\rangle = \sum_{i=1}^{\infty} f(\bar{G})1\{\bar{G} = \bar{G}_i\},$$

where $\bar{G}$ is the set of graphs that are isomorphic to G. Here, we used the fact that f is isomorphism-invariant and again slightly abused notation to write $f(\bar{G})$ where $\bar{G}$ is a set, instead of one graph. Applying Theorem 8 with the function and feature norms above yields the desired result. ■

As with embeddedness tests, isomorphism tests can be naturally extended to depend on the distance r neighborhoods of pairs of nodes, by simply replacing each Γ in the proposition with $\Gamma^{(r)}$ (for constant r). Various network centrality measures, like k-core similarity, betweenness centrality, eigenvalue centrality and others (see, *e.g.*, (Rodrigues, 2019)) may be computed using the induced subgraph of distance r neighborhoods. Similarly, core-periphery measures (Rombach et al., 2014) may be similarly defined for distance r neighborhoods. In each of these cases, care must be taken to ensure that the measure can be computed efficiently and that the function norms are bounded.

21. One could also of course run more sophisticated procedures for isomorphism testing if one desires (*e.g.*, Luks' algorithm (Luks, 1982)), but these are unnecessary for polynomial runtime guarantee in this setting since our distinguisher only examine the local neighborhood of (i,j) which are at most of constant size.

Tests using network topology and neighbors' feature vectors. We end this section by considering distinguishers that examine both the local neighborhood structure, as well as the *features* of individuals in these neighborhoods. (The graph isomorphism tests presented previously only examine the structure of the neighborhood, but not their individual features.)

Theorem 12 provides for OI guarantees that hold with respect to very powerful predictors. For example, we may take $\mathcal{F}$ to be any finite set of graph neural networks, which are currently state-of-the-art for link prediction (Zhang and Chen, 2018; Yun et al., 2019) and any number of other graph-related tasks (see, *e.g.*, (Zhou et al., 2020)) and are widely deployed across digital platforms that host social networks (Zhou et al., 2020; Zhang et al., 2020). Theorem 12 immediately implies that the Any Kernel algorithm yields

$$\left|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} f(u_t)(y_t - p_t)\right| \leqslant \sqrt{mT}.$$

for all $f \in \mathcal{F}$.

R-convolutions (convolutions over relations). This machinery can also be used to guarantee indistinguishability to functions of the form

$$f(i,j,G) = \langle w, \sum_{v\in\Gamma(i)\cup\Gamma(j)} \Phi(z_v)\rangle_{\mathcal{F}} \tag{15}$$

where $\Phi(z) : \mathcal{Z} \to \mathcal{F}$ is a feature mapping and $w \in \mathcal{F}$ is an element in the RKHS. This particular class of functions can be efficiently represented by using the R-convolutional kernel from (Haussler et al., 1999), which, given a feature map Φ and $u = (i,j,G), u' = (i',j',G')$, computes:

$$k(u,u') = \sum_{v\in\Gamma(i)\cup\Gamma(j), v'\in\Gamma'(i')\cup\Gamma'(j')} \langle\Phi(z_v),\Phi(z_{v'})\rangle_{\mathcal{F}}$$

Assuming that the features $\Phi(v)$ and weight w have norm at most 1, and that any node in the graph has degree at most d, the Any Kernel algorithm guaranteees $\mathcal{O}(d\sqrt{T})$ indistinguishability to functions of the form in Eq. (15). The features Φ may include socially salient measures of diversity (Burt, 1982) or bandwidth (Aral and Van Alstyne, 2011).

Appendix C. Online Omniprediction and Applications to Link Prediction

Up until this point, we have focused on designing online algorithms which satisfy online outcome indistinguishability with respect to various classes of tests. In this section, we illustrate how these previous insights and algorithms also imply *loss minimization* with respect to many different objectives $\mathcal{L}$ and infinitely large benchmark classes $\mathcal{H}$.

That is, we show how simple adaptations of techniques developed in the previous section expand the scope of possibilities for *online omniprediction*. We recall the definition of online omnipredictors from Appendix A:

Definition 16 *An algorithm $\mathcal{A}$ is an $(\mathcal{L},\mathcal{H},\mathcal{R}_{\mathcal{A}})$-online omnipredictor if it generates a transcript $\{(x_t,\Delta_t,y_t)\}_{t=1}^{T}$ such that for all $\ell \in \mathcal{L}$ there exists a $\pi_\ell : \mathcal{X}\times[0,1] \to [0,1]$ such that*

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} \ell(x_t,\pi_\ell(x_t,p_t),y_t) \leqslant \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \ell(x_t,h(x_t),y_t) + \mathcal{R}_{\mathcal{A}}(T). \tag{16}$$

where the regret bound, $\mathcal{R}_{\mathcal{A}} : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$, is $o(T)$.

Omnipredictors were initially defined by (Gopalan et al., 2022) for the offline setting and then extended to the online case by (Garg et al., 2024). Intuitively, omnipredictors are efficient "menus of optimality": They provide a single prediction that can be postprocessed (via π_ℓ) to guarantee lower loss than that achievable by any function in some comparator class $\mathcal{H}$. Briefly, the main contribution of this section is we introduce the first algorithm which guarantees online omniprediction with respect to comparator classes $\mathcal{H}$ that are real-valued and of infinite cardinality. These constructions are also unconditionally computationally efficient.

To do this, we build on the insight established by (Gopalan et al., 2023) which shows that, in the distributional (offline) setting, given any set of losses $\mathcal{L}$ and comparator class $\mathcal{H}$, one can always construct a set of distinguishers $\mathcal{F}(\mathcal{H},\mathcal{L})$ such that indistinguishability with respect to $\mathcal{F}(\mathcal{H},\mathcal{L})$ implies omniprediction. We show that such a connection holds in the online setting too, and illustrate computationally efficient ways of achieving the requisite indistinguishability guarantees via the Any Kernel algorithm. Theorem 24 provides a formal statement of this general recipe or meta-theorem for online omniprediction.

The following result (Theorem 17) follows by using machinery of reproducing kernel Hilbert spaces to instantiate this general recipe with various choices of kernels. In the first part, we illustrate how our techniques can be used guarantee omniprediction with respect to common classes of losses and comparator classes. In the second part, we provide a different instantiation of the theorem specialized to the link prediction setting. Although the general framework allows for loss functions that depend on features x, we state the result without dependence on features for simplicity and to enable easier comparisons with prior work.

Theorem 17 *There exists a computationally efficient kernel k, such that the Any Kernel algorithm run with kernel k runs in polynomial time and is a $(\mathcal{H},\mathcal{L},\mathcal{O}(\sqrt{(m+n^d)T}))$-omnipredictor, where*

(a) The comparator class $\mathcal{H} \subseteq \{\{-1,1\}^n \to [-1,1]\}$ contains all regression trees of depth at most d and any pre-specified set of functions $\mathcal{H}_0 \subseteq \{\mathcal{X} \to [-1,1]\}$ where $|\mathcal{H}_0| \leqslant m$.

(b) The set of losses $\mathcal{L}$ contains any function $\ell : [0,1]\times\{0,1\} \to [-1,1]$ that satisfies at least one of the following conditions:

(i) The loss ℓ is a continuous, differentiable proper scoring rule. That is, $p \in \pi_\ell(p)$ and $\ell \in W_1^{1,2}([0,1])$ (see Equation (18) for a formal definition of $W_1^{1,2}([0,1])$.

(ii) The loss $\ell(\hat{y},y)$ strongly convex in $\hat{y}$ and is differentiable in $\hat{y}$ with $|\frac{\partial}{\partial\hat{y}}\partial\ell(\hat{y},y)| \leqslant 1$.

(iii) The loss ℓ is in a pre-speficied finite set $\mathcal{L}_0 \subseteq \{[0,1]\times\{0,1\} \to [-1,1]\}$ where $|\mathcal{L}_0| \leqslant m$.

If the problem domain is link prediction, the loss class $\mathcal{L}$ may instead be a set of functions of the form $\ell_x(u)\ell_y(\hat{y},y)$ where[22]

(a) ℓ_x may be any of the tests described in Appendix B.2 such as indicators for any pair of group memberships or ties with embeddedness c (see Equation 13), and

(b) ℓ_y may be any function described in (b) above, or any finite set of bounded functions rewarding desirable outcomes, such as edge formation (e.g., $\ell_y(\hat{y},y) = 1-y$).

22. Recall that, when we are discussing link prediction, $u = (a,G)$ represents an element of the universe $\mathcal{U}$ where $a = (i,j)$ is an pair of individuals and G is the current state of the graph detailing the existing set of edges and features for every node.

Comparison to prior work. The results we present in this section differ from prior work both in their substance and in the techniques used to prove them. (Garg et al., 2024) considers a more exacting omniprediction definition, called *swap*-omniprediction, for which the function $h \in \mathcal{H}$ that one compares to depends on the current prediction p_t. The paper provides an oracle-efficient algorithm that achieves $\mathcal{O}(T^{7/8})$ swap regret. Furthermore, they prove that $\mathcal{O}(\sqrt{T})$ (or, in fact $o(T^{0.528})$) regret for online swap-omniprediction is in fact impossible.

In the same paper, using ideas rooted in online minimax optimization (Lee et al., 2021), they introduce an algorithm which attains $\mathcal{O}(\sqrt{T \log |\mathcal{H}|})$ vanilla omniprediction regret for the case where $\mathcal{H}$ is a finite set of binary valued functions and $\mathcal{L}$ consists on proper scoring rules or bimonotone loss functions.[23] Their algorithm relies on enumerating the functions in $\mathcal{H}$, and hence has runtime that is linear in $\mathcal{H}$.

In recent, independent work, (Hu et al., 2024) also introduce new omniprediction algorithms for the offline case where $\mathcal{H}$ consists of generalized linear models and $\mathcal{L}$ consists of matching losses. These results are complementary to ours. To the best our knowledge, our work is the first to attain $\mathcal{O}(\sqrt{T})$ regret for vanilla online omniprediction over: *a*) comparator classes $\mathcal{H}$ that are of infinite size or which map onto real values and *b*) arbitrary, bounded losses ℓ.

Outline of the section and preliminaries. In Appendix C.1, we present our main technical results regarding online omniprediction. These rely on the ability to achieve certain online indistinguishability conditions using kernels. We illustrate how to achieve these in Appendices C.2 to C.4. Then, in Appendix C.5 and Appendix C.7 we discuss implications of these results for online regression and performative prediction. Finally, in Appendix C.6, we apply our new technical machinery to the problem of link prediction in a social network.

Before moving on, we review several pieces of notation that we will repeatedly reuse during this section. Given a loss function ℓ, we will use $\partial\ell$ to refer to its discrete derivative:

$$\partial\ell(x, \hat{y}) = \ell(x, \hat{y}, 1) - \ell(x, \hat{y}, 0).$$

Given a set of losses $\mathcal{L}$, we analogosly use $\mathcal{L}$ to refer to the set of discrete derivatives:

$$\partial\mathcal{L} \stackrel{\text{def}}{=} \{\partial\ell \mid \ell \in \mathcal{L}\}$$

Throughout our presentation, we will take always take the post-processing function π_ℓ to be

$$\pi_\ell(x, p) \in \operatorname*{arg\,min}_{\hat{y}\in[0,1]} \mathop{\mathbb{E}}_{y\sim\text{Ber}(p)} [\ell(x, \hat{y}, y)] = \operatorname*{arg\,min}_{\hat{y}\in[0,1]} p \cdot \ell(x, \hat{y}, 1) + (1-p) \cdot \ell(x, \hat{y}, 0). \tag{17}$$

Lastly, we also use the fact that there exists an RKHS for the set of smooth functions over the unit interval. The following observation follows from the fact that the functions in $W_B^{1,2}([0,1])$ are a subset of the well-known Sobolev kernel. See Example 5 for more details.

Fact 18 *Define $W_B^{1,2}([0,1])$ with parameter B to be the set of continuous, differentiable functions $g : [0,1] \to [-1,1]$ satisfying*

$$\int_0^1 g(t)^2 dt + \int_0^1 g'(t)^2 dt \leqslant B^2. \tag{18}$$

$W_B^{1,2}([0,1])$ is contained in the Sobolev space $W^{1,2}([0,1])$. That is, there exists an efficiently computable kernel k with RKHS $\mathcal{F}_k$ such that $W_B^{1,2}([0,1]) \subset \mathcal{F}_k$ and for all $f \in W_B^{1,2}([0,1])$ it holds $\|f\|_{W^{1,2}([0,1])} \leqslant B$ and $\sup_t k(t,t) \leqslant \sqrt{3}$.

23. Informally, bimonotone losses are those which satisfy $\ell(\pi_\ell(p), 1) = \ell(1,1)$ and $\ell(\pi_\ell(p), 0) = \ell(0,0)$. See (Garg et al., 2024).

C.1. Efficient, $\sqrt{T}$ online omiprediction with respect to rich comparison classes $\mathcal{H}$.

In this subsection, we present our main result demonstrating how outcome indistinguishability implies omniprediction in the online setting and illustrating how these indistinguishability conditions can be efficiently achieved via the Any Kernel algorithm.

The following two OI definitions, hypothesis and decision OI, were first introduced (in the batch setting) by (Gopalan et al., 2023). We now adapt them to the online case. Decision outcome indistinguishability (DOI) is defined with respect to a class of losses $\mathcal{L}$. It states that prediction must be approximately indistinguishable with respect to the class of test functions constructed from pairs of loss functions $\ell \in \mathcal{L}$ and post-processed predictions π_ℓ:

Definition 19 (Decision OI) *For a loss class $\mathcal{L}$ and regret bound $\mathcal{R}_{\mathrm{DOI}}(T)$, an algorithm satisfies $(\mathcal{L}, \mathcal{R}_{\mathrm{DOI}}(T))$-decision outcome indistinguishability (DOI) if it generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^T$ such that,*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} \partial \ell(x_t, \pi_\ell(x_t, p_t))(p_t - y_t) \right| \leqslant \mathcal{R}_{\mathrm{DOI}}(T), \qquad \forall \ell \in \mathcal{L}. \tag{19}$$

The second OI condition, hypothesis outcome indistinguishability (HOI), requires that predictions must be approximately indistinguishable with respect to functions constructed from pairs of comparator functions $h \in \mathcal{H}$ and loss functions $\ell \in \mathcal{L}$:

Definition 20 (Hypothesis OI) *For a loss class $\mathcal{L}$, comparator class $\mathcal{H}$, and regret bound $\mathcal{R}_{\mathrm{HOI}}(T)$, an algorithm satisfies $(\mathcal{L}, \mathcal{H}, \mathcal{R}_{\mathrm{HOI}}(T))$-hypothesis outcome indistinguishability (HOI) if it generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^T$ such that:*

$$\left| \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} \partial \ell(x, h(x_t))(p_t - y_t) \right| \leqslant \mathcal{R}_{\mathrm{HOI}}(T), \qquad \forall (\ell, h) \in \mathcal{L} \times \mathcal{H}. \tag{20}$$

Having introduced these two definitions, the result that OI implies omniprediction is almost immediate. The following lemma formally adapts the ideas from (Gopalan et al., 2023) to the online setting.

Lemma 21 *Fix a comparator class $\mathcal{H} \subseteq \{\mathcal{X} \to [0,1]\}$, a class of losses $\mathcal{L} \subseteq \{\mathcal{X} \times [0,1] \times \{0,1\} \to \mathbb{R}\}$ and regret bounds $\mathcal{R}_{\mathrm{DOI}}(T), \mathcal{R}_{\mathrm{HOI}}(T) : \mathbb{N} \to R$. If an algorithm $\mathcal{A}$ satisfies*

1. *$(\mathcal{L}, \mathcal{R}_{\mathrm{DOI}}(T))$-decision OI (Definition 19)*
2. *and $(\mathcal{L}, \mathcal{H}, \mathcal{R}_{\mathrm{HOI}}(T))$-hypothesis OI (Definition 20),*

then, $\mathcal{A}$ is an $(\mathcal{L}, \mathcal{H}, \mathcal{R}_{\mathrm{DOI}}(T) + \mathcal{R}_{\mathrm{HOI}}(T))$-online omnipredictor.

Proof First, we observe that for all $x \in \mathcal{X}$ and any pair $(\hat{y}, y)$ where $y \in \{0,1\}$:

$$\ell(x, \hat{y}, y) = y\ell(x, \hat{y}, 1) + (1-y)\ell(x, \hat{y}, 0) = y(\ell(x, \hat{y}, 1) - \ell(x, \hat{y}, 0)) + \ell(x, \hat{y}, 0)$$

A similar expression holds for the following expectation version,

$$\mathop{\mathbb{E}}_{y \sim \mathrm{Ber}(p)} \ell(x, \hat{y}, y) = p\ell(x, \hat{y}, 1) + (1-p)\ell(x, \hat{y}, 0) = p(\ell(x, \hat{y}, 1) - \ell(x, \hat{y}, 0)) + \ell(x, \hat{y}, 0).$$

Therefore,

$$|\ell(x,\hat{y},y) - \mathop{\mathbb{E}}_{\tilde{y}\sim\mathrm{Ber}(p)} \ell(x,\hat{y},\tilde{y})| = |(y-p)(\ell(x,\hat{y},1)-\ell(x,\hat{y},0))| = |(y-p)\partial\ell(x,\hat{y})|$$

Using this decomposition, by the Decision OI guarantee Definition 19, we know that

$$\sum_{t=1}^{T} \ell(x_t,\pi_\ell(x_t,p_t),y_t) \leqslant \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\tilde{y}_t\sim\mathrm{Ber}(p_t)} \ell(x_t,\pi_\ell(x_t,p_t),\tilde{y}_t) + \mathcal{R}_{\mathrm{DOI}}(T).$$

Furthermore, since π_ℓ is the argmin (see Equation (17)), by definition it satisfies the following inequality for any h,

$$\mathop{\mathbb{E}}_{\tilde{y}_t\sim\mathrm{Ber}(p_t)} \ell(x_t,\pi_\ell(x_t,p_t),\tilde{y}_t) \leqslant \mathop{\mathbb{E}}_{\tilde{y}_t\sim\mathrm{Ber}(p_t)} \ell(x_t,h(x_t),\tilde{y}_t)$$

Lastly, by the Hypothesis OI guarantee (Definition 20),

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\tilde{y}_t\sim\mathrm{Ber}(p_t)} \ell(x_t,h(x_t),\tilde{y}_t) \leqslant \sum_{t=1}^{T} \ell(x_t,h(x_t),y_t) + \mathcal{R}_{\mathrm{HOI}}(T).$$

Combining all three inequalities, we get our desired result:

$$\sum_{t=1}^{T} \ell(x_t,\pi_\ell(x_t,p_t),y_t) \leqslant \sum_{t=1}^{T} \ell(x_t,h(x),y_t) + \mathcal{R}_{\mathrm{DOI}}(T) + \mathcal{R}_{\mathrm{HOI}}(T). \quad \forall h \in \mathcal{H}$$

■

The advantage of this loss OI viewpoint is that it provides a neat template for algorithm design. More specifically, to achieve omniprediction, we only need to design kernels whose corresponding RKHS contain the required distinguishers and then run the Any Kernel algorithm with these kernels. While the main idea is simple, to prove a formal non-asymptotic regret bound we also need to ensure that corresponding function norms of the distinguishers $\|f\|_{\mathcal{F}}$ and feature norms $k((x,p),(x,p)) = \|\Phi(x,p)\|_{\mathcal{F}}^2$ are appropriately bounded. If these quantities are not appropriately bounded, then the guarantees from the Any Kernel algorithm can become vacuous (recall the bound from Theorem 8).

To address this issue, we further specialize the OI definitions above to the RKHS domain. These specializations, kernel decision and hypothesis OI, are representational conditions on the kernel k and the corresponding RKHS $\mathcal{F}_k$. Intuitively, they require that a kernel k be efficiently computable, bounded, and that certain functions are contained (and have small norm) in $\mathcal{F}_k$.

Definition 22 (Kernel Decision OI) *Let $\mathcal{L}$ be a set of loss functions. A kernel k with corresponding RKHS $\mathcal{F}$ is $\mathcal{L}$-kernel decision OI (KDOI) with parameter B if,*

$$\{\partial\ell\circ\pi_\ell \mid \ell\in\mathcal{L}\} \subseteq \mathcal{F} \subseteq \{\mathcal{X}\times[0,1]\to\mathbb{R}\}, \tag{21}$$

where $\partial\ell\circ\pi_\ell(x,p) = \ell(x,\pi_\ell(p),1) - \ell(x,\pi_\ell(p),0)$ and:

$$\sqrt{\sup_{\ell\in\mathcal{L}} \|\partial\ell\circ\pi_\ell\|_{\mathcal{F}}^2 \cdot \sup_{x\in\mathcal{X},p\in[0,1]} k((x,p),(x,p))} \leqslant B.$$

The condition states that the composition of the discrete derivative of each loss composed with its post-processing function is in the corresponding RKHS and that both the function $\|\partial\ell \circ \pi_\ell\|_{\mathcal{F}}^2$ and feature norms $k_{\mathcal{L}}((x,p),(x,p)) = \|\Phi(x,p)\|_{\mathcal{F}}^2$ are uniformly bounded. We note that, by Theorem 55, if a function f is in $\mathcal{F}$, then so is its negation, $-f$ (RKHSs are closed under scalar multiplication). Thus, a sufficient condition for KDOI is that $\ell(\pi_\ell(\cdot), y) \in \mathcal{F}$ for all $\ell \in \mathcal{L}, y \in \{0,1\}$. Next, we define an analogous condition for losses composed with comparator functions.

Definition 23 (Kernel Hypothesis OI) *Let $\mathcal{H}$ be a comparator class and let $\mathcal{L}$ be a class of loss functions. A kernel k with corresponding RKHS $\mathcal{F}$ satisfies $(\mathcal{L},\mathcal{H})$-kernel hypothesis OI (KHOI) with parameter B if,*

$$\{\partial\ell \circ h \mid h \in \mathcal{H}, \ell \in \mathcal{L}\} \subseteq \mathcal{F} \subseteq \{\mathcal{X} \times [0,1] \to \mathbb{R}\}, \tag{22}$$

where $\partial\ell \circ h(x) = \ell(x, h(x), 1) - \ell(x, h(x), 0)$ and

$$\sqrt{\sup_{h\in\mathcal{H},\ell\in\mathcal{L}} \|\partial\ell \circ h\|_{\mathcal{F}}^2 \cdot \sup_{x\in\mathcal{X},p\in[0,1]} k((x,p),(x,p))} \leqslant B.$$

As in the previous setting, a sufficient condition for KHOI is that $\ell(h(\cdot), y) \in \mathcal{F}$ for all $h \in \mathcal{H}, \ell \in \mathcal{L}, y \in \{0,1\}$. We also note that the kernel version of decision and hypothesis OI are qualitatively different from other conditions in the omniprediction literature, since they allow for infinite and real-valued comparison classes but require the existence of a suitable RKHS containing compositions of loss, post-processing and comparator functions.

With these definitions in hand, we can now state our main theorem which provides a general recipe for online omniprediction via the Any Kernel algorithm.

Theorem 24 (Corollary to Theorem 21) *Let $\mathcal{H} \subseteq \{\mathcal{X} \to [0,1]\}$ be a class of comparison functions and let $\mathcal{L} \subseteq \{\mathcal{X} \times [0,1] \times \{0,1\} \to \mathbb{R}\}$ be a set of losses.*

Let $k_{\mathcal{L}}$ and $k_{\mathcal{L},\mathcal{H}}$ be efficient kernels with corresponding RKHSs $\mathcal{F}_{\mathcal{L}}$ and $\mathcal{F}_{\mathcal{L},\mathcal{H}}$ that satisfy $\mathcal{L}$-KDOI and $(\mathcal{L},\mathcal{H})$-KHOI with parameters B_{KDOI} and B_{KDOI}. Then, the Any Kernel algorithm with kernel $k_{\mathcal{L}} + k_{\mathcal{H},\mathcal{L}}$ runs in polynomial time and is an $(\mathcal{L},\mathcal{H}, 2(B_{\text{KDOI}} + B_{\text{KHOI}})\sqrt{T})$-online omnipredictor.

Proof Define the function $k \stackrel{\text{def}}{=} k_{\mathcal{L}} + k_{\mathcal{L},\mathcal{H}}$. From Theorem 56, it holds that k is a kernel and that the functions

$$\{f_1 + f_2 \mid f_1 \in \mathcal{F}_{\mathcal{L}}; f_2 \in \mathcal{F}_{\mathcal{L},\mathcal{H}}\}$$

are in the corresponding RKHS, which we will call $\mathcal{F}$. Also, since $k_{\mathcal{L}}$ and $k_{\mathcal{L},\mathcal{H}}$ can be evaluated in polynomial time, so can k, which implies that the Any Kernel algorithm runs in polynomial time.

Now, by the fact that $\mathcal{F}_{\mathcal{L}}$ and $\mathcal{F}_{\mathcal{L},\mathcal{H}}$ are closed under scalar multiplication (by Theorem 54), the zero function is in $\mathcal{F}_{\mathcal{L}}$ and $\mathcal{F}_{\mathcal{H},\mathcal{L}}$. This implies for all $h \in \mathcal{H}$ and $\ell \in \mathcal{L}$, we have that $\partial\ell \circ \pi_\ell \in \mathcal{F}$ and $\partial\ell \circ h \in \mathcal{F}$, since $\partial\ell \circ \pi_\ell = \partial\ell \circ \pi_\ell + 0$ and $\partial\ell \circ h = 0 + \partial\ell \circ h$.

Now by the main guarantee for the Any Kernel algorithm, since we've assumed that norms and kernels are bounded, we have that,

$$\left|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} (p_t - y_t)(\partial\ell \circ \pi_\ell)(x_t, p_t)\right| \leqslant B_{\text{KDOI}}\sqrt{1 + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} p_t(1-p_t)} \leqslant B_{\text{KDOI}}\sqrt{1 + \frac{1}{4}T},$$

$$\left|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} (p_t - y_t)(\partial\ell \circ h)(x_t)\right| \leqslant B_{\text{KHOI}}\sqrt{1 + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t} p_t(1-p_t)} \leqslant B_{\text{KHOI}}\sqrt{1 + \frac{1}{4}T},$$

which, by Theorem 21, implies the theorem. ■

Discussion. We note that the above theorem establishes a precise, non-asymptotic regret bound. It in particular guarantees that for any $\ell \in \mathcal{L}$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} \ell(x_t, p_t, y_t) \leqslant \min_{h \in \mathcal{H}} \frac{1}{T}\sum_{t=1}^{T} \ell(x_t, h(x_t), y_t) + \frac{(B_{\mathrm{KDOI}} + B_{\mathrm{KHOI}})\sqrt{1 + \sum_{t=1}^{T} \mathbb{E}_{p_t \sim \Delta_t} p_t(1-p_t)}}{T}$$
$$\leqslant \min_{h \in \mathcal{H}} \frac{1}{T}\sum_{t=1}^{T} \ell(x_t, h(x_t), y_t) + 2\frac{B_{\mathrm{KDOI}} + B_{\mathrm{KHOI}}}{\sqrt{T}}$$

for every value of T greater than 1. Note that the bound adapts to the variance of the predictions p_t. Furthermore, the algorithm is very simple and easy to implement. As presented previously in Appendix B.1, you only need to be able to evaluate the kernel and solve a small binary search problem at every iteration. In the next sections, we instantiate our results for several common comparator and loss classes and show how the relevant parameters B_{KHOI} and B_{KDOI} are reasonably bounded in natural settings.

More specifically, in Appendix C.2, we demonstrate how to construct kernels that satisfy KDOI and in Appendix C.3, we demonstrate how to construct kernels to satisfy KHOI. Since the kernels for each condition can be constructed separately and then combined (added) to create a kernel to pass into the Any Kernel algorithm that satisfies both conditions jointly, the constructions in each subsection can be mixed and matched according to the prediction problem at hand.

C.2. Loss classes satisfying kernel decision OI.

In this subsection, we present several broad classes of loss functions satisfying kernel decision OI, which says that the composition of the discrete derivatives of loss functions with their associated post-processing functions must be in an RKHS and have bounded function and feature norms.

Throughout these next two subsections, we restrict our attention to a particular class of losses: those that depend only on decisions $\hat{y}$ and outcomes y, and not on features x. We will call these loss classes *feature-invariant*. This is the typical setting for omniprediction in prior work (Gopalan et al., 2022; Garg et al., 2024) (and for loss or regret minimization). Since all of the loss functions in this section will be assumed to be invariant to the feature vectors, we will drop x from the notation and consider $\mathcal{L} \subset \{[0,1] \times \{0,1\} \to \mathbb{R}\}$. We will also drop the argument for x from each post-processing function π_ℓ. Later on, in Appendix C.4, we will bring the dependence on x back in when we generalize these constructions to *separable* losses.

A naive strategy. A first attempt to achieve kernel decision OI is to find a rich, expressive RKHS $\mathcal{F}$ such that $\partial\ell \in \mathcal{F}$ then hope that the composition $\partial\ell \circ \pi_\ell$ is also in $\mathcal{F}$.[24] In fact, it is generally straightforward to find such RKHSs that contain $\partial\ell$ for many natural loss classes. For example, the set of losses where $\ell(\hat{y}, y)$ is Lipschitz in $\hat{y}$ for each $y \in \{0,1\}$ is contained in an RKHS. This is the Sobolev space mentioned in the preliminaries of this section. Lipschitz loss functions include squared/absolute error on a bounded domain, Huber, exponential, and the hinge loss, among others.

Unfortunately, the mere fact that $\partial\mathcal{L}$ is contained in an RKHS $\mathcal{F}$ does not imply that $\partial\ell \circ \pi_\ell$ is in $\mathcal{F}$. Theorem 25 shows a formal counterexample for the case where $\mathcal{F}$ is the Sobolev space.

24. Recall that $\partial\mathcal{L}$ is defined as the set $\{\partial\ell \mid \ell \in \mathcal{L}\}$, and $\partial\ell(x,p)$ is defined as $\ell(x,p,1) - \ell(x,p,0)$.

Proposition 25 *There exists a kernel k with RKHS $\mathcal{F}$ and a set of losses $\mathcal{L}$ such that $\partial\mathcal{L} \subseteq \mathcal{F}$, but*

$$\{\partial\ell \circ \pi_\ell \mid \ell \in \mathcal{L}\} \not\subseteq \mathcal{F}.$$

Proof Let $\mathcal{L} \subseteq \{[0,1]\times\{0,1\} \to \mathbb{R}\}$ be the set of functions that just depend on $\hat{y}$ and y such that for all $\ell \in \mathcal{L}$ and $y \in \{0,1\}$, $\ell(\cdot, y) : [0,1] \to \mathbb{R}$ is differentiable and for which both ℓ and its derivative with respect to $\hat{y}$ are square integrable over $[0,1]$:

$$\int_0^1 \ell(t,y)^2 dt + \int_0^1 \ell'(t,y)^2 dt < \infty$$

Notice that $\partial\mathcal{L}$ is the Sobolev space $W^{1,2}([0,1])$, which is an RKHS that has an efficient kernel. (See Example 5 for a definition of Sobolev spaces relevant to our context.) We will show that the postprocessing of a function $\partial\ell \circ \pi_\ell \in \mathcal{L}$ may *not* be in the Sobolev space. Take $\ell(x,\hat{y},1) = -(\hat{y}-1/2)^2$ and $\ell(x,\hat{y},0) = (\hat{y}-1/2)^2$, which are each in $W^{1,2}([0,1])$. Next, we will argue the postprocessing π_ℓ is not a continuous function of p. In particular,

$$\begin{aligned}\pi_\ell(x,p) &= \operatorname*{arg\,min}_{\hat{y}\in[0,1]} \; p\cdot(-(\hat{y}-1/2)^2) + (1-p)\cdot(\hat{y}-1/2)^2 \\ &= \operatorname*{arg\,min}_{\hat{y}\in[0,1]} \; (1-2p)(\hat{y}-1/2)^2.\end{aligned}$$

is discontinuous in p. In particular for $p < 1/2$, the function evaluates to $c(\hat{y}-1/2)^2$ for some $c > 0$ and hence is minimized at $1/2$. For $p > 1/2$ the function evaluates to $c(\hat{y}-1/2)^2$ for some $c < 0$ and is hence minimized at either of the end points $\{0,1\}$. Then,

$$\partial\ell \circ \pi_\ell(x,p) = \begin{cases} 0 & \text{if } p < 1/2, \text{ and} \\ -1/2 & \text{otherwise,}\end{cases}$$

which is discontinuous and hence not in the Sobolev space since the space only contains continuous functions. ■

Thus, additional conditions on $\partial\mathcal{L}$ are necessary to ensure that $\partial\mathcal{L} \subseteq \mathcal{F}$ implies KDOI. In our main result in this subsection, Theorem 26, we identify natural conditions on $\mathcal{L}$ which do guarantee decision OI:

Proposition 26 *The following statements are true:*

(1) Let $\mathcal{L}_{\text{PS}}$ be the set of continuous and differentiable proper scoring rules $\ell(\hat{y}, y)$. That is,

$$\mathcal{L}_{\text{PS}} = \{\ell(\hat{y},y) : p \in \pi_\ell(p),\ \partial\ell \in W_1^{1,2}([0,1])\}$$

Then, there exists an efficient kernel k_{PS} satisfying $\mathcal{L}_{\text{PS}}$-KDOI with parameter $B_{\text{KDOI}} \leqslant \sqrt{3}$.

(2) Let $\mathcal{L}_{\text{SC}}$ be the set of continuous, smooth, strongly convex losses $\ell(\hat{y}, y)$. That is,

$$\mathcal{L}_{\text{SC}} = \{\ell(\hat{y},y) : \ell(\hat{y},y) \text{ is } \gamma \text{ strongly convex in } \hat{y},\ \forall\ y \in \{0,1\}, \text{ and } |\frac{d}{d\hat{y}}\ell(\hat{y},y)| \leqslant 1\}$$

Then, there exists an efficient kernel k_{SC} satisfying $\mathcal{L}_{\text{SC}}$-KDOI with parameter $B_{\text{KDOI}} \leqslant 2\sqrt{3}(3 + 2\gamma^{-1})$.

(3) Let $\mathcal{L}_m = \{\ell_1(\hat{y}, y), \ldots, \ell_m(\hat{y}, y)\}$ be any finite set of bounded functions with $|\mathcal{L}_m| \leqslant m$. Then, there exists an efficient kernel k_m satisfying $\mathcal{L}_m$-KDOI with parameter $B_{\text{KDOI}} \leqslant m$.

Moveover, if $\mathcal{L} = \mathcal{L}_{\text{PS}} \cup \mathcal{L}_{\text{SC}} \cup \mathcal{L}_{\text{m}}$, then the efficient kernel $k = k_{\text{PS}} + k_{\text{SC}} + k_m$ satisfies KDOI with constant $4\sqrt{3m}(2 + \gamma^{-1})$.

Proof We prove that each of the statements separately. Then, applying Theorem 56, which says that the union of RKHSs is an RKHS associated with the sum of each kernel function, implies the last statement.

Proof for $\mathcal{L}_{\text{PS}}$. π_ℓ is the identity function, so $\partial\ell \circ \pi_\ell = \partial\ell$. The result follows from the assumption that $\partial\ell$ is in $W^{1,2}([0,1])$ (see Example 5 for discussion) and the function norm is bounded by 1 and the feature norms are bounded by 3.

Proof for $\mathcal{L}_{\text{PS}}$. Our strategy will be to show that $\Pi_{\mathcal{L}} = \{\pi_\ell : \ell \in \mathcal{L}\}$ consist of functions in the Sobolev space $W^{1,2}([0,1])$. Then, we will apply Theorem 60, which states that the composition of functions in a Sobolev space $W^{1,2}([0,1])$ are in the space and the norm of the composition of functions in the space with bounded norm is bounded.

The convexity of ℓ in its second argument implies ℓ is differentiable almost everywhere and continuous. This implies that the discrete derivative function $\partial\ell$ is differentiable almost everywhere and continuous, which implies that $\partial\ell$ is in $W^{1,2}([0,1])$. Also, since $|\frac{d}{d\hat{y}}\ell(\hat{y}, y)| \leqslant 1$ and the range of ℓ is in $[-1,1]$

$$\begin{aligned}\|\partial\ell\|_{W^{1,2}([0,1])} &\leqslant \|\ell(\cdot,0)\|_{W^{1,2}([0,1])} + \|\ell(\cdot,1)\|_{W^{1,2}([0,1])} \\ &\leqslant 4\end{aligned}$$

Next, we show that π_ℓ is a Lipschitz function of p. The intuition is that, since ℓ is strongly convex, it has a unique minimum, and small changes to p cannot induce large changes in π_ℓ. Lipschitzness of π_ℓ implies $\pi_\ell \in W^{1,2}([0,1])$ since Lipschitz functions are absolutely continuous and hence differentiable almost everywhere and equal to their Lebesgue integral almost everywhere. The proof of Lipschitzness follows by using the same analysis used in Theorem 3.5 of (Perdomo et al., 2020) (albeit with slightly different assumptions). Let p and $\tilde{p}$ be two different predicted probabilities in $[0,1]$. Also, define:

$$f(\hat{y}) = p \cdot \ell(\hat{y}, 1) + (1-p) \cdot \ell(\hat{y}, 0) \tag{23}$$
$$\tilde{f}(\hat{y}) = \tilde{p} \cdot \ell(\hat{y}, 1) + (1-\tilde{p}) \cdot \ell(\hat{y}, 0) \tag{24}$$

and $f' = \partial f / \partial \hat{y}$. With this notation, we have that $\pi_\ell(p) \in \arg\min_{\hat{y}} f(\hat{y})$ and likewise $\pi_\ell(\tilde{p}) = \arg\min_{\hat{y}} \tilde{f}(\hat{y})$. First, we have that,

$$\begin{aligned}f(\pi_\ell(p)) - f(\pi_\ell(\tilde{p})) &\geqslant f'(\pi_\ell(\tilde{p}))(\pi_\ell(p) - \pi_\ell(\tilde{p})) + \frac{\gamma}{2}(\pi_\ell(p) - \pi_\ell(\tilde{p}))^2 \\ f(\pi_\ell(\tilde{p})) - f(\pi_\ell(p)) &\geqslant \frac{\gamma}{2}(\pi_\ell(p) - \pi_\ell(\tilde{p}))^2,\end{aligned}$$

where the first line follows by strong convexity of f, and the second line follows by strong convexity of f and the fact that $\pi_\ell(p)$ is the unique minimizer of f so $f'(\pi_\ell(p)) = 0$. Combining these two inequalities, we get that:

$$-\gamma(\pi_\ell(p) - \pi_\ell(\tilde{p}))^2 \geqslant f'(\pi_\ell(\tilde{p}))(\pi_\ell(p) - \pi_\ell(\tilde{p})). \tag{25}$$

Next, we derive a lower bound for $f'(\pi_\ell(\tilde{p}))(\pi_\ell(p) - \pi_\ell(\tilde{p}))$ in terms of $p, \tilde{p}$. Observe that, by definition,

$$f'(\pi_\ell(\tilde{p})) - \tilde{f}'(\pi_\ell(\tilde{p})) = (p - \tilde{p})\ell'(\pi_\ell(\tilde{p}), 1) + (1 - p - (1 - \tilde{p}))\ell'(\pi_\ell(\tilde{p}), 0).$$

Hence, $|f'(\pi_\ell(\tilde{p})) - \tilde{f}'(\pi_\ell(\tilde{p}))| \leqslant 2|p - \tilde{p}|$. Then, we get that,

$$\begin{aligned}(\pi_\ell(p) - \pi_\ell(\tilde{p}))f'(\pi_\ell(p)) &\geqslant (\pi_\ell(p) - \pi_\ell(\tilde{p}))f'(\pi_\ell(p)) - (\pi_\ell(p) - \pi_\ell(\tilde{p}))\tilde{f}'(\pi_\ell(\tilde{p})) \\ &\geqslant |\pi_\ell(p) - \pi_\ell(\tilde{p})| \cdot |f'(\pi_\ell(\tilde{p})) - \tilde{f}'(\pi_\ell(\tilde{p}))| \\ &\geqslant -2|\pi_\ell(p) - \pi_\ell(\tilde{p})| \cdot |p - \tilde{p}|\end{aligned}$$

where the first line follows from the fact that $\tilde{f}'(\pi_\ell(\tilde{p})) = 0$, and the second line follows from the first order optimality conditions for convex functions, $(\pi_\ell(p) - \pi_\ell(\tilde{p}))\tilde{f}'(\pi_\ell(\tilde{p})) \geqslant 0$. Combining this last chain of inequalities with Eq. (25), we get that

$$-\gamma(\pi_\ell(p) - \pi_\ell(\tilde{p}))^2 \geqslant -2|\pi_\ell(p) - \pi_\ell(\tilde{p})| \cdot |p - \tilde{p}|.$$

After simplifying and rearranging, we get $|\pi_\ell(p) - \pi_\ell(\tilde{p})| \leqslant 2\gamma^{-1}|p - \tilde{p}|$, so $\|\pi_\ell\|_{W^{1,2}([0,1])} \leqslant 2(1 + 2\gamma^{-1})$. Finally, using the kernel associated with $W^{1,2}([0,1])$, the feature norm is upper bounded by 3.

Proof for $\mathcal{L}_{(3)}$. We apply Theorem 59, which says that finite sets of functions taking values in $[-1,1]$ are in an RKHS with function and feature norms bounded by 1. Let the $\mathcal{X}$ in the lemma be $[0,1]$ and let $\mathcal{C} = \mathcal{L}_{(3)}$. Denote the induced RKHS $\mathcal{F}$. Then the lemma implies that $\|\ell\|_{\mathcal{F}} \leqslant 1$, and by the fact that $|\mathcal{F}_{(3)}| \leqslant m$ and losses are assumed to be bounded in $[-1,1]$, the feature norm must be bounded by m. ■

Intuitively, the previous says that if a loss class satisfies common regularity conditions like truthfulness (i.e. a proper scoring rule), smoothness/convexity, or is finite, then there exists a kernel satisfying KDOI. Additionally, it says that we can combine any sets of losses satisfying the above conditions and still satisfy KDOI. Notice that the Sobolev proper scoring losses include, for example, squared error, while the continuous, smooth and strongly convex losses $\mathcal{L}_{\text{SC}}$ include (ℓ_2 regularized) absolute error, Huber loss, and exponential loss. Losses that don't fit into the previous categories, such as the truncated cross-entropy loss, the 0-1 loss or the hinge loss may be included in the finite set of losses $\mathcal{L}_m$.

C.3. Comparator and loss classes satisfying kernel hypothesis OI.

Having analyzed how one can guarantee kernel *decision* OI with respect to common classes of losses, we now move only to analyze pairs $\mathcal{L}, \mathcal{H}$ that satisfy kernel *hypothesis* OI. That is, we aim to design kernels k with functions spaces $\mathcal{F}$ such that the functions $\ell \circ h \in \mathcal{F}$ (see Definition 23).

Regression trees. Our first result in this section shows one can guarantee kernel hypothesis OI for the class $\mathcal{H}$ of bounded-depth regression trees on binary features (an infinite comparator class) and $\mathcal{L}$ that consists of all bounded losses functions:

Proposition 27 *Let $\mathcal{H} \subseteq \{\{\pm 1\}^n \to \mathbb{R}\}$ be the set of all regression trees of depth at most $d \in \mathbb{N}$ over the boolean hypercube an let $\mathcal{L}$ be a set of all loss functions $\ell(\hat{y}, y)$ bounded in $[-1,1]$. There exists an computationally efficient kernel satisfying $(\mathcal{L}, \mathcal{H})$-KHOI with parameter B bounded by $(n+1)^{d/2} \cdot 2^d$.*

Proof We first note that regression trees on binary features are low degree polynomials, which are contained an RKHS $\mathcal{F}$ associated with the degree d polynomial kernel (see Example 2 for a definition and discussion of polynomial kernels).

To see this, we can write each tree in the following form: For a given regression tree, let $b \in \{0,1\}^d$ represent the path down the regression tree with mth element b_m. Let c_b be the leaf value assigned to path b. Let $i_{b,j}$ represent the index of the decision variable at the jth decision on path b. Then, any regression tree can be written in terms of $\{c_b\}_{b\in\{0,1\}^d}$ and $\{i_{b,j}\}_{b\in\{0,1\}^d, j\in[d]}$:

$$h(x) = \sum_{b\in\{0,1\}^d} c_b \prod_{m=0}^{d-1} ((1 - x_{i_{b,m}})(1 - b_m) + x_{i_{b,m}} b_m) \tag{26}$$

By distributing each product, combining like terms, and using the notation $x_I \stackrel{\text{def}}{=} \prod_{i\in I} x_i$, we can recover the following more concise expression:

$$h(x) = \sum_{I\in\mathcal{I}} a_I x_I \tag{27}$$

where $\mathcal{I} \subseteq 2^{\{0,1\}^d}$, $a_I \in \mathbb{R}$ for all $I \in \mathcal{I}$. Moreover, the latter form reveals that each nonzero a_I corresponds to some I with no more than d terms. Thus, $\mathcal{H} \subseteq \mathcal{F}$. (See Definition 3.13 in (O'Donnell, 2021) for more discussion of representing decision trees on Boolean inputs as polynomial functions.)

Next, notice that functions $\ell(h(\cdot), 1)$ and $\ell(h(\cdot), 0)$ for $\ell \in \mathcal{L}$ and $h \in \mathcal{H}$ can themselves be written as depth-r regression trees by taking each leaf value c_b of h and replacing it with $\ell(c_b, 0)$ and $\ell(c_b, 1)$, respectively. That is, for each $h \in \mathcal{H}$, we create two new trees $h_0, h_1 \in \mathcal{F}$ to be h with its leaf values replaced with the corresponding value of $\ell(c_b, y)$ for $y \in \{0, 1\}$. Finally, using Theorem 56 and Theorem 55, this implies that $\{\partial\ell \circ h \mid h \in \mathcal{H}, \ell \in \mathcal{L}\} \subseteq \mathcal{F}$.

Since there are 2^d leaves and each leaf has absolute value bounded by 1, $\|h_y\|_{\mathcal{F}} \leqslant 2^d$. Also, since the kernel function associated with $\mathcal{F}$ is $(1 + \langle x, x'\rangle)^d$, then $k(x, x)$ is bounded by $(1 + n)^d$. ■

Any finite set of real-valued functions $\mathcal{H}$. In our next construction, we show how to guarantee kernel hypothesis OI for the case where $\mathcal{H}$ is any finite set of comparator functions and $\mathcal{L}$ is a set of losses that can be represented in an RKHS.

This could of interest in setting where there are pre-specified predictors (like an existing link prediction system) that we would like the Any Kernel algorithm to compete with.

Proposition 28 *Let $\mathcal{H} = \{h_1, \ldots, h_m\}$ be any finite set of real-valued functions on $\mathcal{X}$ and let $\mathcal{L}$ be any set of loss functions $\ell(\hat{y}, y)$. Let k be a kernel with RKHS $\mathcal{F}$ such that $\mathcal{L} \subseteq \mathcal{F}$, $\|\ell\|_{\mathcal{F}} < 1$ for all $\ell \in \mathcal{L}$, and $\sup_t k(t, t) \leqslant 1$. Then,*

1. *There exists a kernel k' that is $(\mathcal{L}, \mathcal{H})$-KHOI with parameter B_{KHOI} at most $2\sqrt{m}$.*
2. *The kernel k' is computable in time at most $\mathcal{O}(m \cdot \mathsf{time}(k) \cdot \mathsf{time}(\mathcal{H}))$ where $\mathsf{time}(k)$ is a uniform upper bound on the runtime of the kernel k and $\mathsf{time}(\mathcal{H})$ is a uniform upper bound on the runtime of computing any function $h \in \mathcal{H}$.*

Proof

The main idea is that one can compose kernels in the following fashion. Let $k(t, t') : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a kernel with corresponding RKHS $\mathcal{F}$ such that $\ell(\cdot, 1)$ and $\ell(\cdot, 0)$ are both in $\mathcal{F}$ for all $\ell \in \mathcal{L}$. Then, for any fixed function $h_i : \mathcal{X} \to [0, 1]$, the kernel $k_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as:

$$k_i(x, x') = k(h_i(x), h_i(x'))$$

has an RKHS $\mathcal{F}_i$ which contains $\ell(h_i(x), 1)$ and $\ell(h_i(x), 0)$ for all $\ell \in \mathcal{L}$. Furthermore, if the functions $\ell(\cdot, 1)$ and $\ell(\cdot, 0)$ have norm at most 1 in $\mathcal{F}$, then the composed functions $\ell(h_i(\cdot), 1), \ell(h_i(\cdot), 0)$ will also have norm at most 1 in $\mathcal{F}_i'$ This is a neat fact from the theory of RKHSs (Theorem 58).

Since we can construct an RKHS for each $\partial\ell \circ h_i$ individually, we can construct an RKHS that contains all of the h_i simultaneously simply by summing the individual kernels together.

In particular, by Theorem 56, the kernel,

$$k'(x, x') = \sum_{h_i \in \mathcal{H}} k(h_i(x), h_i(x')) \tag{28}$$

contains $\partial\ell \circ h = \ell(h(\cdot), 1) - \ell(h(\cdot), 0)$ for all $h \in \mathcal{H}$ and $\ell \in \mathcal{L}$. Moreover, since each $\ell(h(x), y)$ has norm at most 1 (for $y \in \{0, 1\}$), then (by the triangle inequality) the functions $\partial\ell \circ h$ have norm at most 2 in the RKHS corresponding to k'. Furthermore,

$$\sup_x k'(x, x) = \sum_{h \in \mathcal{H}} k(h(x), h(x)) \leqslant m,$$

so the kernel k' is $(\mathcal{L}, \mathcal{H})$-KHOI with parameter B_{KHOI} bounded by $2\sqrt{m}$. ■

This result in particular implies that given any finite set of real-valued functions $\mathcal{H}$, we can guarantee kernel hypothesis OI when for all losses $\ell(\hat{y}, y)$ that are continuous and differentiable in $\hat{y}$. Given the previous construction in Theorem 26 showing that one can also guarantee kernel decision OI with respect to any finite class $\mathcal{H}$, this establishes that one can in fact guarantee omniprediction with respect to any finite set $\mathcal{H}$ and smooth losses $\mathcal{L}$ at rates $\mathcal{O}(\sqrt{T|\mathcal{H}|})$.

Asymptotic KHOI for all continuous functions. RKHSs can contain very rich function classes which can be used as benchmark classes. Indeed, some RKHSs are *universal approximators* in the sense that they contain arbitrarily precise approximations of all continuous functions.

Formally, an RKHS $\mathcal{F}$ is a universal approximator if, for all ε and continuous $g : \mathcal{X} \to \mathbb{R}$, there exists some $f \in \mathcal{F}$ such that $\sup_x |f(x) - g(x)| \leqslant \varepsilon$. Several common kernels like the Gaussian (or RBF) kernel, $k(x, x') = \exp(-\|x - x'\|^2)$ fall into this class. We refer the reader to (Steinwart, 2008), Section 4.6 for further examples and background.

Universal approximators can be used to guarantee KHOI with respect to any continuous benchmark function h and loss ℓ. However, the result is best understood in an asymptotic sense since it is not always tractable to control relevant function norms in the RKHS.

Here, we outline a general approach for doing so. The template matches those of similar results in the literature (see e.g. the discussion in Section C of (Foster and Kakade, 2006)). Let $\mathcal{H}$ be a comparison class of continuous functions and $\mathcal{L}$ be a class of continuous losses. Since the composition of continuous functions is continuous, the functions in $\partial\mathcal{L} \circ \mathcal{H}$ are also continuous. For a universal approximator $\mathcal{F}$, denote by $\mathcal{F}_\varepsilon \subseteq \mathcal{F}$ a set such that for all $\partial\ell \circ h \in \partial\mathcal{L} \circ \mathcal{H}$, there exists some $f \in \mathcal{F}_\varepsilon$ such that $\|f - \partial\ell \circ h\|_\infty \leqslant \varepsilon$. Define

$$B_\varepsilon = \inf_{\mathcal{F}_\varepsilon \subseteq \mathcal{F}} \sup_{f \in \mathcal{F}_\varepsilon} \|f\|_{\mathcal{F}}$$

be the infimum of a uniform upper bound on the norm of subsets $\mathcal{F}_\varepsilon$ satisfying the property. Notice that $B_\varepsilon \geqslant B_{\varepsilon'}$ for all $\varepsilon \leqslant \varepsilon'$ since any $\mathcal{F}_{\varepsilon'}$ satisfying the ε'-approximation property also satisfies ε-approximation. Then, one can chose a sequence ε_T for $T = 1, 2, \ldots$ such that $\lim_{T\to\infty} \varepsilon = 0$ and $B_{\varepsilon_T} = o(\sqrt{T})$. Then, the universal approximator can be used to satisfy an asymptotic, approximate version of KHOI with respect to $\mathcal{H}$ and $\mathcal{L}$.

C.4. Generalizing kernel OI to separable losses.

So far, we've established structural properties of losses $\ell(\hat{y}, y)$ that guarantee kernel decision and hypothesis OI. Here, we generalize these analyses to include losses that also depend on the features x. In particular, we prove that these requisite OI conditions also for a wide variety of *separable* loss functions $\ell(x, \hat{y}, y)$: those where each loss function can be factorized into a function of the feature vector x and of the decision-outcome pair $(\hat{y}, y)$.

Definition 29 (Separable Losses) *A loss function $\ell(x, \hat{y}, y)$ is separable if there exists functions $\ell_x : \mathcal{X} \to \mathbb{R}$ and $\ell_y : [0,1]^2 \to \mathbb{R}$ such that for all $(x, \hat{y}, y)$,*

$$\ell(x, \hat{y}, y) = \ell_x(x)\ell_y(\hat{y}, y).$$

Similarly, we say that a set of losses $\mathcal{L}$ For a separable loss class $\mathcal{L}$, we will define two new sets $\mathcal{L}_x$ and $\mathcal{L}_y$ to consist of the sets of the feature and decision-outcome components of the losses, respectively:

$$\mathcal{L} = \{\ell_x(x)\ell_y(\hat{y}, y) : \ell_x \in \mathcal{L}_x, \ell_y \in \mathcal{L}_y\}.$$

We refer to $\mathcal{L}_x$ and $\mathcal{L}_y$ as the factors *of the separable class $\mathcal{L}$.*

Separable loss classes capture many important examples of loss functions that depend on features. For example, $\mathcal{L}_x$ may consist of indicator functions for set membership, so that the loss only accumulates for members of a certain set. More generally, $\mathcal{L}_x$ can be interpreted to consist of any (re)weighting of the loss function over feature vectors x. These kinds of losses will be important for our results on link prediction at the end of this section.

We next state a simple result showing how to construct kernels for separable loss classes. Intuitively, the result says that any of the feature-invariant losses in the previous subsection can be reweighted by functions of the features x, as long as these functions are themselves in an RKHS with bounded norms.

Proposition 30 (Corollary to Theorem 57) *Let $\mathcal{L}$ be a separable class of losses with factors $\mathcal{L}_x, \mathcal{L}_y$ and let $\mathcal{H}$ be a comparator set of functions. Assume that k_x has an RKHS $\mathcal{F}_x$ such that $\mathcal{L}_x \subseteq \mathcal{F}_x$ and*

$$\sqrt{\sup_{\ell_x \in \mathcal{L}_x} \|\ell_x\|^2_{\mathcal{F}_x} \cdot \sup_{x \in \mathcal{X}} k_x(x, x)} \leqslant B_x.$$

1. *If k_y is a kernel that is $(\mathcal{L}_y, \mathcal{H})$-KHOI with parameter B_y. Then, then the product kernel,*

$$k((x,p),(x',p')) = k_x((x,p),(x',p')) \cdot k_y((x,p),(x',p')),$$

   is $(\mathcal{L}, \mathcal{H})$-KHOI with parameter $B_x B_y$.

2. *If k_y is a kernel that is $(\mathcal{L}_y)$-KDOI with parameter B_y. Then, then the same product kernel is $(\mathcal{L}, \mathcal{H})$-KDOI with parameter $B_x B_y$.*

Proof The result follows directly from Theorem 57, which says that the product of functions in an RKHS are contained in an RKHS and that the norm of the product function is no more than the product of norms of component the functions. ■

Letting the separable loss class be functions where $\mathcal{L}_x$ is composed of a set membership kernel (as described in Theorem 59 or any of the examples in Appendix B) and letting $\mathcal{L}_y$ consist of loss functions

$\ell_y(\hat{y}, y)$ which we know satisfy KDOI or KHOI from our previous analyses in Appendices C.2 and C.3 illustrates the expressive power of separable loss classes. In particular, $\mathcal{F}_x$ could consist of any collection of functions indexed by a set $\mathcal{I}$ where for all $x \in \mathcal{X}$ and $\ell_x \in \mathcal{L}_x \subseteq \mathcal{F}_x$, it holds $\sum_{i \in \mathcal{I}} \ell_i(x)^2 \leqslant B$. These could include, but are not limited to any finite set of group membership indicators. In this case, $k_x(x, x) \leqslant m$ and $\|\ell_x\|_{\mathcal{F}_x} \leqslant 1$. $\mathcal{F}_y$ could consist of any of the classic loss functions considered in Theorem 26 such as squared loss, log loss, or any bounded loss function.

We leave exploration of non-separable loss functions where $\ell(x, \hat{y}, y)$ cannot be written as a product to future work.

C.5. Guarantees for online regression.

Before moving onto to discussing the application of these techniques in the link prediction context, we briefly remark on how these ideas apply to the specific problem of online regression.

Online squared loss regression oracles are algorithms which generate a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^T$ satisfying the following guarantee:

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (p_t - y_t)^2 \leqslant \min_{h \in \mathcal{H}} \sum_{t=1}^{T} (h(x_t) - y_t)^2 + o(T). \tag{29}$$

In addition to being their intrinsic guarantees, online regression is a fundamental building block in the design of algorithms for other online learning problems like contextual bandits (Foster and Rakhlin, 2020) and online omniprediction (Garg et al., 2024).

Here, we show that whenever there exists a kernel k whose RKHS $\mathcal{F}$ contains a comparator class of functions $\mathcal{H} \subseteq \{\mathcal{X} \to \mathbb{R}\}$, then the Any Kernel algorithm run with the kernel k solves online regression.

Proposition 31 *Let $\mathcal{H}$ be a set of comparator functions and let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be an efficient kernel whose RKHS $\mathcal{F}$ satisfies, $\mathcal{H} \subseteq \mathcal{F}$ and $\|h\|_{\mathcal{F}} \leqslant 1$ for all $h \in \mathcal{H}$. Then, the Any Kernel algorithm algorithm instantiated with the kernel,*

$$k((x, p), (x', p')) = k(x, x') + pp' + 1$$

runs in polynomial time and generates a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^T$ satisfying,

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (p_t - y_t)^2 \leqslant \min_{h \in \mathcal{H}} \sum_{t=1}^{T} (h(x_t) - y_t)^2 + 6 \frac{\sqrt{1 + \sum_{t=1}^{T} \mathbb{E}_{p_t \sim \Delta_t} p_t(1 - p_t) k((x_t, p_t), (x_t, p_t))}}{T} \tag{30}$$

Proof The proof follows almost directly from Theorem 21. For the case of squared loss,

$$\begin{aligned} \partial\ell(x, \hat{y}) &= \ell(x, \hat{y}, 1) - \ell(x, \hat{y}, 0) \\ &= (\hat{y} - 1)^2 - (\hat{y} - 0)^2 \\ &= 1 - 2\hat{y}. \end{aligned}$$

Therefore, $\partial\ell(x, h(x)) = 1 - 2h(x)$ and $\partial\ell(x, \pi_\ell(p)) = 1 - 2p$ (since $\pi_\ell(p) = p$ for the squared loss).

By assumption the RKHS for k contains $h(x)$ and hence $2h(x)$ since RKHS are closed under scalar multiplication. Furthermore, the linear kernel $k_{\text{lin}}(p, p') = 1 + pp'$ has an RKHS that contains all affine functions $a + bp$. Moreoverv, both of these functions $1 - 2h(x)$ and $1 - 2p$ have norm at most 3 in the corresponding RKHS.

By adding these two kernels together, we can guarantee online OI with respect to the union of both distinguishers by Theorem 8. ■

In short, by specializing our omniprediction analysis to the case where $\mathcal{L}$ is a singleton set containing the squared loss, we show how to perform online regression with respect to any RKHS. Furthermore, the bounds have the advantage that they depend on the variance of the predictions p_t.[25] This result implies that the algorithms in (Garg et al., 2024) are unconditionally computationally efficient whenever the class $\mathcal{H}$ is contained in an RKHS.

It has been previously observed that, since online gradient descent kernelizes, any time $\mathcal{H}$ is in an RKHS, one can run online gradient descent (OGD) to produce an online squared error regression predictor (Foster and Rakhlin, 2020). And, in fact, there are various other algorithms for online regression (Azoury and Warmuth, 2001; Vovk, 2001), some of which achieve $\mathcal{O}(\log(T))$ regret (Hazan et al., 2007). The point of this analysis is that the Any Kernel algorithm is yet another alternative. Each algorithm has different trade-offs in terms of computational complexity and regret that justify use of one or the other in different contexts.

C.6. Specializing regret minimization to online link prediction.

As we outlined in the introduction to this paper, the link prediction problem has several distinctive properties that make it different from the traditional problems considered in prior work in online omniprediction (Garg et al., 2024; Gupta et al., 2022). In particular, the link prediction problem involves

(a) objectives that depend on characteristics of individuals or their communities;

(b) diverse and time-varying objectives, such as high predictive performance and encouraging desirable outcomes; and

(c) comparator classes that are particularly suited to graph settings, either because they are expressive, such as graph neural networks, or they leverage some interpretable structure of graphs, such as R-convolution kernels.

In the remainder of this section, we demonstrate how the results developed thus far can be instantiated so that the Any Kernel algorithm solves online omniprediction in the link prediction context.

Feature-dependent objectives. Depending on the way social networks affect outcomes, different properties of networks may be socially desirable. For example, platform may want to facilitate integration (Abebe et al., 2022; Calvo-Armengol and Jackson, 2004; Zeltzer, 2020; Stoica et al., 2018; Okafor, 2020) or encourage homophily or heterophily along different dimensions (McPherson et al., 2001; Kossinets and Watts, 2009; Zeltzer, 2020). It may be desirable to take into account structural cohesion measures (Eagle et al., 2010; Reagans and McEvily, 2003; Ugander et al., 2012; Granovetter, 1985) such as embeddedness. Our next result provides such a guarantee.

Proposition 32 *Suppose the sequence of graphs $\mathcal{G}_t$ is known to have nodes of degree bounded by a constant m and $\mathcal{L}$ consists of functions of the form $\ell(x, \hat{y}, y) = \nu(x)\gamma(\hat{y}, y)$, where*

(a) $\{\gamma : \ell = \nu \cdot \gamma, \ell \in \mathcal{L}\} \subseteq \mathcal{F}$ for an RKHS $\mathcal{F}$ associated with computationally efficient kernel k where $\mathcal{F}$ is KDOI with constant B_1, and

25. Bounds with this property are often referred to as second order bounds in the literature.

(b) ν *may be any of the tests described in Appendix B.2 (dropping dependence on the prediction* p*), including*

(i) any set of measures $\mathcal{F}' \subseteq \{\mathcal{U}^2 \to \mathbb{R}\}$ *of (dis)similarity of individuals where* $\sum_{f' \in \mathcal{F}'} \nu(u) \leqslant m$*, or*

(ii) any c*-embeddedness test for* $c \in \mathbb{N}$*:* $\nu(u,u') = 1\{\mathsf{Em}_t(u) = c\}$ *(or, more generally, any isomorphism indicator function* $1\{G_t \in \bar{G}\}$*).*

Additionally, suppose the exists an efficient kernel k *that is* $(\mathcal{L},\mathcal{H})$*-KHOI with parameter* B_{KHOI}*. Then there exists a computationally efficient kernel* k' *such that the Any Kernel algorithm instantiated with the kernel* k' *is an* $(\mathcal{L},\mathcal{H},(B_{\mathrm{KHOI}} + B_1(1+\sqrt{m}))\sqrt{T+1})$*-online omnipredictor.*

Proof We will show that $\mathcal{L}$ is KDOI with constant $B_1(1+\sqrt{m})$. With, Theorem 24, this will imply the result. Indeed, from Theorem 30 that, since $\mathcal{F}$ is KDOI with constant B_1, all we need to show is that functions in (i) have function and feature norm $\sqrt{m}$ and functions in (ii) by 1. Then, we can combine the RKHS for (i) with the one from (ii) with Theorem 56. The bound for (i) is proved in Theorem 13 and (ii) in Theorem 14, Theorem 15 for embeddedness tests and isomorphism indicators, respectively. ■

Diverse and time-varying objectives. Platforms may need to make predictions for a class of loss functions if they are taking multiple actions on the basis of a single prediction, or the loss function is not known until decision time, perhaps because a platform is running experiments to learn which of a class of losses is best to optimize for long-term objectives.

For a digital platform making link predictions, it may be important either to *forecast* how link formation will affect relevant properties of networks, or to *steer* the outcomes appropriately using recommendations. Many of the properties above can be encoded as loss functions in our setting, especially as separable losses Appendix C.4.

Proposition 33 *Suppose* $\mathcal{L}$ *consists of functions of the form* $\ell(x,\hat{y},y) = \nu(x)\gamma(\hat{y},y)$*, where*

(a) $\{\nu : \ell = \nu \cdot \gamma, \ell \in \mathcal{L}\} \subseteq \mathcal{F}$ *for an RKHS* $\mathcal{F}$ *associated with computationally efficient kernel* k *where* $\mathcal{F}$ *is KDOI with constant* B_1*, and*

(b) γ *may be*

(i) any of the feature-invariant losses described in Theorem 26,

(ii) any polynomial function $f : \{0,1\} \to [-1,1]$ *of outcomes* y *of degree no more than* d*, or*

(iii) any finite convex combination of functions $\{\gamma : \ell = \nu \cdot \gamma\}$ *satisfying (a) or (b).*

Additionally, suppose there exists a kernel k *that is* $(\mathcal{L},\mathcal{H})$*-KHOI with parameter* B_{KHOI}*. Then there exists a kernel* k' *such that the Any Kernel algorithm instantiated with the kernel is an* $(\mathcal{L},\mathcal{H},(B_{\mathrm{KHOI}} + B_1\sqrt{3+2^d}((4\sqrt{m}(3+\gamma^{-1}))+1)\sqrt{T+1})$*-online omnipredictor.*

Proof As in the proof of the previous proposition, we simply need to prove that $\mathcal{L}$ is in an RKHS that is KDOI with constant $B_1\sqrt{3+2^d}((4\sqrt{m}(3+\gamma^{-1}))+1)$, which implies the result. The bound on functions in (i) is $4\sqrt{m}(2+\gamma^{-1})$ from Theorem 26 and the bound on features is $\sqrt{3}$. For (ii), since the dimension of y is 1, the bound on functions is $1^d = 1$ for any polynomial of degree d by Theorem 9. The bound on the features is $2^{d/2}$, since $(1+\langle y,y'\rangle)^d \leqslant 2^d$. We do not need to add any constant for the functions in (iii) because of the fact that convex combinations and the triangle inequality imply that the norm of

any such function is no more than the norm of a function in parts (i) or (ii). We can combine the RKHSs associated with (i) and (ii) using Theorem 56: the function norm associated with this combined RKHS is $4\sqrt{m}(2+\gamma^{-1})+1$, and the feature norm is $\sqrt{3+2^d}$ By the Moore-Aronszajn theorem (Theorem 54) the functions in (iii) are in the RKHS that contains those in (i) and (ii) by the fact that RKHSs are closed under linear combinations and the triangle inequality.

■

Of course, in our setting, loss functions can only depend on features, decisions and outcomes, so networks can only hope to steer networks towards more desirable outcomes on a decision-by-decision basis. Elsewhere, this local optimization has been described as a best response in a game-theoretic formulation of the problem (Noarov et al., 2023), or a greedy algorithm for steering the network towards desirable outcomes. We leave an exploration of non-greedy, global approaches to network optimization to future work.

Graph-specific comparator classes. Link prediction has a long history and a rich literature (see e.g., (Martínez et al., 2016; Kumar et al., 2020), which we can use to build comparator classes in our kernel omniprediction framework. Broadly, comparator classes fall into two categories: those containing flexible, expressive models, and those containing simple, interpretable ones. Expressive classes can be used to show that the Any Kernel algorithm, instantiated with an appropriate kernel, can be used to compete with state-of-the-art and tailor-made models for a particular context, while the latter classes can be used to validate known dynamics, pass sanity checks, or guarantee trustworthiness with respect to the predictor.

For expressive comparator classes, any finite set of pre-existing graph neural network link predictors (Zhang and Chen, 2018; Yun et al., 2019) or other powerful predictive models can be used to instantiate Theorem 28, which, informally, says that the Any Kernel algorithm can compete with any finite set of pre-existing functions. Prior work (e.g., (Garg et al., 2024)) could not provide such guarantees because it required comparators to have binary rather than real-valued outputs.

On the other hand, especially in socially sensitive contexts or high stakes decisions, interpretable models can be important (see, e.g., (Rudin, 2019; Hays et al., 2023) for further discussion of interpretability in socially salient prediction). Interpretable function classes may include regression trees on pairs of node features or linear or polynomial regressions. They may also include the graph-specific models, like convolution kernels or other regression methods based on network topology as discussed in Appendix B.2.

C.7. Connections to Performative Prediction

We close this section with some brief remarks interpreting these loss minimization guarantees within the context performative prediction.

Recall that in the online prediction protocol, $x_t \in \mathcal{X}$ can chosen arbitrarily and in particular as a function of the history $\pi_{t-1} = \{(x_i, p_i, y_i)\}_{i=1}^{t-1}$. Outcomes y_t can be chosen as a function both of the history π_{t-1} and the current distribution over predictions Δ_t. Hence in this setup, both the features x_t and the outcomes y_t can be performative. That is, they can be a *function of the predictive model*. Furthermore, no restrictions are made regarding how Real Life responds to realized sequence of predictions. Please see (Perdomo et al., 2020; Hardt and Mendler-Dünner, 2023; Perdomo Silva, 2023) for further background on the performative prediction literature.

In particular, given an algorithm $\mathcal{A}$, let $\{(x_t(\mathcal{A}), \hat{y}_t(\mathcal{A}), y_t(\mathcal{A}))\}_{t=1}^{T}$ be the sequence of features, decisions and outcomes that are induced by making predictions $p_t \sim \Delta_t$ according to $\mathcal{A}$ in the online protocol where $\hat{y}_t = \pi_\ell(x_t, p_t)$. Similarly,let $\{(x_t(h), \hat{y}_t(h), y_t(h))\}_{t=1}^{T}$ be the sequence of features, predictions and outcomes that are induced by making predictions according to some other function h. The algorithms we

introduce in this section satisfy the following guarantee:

$$\frac{1}{T}\sum_{t=1}^{T}\ell(x_t(\mathcal{A}),\hat{y}_t(\mathcal{A}),y_t(\mathcal{A})) \leqslant \min_{h\in\mathcal{H}}\frac{1}{T}\sum_{t-1}^{T}\ell(x_t(\mathcal{A}),h(x_t(\mathcal{A})),y_t(\mathcal{A})) + o(1)$$

This condition states that, in hindsight over the sequence of data induced by the algorithm $\mathcal{A}$, no alternative h in the comparator class would have higher loss. We think of this as a version of online performative stability (see (Perdomo et al., 2020) for a formal definition of performative stability).

This is different than performative optimality.[26] The most natural definition for an algorithm $\mathcal{A}$ to guarantee performative optimality would be the following statement where we change the dependency structure on the right hand side of the bound above:

$$\frac{1}{T}\sum_{t=1}^{T}\ell(x_t(\mathcal{A}),\hat{y}_t(\mathcal{A}),y_t(\mathcal{A})) \leqslant \min_{h\in\mathcal{H}}\frac{1}{T}\sum_{t-1}^{T}\ell(x_t(h),h(x_t(h)),y_t(h)) + o(1). \tag{31}$$

While stability is about making good predictions in hindsight over the data that you induce, optimality is inherently a counterfactual statement. To achieve performative optimality, one compares performance not on the same data sequence, but on the data that *would have resulted* by making decisions according to some other function h. Our algorithms guarantee the former, but not the latter.

In the batch setting, we by now know how to achieve performative optimality (see *e.g.* (Miller et al., 2021)) and even performative omniprediction (Kim and Perdomo, 2023). We believe it is an interesting direction for future work to understand how one might guarantee online performative omniprediction. That is, algorithms which achieve the guarantee in Equation (31) simultaneously over many losses.

Appendix D. New Algorithms for Online Quantile & Vector Regression, Distance to Multicalibration, and Extensions to the Batch Case

As an addeded benefit of our investigation into kernel methods for online indistinguishability and omniprediction, we obtain algorithms for other, seemingly different, online prediction problems. In this section, we illustrate how to generalize the ideas presented previously beyond the binary setting to quantile regression and vector-valued predictions. As was true previously, the RKHS perspective provides a computationally efficient way to generate predictions that are indistinguishable with respect to rich classes of real-valued test functions in these settings.

In addition to these new algorithms, we also initiate the study of distance to multicalibration and prove that the classical problem of weak agnostic learning of a function class $\mathcal{F}$ can be solved efficiently whenever $\mathcal{F}$ is a reproducing kernel Hilbert space.

D.1. Quantile regression.

Unlike the binary case where means (i.e $\mathbb{E}[y \mid X = x]$) provide a complete description of the conditional distribution over outcomes, knowing the mean of a real-valued outcome y often provides a misleading picture of the future. In domains like finance and weather prediction where outcomes are noisy and heavy-tailed, y and $\mathbb{E}[y \mid x]$ can be very different. In these cases, we often often want estimates of best or worst case outcomes for y_t. Quantile prediction provides a rigorous way to estimate these best/worst case outcomes and quantify uncertainty.

26. Also note that both guarantees are the same if the data sequence (x_t, y_t) is not influenced by the predictions.

Prediction protocol. The online protocol for quantile calibration mirrors that of binary prediction. At every round t, Real Life chooses features $x_t \in \mathcal{X}$ arbitrarily, the learner chooses a distribution Δ_t over outcomes $p_t \in \mathbb{R}$. Finally, Nature selects a distribution o_t over outcomes $y_t \in [Y_{\min}, Y_{\max}]$, possibly as a function of Δ_t and x_t. Throughout this section, we will assume that Real Life selects outcomes from a Lipschitz distribution. This is a technical condition, standard in online quantile prediction (Roth, 2022), which requires that small changes in predictions also imply small changes in the CDF of y:

Definition 34 (Lipschitz Distribution) *A conditional label distribution o over outcomes $y \in [Y_{\min}, Y_{\max}]$ is ρ-Lipschitz continuous for some parameter $\rho > 0$ if for all $p_1, p_2 \in [Y_{\min}, Y_{\max}]$,*

$$\Pr_{y\sim o}[y \leqslant p_1] - \Pr_{y\sim o}[y \leqslant p_2]\| \leqslant \rho \cdot |p_1 - p_2|.$$

We aim to design online algorithms which satisfy the following guarantee:

Definition 35 (Online Quantile Indistinguishability)
An algorithm $\mathcal{A}$ guarantees online quantile indistinguishability with respect to class of functions $\mathcal{F}\{\mathcal{X} \times \mathbb{R} \to \mathbb{R}\}$ if it is guaranteed to generate a transcript $\{(x_t, \Delta_t, y_t)\}_{t=1}^T$ satisfying

$$\Big|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t\sim\Delta_t, y_t\sim o_t} (1\{y_t \leqslant p_t\} - q) f(x_t, p_t)\Big| \leqslant \mathcal{R}_{\mathcal{A}}(T, f)$$

for all $f \in \mathcal{F}$ where $\mathcal{R}_{\mathcal{A}}(T, f)$ is $o(T)$ for every f.

As discussed in previous sections, we refer to the above guarantee as indistinguishability instead of as multicalibration since we generally assume that the functions f are real-valued rather than binary valued. However, both terms are essentially interchangeable (Dwork et al., 2021).

Algorithm. The algorithm to guarantee online quantile calibration is almost identical to (randomized) version of the K29* algorithm for binary calibration. The only difference is that function S_t which the learner optimizes is slightly different.

$$S_t^q(p) \stackrel{\text{def}}{=} \sum_{i=1}^{t-1} k((x_t, p), (x_i, p_i))(1\{y_i \leqslant p_i\} - q) + \frac{1}{2} k((x_t, p), (x_t, p))(1 - 2q).$$

Guarantees. The proof for why this algorithm guarantees online quantile indistinguishability matches the template from previous analyses. The main idea is again to use the representer theorem to show that it suffices to bound the correlation between the quantile errors, $1\{y_t \leqslant p_t\} - q$, and the feature maps $\Phi(x_t, p_t)$:

$$\Big|\mathop{\mathbb{E}}_{p_t\sim\Delta_t, y_t\sim o_t} \sum_{t=1}^{T} (1\{y_t \leqslant p_t\} - q) f(x_t, p_t)\Big| = \Big|\mathop{\mathbb{E}}_{p_t\sim\Delta_t, y_t\sim o_t} \langle \sum_{t=1}^{T} (1\{y_t \leqslant p_t\} - q)\Phi(x_t, p_t), c\rangle_{\mathcal{F}}\Big| \tag{32}$$

$$\leqslant \|f\|_{\mathcal{F}} \cdot \mathbb{E}\Big\| \sum_{t=1}^{T} (1\{y_t \leqslant p_t\} - q)\Phi(x_t, p_t)\Big\|_{\mathcal{F}} \tag{33}$$

From this decomposition, we can leverage the defensive forecasting approach (Vovk et al., 2005; Shafer and Vovk, 2005; Vovk, 2007) to find a prediction strategy which guarantees that the last term,

$$\mathbb{E}\Big\| \sum_{t=1}^{T} (1\{y_t \leqslant p_t\} - q)\Phi(x_t, p_t)\Big\|_{\mathcal{F}},$$

The Quantile Any Kernel Algorithm

Input: A kernel $k : (\mathcal{X} \times [Y_{\min}, Y_{\max}])^2 \to \mathbb{R}$, quantile $q \in (0,1)$, bounds on outcome $[Y_{\min}, Y_{\max}]$

For $t = 1, 2, \ldots$:

1. Given $\{(x_i, p_i, y_i)\}_{i=1}^{t-1}$ and current features x_t define

$$S_t^q(p) \stackrel{\text{def}}{=} \sum_{i=1}^{t-1} k((x_t,p),(x_i,p_i))(1\{y_i \leqslant p_i\} - q) + \frac{1}{2}k((x_t,p),(x_t,p))(1-2q).$$

2. If $S_t^q(Y_{\min}), S_t^q(Y_{\max}) \geqslant 0$, **return** $\Delta_t = p_t = Y_{\min}$.
3. Else if, $S_t^q(Y_{\min}), S_t^q(Y_{\max}) \leqslant 0$, **return** $\Delta_t = p_t = Y_{\max}$.
4. Otherwise, let $B_t = \max_{t' \leqslant t} k((x_{t'}, p_{t'}), (x_{t'}, p_{t'}))$,
   - Run binary search to find $p_{t,1}$ and $p_{t,2}$ such that $S_t^q(p_{t,1})$ and $S_t^q(p_{t,2})$ have opposite signs and $|p_{t,1} - p_{t,2}| \leqslant 1/(10 \cdot B_t \cdot t^3)$.
   - **return**

$$\Delta_t = \begin{cases} p_{t,1} & \text{with probability } \tau \\ p_{t,2} & \text{with probability } 1-\tau. \end{cases} \qquad \text{for } \tau = \frac{|S_t(p_{t,2})|}{|S_t(p_{t,1})| + |S_t(p_{t,2})|} \in [0,1]$$

Figure 2: Extension of Any Kernel algorithm for quantiles. The algorithm is essentially identical to the Any Kernel algorithm, except that the S_t function has been defined slightly differently. As before, the algorithm is near-deterministic. The distribution Δ_t is either a point mass, or supported on two points that are very close together.

grows sublinearly, *i.e.* is bounded by $\mathcal{O}(\sqrt{T})$. As we now formalize in the following lemma, this is ensured by carefully choosing the $S_t^q(\cdot)$ function in the Quantile Any Kernel algorithm and incorporating the forecasting hedging ideas from (Foster and Hart, 2021). We break the analysis up into a series of lemmas:

Lemma 36 *Assume that the learner makes predictions in such a way that, for all choices of Nature,*

$$\mathbb{E}[S_t^q(p_t)(1\{y_t \leqslant p_t\} - q)] \leqslant \varepsilon_t$$

for all $t \geqslant 1$. Then,

$$\mathbb{E}\left\| \sum_{t=1}^{T} (1\{y_t \leqslant p_t\} - q) \cdot \Phi(x_t, p_t) \right\|_{\mathcal{F}}^2 \leqslant 2 \sum_{t=1}^{T} \varepsilon_t + \mathbb{E} \sum_{t=1}^{T} q(1-q) \left\| \Phi(x_t, p_t) \right\|_{\mathcal{F}}^2.$$

Proof By definition of S_t^q we have that $\sum_{t=1}^{T} \mathbb{E}[S_t^q(p_t)(1\{y_t \leqslant p_t\} - q)]$ is equal to:

$$\sum_{t=1}^{T} \sum_{i=1}^{t-1} k((x_t,p_t),(x_i,p_i))(1\{y_t \leqslant p_t\} - q)(1\{y_i \leqslant p_i\} - q) + \frac{1}{2} \sum_{t=1}^{T} k((x_t,p_t),(x_t,p_t))(1-2q)(1\{y_t \leqslant p_t\} - q).$$

Increasing the top limit of the first sum from $t-1$ to T, we can rewrite this as:

$$\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{T}k((x_t,p_t),(x_i,p_i))(1\{y_t \leqslant p_t\}-q)(1\{y_i \leqslant p_i\}-q) - \frac{1}{2}\sum_{t=1}^{T}k((x_t,p_t),(x_t,p_t))(1\{y_t \leqslant p_t\}-q)^2$$
$$+\frac{1}{2}\sum_{t=1}^{T}k((x_t,p_t),(x_t,p_t))(1-2q)(1\{y_t \leqslant p_t\}-p_t)$$

Now, using the identity that for binary v, $(v-q)^2 = q(1-q)+(1-2q)(v-q)$, we get:

$$\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{T}k((x_t,p_t),(x_i,p_i))(1\{y_t \leqslant p_t\}-q)(1\{y_i \leqslant p_i\}-q) - \frac{1}{2}\sum_{t=1}^{T}k((x_t,p_t),(x_t,p_t))q(1-q).$$

Finally, since $k((x_t,p_t),(x_i,p_i))(1\{y_t \leqslant p_t\}-q)(1\{y_i \leqslant p_i\}-q)$ is equal to

$$\langle \Phi(x_i,p_i)(1\{y_i \leqslant p_i\}-q), \Phi(x_t,p_t)(1\{y_t \leqslant p_t\}-q)\rangle_{\mathcal{F}},$$

we arrive at the identity that:

$$\sum_{t=1}^{T}\mathbb{E}[S_t^q(p_t)(1\{y_t \leqslant p_t\}-q)] = \frac{1}{2}\left\|\sum_{t=1}^{T}(1\{y_t \leqslant p_t\}-q)\cdot\Phi(x_t,p_t)\right\|_{\mathcal{F}}^2 - \frac{1}{2}\sum_{t=1}^{T}q(1-q)\left\|\Phi(x_t,p_t)\right\|_{\mathcal{F}}^2.$$

Lastly, by our assumption that $\mathbb{E}[S_t^q(1\{y_t \leqslant p_t\}-q)] \leqslant \varepsilon_t$, we get our desired result:

$$\mathbb{E}\left\|\sum_{t=1}^{T}(1\{y_t \leqslant p_t\}-q)\cdot\Phi(x_t,p_t)\right\|_{\mathcal{F}}^2 \leqslant 2\sum_{t=1}^{T}\varepsilon_t + \mathbb{E}\sum_{t=1}^{T}q(1-q)\left\|\Phi(x_t,p_t)\right\|_{\mathcal{F}}^2.$$

■

Given this result, the final step in the analysis is to show that the Quantile Any Kernel Algorithm generates predictions such that $\mathbb{E}[S_t^q(p_t)(1\{y_t \leqslant p_t\}-q)] \approx 0$.

Lemma 37 *Assume that the learner makes predictions $p_t \sim \Delta_t$ according to the Quantile Any Kernel algorithm and that Real Life selects outcomes y_t from a ρ-Lipschitz conditional distribution o_t, then*

$$\left|\mathop{\mathbb{E}}_{y_t\sim\Delta_t, p_t\sim o_t} S_t^q(p_t)(1\{y_t \leqslant p_t\}-q)\right| \leqslant \frac{1}{10t^2}\rho.$$

Proof If $S_t^q(Y_{\min})$ and $S_t^q(Y_{\max})$ are both non-negative or non-positive then the inequality,

$$S_t^q(p_t)(1\{y_t \leqslant p_t\}-q) \leqslant 0,$$

holds trivially regardless of the outcome y_t. If they have opposite signs, recall that by definition of the algorithm, the learner plays $p_{t,1}$ with probability $r_1 = \tau$ and $p_{t,2}$ with probability $r_2 = 1-r_1$. With his in mind,

$$\mathop{\mathbb{E}}_{y_t,p_t}\left[S_t^q(p)(1\{y_t \leqslant p_t\}-q)\right] = r_1\cdot S_t^q(p_{t,1})\,\mathbb{E}[1\{y_t \leqslant p_{t,1}\}-q] + r_2\cdot S_t^q(p_{t,2})\,\mathbb{E}[1\{y_t \leqslant p_{t,2}\}-q].$$

By adding and subtracting, $r_1 \cdot S_t^q(p_{t,1})\,\mathbb{E}[1\{y_t \leqslant p_{t,2}\} - q]$, we can rewrite this as,

$$[r_1 S_t^q(p_{t,1}) + r_2 S_t^q(p_{t,2})] \cdot \mathbb{E}[1\{y_t \leqslant p_{t,2}\} - q] + r_1 S_t^q(p_{t,1})\,\mathbb{E}[1\{y_t \leqslant p_{t,1}\} - 1\{y_t \leqslant p_{t,2}\}].$$

By choice of $r_1, p_{t,1}$ and $p_{t,2}$, we have that, $r_1 S_t^q(p_{t,1}) + r_2 S_t^q(p_{t,2}) = 0$, so the first term drops out. Then, since Real Life is required to select outcomes from a Lipschitz distribution,

$$\begin{aligned} r_1 S_t^q(p_{t,1})\,\mathbb{E}[1\{y_t \leqslant p_{t,1}\} - 1\{y_t \leqslant p_{t,2}\}] &\leqslant |S_t^q(p_{t,1})| \cdot |\Pr[y_t \leqslant p_{t,1}] - \Pr[y_t \leqslant p_{t,2}]| \\ &\leqslant |S_t^q(p_{t,1})| \cdot \rho \cdot |p_{t,1} - p_{t,2}|. \end{aligned}$$

The bound follows from the fact that $|S_t^q(p_{t,1})| \leqslant B_t$ and $|p_{t,1} - p_{t,2}| \leqslant 1/(10 \cdot B_t \cdot t^3)$. ■

Taken together, these lemmas establish the following theorem which summarizes the final guarantee of the Quantile Any Kernel algorithm.

Theorem 38 *Let k be a kernel with associated reproducing kernel Hilbert space $\mathcal{F}$. If outcomes y_t are drawn from a ρ-Lipschitz conditional distribution, then, the Quantile Any Kernel algorithm generates a transcript $\{(x_t, \Delta_t, y_t)\}$ such that for all $f \in \mathcal{F}$,*

$$\Big|\sum_{t=1}^{T} \mathbb{E}(1\{y_t \leqslant p_t\} - q) f(x_t, p_t)\Big| \leqslant \|f\|_{\mathcal{F}} \sqrt{\rho + \sum_{t=1}^{T} q(1-q)\,\mathbb{E}\, k((x_t, p_t), (x_t, p_t))}$$

Furthermore, if the kernel is bounded by B,

$$\sup_{(x,p) \in \mathcal{X} \times [0,1]} k((x,p),(x,p)) \leqslant B,$$

then the per round runtime of the algorithm is bounded by $\mathcal{O}(t \cdot \log(tB) \cdot \mathsf{time}(k))$, where $\mathsf{time}(k)$ is a uniform upper bound on the runtime of computing the kernel function k.

Discussion. To the best of our knowledge this is the first online algorithm for quantile regression with respect to functions spaces $\mathcal{F}$ that are an RKHS. As was the case with the Any Kernel algorithm, the algorithm is very simple to implement. At every time step, one only needs to solve a binary search problem over the unit interval. Furthermore, the guarantees are adaptive and illustrates how certain quantiles q (those closer to 0 or 1) lead to lower OI error bounds than those closer to 1/2. Lastly, the algorithm is hyperparameter free, one does not need to know the Lipschitz constant ρ ahead of time. The only requirement is that we know bounds $Y_{\min}, Y_{\max}$ on the outcome y.

D.2. Vector-valued, high-dimensional regression.

In addition to quantile regression, the RKHS and defensive forecasting viewpoint also provides a simple way of generating indistinguishable predictions in settings where outcomes are high-dimensional. That is, instead of binary or scalar-valued outcomes, in this subsection we consider the case where $y_t \in \mathcal{Y} \subset \mathbb{R}^d$ and $\mathcal{Y}$ is a compact, convex set (e.g $\mathcal{Y} = [-1, 1]^d$).

Formal setup. The online protocol is identical to that of scalar prediction. At every round t, Real Life chooses features $x_t \in \mathcal{X}$ arbitrarily, the learner chooses a distribution Δ_t over $p_t \in \mathcal{Y}$. Finally, Nature selects a distribution o_t over outcomes $y_t \in \mathcal{Y}$, possibly as a function of Δ_t and x_t.

Definition 39 (Online Vector-Valued Indistinguishability) *An algorithm $\mathcal{A}$ guarantees online high-dimensional indistinguishability with respect to class of functions $\mathcal{F} \subseteq \{\mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d\}$ if it is guaranteed to generate a transcript satisfying the following guarantee,*

$$\Big|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t, y_t \sim o_t} (y_t - p_t)^\top f(x_t, p_t)\Big| \leqslant \mathcal{R}_{\mathcal{A}}(T, f)$$

where $\mathcal{R}_{\mathcal{A}} : \mathbb{N} \times \mathcal{F} \to \mathbb{R}$ is $o(T)$ for every f.

Note that in this setting the test functions $c(x_t, p_t)$ are *vector-valued*. High-dimensional indistinguishability asks that, when averaged over the sequence, prediction errors $y_t - p_t$ are uncorrelated with any test function $f \in \mathcal{F}$,

$$\lim_{t\to\infty} \frac{1}{T} \sum_{t=1}^{T} (y_t - p_t)^\top f(x_t, p_t) = 0.$$

Background on vector-valued RKHSs As was the case previously, the algorithm has guarantees with respect to set functions that form an RKHS, but in this functions take values in $\mathbb{R}^d$ rather than $\mathbb{R}$. A vector-valued RKHS is a set of functions $\mathcal{F} \subset \{\mathcal{X} \to \mathbb{R}^d\}$, where the set $\mathcal{F}$ is itself a Hilbert space, equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$.

A kernel K for a vector-valued RKHS is a mapping from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}^{d\times d}$. To disambiguate from the scalar case, we use capital K to denote matrix-valued kernels and lower case k to denote a scalar-valued kernel.

For a more comprehensive background on vector-valued kernels, we refer the reader to the excellent survey by Alvarez, Rosasco, and Lawrence (Álvarez et al., 2012). For the context of our results, we will only need two main facts. First, as with the scalar case, the kernel K has the reproducing property such that for any function $f : \mathcal{X} \to \mathbb{R}^d$ in the RKHS and vector $v \in \mathbb{R}^d$.

$$f(z)^\top v = \langle f, \Phi(z)v \rangle_{\mathcal{F}} \tag{34}$$

Here $\Phi(x)$ is the feature map of x. For any fixed x, $\Phi(x)$ is a mapping from $\mathbb{R}^d$ to $\mathcal{F}$. The last property we need is part a) from Proposition 2.1 in (Micchelli and Pontil, 2005) which states that for any $x, x' \in \mathcal{X}$ and $v, v' \in \mathbb{R}^d$:

$$v^\top K(z, z')v' = \langle \Phi(z')v', \Phi(z)v \rangle_{\mathcal{F}} \tag{35}$$

Algorithmic guarantees. As before, the advantage of this approach is that the final algorithm has strong guarantees of performance, and is additionally very simple to state and analyze. The main computational difference relative to previous settings is that the learner needs to solve a *variational inequality* (Eqs. (36) and (37)). Variational inequalities are a rich and well-developed area of research within the optimization literature (Kinderlehrer and Stampacchia, 2000; Noor, 1988), with earliest work dating back to the papers by Signori and Fichera (Fichera, 1963). These optimization problems always have a solution. Furthermore, these solutions can be found efficiently in various settings.

However, before discussing these ideas further, we state the final end-to-end result for the Vector Any Kernel algorithm:

Theorem 40 *Let K be a kernel for a vector-valued reproducing kernel Hilbert space $\mathcal{F}$. Then, the Vector Any Kernel algorithm is guaranteed to generate a transcript such that for any $f \in \mathcal{F}$,*

$$\left|\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (y_t - p_t)^\top f(x_t, p_t)\right| \leqslant \|f\|_{\mathcal{F}} \sqrt{\sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_t \sim \Delta_t} (y_t - p_t)^\top K((x_t, p_t), (x_t, p_t))(y_t - p_t)}.$$

The Vector Any Kernel Algorithm

Input: A compact, convex set $\mathcal{Y} \subseteq \mathbb{R}^d$, a kernel $K : (\mathcal{X} \times \mathcal{Y})^2 \to \mathbb{R}^{d \times d}$

For $t = 1, \ldots,$:

1. Given history $\{(x_i, p_i, y_i)\}_{i=1}^{t-1}$ and current features x_t define

$$S_t(p) \stackrel{\text{def}}{=} \sum_{i=1}^{t-1} K((x_t, p), (x_i, p_i))(y_i - p_i) \in \mathbb{R}^d$$

2. If K is continuous in p, **return** $\Delta_t = p_t \in \mathcal{Y}$ that solves the variational-inequality:

$$\sup_{y \in \mathcal{Y}} (y - p_t)^\top S_t(p) \leqslant 0 \tag{36}$$

For discontinuous kernels, **return** Δ_t, where Δ_t is a distribution over $p_t \in \mathcal{Y}$ satisfying

$$\mathop{\mathbb{E}}_{p_t \sim \Delta_t} \sup_{y \in \mathcal{Y}} (y - p_t)^\top S_t(p) \leqslant 0 \tag{37}$$

Figure 3: Extension of the Any Kernel algorithm for high-dimensional prediction. For simplicity, we state the algorithm assuming that the variational inequalities are solved exactly. However, as illustrated previously in quantile and binary prediction, the analysis can be easily modified to accomodate approximate solutions. The behavior of the algorithm for continuous kernels is the same as in (Vovk et al., 2005). The extension to the discontinuous case is new.

If we further assume that the kernel K is uniformly bounded by B over $\mathcal{X} \times \mathcal{Y}$, and that the diameter of the set $\mathcal{Y}$ is at most D,

$$\sup_{x \in \mathcal{X}, p \in \mathcal{Y}} \|K((x,p),(x,p))\|_{\text{op}} \leqslant B, \qquad \sup_{p, p' \in \mathcal{Y}} \|p - p'\|_2^2 \leqslant D$$

then, the above guarantee implies that:

$$\left| \sum_{t=1}^{T} \mathbb{E}(y_t - p_t)^\top c(x_t, p_t) \right| \leqslant \|c\|_{\mathcal{F}} \sqrt{BDT}.$$

Furthermore, the per round runtime of the algorithm is at most $\mathcal{O}(t\mathsf{timeVE})$ where timeVE is an upper bound on the time it takes solve the variational inequality problems in Equation (36) and Equation (37).

Proof We start the analysis by again showing that it suffices to bound the correlation between the features $\Phi(x_t, p_t)$ and the errors $(y_t - p_t)$. Using the reproducing property for vector-valued RKHSs, Eq. (34), we

first show the following bound:

$$\begin{aligned}\left|\mathbb{E}\sum_{t=1}^{T}(y_t-p_t)^\top c(x_t,p_t)\right| &= \left|\mathbb{E}\sum_{t=1}^{T}\langle c,\Phi(x_t,p_t)(y_t-p_t)\rangle_{\mathcal{F}}\right| \\ &= \left|\left\langle c,\sum_{t=1}^{T}\mathbb{E}\,\Phi(x_t,p_t)(y_t-p_t)\right\rangle_{\mathcal{F}}\right| \\ &\leqslant \|c\|_{\mathcal{F}}\cdot\left\|\sum_{t=1}^{T}\mathbb{E}\left[\Phi(x_t,p_t)(y_t-p_t)\right]\right\|_{\mathcal{F}}. \end{aligned} \tag{38}$$

Next, we show that the Vector Any Kernel algorithm bounds the second term. In particular, by construction, the algorithm guarantees that:

$$\mathop{\mathbb{E}}_{p_t\sim\Delta_t,\Delta_t\sim o_t}(y_t-p_t)^\top S_t(p_t)\leqslant 0 \text{ where } S_t(p)=\sum_{i=1}^{t-1}K((x_t,p),(x_i,p_i))(y_i-p_i).$$

Summing up this quantity over all T rounds,

$$\begin{aligned}0 &\geqslant \sum_{t=1}^{T}\sum_{i=1}^{t-1}\mathbb{E}\Big[(y_t-p_t)^\top K((x_t,p_t),(x_i,p_i))(y_i-p_i)\Big] \\ &= \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{T}\mathbb{E}\Big[(y_t-p_t)^\top K((x_t,p_t),(x_i,p_i))(y_i-p_i)\Big] \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}\mathbb{E}\Big[(y_t-p_t)^\top K((x_t,p_t),(x_t,p_t))(y_t-p_t)\Big].\end{aligned}$$

Hence,

$$\sum_{t=1}^{T}\sum_{i=1}^{T}\mathbb{E}\Big[(y_t-p_t)^\top K((x_t,p_t),(x_i,p_i))(y_i-p_i)\Big]\leqslant\sum_{t=1}^{T}\mathbb{E}\Big[(y_t-p_t)^\top K((x_t,p_t),(x_t,p_t))(y_t-p_t)\Big]. \tag{39}$$

Now, by applying Eq. (35), we see that,

$$\begin{aligned}(y_t-p_t)^\top K((x_t,p_t),(x_t,p_t))(y_t-p_t) &= \langle\Phi(x_t,p_t)(y_t-p_t),\Phi(x_t,p_t)(y_t-p_t)\rangle_{\mathcal{F}} \\ &= \left\|\Phi(x_t,p_t)(y_t-p_t)\right\|_{\mathcal{F}}^2\end{aligned} \tag{40}$$

And,

$$\sum_{t=1}^{T}\sum_{i=1}^{T}(y_t-p_t)^\top K((x_t,p_t),(x_i,p_i))(y_i-p_i)=\left\|\sum_{t=1}^{T}\Phi(x_t,p_t)(y_t-p_t)\right\|_{\mathcal{F}}^2 \tag{41}$$

Combining Eqs. (39) to (41) (and Jensen's inequality) we get that the Vector Any Kernel algorithm generates sequence satisfying,

$$\left\|\mathbb{E}\left[\Phi(x_t,p_t)(y_t-p_t)\right]\right\|_{\mathcal{F}}^2\leqslant\mathbb{E}\left[\left\|\sum_{t=1}^{T}\Phi(x_t,p_t)(y_t-p_t)\right\|_{\mathcal{F}}^2\right]\leqslant\sum_{t=1}^{T}\mathbb{E}\left[\left\|\Phi(x_t,p_t)(y_t-p_t)\right\|_{\mathcal{F}}^2\right]$$

Together with the first inequality, Eq. (38), we get our desired data-dependent guarantee,

$$\left|\sum_{t=1}^{T}\mathbb{E}(y_t - p_t)^\top c(x_t, p_t)\right| \leqslant \|c\|_{\mathcal{F}}\sqrt{\sum_{t=1}^{T}\mathbb{E}\left[\left\|\Phi(x_t, p_t)(y_t - p_t)\right\|_{\mathcal{F}}^2\right]}.$$

■

Variational inequalities. As seen from the description of the algorithm, the main computational step is the Vector Any Kernel algorithm is to solve for a vector p_t, or a distribution Δ_t over vectors p_t that satisfies,

$$(y - p_t)^\top S_t(p) \leqslant 0 \quad \forall y \in \mathcal{Y}.$$

From a first glance, it is not obvious that such a p_t exists. However, in a recent, related paper on online calibration, Foster and Hart show that these "outgoing fixed points" exists under very mild conditions. We restate their result below:

Proposition 41 (Theorem 4 & Corollary 6 in (Foster and Hart, 2021)) *Let $\mathcal{Y} \subset \mathbb{R}^d$ be a compact, convex set and let $S : \mathcal{Y} \to \mathbb{R}^d$ be a continuous function. Then, there exists a point $p_* \in \mathcal{Y}$ such that,*

$$(y - p_*)^\top S(p_*) \leqslant 0 \quad \forall y \in \mathcal{Y}.$$

If $S : \mathcal{Y} \to \mathbb{R}^d$ is not necessarily continuous, but bounded in the sense that,

$$\sup_{y \in \hat{\mathcal{Y}}} \|S(y)\|_2 < \infty,$$

then, for all $\varepsilon > 0$ there exists a distribution Δ supported on at most $d + 3$ points in $\hat{\mathcal{Y}}$ such that,

$$\mathop{\mathbb{E}}_{p_* \sim \Delta}(y - p_*)^\top S(p_*) \leqslant \varepsilon \quad \forall y \in \hat{\mathcal{Y}}.$$

Not only do these fixed points exists, but by now there is an increasingly extensive literature on algorithms for finding them (Censor et al., 2012; Burachik and Iusem, 1998; Kinderlehrer and Stampacchia, 2000; Noor, 1988) under various regularity conditions on the function S.

Discussion. The Vector Any Kernel algorithm is most closely related to the K29 (not star) algorithm from Vovk (Vovk et al., 2005; Vovk, 2007). By using the forecast hedging idea from (Foster and Hart, 2021), we extend the algorithm so that it works for any matrix valued kernel. Modulo this extension, the regret guarantees are nearly identical.

To the best of our knowledge, the other most closely related work is the recent paper by Noarov, Ramalingam, Roth, and Xie (Noarov et al., 2023). Using different techniques to ours (from online minimax optimization), they introduce an algorithm that achieves the following guarantee,

$$\|\sum_{t=1}^{T}(y_t - p_t)^\top f(x_t, p_t)\|_\infty \leqslant \mathcal{O}(\sqrt{T}).$$

This is essentially the same goal we consider (up to poly d factors). However, their result holds with respect to functions f taking values in $\{0, 1\}^d$ (they refer to f as events) and sets $\mathcal{F}$ which are finite. In our case, $|\mathcal{F}|$ is infinite and $\mathcal{F}$ is real-valued since it is an RKHS.

Furthermore, their runtime is guarateed to be polynomial whenever $|\mathcal{F}|$ is polynomially sized whereas our results are best understood as being oracle efficient. The algorithm runs in polynomial time whenever there exists an efficient oracle that can solve the corresponding variational inequality. These efficent algorithms exist for instance when the functions S are *monotone*, however they may be computationally difficult in general.

Please see the supplemantary material for results on how one can design matrix valued kernels whose corresponding RKHS contain an arbitrary finite set of functions $\mathcal{F} \subseteq \{\mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d\}$.

D.3. Distance to online multicalibration.

In this subsection, we show that instantiating the Any Kernel algorithm with a particular kernel k achieves small *distance to online multicalibration*, a novel extension of the canonical notion of *distance to (online) calibration* from (Błasiok et al., 2023; Qiao and Zheng, 2024) which we introduce in this paper.

To start, we start by recalling what it means for a predictor to be perfectly calibrated and restate the definition of distance to calibration from (Błasiok et al., 2023; Qiao and Zheng, 2024).

Definition 42 (Perfect Online (Multi)Calibration) *Suppose we are given fixed sequences of predictions $\boldsymbol{p} = (p_1, \ldots, p_T) \in [0,1]^T$, features $\boldsymbol{x} = (x_1, \ldots, x_T) \in \mathcal{X}^T$, outcomes $\boldsymbol{y} = (y_1, \ldots, y_T) \in \{0,1\}^T$, and a collection $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ of group indicator functions. We say that $\boldsymbol{p}$ is* perfectly multicalibrated with respect to the collection $\mathcal{C}$ *if for all $v \in [0,1]$ and $c \in \mathcal{C}$,*

$$\sum_{t=1}^{T} (y_t - v) c(x_t) \mathbf{1}[p_t = v] = 0.$$

Likewise, we say that a prediction is perfectly calibrated if it is multicalibrated with respect to the collection $\mathcal{C}$ that just contains the constant 1 function.

Given a function $c : \mathcal{X} \to \{0,1\}$, let $\mathrm{PC}(c)$ denote the set of prediction sequences $\boldsymbol{q} = (q_1, \ldots, q_T) \in [0,1]^T$ that are perfectly calibrated on c. Let $\mathrm{PC}(\mathcal{C})$ be the intersection of $\mathrm{PC}(c)$ for all $c \in \mathcal{C}$.

While defining perfect calibration is relatively straightforward, defining distance to calibration is not. In their recent work, (Błasiok et al., 2023) propose a unifying notion of distance to calibration. Here, we state the online version of their definition as presented in (Qiao and Zheng, 2024).

Definition 43 (Distance to Online Calibration (Qiao and Zheng, 2024)) *Suppose we are given fixed sequences of predictions $\boldsymbol{p} = (p_1, \ldots, p_T) \in [0,1]^T$, features $\boldsymbol{x} = (x_1, \ldots, x_T) \in \mathcal{X}^T$, outcomes $\boldsymbol{y} = (y_1, \ldots, y_T) \in \{0,1\}^T$. The* distance to online calibration *is*

$$\mathrm{dCE}_{\boldsymbol{y}}(\boldsymbol{p}) = \inf_{\boldsymbol{q} \in \mathrm{PC}(1)} \sum_{t=1}^{T} |p_t - q_t|,$$

where $1 : \mathcal{X} \to \{0,1\}$ denotes the all-ones function.

With these definitions in hand, we now introduce our definition of distance to (online) *multicalibration*. Given a collection $\mathcal{C}$ of group indicator functions there are several ways of defining distance to multicalibration. Here, we present two such versions, showing how one is efficiently achievable and the other is in fact impossible to achieve in general.

Definition 44 (Distance to Online Multicalibration, Standard and Strong Variants) *Suppose we are given fixed sequences of predictions $\boldsymbol{p} = (p_1, \ldots, p_T) \in [0,1]^T$, features $\boldsymbol{x} = (x_1, \ldots, x_T) \in \mathcal{X}^T$, outcomes $\boldsymbol{y} = (y_1, \ldots, y_T) \in \{0,1\}^T$, and a collection $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ of group indicator functions.*

We define the distance to online multicalibration $\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}$ *and* strong distance to online multicalibration $\mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}}$ *as follows:*

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) = \sup_{c \in \mathcal{C}} \inf_{\boldsymbol{q} \in \mathrm{PC}(c)} \sum_{t=1}^{T} |p_t - q_t|,$$

$$\mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) = \inf_{\boldsymbol{q} \in \mathrm{PC}(\mathcal{C})} \sum_{t=1}^{T} |p_t - q_t|,$$

where $\mathrm{PC}(\mathcal{C})$ *is as defined in Definition 42.*

Several remarks about Definition 44 are in order. First, it is easy to see that even the first of these two notions of distance to multicalibration is still stronger than a global notion of distance to calibration. For example, in the online setting, consider a single subsequence indicator $c : \mathcal{X} \to \{0,1\}$ such that for each $t = 1, \ldots, T$,

$$c(x_t) = \begin{cases} 1 & \text{if } t \text{ is odd} \\ 0 & \text{if } t \text{ is even.} \end{cases}$$

Suppose the outcome sequence y follow the same pattern, so $y_t = c(x_t)$, but we predict $p_t = 1/2$ for all time steps $t \in [T]$. In this case, $\boldsymbol{p}$ will be perfectly calibrated with respect to $\boldsymbol{y}$ in a global sense, but $\mathrm{dMCE}_{\boldsymbol{y},\{c\}}(\boldsymbol{p}) = T/4 = \Omega(T)$.

Next, observe that in the definition of distance to online multicalibration, the constraint $q \in \mathrm{PC}(c)$ only restricts the values that q takes during time steps $t \in [T]$ such that $c(x_t) = 1$. In other words, during time steps for which $c(x_t) = 0$, it is clearly optimal to take $q_t = p_t$ if the goal is to minimize the sum on the right side, because this ensures that the t^{th} term satisfies $|p_t - q_t| = 0$. Consequently, we have the equality

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) = \sup_{c \in \mathcal{C}} \inf_{\boldsymbol{q} \in \mathrm{PC}(c)} \sum_{t=1}^{T} |p_t - q_t| c(x_t)$$

Next, we establish the relationship between our standard and strong notions of distance to online multicalibration:

Theorem 45 *For any prediction, feature, and outcome sequences, and for any collection $\mathcal{C}$,*

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) \leqslant \mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}).$$

Moreover, this inequality can be strict; in fact, there exists a distribution over feature and outcome sequences, as well as a collection $\mathcal{C}$, such that for any prediction algorithm used to generate $\boldsymbol{p}$,

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) \leqslant O(1)$$

but with high probability,

$$\mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) \geqslant \Omega(T).$$

Proof Using the fact that $\boldsymbol{q} \in \mathrm{PC}(\mathcal{C})$ necessarily belongs to $\mathrm{PC}(c)$ for each $c \in \mathcal{C}$, it is clear that

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) \leqslant \mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p})$$

for any prediction sequence $\boldsymbol{p}$. To see that this inequality can be strict, consider a setting in which $\mathcal{X} = \mathbb{N}$ and $x_t = t$ at each time step $t \in [T]$. Consider the collection $\mathcal{C}_{\mathrm{singleton}}$ consisting of all "singleton" indicator functions c_t of the form $c_t(s) = \mathbf{1}[s = t]$ for some fixed $t \in [T]$. In this case, being perfectly calibrated on the set $\{t\}$ amounts to exactly predicting the t^{th} bit—in other words, the event that $p_t = y_t \in \{0, 1\}$. Consequently, the set $\mathrm{PC}(\mathcal{C}_{\mathrm{singleton}})$ of perfectly $\mathcal{C}$-multicalibrated prediction sequences is a singleton set that only contains the true outcome sequence $\boldsymbol{y}$, which implies that

$$\mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}_{\mathrm{singleton}}}(\boldsymbol{p}) = \sum_{t=1}^{T} |p_t - y_t|.$$

On the other hand, using the aforementioned characterization of the standard notion of distance to online multicalibration, we see that

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}_{\mathrm{singleton}}}(\boldsymbol{p}) = \max_{t \in [T]} |p_t - y_t|,$$

the *maximum* error made at any particular time step. In particular, in this example, we have that $\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}_{\mathrm{singleton}}}(\boldsymbol{p}) \leqslant 1$ for any prediction sequence $\boldsymbol{p}$. However, if $y_t \in \{0, 1\}$ is sampled uniformly and independently of the history of predictions and outcomes before time step t, we have $\mathrm{dMCE}^{\mathrm{strong}}_{\boldsymbol{y},\mathcal{C}_{\mathrm{singleton}}}(\boldsymbol{p}) \geqslant \Omega(T)$ with high probability, regardless of the algorithm used to make the predctions at each time step. ■

To conclude this section, we show that the Any Kernel algorithm can be used to achieve small distance to online multicalibration, provided that we aim for the standard notion, as opposed to the strong notion.

Theorem 46 *Given a collection $\mathcal{C}$ of indicator functions for subpopulations of a population $\mathcal{X}$, let $k_{\mathrm{Lap}} = k_{\mathbb{R}}$ be the* Laplace kernel *as defined in Example 5, let $\mathsf{Int}_{\mathcal{C}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ denote the intersection kernel*

$$\mathsf{Int}_{\mathcal{C}}(x, x') = |\{c \in \mathcal{C} : c(x) = c(x') = 1\}|,$$

and let $k_{\mathrm{MC}} : (\mathbb{R} \times \mathcal{X}) \times (\mathbb{R} \times \mathcal{X}) \to \mathbb{R}$ denote the product kernel

$$k_{\mathrm{MC}}((p, x), (p', x')) = k_{\mathrm{Lap}}(p, p') \cdot \mathsf{Int}_{\mathcal{C}}(x, x'),$$

which is uniformly bounded by

$$m = \max_{x \in \mathcal{X}} |\{c \in \mathcal{C} : c(x) = 1\}|.$$

Let $\pi_{1:T} = \{(x_t, p_t, y_t)\}_{t=1}^{T}$ denote the transcript at the end of the Any Kernel algorithm when instantiated with the kernel k_{MC}. Then,

$$\mathrm{dMCE}_{\boldsymbol{y},\mathcal{C}}(\boldsymbol{p}) \leqslant \mathcal{O}(\sqrt{mT}).$$

Proof Theorem 8 guarantees that the transcript ultimately satisfies

$$\left| \sum_{t=1}^{T} (p_t - y_t) f(p_t) c(x_t) \right| \leqslant \sqrt{mT + 1}$$

for all f with norm at most 1 in the RKHS corresponding to k_{Lap}, and for all $c \in \mathcal{C}$ (these have norm at most 1 in the RKHS corresponding to $\mathsf{Int}_{\mathcal{C}}$ by Theorem 59). Next, we fix a particular function $c \in \mathcal{C}$ and rewrite this inequality as

$$\left| \sum_{\substack{t \in [T] \\ c(x_t)=1}} (p_t - y_t) f(p_t) \right| \leqslant \sqrt{mT+1}.$$

Letting $\boldsymbol{y}_c, \boldsymbol{p}_c \in [0,1]^{|S|}$ denote the restriction of $\boldsymbol{y}, \boldsymbol{p} \in [0,1]^T$ to the set S of $t \in [T]$ for which $c(x_t) = 1$, this implies that the *kernel calibration error*, defined as follows, also is at most $\sqrt{mT+1}$:

$$\text{kCE}^{k_{\text{Lap}}}_{\boldsymbol{y}_c}(\boldsymbol{p}_c) := \sup_{f : \|f\|_{\text{Lap}} \leqslant 1} \left| \sum_{\substack{t \in [T] \\ c(x_t)=1}} (p_t - y_t) f(p_t) \right| \leqslant \sqrt{mT+1}.$$

By Lemma 7.3 of (Błasiok et al., 2023), Theorem 8.5 of (Błasiok et al., 2023), and Theorem 2 of (Qiao and Zheng, 2024), we deduce that there exists a prediction sequence $q \in \text{PC}(c)$ (which may depend on $\pi_{1:t}$) such that

$$\sum_{\substack{t \in [T] \\ c(x_t)=1}} |p_t - q_t| \leqslant \mathcal{O}(\sqrt{mT}).$$

Since our initial choice of $c \in \mathcal{C}$ was arbitrary, we conclude that

$$\text{dMCE}_{y,\mathcal{C}}(\boldsymbol{p}) = \sup_{c \in \mathcal{C}} \inf_{\boldsymbol{q} \in \text{PC}(c)} \sum_{t=1}^{T} |p_t - q_t| = \mathcal{O}(\sqrt{mT}).$$

■

We remark that if $\mathcal{C} = \{1\}$ just has the constant one function, then the Any Kernel algorithm guarantees an asymptotic bound of $\mathcal{O}(\sqrt{T})$ distance to online calibration. See (Arunachaleswaran et al., 2025) for a different algorithm that guarantees a non-asymtotic bound.

On measuring distance to multicalibration. A priori, it is not clear from Definition 44 how, given a prediction sequence $\boldsymbol{p}$, one would go about measuring its distance to online multicalibration. For our standard notion of distance, Theorem 46 gives a useful, computable metric for this purpose. Indeed, by Theorem 46, one can upper bound the distance by the kernel calibration error with respect to k_{MC}, given by the following formula:

$$\sup_{\substack{f \in \mathcal{F}_{\text{MC}} \\ \|f\|_{\mathcal{F}} \leqslant 1}} \sum_{t=1}^{T} f(x_t, p_t)(y_t - p_t) = \sqrt{\sum_{t=1}^{T} \sum_{s=1}^{T} (y_t - p_t)(y_s - p_s) k_{\text{MC}}((x_t, p_t), (x_s, p_s))}.$$

D.4. Offline results: weak agnostic learning and online to batch conversions.

In this section, we shift our attention to the offline setting where samples are drawn i.i.d from some fixed distribution $\mathcal{D}$. We prove two main results.

The first shows that one can efficiently solve weak agnostic learning over function classes $\mathcal{F}$ that are an RKHS, matching a result of (Gopalan et al., 2024a) through a different argument. Given the

tight connection between weak agnostic learning and multicalibration (Hébert Johnson et al., 2018), this result shows that any multicalibration algorithm that relies on the existence of a weak agnostic learner is unconditionally efficient whenever $\mathcal{F}$ is an RKHS.

Second, we show to convert the online learning algorithms into offline algorithms with strong guarantees for the batch setting. This adaptation in particular implies omniprediction and outcome indistinguishability algorithms for the batch case with end-to-end computational efficiency and near-optimal statistical guarantees.

Efficient (strong) learning over an RKHS. We start by recalling the definition of weak agnostic learning. Here, we state the definition as presented in (Gopalan et al., 2024b):

Definition 47 (Weak Agnostic Learning) *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times [-1,1]$. Given a comparator class $\mathcal{H} \subseteq \{\mathcal{X} \to [-1,1]\}$, a weak agnostic learner for $\mathcal{H}$ solves the following promise problem: Given an accuracy parameter γ, if there exists $h \in \mathcal{H}$ such that*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[h(x)y] \geqslant \gamma$$

then weak agnostic learner returns a function $h' : \mathcal{X} \to [-1,1]$ (not necessarily in $\mathcal{H}$) such that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[h(x)y] \geqslant \mathrm{poly}(\gamma).$$

Using the representer theorem, we prove that one can efficient solve a stronger version of the optimization problem above when $\mathcal{H}$ is an RKHS.

Proposition 48 (Existence of a Strong Learner over an RKHS) *Let k be a efficiently computable kernel with associated RKHS $\mathcal{F} \subseteq \{\mathcal{X} \to \mathbb{R}\}$ with $\sup_x k(x,x) \leqslant 1$ and let $\mathcal{F}_B \subseteq \mathcal{F}$ be the subset of functions with norm at most B,*

$$\mathcal{F}_B = \{f \in \mathcal{F} : \|f\|_{\mathcal{F}} \leqslant B\}.$$

Then, there exists a polynomial-time algorithm such that for any $\gamma \geqslant 0$, given $n \geqslant \mathrm{poly}(1/\gamma, \log(1/\delta))$ samples $(x,y) \sim \mathcal{D}$, returns a function $f' \in \mathcal{F}$ such that:

$$\Pr\left[\max_{f\in\mathcal{F}_B} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[f(x)y] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[f'(x)y] \geqslant \gamma\right] \leqslant \delta.$$

Proof The proof consists of two parts. First, we show that the corresponding empirical risk minimization problem can be solved in polynomial time. Second, we prove a uniform convergence bound showing that the empirical risk and the true risk of the functions in this class are close. Let $S_n = \{(x_i, y_i)\}_{i=1}^n$ for $x_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$ be a dataset.

Starting with the first part, let $\{(x_i, y_i)\}_{i=1}^n$ be set of samples drawn i.i.d from $\mathcal{D}$. By the Moore-Aronszajn theorem (Theorem 54), we can write any function $f \in \mathcal{F}$ as $\sum_{i=1}^n \alpha_i \Phi(x_i) + v$ where v lies in the orthogonal complement to

$$\overline{\mathrm{span}}\{\Phi(x) : x \in \{x_i\}_{i=1}^n\}.$$

Therefore, using the representer theorem, $f(x_i) = \langle f, \Phi(x_i)\rangle_{\mathcal{F}}$,we can write the following optimization problem over a Hilbert space $\mathcal{F}$

$$\arg\max_{f\in\mathcal{F}_B} \frac{1}{n}\sum_{i=1}^n f(x_i)y_i$$

as an optimization problem over $\mathbb{R}^n$:

$$\arg\max_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\langle\sum_{j=1}^{n}\alpha_j\Phi(x_j)+v,\Phi(x_i)\rangle_{\mathcal{F}}$$
$$\text{s.t}\quad \langle\sum_{i=1}^{n}\alpha_i\Phi(x_i),\sum_{i=1}^{n}\alpha_i\Phi(x_i)\rangle_{\mathcal{F}}\leqslant B^2.$$

If we let $K\in\mathbb{R}^{n\times n}$ be the matrix with $k(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle_{\mathcal{F}}$ as its (i,j)th entry, this becomes,

$$\arg\max_{\alpha\in\mathbb{R}^n} \frac{1}{n}\alpha^\top K y \tag{42}$$
$$\text{s.t}\quad \alpha^\top K\alpha\leqslant B^2.$$

This is a convex optimization problem (linear objective, quadratic constraints) and can hence be solved to any tolerance γ in time polynomial in n and $1/\gamma$.

To finish the proof, we prove a uniform convergence bound showing that all of the functions in $\mathcal{F}_B$ are close to their empirical counterparts with high probability:

$$\Pr\left[\sup_{f\in\mathcal{F}_B}|\frac{1}{n}\sum_{i=1}^{n}f(x_i)y_i-\mathbb{E}\,f(x)y|\geqslant B\sqrt{\frac{2\log(1/\delta)}{n}}\right]\leqslant\delta. \tag{43}$$

The proof of this fact follows from observing that by applying the representer theorem and linearity of inner products, we can avoid union bounding over all $f\in\mathcal{F}_B$ and instead just bound a quantity involving the feature vectors:

$$\begin{aligned}|\frac{1}{n}\sum_{i=1}^{n}f(x_i)y_i-\mathbb{E}[f(x)y]| &= |\frac{1}{n}\sum_{i=1}^{n}\langle f,\Phi(x_i)\rangle_{\mathcal{F}}y_i-\mathbb{E}[\langle f,\Phi(x)\rangle_{\mathcal{F}}y]|\\ &\leqslant \|f\|_{\mathcal{F}}\|\frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)y_i-\mathbb{E}\,\Phi(x)y\|.\end{aligned}$$

Now, since $\sup_x k(x,x)=\|\Phi(x)\|_{\mathcal{F}}\leqslant 1$ and $y\in[-1,1]$, the vectors $z=\Phi(x)y$ are sub-Gaussian (have norm bounded by 1 a.s). Therefore, we can just apply standard concentration bounds for sub-Gaussian vectors. In particular, we apply Proposition 7 in (Maurer and Pontil, 2021) (Theorem 51) to get that with probability $1-\delta$,

$$\|\frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)y_i-\mathbb{E}\,\Phi(x)y\|_{\mathcal{F}}\leqslant 8e\sqrt{\frac{2\log(1/\delta)}{n}}.$$

This completes the proof of the claim in Equation (43). The proof of the main result then follows directly by combining this concentration result with the optimization fact from Equation (42). In particular, let f' be an γ approximate optima for Equation (42) (which can be computed in polynomial time), and let f be any other function in $\mathcal{F}_B$. Then,

$$\begin{aligned}\mathbb{E}[f'(x)y] &\geqslant \frac{1}{n}\sum_{i=1}^{n}f'(x_i)y_i-\mathcal{O}(B\sqrt{\log(1/\delta)/n})\\ &\geqslant \frac{1}{n}\sum_{i=1}^{n}f(x_i)y_i-\mathcal{O}(B\sqrt{\log(1/\delta)/n})-\gamma\\ &\geqslant \mathbb{E}[f(x_i)y_i]-\mathcal{O}(B\sqrt{\log(1/\delta)/n})-\gamma.\end{aligned}$$

Letting $n \geqslant \mathrm{poly}(B, \gamma^{-1}, \log(1/\delta)))$, we get that $\mathbb{E}[f'(x)y] \geqslant \sup_{f \in \mathcal{F}_B} \mathbb{E}[f(x)y] - \mathcal{O}(\gamma)$. ■

Online to batch conversions. For the sake of completeness, we also illustrate how one can convert any of the online algorithms we study in this paper into batch algorithms. The proof of the following result is somewhat standard and uses classical martingale decompositions, but we include it for completeness.

Proposition 49 *Let k be a kernel with RKHS $\mathcal{F}$ satisfying*

$$\sup_{x \in \mathcal{X}, p \in [0,1]} k((x,p),(x,p)) \leqslant B < \infty$$

and let $\{(x_i, y_i)\}_{i=1}^n$ be a dataset of i.i.d samples drawn from a fixed distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$.

Furthermore, let $S = \{(x_i, y_i, p_i\}_{i=1}^n$ be transcript generated from running the Any Kernel algorithm on the samples (x_i, y_i) and $h_i : \mathcal{X} \to [0,1]$ be the randomized function induced by the Any Kernel algorithm conditioned on $\pi_{1:i-1} = \{(x_j, y_j)\}_{j=1}^{i-1}$.

If we define $\overline{h}_S$ be the randomized predictor which selects a function from the set $\{h_i\}_{i=1}^n$ uniformly, then with probability $1-\delta$ over the randomness of the n samples and the predictor $\overline{h}_S$, the following inequality holds for all $f \in \mathcal{F}$,where c_0 and c_1 are universal constants:

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^{(n)}, (x,y) \sim \mathcal{D}} [(y - \overline{h}_S(x)) f(x, \overline{h}_S(x))] \leqslant c_0 \frac{1}{\sqrt{n}} \|f\|_{\mathcal{F}} B + c_1 \sqrt{\frac{1 + \log(1/\delta)}{n}}.$$

Proof We use a similar decomposition as in the previous results. We start by using the reproducing property of the RKHS, linearity of expectation and then applying Cauchy-Schwarz:

$$\begin{aligned}
\mathbb{E}[(y - \bar{h}_S(x)) f(x, \bar{h}_S(x))] &= \mathbb{E}[(y - \bar{h}_S(x)) \langle f, \Phi(x, \bar{h}_S(x)) \rangle_{\mathcal{F}}] \\
&= \langle f, \mathbb{E}[(y - \bar{h}_S(x)) \Phi(x, \bar{h}_S(x))] \rangle_{\mathcal{F}} \\
&\leqslant \|f\|_{\mathcal{F}} \cdot \|\mathbb{E}[(y - \bar{h}_S(x)) \Phi(x, \bar{h}_S(x))]\|_{\mathcal{F}}.
\end{aligned}$$

Having done this, the proposition follows by combining the following two statements:

$$\mathbb{E}[(\overline{h}_S(x) - y) \Phi(\overline{h}_S(x), x)] \lesssim \|\frac{1}{n} \sum_{i=1}^n (p_i - y_i) \Phi(x_i, p_i)\|_{\mathcal{F}} + \sqrt{\frac{1 + \log(1/\delta)}{n}}, \tag{44}$$

$$\|\sum_{i=1}^n (p_i - y_i) \Phi(x_i, p_i)\|_{\mathcal{F}} \leqslant \sqrt{\sum_{i=1}^n \mathbb{E}\, p_i(1 - p_i)} \leqslant \sqrt{n}$$

where the second one is exactly the guarantee shown for the Any Kernel algorithm from Theorem 8 (see Equation (9)). We hence now focus on establishing the bound in Equation (44). By definition of $\bar{h}_S$,

$$\begin{aligned}
\mathbb{E}[(\overline{h}_S(x) - y) \Phi(\overline{h}_S(x), x)] &= \sum_{i=1}^n \mathbb{E}[(h_i(x) - y) \Phi(x, h_i(x)) \mid h_i] \Pr[\overline{h}_S = h_i] \qquad (45) \\
&= \frac{1}{t} \sum_{s=1}^t \mathbb{E}[(h_s(x) - y) \Phi(x, h_s(x)) \mid h_s].
\end{aligned}$$

Now consider the following Hilbert-space valued martingale sequence V_i adapted to the filtration $\mathcal{B}_i = \sigma(\{(x_i, y_i), p_o\}_{i=1}^n)$ where $V_0 = 0$ and

$$V_{i+1} = V_i + \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(h_i(x) - y)\Phi(x, h_i(x)) \mid \mathcal{B}_{i-1}] - (p_i - y_i)\Phi(x_i, p_i).$$

We can easily check that this process is indeed a martingale. Clearly, V_t is adapted to $\mathcal{B}_t$. Furthermore, since $\|(p_t - y_t)\Phi(x_t, p_t)\|_{\mathcal{F}} \leqslant B$, then $\mathbb{E}\|V_i\|_{\mathcal{F}} < \infty$. Lastly, since

$$\mathbb{E}[(p_i - y_i)\Phi(x_i, p_i) \mid \mathcal{B}_{i-1}] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(h_i(x) - y)\Phi(x, h_i(x)) \mid \mathcal{B}_{i-1}],$$

then,

$$\mathbb{E}[V_{i+1} \mid \mathcal{B}_i] = \mathbb{E}[V_i \mid \mathcal{B}_i] + 0 = V_i.$$

Rewriting V_i as

$$\sum_{i=1}^{n} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(h_i(x_i) - y_i)\Phi(x_i, h_i(x_i)) \mid \mathcal{B}_{i-1}] - (p_i - y_i)\Phi(x_i, p_i)$$

Using the Azuma-Hoeffding deviation inequality from (Naor, 2012) (Theorem 50), there exists a universal constant c' such that with probability $1 - \delta$,

$$\|V_i\|_{\mathcal{F}} \leqslant c'\sqrt{t\log(e^3/\delta)},$$

and hence by the reverse triangle inequality,

$$\|\sum_{s=1}^{t} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(h_s(x) - y)\Phi(x, h_s(x)) \mid \mathcal{B}_{s-1}]\|_{\mathcal{F}} \leqslant \|\sum_{s=1}^{t} (\tilde{p}_s - y_s)\Phi(x_s, \tilde{p}_s)\|_{\mathcal{F}}.$$

Plugging this into the decomposition from Equation (45), we get that with probability $1 - \delta$,

$$\mathbb{E}[(\overline{h}_t(x) - y)\Phi(\overline{h}(x), x)] \leqslant \|\frac{1}{t}\sum_{s=1}^{t} (\tilde{p}_s - y_s)\Phi(x_s, \tilde{p}_s)\|_{\mathcal{F}} + c_0'\sqrt{\frac{\log(e^3/\delta)}{t}}.$$

This establishes our two previous conditions and hence concludes the proof of the result. ■

Lemma 50 (Theorem 1.5 in (Naor, 2012)) *Let $\mathcal{F}$ be a Hilbert space and let $\{V_t\}_{t=0}^{\infty}$ be an $\mathcal{F}$-valued martingale satisfying $\|V_{t+1} - V_t\|_{\mathcal{F}} \leqslant 2$ for all $t \geqslant 0$. Then, there exists a universal constant c_0 such that for all $u > 0$ and positive integers $t \geqslant 0$,*

$$\Pr[\|V_t - V_0\|_{\mathcal{F}} \geqslant u] \leqslant e^3 \exp\left(\frac{-cu^2}{4t}\right).$$

Lemma 51 (Proposition 7 in (Maurer and Pontil, 2021)) *If $\mathcal{F}$ is a Hilbert space and $\{X_i\}_{i=1}^n$ are i.i.d random variables taking values in $\mathcal{F}$ such that $\|X_i\|_{\mathcal{F}} \leqslant B$. If $n \geqslant \log(1/\delta) \geqslant \log(2)$, then with probability $1 - \delta$,*

$$\|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[X]\|_{\mathcal{F}} \leqslant 8eB\sqrt{\frac{2\log(1/\delta)}{n}}.$$

Appendix E. Background on Reproducing Kernel Hilbert Spaces

E.1. Definition and properties.

We start with a more detailed definition of an RKHS and some of its key properties.

Definition 52 (Reproducing Kernel Hilbert Spaces) *A set of functions $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathbb{R}\}$ is a* reproducing kernel Hilbert space (RKHS) *if it satisfies the following properties.*

1. *There exists an inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$. That is, $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is symmetric, linear in its first argument, and positive definite (for all f, $\langle f, f \rangle_{\mathcal{F}} \geqslant 0$, and $\langle f, f \rangle_{\mathcal{F}} = 0$ if and only if $f = 0$).*
2. *The space is complete with respect to the norm $\|f\|_{\mathcal{F}} \stackrel{\text{def}}{=} \sqrt{\langle f, f \rangle_{\mathcal{F}}}$. That is, for all Cauchy sequences $f_1, f_2, \cdots \in \mathcal{F}$, it holds $\lim_{i \to \infty} f_i \in \mathcal{F}$.*
3. *For all $x \in \mathcal{X}$, there exists a function $K_x \in \mathcal{F}$ such that*

$$f(x) = \langle f, K_x \rangle_{\mathcal{F}}$$

for all $f \in \mathcal{F}$ where $\langle \cdot, K_x \rangle_{\mathcal{F}}$ is continuous.

The map $\langle \cdot, K_x \rangle_{\mathcal{F}} : \mathcal{F} \to \mathbb{R}$ is called the evaluation functional. The function $K(x, x') \stackrel{\text{def}}{=} \langle K_x, K_{x'} \rangle_{\mathcal{F}}$ is called the *reproducing kernel* (or *kernel* for short) of $\mathcal{F}$. Next, we define positive semi-deminite functions, which will be used in Theorem 54.

Definition 53 (PSD function) *A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite if for all $n \in \mathbb{N}$:*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j k(x_i, x_j) \geqslant 0$$

for all $x_1, \ldots, x_n \in \mathcal{X}$ and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$.

The next theorem states that each positive semi-definite function corresponds to a unique RKHS.

Theorem 54 (Moore-Aronszajn Theorem) *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive semi-definite function. Then, there is a unique RKHS $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathbb{R}\}$ for which k is the reproducing kernel. Moreover, $\mathcal{F}$ consists of the completion of the linear span of $\{k(\cdot, x) \mid x \in \mathcal{X}\}$, i.e., the set*

$$\left\{ \sum_{i=1}^{\infty} \alpha_i k(\cdot, x_i) \;\middle|\; \alpha_i \in \mathbb{R}, x_i \in \mathcal{X}, \lim_{m \to \infty} \sup_{n \geqslant m} \left\| \sum_{i=m}^{n} \alpha_i k(\cdot, x_i) \right\|_{\mathcal{F}} = 0 \right\}.$$

For example, if $|\mathcal{X}| < \infty$ then, the RKHS induced by k is

$$\mathcal{F} \stackrel{\text{def}}{=} \left\{ \sum_{x_i \in \mathcal{X}} \alpha_i k(\cdot, x_i) : \alpha_i \in \mathbb{R} \right\}.$$

Next, we state several lemmas that are useful for our analysis.

Lemma 55 (Corollary to Theorem 54) *Let $\mathcal{F}$ be a RKHS on $\mathcal{X}$. Then the zero function $x \mapsto 0$ is in $\mathcal{F}$, and, more generally, for all $f \in \mathcal{F}$ and $\alpha \in \mathbb{R}$, any linear function $x \mapsto \alpha f(x)$ is in $\mathcal{F}$.*

Lemma 56 (Theorem 5.4, (Paulsen and Raghupathi)) *Let k_1 and k_2 be positive semi-definite kernels on $\mathcal{X}$ with associated RKHSs $\mathcal{F}_1$ and $\mathcal{F}_2$ then $k = k_1 + k_2$ is a valid kernel with associated RKHS $\mathcal{F}$ equal to the completion of the span of*

$$\{f_1 + f_2 : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}.$$

Moreover, direct implication of the above result is that, for $f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2$, $\|f_1 + f_2\|_{\mathcal{F}} \leqslant \|f_1\|_{\mathcal{F}_1} + \|f_2\|_{\mathcal{F}_2}$.

A direct implication of the above result, since the zero function $x \mapsto 0$ is in every RKHS, is that $\mathcal{F}_1 \cup \mathcal{F}_2 \subseteq \mathcal{F}$.

Lemma 57 (Theorem 5.11, (Paulsen and Raghupathi)) *Let $k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_2 : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be positive semi-definite kernels with associated RKHSs $\mathcal{F}_1$ and $\mathcal{F}_2$ then $k((x,y),(x,y')) = k_1(x,x')k_2(y,y')$ is a valid kernel. Furthermore, its associated function space is the completion of the span of the set*

$$\{f_1 \cdot f_2 \ : \ f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}$$

where for any $f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2$ we define $f_1 \cdot f_2 \ : \ \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to be the function $(f_1 \cdot f_2)(x,y) = f_1(x)f_2(y)$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. Moreover, for $f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2$, $\|f_1 \cdot f_2\|_{\mathcal{F}} \leqslant \|f_1\|_{\mathcal{F}_1}\|f_2\|_{\mathcal{F}_2}$.

Lemma 58 (Theorem 5.7, (Paulsen and Raghupathi)) *For any function $\phi \ : \ \mathcal{X} \to \mathbb{R}$ and RKHS $\mathcal{F}_0 \subseteq \{f \ : \ \mathbb{R} \to \mathbb{R}\}$ associated with kernel k, there exists an RKHS $\mathcal{F}_1$ equal to the completion of the span of the set $\{f \circ \phi \ : \ f \in \mathcal{F}_0\}$ and associated with kernel $k \circ \phi \stackrel{\text{def}}{=} k(\phi(\cdot), \phi(\cdot))$. Moreover, it holds $\|f \circ \phi\|_{\mathcal{F}_1} \leqslant \|f\|_{\mathcal{F}_0}$.*

Lemma 59 *Let $\mathcal{X}$ be any set and let $\mathcal{I}$ be any index set. Let $\mathcal{F} = \{f_i\}_{i \in \mathcal{I}}$ be a collection of functions $f_i : \mathcal{X} \to \mathbb{R}$ indexed by $\mathcal{I}$. Suppose that for each $x \in \mathcal{X}$, we have*

$$\sum_{i \in \mathcal{I}} f_i(x)^2 < m \tag{46}$$

for some constant m, in which case the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by

$$k(x,y) = \sum_{i \in \mathcal{I}} f_i(x)\, f_i(y)$$

is a valid kernel. Then, the RKHS $\mathcal{F}$ corresponding to k contains $\mathcal{F}$, and $\|f_i\|_{\mathcal{F}} \leqslant 1$ for each $i \in \mathcal{I}$.

Proof [Proof of Theorem 59] We introduce several pieces of notation:

- Let $\mathcal{H}$ be the Hilbert space of "coefficient sequences" $\alpha : \mathcal{I} \to \mathbb{R}$ that are L^2 bounded by m with respect to the counting measure on $\mathcal{I}$, which means that $\sum_{i \in \mathcal{I}} \alpha(i)^2 < m$.
- For each $x \in \mathcal{X}$, define a coefficient sequence $\Phi_x : \mathcal{I} \to \mathbb{R}$ by the formula $\Phi_x(i) = f_i(x)$. Note that $\Phi_x \in \mathcal{H}$ by the assumption that $\sum_{i \in \mathcal{I}} f_i(x)^2$ is finite. Note also that the kernel function k satisfies

  $$k(x,y) = \langle \Phi_x, \Phi_y \rangle_{\mathcal{H}}$$

  for any $x, y \in \mathcal{X}$.

- Given a coefficient sequence $\alpha \in \mathcal{H}$, let $f_\alpha : \mathcal{X} \to \mathbb{R}$ denote the function
$$f_\alpha(x) = \langle \alpha, \Phi_x \rangle_{\mathcal{H}} = \sum_{i \in \mathcal{I}} \alpha(i) f_i(x).$$
- Let $V \subseteq \mathcal{H}$ be the closure in $\mathcal{H}$ of the subspace $\text{span}\{\Phi_x : x \in \mathcal{X}\}$. In other words, let V be the set of all finite linear combinations of coefficient sequences Φ_x for $x \in \mathcal{X}$, together with their limit points in $\mathcal{H}$. Relatedly, let proj_V denote the orthogonal projection of $\mathcal{H}$ onto V, which satisfies $\text{proj}_V(\alpha) \in V$ and
$$\left\langle \alpha - \text{proj}_V(\alpha), \Phi_x \right\rangle_{\mathcal{H}} = 0 \tag{47}$$
for each $\alpha \in \mathcal{H}$ and $x \in \mathcal{X}$.

Rephrased in this language, Moore-Aronszajn theorem and its proof simply show that the map $\alpha \mapsto f_\alpha$ is a distance-preserving, one-to-one correspondence (*i.e.* an isometric isomorphism) between V and the RKHS $\mathcal{F}$ corresponding to the kernel k. Next, by Eq. (47) with $\alpha = e_i$, we see that for all $x \in \mathcal{X}$ and $i \in \mathcal{I}$,
$$f_i(x) = \langle e_i, \Phi_x \rangle_{\mathcal{H}} = \left\langle \text{proj}_V(e_i), \Phi_x \right\rangle_{\mathcal{H}} = f_{\text{proj}_V(e_i)}(x).$$
Here, e_i denotes the i^{th} standard basis coefficient sequence
$$e_i(j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$
Using the aforementioned distance-preserving correspondence between V and $\mathcal{F}$, we see that
$$\|f_i\|_{\mathcal{F}} = \left\| f_{\text{proj}_V(e_i)} \right\|_{\mathcal{F}} = \|\text{proj}_V(e_i)\|_{\mathcal{H}} \leqslant \|e_i\|_{\mathcal{H}} = 1,$$
which concludes the proof. ■

We also remark that if $\mathcal{F}$ is a (not necessarily finite) collection of indicator functions for subsets $S_i \subseteq \mathcal{X}$ but each $x \in \mathcal{X}$ belongs to at most finitely many such S_i, then Eq. (10) is satisfied, so Theorem 59 implies that the RKHS corresponding to the intersection kernel
$$k(x, y) = \text{Int}_{\mathcal{F}}(x, y) = \left| \{ i \in \mathcal{I} : x, y \in S_i \} \right|$$
contains all functions in $\mathcal{F}$ and that their norms in $\mathcal{F}$ are at most 1.

E.2. Key examples.

Example 1 (Linear functions) *Let $\mathcal{X} = \mathbb{R}^d$, then $\mathcal{F}_{\text{lin}}$, the space of all linear functions from $\mathbb{R}^d$ to $\mathbb{R}$, defined as,*
$$\mathcal{F}_{\text{lin}} = \{ f_w : w \in \mathbb{R}^d, f(x) = x \cdot w \} \subseteq \{ \mathbb{R}^d \to \mathbb{R} \}$$
is an RKHS with corresponding kernel $k_{\text{lin}}(x, x') = x \cdot x' = \sum_{i=1}^d x_i x_i'$ equal to the standard inner product. The feature mapping is just the identity function $\Phi(x) = x$. Note that each element $f \in \mathcal{F}$ could be thought of both as a function from $\mathbb{R}^d$ to $\mathbb{R}$ as well as an element in the Hilbert space (which in this case is just $\mathbb{R}^d$). However, going back to our earlier comment, we see that we could have equivalently written out $\mathcal{F}_{\text{lin}}$ as,
$$\mathcal{F}_{\text{lin}} = \text{span}\left\{ \sum_{x_i \in \mathcal{X}} \alpha_i k_{\text{lin}}(\cdot, x_i) \ : \ \alpha_i \in \mathbb{R} \right\} = \text{span}\left\{ \sum_{x_i \in \mathbb{R}^d} \alpha_i x_i \ : \ \alpha_i \in \mathbb{R} \right\}.$$

Example 2 (Polynomial functions) *Consider the set of polynomials of degree $\leqslant k$ on d variables with the inner product defined as the inner product of the coefficients on each monomial. In this case, $\mathcal{X} = \mathbb{R}^d$. Since the space of coefficients is just $\mathbb{R}^\ell$ for some appropriate ℓ (depending on the dimension of the input space d and k), it is complete and the inner product satisfies all the necessary properties.*

Then, to show that this has the reproducing property, let K_x be the polynomial where the coefficient on a given monomial is determined by multiplying together the corresponding entries of x. So the coefficient on the $x_1 x_2^3$ term is the first entry of x times the cube of the second entry of x. Then, notice that for all $f \in \mathcal{H}$, $f(x) = \langle f, K_x \rangle_{\mathcal{H}}$. It can be shown that the corresponding kernel is

$$k(x, y) = (1 + \langle x, y \rangle)^k.$$

Example 3 (Boolean functions) *Consider the set of functions taking the form $f : \{-1, 1\}^d \to \{-1, 1\}$. First, notice that we can write f as a polynomial. For $a, x \in \{-1, 1\}^d$, define the indicator polynomial*

$$\begin{aligned} 1_a(x) &= \left(\frac{1 + a_1 x_1}{2}\right)\left(\frac{1 + a_2 x_2}{2}\right) \cdots \left(\frac{1 + a_d x_d}{2}\right) \\ &= \begin{cases} 1 & \text{if } a = x \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Then, notice

$$f(x) = \sum_{a \in \{-1,1\}^d} f(a) 1_a(x).$$

This is just the sum of 2^d different order d polynomials and therefore a polynomial of order d. Thus, Boolean functions are a subset of the polynomials and we can use the kernel $k(x, y) = (1 + \langle x, y \rangle)^d$. The inner product is also the same as for the polynomials: the inner product is just the inner product of the coefficients on each monomial.

In fact, if we distribute the products in $1_a(x)$, we can see that every Boolean function can be written as

$$f(x) = \sum_{I \in 2^d} \alpha_I x_I$$

for $\alpha_I \in \mathbb{R}$ a constant and $x_I \stackrel{\text{def}}{=} \prod_{i \in I} x_i$. See (O'Donnell, 2021) for more discussion of Boolean functions.

Example 4 (Regression trees) *As a special case of Boolean functions, we will write down the functions representing regression trees on Boolean inputs. For a given regression tree, let $b \in \{0, 1\}^k$ represent the path down the decision tree, where $b_i = 0$ means go to the left child (i.e., the the decision variable in the ith decision following path b is 0) and $b_i = 1$ means go to the right child at depth i. Let c_b be the leaf assigned to path b. Let $i_{b,j}$ represent the index of the decision variable at the jth decision following path b. Then any decision tree can be specified by $\{c_b\}_{b \in \{0,1\}^k}$ and $\{i_{b,j}\}_{b \in \{0,1\}^k, j \in [k]}$:*

$$f(x) = \sum_{b \in \{0,1\}^k} c_b \prod_{\ell=0}^{k-1} ((1 - x_{i_{b,\ell}})(1 - b_\ell) + x_{i_{b,\ell}} b_\ell)$$

Example 5 (Sobolev spaces $W^{1,2}(\Omega)$ for $\Omega \in \{[0, 1], \mathbb{R}\}$,) *This example comes from (Berlinet and Thomas-Agnan, 2011), Section 7.4, Examples 13 and 24. Consider the set of functions $\mathcal{F}_0 \subseteq \{\Omega \to \mathbb{R}\}$ for $\Omega \in \{[0, 1], \mathbb{R}\}$ such that*

(a) each function is differentiable almost everywhere and continuous, and

(b) each function and its derivative are square integrable.

The completion of $\mathcal{F}_0$ with respect to the norm

$$\|f\|_{\mathcal{F}_0}^2 = \int_\Omega (f(x))^2\,dx + \int_\Omega (f'(x))^2\,dx.$$

is an RKHS $\mathcal{F}$ (usually denoted $W^{1,2}(\Omega)$) where, if $\Omega = [0,1]$, the kernel is

$$k_{[0,1]}(x,x') = \frac{(e^x + e^{-x})(e^{1-x'} + e^{x'-1})}{2(e - e^{-1})} < 3.$$

for $0 \leqslant x \leqslant x' \leqslant 1$ and $k_{[0,1]}(x,x') = k_{[0,1]}(x',x)$ if $0 \leqslant x' \leqslant x \leqslant 1$. If $\Omega = \mathbb{R}$, the kernel is

$$k_{\mathbb{R}}(x,x') = \exp\{-|x - x'|\}.$$

The inner product in $\mathcal{F}$ for differentiable functions $f, g \in \mathcal{F}$ is

$$\langle f, g\rangle_{\mathcal{F}} = \int_\Omega f(x)g(x)\,dx + \int_\Omega f'(x)g'(x)\,dx.$$

Next, we state the following simple lemma about the composition of functions in $W^{1,2}([0,1])$. For a set of differentiable functions $\mathcal{F}$, let $\mathcal{F}' = \{f' \mid f \in \mathcal{F}\}$ denote the set of derivatives.

Lemma 60 *Suppose that there exists a universal constant $B \geqslant 1$ and sets of differentiable functions $\mathcal{F}_0, \mathcal{F}_1$ with $\mathrm{Im}(\mathcal{F}_0) \subseteq [0,1]$, $\|\mathcal{F}_0\|_{W^{1,2}([0,1])} \leqslant B$, $\mathrm{Im}(\mathcal{F}_1) \subseteq [-B,B]$, and $\mathrm{Im}(\mathcal{F}_1') \subseteq [-B,B]$. Then, $\{f_1 \circ f_0 \mid f_0 \in \mathcal{F}_0, f_1 \in \mathcal{F}_1\} \subseteq \mathcal{F}$ and $\|f_1 \circ f_0\|_{\mathcal{F}} \leqslant 2B^2$.*

Proof Fix $f_0 \in \mathcal{F}_0, f_1 \in \mathcal{F}_1$. Notice that by the uniform boundedness of f_1, $\|f_1 \circ f_0\|_{L^2([0,1])} \leqslant B$. Also, $\|f_0'\|_{L^2([0,1])} \leqslant \|f_0'\|_{W^{1,2}([0,1])} \leqslant B$. Then,

$$\begin{aligned}\|(f_1' \circ f_0)f_0'\|_{L^2([0,1])} &\leqslant \|f_1' \circ f_0\|_{L^2([0,1])}\|f_0'\|_{L^2([0,1])} \\ &\leqslant B^2\end{aligned}$$

where the first line comes from the Cauchy-Schwarz inequality and the second line comes from the plugging in the bounds on each norm. Also, by the uniform boundedness of $\mathcal{F}_1$, $\|f_1 \circ f_0\| \leqslant B$, which implies the desired bound. See, e.g., (Evans, 2018), Theorem 4.4, part (ii) for more general conditions on the composition of functions in a Sobolev space. ■

Example 6 (Low-degree functions on $\{0,1\}^n$, (Shawe-Taylor and Cristianini, 2004), Section 9.2) *Consider the set of functions $\mathcal{F}_0 \subseteq \{\{-1,1\}^n \to [-1,1]\}$ whose Fourier spectrum is supported on monomials of degree at most d. The kernel associated with the completion of $\mathcal{F}_0$ is*

$$k(x,x') = \sum_{S \subseteq [n], |S| \leqslant d} x_S x'_S.$$

E.3. Matrix-valued kernels

We now introduce two standard definitions related to *matrix-valued* kernels and their corresponding vector valued reproducing kernel Hilbert spaces. These standard facts can be found, for example, in (Álvarez et al., 2012; Minh, 2016).

Definition 61 *We say that a matrix-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d\times d}$ is a valid* kernel *if the following two "positive semidefiniteness" properties hold:*

- *For all $x, y \in \mathcal{X}$, we have $k(x,y) = k(y,x)^\top$.*
- *For all $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathcal{X}$ and $w_1, \ldots, w_n \in \mathbb{R}^d$, we have*

$$\sum_{a=1}^{n}\sum_{b=1}^{n}\langle w_a, k(x_a,x_b)w_b\rangle \geqslant 0.$$

Definition 62 *Given a matrix-valued kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d\times d}$, the* reproducing kernel Hilbert space (RKHS) $\mathcal{F}$ corresponding to k *is a Hilbert space consisting of vector-valued functions $f : \mathcal{X} \to \mathbb{R}^d$. Specifically, $\mathcal{F}$ is the completion of the space of all linear combinations of functions of the form*

$$x \mapsto \sum_{a=1}^{n} k(x,x_a)w_a$$

for some $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathcal{X}$ and $w_1, \ldots, w_n \in \mathbb{R}^d$. It is imbued with the unique inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ satisfying the following property: for all $x_1, x_2 \in \mathcal{X}$ and $w_1, w_2 \in \mathbb{R}^d$, the inner product of the functions $f_1(x) = k(x,x_1)w_1$ and $f_2(x) = k(x,x_2)w_2$ is

$$\langle f_1, f_2\rangle_{\mathcal{F}} = \langle w_1, k(x_1,x_2)w_2\rangle,$$

where the inner product on the right hand side is the standard inner product on $\mathbb{R}^d$.

The following result illustrates how one might represent any finite set of vector valued functions using a matrix valued kernel:

Lemma 63 *Let $\mathcal{X}$ be any (not necessarily finite) population set and let $\mathcal{I}$ be any (not necessarily finite) index set. Let $\mathcal{C} = \{c_i\}_{i\in\mathcal{I}}$ be a collection of functions $c_i : \mathcal{X} \to \mathbb{R}^d$ indexed by $\mathcal{I}$. Suppose that for each $x \in \mathcal{X}$, we have*

$$\sum_{i\in\mathcal{I}} \|c_i(x)\|^2 < \infty, \tag{$*$}$$

in which case the matrix-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d\times d}$ given by

$$k(x,y) = \sum_{i\in\mathcal{I}} c_i(x)\, c_i(y)^\top$$

is a valid kernel. Then, the RKHS $\mathcal{F}$ corresponding to k contains $\mathcal{C}$, and $\|c_i\|_{\mathcal{F}} \leqslant 1$ for each $i \in \mathcal{I}$.

Proof Given a fixed element $y \in \mathcal{X}$ and $a \in \mathbb{R}^d$, consider the following vector-valued function from $\mathcal{X}$ to $\mathbb{R}^d$:

$$x \mapsto k(x,y)a.$$

By Definition 62, we know that the RKHS $\mathcal{F}$ corresponding to the matrix-valued kernel k is the completion of the set of all linear combinations of vector-valued functions of the above form. Next, consider the following related *scalar-valued* kernel $k_{\text{scalar}} : (\mathcal{X} \times [d]) \times (\mathcal{X} \times [d]) \to \mathbb{R}$, defined as follows:

$$k_{\text{scalar}}((x,a),(y,b)) = k(x,y)_{ab}.$$

The RKHS $\mathcal{F}_{\text{scalar}}$ corresponding to k_{scalar} is given by the Moore-Aronszajn Theorem (Theorem 54), and comparing this description to the aforementioned description of $\mathcal{F}$, it becomes clear that $\mathcal{F}$ and $\mathcal{F}_{\text{scalar}}$ are isometrically isomorphic, i.e. there is a one-to-one, length-preserving correspondence between elements of $\mathcal{F}$ and elements of $\mathcal{F}_{\text{scalar}}$. Specifically, the isomorphism maps a function $f : \mathcal{X} \to \mathbb{R}^d$ in $\mathcal{F}$ to the function $f_{\text{scalar}} : \mathcal{X} \times [d] \to \mathbb{R}$ given by

$$f_{\text{scalar}}(x,a) = f(x)_a$$

for each $x \in \mathcal{X}$ and $a \in [d]$. By Theorem 59, the space $\mathcal{F}_{\text{scalar}}$ contains the function $c_{\text{scalar}} : \mathcal{X} \times [d] \to \mathbb{R}$ for each $c \in \mathcal{C}$, and these functions all have norm $\|c_{\text{scalar}}\|_{\mathcal{F}_{\text{scalar}}} \leqslant 1$. Consequently, $\mathcal{C} \subseteq \mathcal{F}$ and $\|c\|_{\mathcal{F}} \leqslant 1$ for each $c \in \mathcal{C}$, as well. ■