Frontiers of Mathematical Finance
Vol. 8, 2026, pp. 23-77 (January 28, 2026)
doi:10.3934/fmf.2026001

SYNCHRONIZING PRETRAINED KERNEL REGRESSORS WITH APPLICATIONS TO AMERICAN OPTION PRICING

Xuwei Yang[1], Anastasis Kratsios[2], Florian Krach[3], Matheus Grasselli[1] and Aurelien Lucchi[4]

[1]Department of Mathematics and Statistics, McMaster University, Canada

[2]Department of Mathematics and Statistics, McMaster University, and Vector Institute, Canada

[3]Department of Mathematics, ETH Zürich, Switzerland

[4]Department of Mathematics and Computer Science, University of Basel, Switzerland

(Communicated by Xin Guo)

Abstract. American option pricing is a well-studied, computationally-heavy optimal stopping problem, which has motivated the need for *light* and accurate computational pipelines. Currently, the state-of-the-art *light* American option prices are built on randomly generated kernel regressors such as random neural networks; however, a single random (or pre-trained) kernel can yield unstable performance on new tasks. With this motivation, we study the problem of *fine-tuning* several pre-trained kernel regressors, trained on distinct datasets to a given novel dataset with a downstream view toward light American option pricing. This paper addresses the *meta-optmization* problem of the task of designing optimal transfer learning algorithms, which minimize a pathwise regret functional. Using techniques from optimal control, we construct the unique regret-optimal optimization algorithms for fine-tuning mixtures of pre-trained kernel regressors on $\mathbb{R}^d$ sharing the same feature map. Our regret functional balances the objects of predictive power on the new dataset against transfer learning from other datasets under an algorithmic stability penalty. We show that an adversary which perturbs a proportion $0 \leq q \leq 1$ of training pairs by at most $\varepsilon > 0$, across all training sets, cannot reduce the regret-optimal algorithm's regret by more than $\mathcal{O}(\varepsilon(q\bar{N})^{1/2})$, where $\bar{N}$ is the aggregate number of training pairs. Further, we derive estimates of the computational complexity of our regret-optimal optimization algorithm. Our theoretical findings are used to improve state-of-the-art (SOTA) in computationally *light* American option pricing problems using random feature models, where we achieve SOTA performance compared to the benchmark light American option pricer of [32].

1. **Introduction.** In transfer learning, one is usually confronted with a focal task defined by a dataset $\mathcal{D}_1$ drawn from a distribution μ_1 on $\mathbb{R}^{d+1}$ and additional (more or less) related datasets $\mathcal{D}_2, \dots, \mathcal{D}_N$, where each $\mathcal{D}_i$ is drawn from (possibly different) distributions μ_i on $\mathbb{R}^{d+1}$. The goal is to maximize the performance of a model $f_\theta : \mathbb{R}^d \to \mathbb{R}$ on $\mathcal{D}_1$ by optimizing the parameters $\theta \in \mathbb{R}^p$ with the possibility to leverage the knowledge (about the focal task) that is enclosed in the additional datasets $\mathcal{D}_2, \dots, \mathcal{D}_N$. The performance of a model is quantified using a

2020 *Mathematics Subject Classification.* 91-10, 49N90, 68T01.

Key words and phrases. Finite-rank kernel ridge regressors, meta-optimization, transfer learning, optimal control, American option pricing.

user-specified L_ℓ-Lipschitz loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, with the intent being to minimize the out-of-sample loss on μ_1 of any learner $f_\theta : \mathbb{R}^d \to \mathbb{R}$, as quantified by its *true risk* $\mathcal{R}(f_\theta)$. In practice, we minimize a *weighted* version of the empirical risk $\hat{\mathcal{R}}_w^{\mathcal{D}}(f_\theta)$ proportionally accounting for each dataset, using a *mixture coefficient* $w \in \Delta_N \overset{\text{def.}}{=} \{u \in [0,1]^N : \sum_{n=1}^N u_n = 1\}$. These risks are defined by

$$\mathcal{R}(f_\theta) \overset{\text{def.}}{=} \mathbb{E}_{(X,Y)\sim\mu_1}\big[\ell(f_\theta(X), Y)\big]$$

and

$$\hat{\mathcal{R}}_w^{\mathcal{D}}(f_\theta) \overset{\text{def.}}{=} \sum_{i=1}^N \frac{w_i}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \ell(f_\theta(X_j^i), Y_j^i).$$

Our first main result (Theorem 3.1) is a *statistical guarantee*, which shows that the following choice of weights $w^\star = (w_i^\star)_{i=1}^N$ minimizes the worst-case generalization gap between the true risk and the weighted empirical risk of any L-Lipschitz learner (for a given $L \geq 0$)

$$w_i^\star = \frac{e^{-\gamma\, \mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i) - \frac{\gamma}{|\mathcal{D}_i|^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}{\sum_{u=1}^N e^{-\gamma \mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u) - \frac{\gamma}{|D_u|^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\leq\eta}}, \tag{1}$$

where $\mathcal{W}_1$ is the 1-Wasserstein distance and $\hat{\mu}_i \overset{\text{def.}}{=} \frac{1}{|\mathcal{D}_i|}\sum_{(x,y)\in\mathcal{D}_i} \delta_{(x,y)}$is the empirical distribution associated to the i^{th} dataset, and $\eta \geq 0$ is the *sparse thresholding* hyperparameter controlling how similar a dataset must be to the focal task's dataset $\mathcal{D}_1$ to be considered. Moreover, the weighted empirical risk is a consistent estimator of the true risk, converging at the non-parametric rate of $\mathcal{O}\big((N^\star)^{-1/(d+D)}\big)$.

Given the optimal mixture weights, defining the optimal weighted empirical risk for the general transfer learning problem, one may ask: *How does one select optimal parameters θ given $w^\star$ as efficiently as possible?* We will answer this *meta-optimization* question in the specialized case where f_θ are kernel ridge regressors, the loss function ℓ, and all the data generating distributions are compactly supported. Our *meta-optimization* problem is thus to select, amongst a class of deterministic iterative optimization algorithms, those algorithms which minimize a fixed performance metric along their path. We tailor our analysis to the following transfer learning problem.

We identify deterministic algorithms defining and jointly optimizing parameters $\Theta(t) \overset{\text{def.}}{=} (\theta_1^\top(t), \ldots, \theta_N^\top(t))^\top$ associated with each task $\mathcal{D}_i$, focusing on the focal task $\mathcal{D}_1$. We define the parameter trajectories $\Theta^{0\cdots T} \overset{\text{def.}}{=} \{\Theta(t)\}_{t=0}^T$ in $(\mathbb{R}^{Np})^T$, and $\theta_i^\star$ to be the greedy parameter choices optimizing the empirical risk *calculated only over the i^{th} dataset*. All algorithms will be initialized at $\Theta(0) \overset{\text{def.}}{=} \Theta^\star \overset{\text{def.}}{=} (\theta_1^\star, \ldots, \theta_N^\star) \in \mathbb{R}^{Np}$, and are terminated after $T \in \mathbb{N}_+$ iterations. Denote $\theta^w(T) \overset{\text{def.}}{=} \sum_{i=1}^N w_i\theta_i(T)$ for each $w \in \Delta_N$ and $\Theta(T) = (\theta_1^\top(T), \ldots, \theta_N^\top(T))^\top$. Building on [15, 17], given a $w \in \Delta_N$, we use the following *regret*-based synchronization criterion to quantify the performance of an algorithm:

$$\mathcal{R}(\Theta^{0\cdots T}) \overset{\text{def.}}{=} \underbrace{\sum_{i=1}^N w_i \sum_{j=1}^{|\mathcal{D}_i|} (f_{\theta^w(T)}(x_j^i) - y_j^i)^2}_{\text{Performance: Weighted Empirical Risk}} - l^\star$$

$$+\sum_{t=0}^{T-1}\Big(\underbrace{\lambda\left\|\Theta(t+1)-\Theta^{\star}\right\|_2^2}_{\text{Transfer Learning Level}}+\underbrace{\beta\left\|\Delta\Theta(t)\right\|_2^2}_{\text{Stability}}\Big), \tag{2}$$

where the hyperparameter $\lambda \geq 0$ controls how far the algorithm's iterates can deviate from the pre-trained model parameters $\Theta^{\star}$, the hyperparameter $\beta \geq 0$ controls the stability of the training algorithm by penalizing rapidly changing increments $\Delta\Theta(t) \stackrel{\text{def.}}{=} \Theta(t)-\Theta(t-1)$, where $t \geq 1$, and $l^{\star} \stackrel{\text{def.}}{=} \min_{\Theta\in\mathbb{R}^{Np}} \sum_{i=1}^{N} w_i \sum_{j=1}^{|\mathcal{D}_i|}(f_{\theta^w}(x_j^i)-y_j^i)^2$ captures the achievable error across all tasks.

The pathwise regret functional $\mathcal{R}(\Theta^{0\cdots T})$ measures not only the final empirical performance, but also the quality of the learning trajectory throughout fine-tuning. The weighted empirical risk term evaluates the aggregated model's predictive performance across all tasks. The term of transfer learning level, regulated by λ, constrains deviations from the pre-trained initialization $\Theta^{\star}$, thereby moderating task-specific drift and mitigating catastrophic forgetting. The stability term, scaled by β, enforces smooth parameter updates, encouraging stable and consistent adaptation. Together, these components quantify how well an algorithm synchronizes multi-task learning while balancing performance, transfer, and stability. In this paper, we select the optimal weights w with Algorithm 1, using a procedure similar to the PAC-Bayesian approach of [2], often used in meta-learning [56, 59, 60].

Our main motivation for this setting stems from the now typical setting, where one has access to a pre-trained foundation model, whose final linear layer is to be fine-tuned on several different datasets. This leads to pre-trained deep learning models $f^{\star}_{\theta_1},\ldots,f^{\star}_{\theta_N}$ whose hidden layers all coincide. In this case, f_θ factors as a kernel regressor where its *frozen* hidden layers define a feature map, and the final linear layer is the only-trainable parameter. For analytical tractability, we require that each neural network has the same frozen hidden layers. Thus, all our models are finite-rank kernel ridge regressors of the form

$$f_\theta(x) = \phi(x)^{\top}\theta. \tag{3}$$

In reservoir computing [29, 50] and randomized neural networks [28], this assumption is standard since random reservoirs/feature maps are used. Moreover, in settings where each neural network has a different set of hidden layers, one can achieve the representation (3) simply by concatenating each of the frozen hidden layers into a joint feature map. Thus, the structural condition (3) is made without any loss of generality.

1.1. **Contributions.** In Theorem 3.1, we obtain a guarantee that the weighting scheme in (1) (and implemented by Algorithm 1) minimizes an upper-bound on the *worst-case* gap between the empirical risk for the focal dataset $\mathcal{D}_1$ and the (optimal) *weighted* empirical risk over all datasets. We emphasize that the first result is fully general and holds for arbitrary deep learning models optimizing a general Lipschitz loss, just as much as it does for specialized kernel ridge regressors optimizing the MSE loss.

Meta-Optimality. We present a *regret-optimal* algorithm, by which we mean an optimizer of (2). Aligned with recent trends in optimization [15, 47], we leverage techniques from optimal control theory to characterize the regret-optimal algorithm. We show that this is possible due to the observation that minimizers of (2) coincide

with the minimizers of the energy functional

$$\mathcal{L}(\Theta^{0\cdots T}) \overset{\text{def.}}{=} \sum_{t=0}^{T-1} \underbrace{\left[\lambda \left\|\Theta(t+1) - \Theta^{\star}\right\|_2^2 + \beta \left\|\Delta\Theta(t)\right\|_2^2\right]}_{\text{Running-Cost}} + \underbrace{l(\Theta(T), w; \mathcal{D})}_{\text{Terminal Cost}}, \tag{4}$$

where we abbreviate $l(\Theta, w; \mathcal{D}) \overset{\text{def.}}{=} \sum_{i=1}^{N} w_i \sum_{j=1}^{|\mathcal{D}_i|} (f_{\theta^w}(x_j^i) - y_j^i)^2$. From this control-theoretic perspective, every algorithm can be regarded as driven by a "control", and the functional $\mathcal{L}$ has the form of a "cost function" of an optimal control problem, decomposable into the sum of a "running/operational cost" and a "terminal/objective cost". This allows us to leverage tools from optimal control to derive a closed-form expression of the unique regret-optimal algorithm driven by the unique "optimal control" (Algorithm 2).

Adversarial Robustness. We prove that this procedure is adversarially robust, meaning that the value of (4) varies by $\mathcal{O}(\varepsilon\sqrt{\bar{N}}q)$ if a malicious adversary can perturb at most $1/q$ percent of all the data in each of the datasets $\mathcal{D}_i$ by at-most $\varepsilon \geq 0$, where $\bar{N} = \sum_{i=1}^{N} |\mathcal{D}_i|$. Furthermore, we show that our regret-optimal algorithm has a computational complexity of $\mathcal{O}(N^2p^3 + T\,(Np)^{2.373})$. Moreover, we characterize the optimal choice of the weights w which the central planner must use to maximize the selected model's performance on the primary/focal dataset $\mathcal{D}_1$.

SOTA Light American Option Pricer. We then use our meta-optimization procedure to improve upon the performance of the randomized neural network approach to American option pricing, recently proposed in [32] as a computationally light alternative to the methodology of [9, 10]. Our main method remains computationally light, such as [32], as it also relies on a generated ensemble of random neural network kernels as its backbone while performing a meta-optimization atop. Thus, we inherit the expressivity [28] and statistical consistency [27] of *individual* randomly generated neural networks, while gaining from an optimally synchronized ensemble of them. When deploying our procedure numerically, we are able to obtain superior predictive performance over the aforementioned random neural network approach to *light* American option pricing for only a marginal increase in computational load.

Outline of paper. In Section 3, we introduce the regret-optimal algorithm and demonstrate the optimality, computational complexity, and adversarial robustness of the algorithm. Further, we introduce an accelerated algorithm of lower computational complexity that can achieve near-regret optimality. In Section 4, we conduct a convergence analysis of the regret-optimal algorithm and compare it to the standard gradient descent approach for American option pricing in quantitative finance.

A game theoretic interpretation of $\mathcal{R}$ is given in Appendix A. Appendix B is devoted to proving the optimality, adversarial robustness, and computational complexity of the regret-optimal algorithm. The near-regret optimality and computational complexity of the accelerated algorithm are also proven. Appendix E includes background on American option pricing in quantitative finance, and on finite-rank kernel ridge regressors.

1.2. **Related work.** Examples of f_θ of the form (3) include extreme learning machines [25, 27, 28, 54] or any finite-rank kernel ridge regressor (fKRR) [3, 4]. Random feature models and finite-rank ridge regressors are typical in contemporary quantitative finance [26, 32] or reservoir computing [21], where it is favorable to rapidly deploy a deep learning model due to time or computing-power limitations/constraints [20].

Finite-rank kernel ridge regressors are also typical in the setting of standard transfer [35] and multi-task learning [39, 58] pipelines where one often has a pre-trained deep neural network model and, for any novel task, the user freezes the hidden layers in the deep neural network model and only fine-tunes/trains the deep neural network's final linear layer. This is useful since the hidden layers of most deep neural network architectures are known to process general, cruder features, and the final layer of any such model decodes these features in a task-specific manner [71, 72].

The optimization of a model trained from multiple data sources lies at the heart of multi-task learning. In [61], the authors cast the task of optimizing (2) with $T = 0 = \lambda = \beta$ as a multi-objective optimization problem. They propose to solve it via a multi-gradient descent algorithm, which, under mild conditions, converges to a Pareto stationary point; i.e., which is essentially a critical point of each of the ith player's penalized SSE. Though iterative schemes for multi-objective optimization can efficiently reach critical points [23], this problem's critical point is not necessarily a minimum since problem (2) is not convex when w and Θ are jointly optimizable. The regularization hyperparameter λ in (2) controls the level of leveraging the information from the source tasks and has been studied in [44, 49, 63, 66].

This proposed algorithm reflects the ideas of the federated stochastic gradient descent (FedSGD) algorithm of [53], its online counterpart FedOGD, which is an online federated version of online gradient descent[1], and more efficient online versions thereof, such as [45]. From the federated learning perspective, the problem of training individual fKRR on each dataset and then centralizing them by optimizing the mixing parameter w in (2) and its origins date back to local gradient descent introduced in the optimization and control literature in [51]. Though several authors (e.g. [40]) have proposed various progressive improvements of federated gradient-descent-type algorithms for training a learner from several data sources, the problem of studying *the* most efficient iterative federated learning algorithm for regret minimization has not yet been tackled.

We note that (stochastic) optimal control tools have revealed new insights into machine learning. Examples include an optimal step-size control [48] and batch-size control [74], developing online subgradient methods that adapt to the data's observed geometry [22], developing new maximum principle-based training algorithms for deep neural networks [47], or unifying (stochastic) optimization frameworks [16].

Our experiments focus on quantitative finance problems. Thus, we add to the emerging literature, e.g. [13, 38, 43], recognizing the potential of transfer learning in mathematical finance, especially in situations where data is scarce.

Notation. We introduce some additional notation. Each dataset has a finite number of samples which we write as $\mathcal{D}_i = \{(x_j^i, y_j^i) \,|\, 1 \leq j \leq |\mathcal{D}_i|\}$. We assume that some feature map ϕ, is fixed and denote the features of each sample of $\mathcal{D}_i$ as $u_j^i \overset{\text{def.}}{=} \phi(x_j^i)$. We denote the joint inputs and outputs for each dataset as $X^i \overset{\text{def.}}{=} (x_1^i, \cdots, x_{|\mathcal{D}_i|}^i)^\top \in \mathbb{R}^{|\mathcal{D}_i| \times d}$, $Y^i \overset{\text{def.}}{=} (y_1^i, \cdots, y_{|\mathcal{D}_i|}^i)^\top \in \mathbb{R}^{|\mathcal{D}_i|}$, and the joint features as $U^i \overset{\text{def.}}{=} \Phi(X^i) \overset{\text{def.}}{=} (\phi(x_1^i), \cdots, \phi(x_{|\mathcal{D}_i|}^i))^\top \in \mathbb{R}^{|\mathcal{D}_i| \times p}$. We use I_p to denote a $p \times p$ identity matrix.

2. Algorithms.

We begin by describing our procedure for determining the optimal mixture weights used in (2). This procedure is detailed in Algorithm 1.

[1] See [30, Chapter 3] for details on online gradient descent.

[2] Different ridge hyperparameters κ can be used to define each regressor in Algorithm 1.

Algorithm 1: Optimal Information Sharing - Initialization to Algorithm 2

Require: Datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$, finite-rank kernel ϕ, hyperparameters[2] $\kappa > 0$, $\eta > 0$.

```
// Initialize Locally-Optimal Learners and Record Scores on Main Dataset
```

For $i : 1, \dots, N$ **in parallel**

$s^i \leftarrow \gamma\, \mathcal{W}_1(\hat{\mu}_1, \hat{\mu}_i) - \frac{\gamma}{|\mathcal{D}_i|^{1/(d+1)}}$ // Score i^{th} dataset

$T^i \leftarrow I_{s^i \leq \eta}$ // Test: Is Dataset i Similar to Focal Dataset?

end

$w^\star \leftarrow \text{Softmin}\left(s\right)^\top T$

return Optimal Weights $w^\star$.

Suppose now that we have used Algorithm 1 to compute weights $w^\star = (w_1^\star, \cdots, w_N^\star)$, and the locally optimal fKRR parameters $\theta_1^\star$, ..., $\theta_N^\star$ and $w^\star$. The key novelty in the next algorithm follows in the spirit of [47] and [17] by casting the problem of constructing a regret-optimal algorithm as an optimal control problem. We denote

$$\Theta(t) \overset{\text{def.}}{=} [\theta_1^\top(t), \cdots, \theta_N^\top(t)]^\top, \quad \Theta^\star \overset{\text{def.}}{=} [\theta_1^{\star\top}, \cdots, \theta_N^{\star\top}]^\top,$$
$$\boldsymbol{\alpha}(t) \overset{\text{def.}}{=} [\alpha_1^\top(t), \cdots, \alpha_N^\top(t)]^\top,$$

which are all elements of $\mathbb{R}^{Np}$, and introduce the dynamics that Θ follows

$$\Theta(0) = \Theta^\star, \quad \Theta(t+1) = \Theta(t) + \boldsymbol{\alpha}(t), \quad 0 \leq t \leq T-1, \tag{5}$$

which is equivalently written as $\Theta(t+1) = \Theta^\star + \sum_{u=0}^{t} \boldsymbol{\alpha}(u)$ for $t = 0, \dots, T-1$.

Then, since the optimizers of the energy (4) coincide with the optimizers of the systemic regret functional (2), searching for a regret-optimal algorithm reduces to solving for an optimal control α to minimize the cost (4) subject to (5). The optimal control problem (4) and (5) is a discrete-time linear quadratic optimal control problem, and can be solved by dynamic programming methods that are standard in the literature. For instance, by [7, Proposition 5.1], we obtain the solution for the regret-optimal algorithm in closed-form as solutions of a system of Riccati equations (13) and (14) given in Theorem 3.2.

Algorithm 2 below implements the optimal control approach in two steps. In Step 1, we solve the equation system (13) and (14) backwards for $t = T$, $T-1$, ..., 1. Then, in Step 2, we obtain the optimal control $\boldsymbol{\alpha}(t)$ and update the parameter $\Theta(t)$ for $t = 0$, 1, ... T.

Remark 2.1. By decoupling the optimization of the "dataset-relevant weights" w in Algorithm 1 from the optimization of the parameter θ in Algorithm 2, we avoid open-loop forward-backward iterations, which iterate between optimizing w and θ.

In the next section, we state our primary results, whose proofs are presented in Appendix B. We first quantify the optimality of Algorithm 1. Next, we establish the optimality, complexity, and adversarial robustness of Algorithm 2.

3. Main guarantees.

We now present our theoretical guarantees for Algorithms 1 and 2.

Algorithm 2: Regret-Optimal Optimization Algorithm

Require: Datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$, N. Iterations $T \in \mathbb{N}_+$, finite-rank kernel ϕ and hyperparameters $\lambda, \beta, \kappa, \eta > 0$.

// Get Initialize Weights and Locally-Optimal fKRR Parameters

$\theta_1^\star, \dots, \theta_N^\star, w^\star \leftarrow$ Run: Algorithm 1 with $\mathcal{D}_1, \dots, \mathcal{D}_N$, ϕ, and κ, η.

// Initialize Updates

$P(T) = [w_1^\star I_p, \cdots, w_N^\star I_p]^\top \big(\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \big) [w_1^\star I_p, \cdots, w_N^\star I_p]$

$S(T) = -[w_1^\star I_p, \cdots, w_N^\star I_p]^\top \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i$

$\Theta(0) = \Theta^\star \stackrel{\text{def.}}{=} (\theta_1^{\star\top}, \cdots, \theta_N^{\star\top})^\top$

// Generate Iterates

for $t = T-1, \dots, 1$ **do**

  // Update Driving Parameters

  $P(t) = \beta I_{Np} - \beta^2 [(\lambda+\beta) I_{Np} + P(t+1)]^{-1}$

  $S(t) = \beta [(\lambda+\beta) I_{Np} + P(t+1)]^{-1} (S(t+1) - \lambda \Theta^\star)$

  end for

for $t = 0, ..., T-1$ **do**

  // Update Control

  $\boldsymbol{\alpha}(t) = -[(\lambda+\beta) I_{Np} + P(t+1)]^{-1} [(\lambda I_{Np} + P(t+1)) \Theta(t) - \lambda \Theta^\star + S(t+1)]$

  $\Theta(t+1) = \Theta(t) + \boldsymbol{\alpha}(t)$

  end for

return Return the Optimized fKRR $f_{\theta^{w^\star}}$

3.1. **Statistical guarantees - Algorithm 1.** Assume that either d or D are greater than 1. Let $\mathcal{Z} \subset \mathbb{R}^d \times \mathbb{R}^D$ be compact of diameter 1 (with respect to the Euclidean metric). Let $\mu_1, \dots, \mu_N \in \mathcal{P}_1(\mathcal{Z})$, $|\mathcal{D}_1|, \dots, |\mathcal{D}_N| \in \mathbb{N}_+$, let $\mathcal{D}_i \stackrel{\text{def.}}{=} \{(X_j^i, Y_j^i)\}_{j=1}^{|\mathcal{D}_i|}$ and $\mathcal{D} \stackrel{\text{def.}}{=} \{\mathcal{D}_i\}_{i=1}^N$ be independent random variables where, for $i = 1, \dots, N$, $(X_j^i, Y_j^i) \sim \mu_i$, and consider the empirical measure

$$\hat{\mu}_i \stackrel{\text{def.}}{=} \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \delta_{(X_j^i, Y_j^i)}.$$

Let $\ell : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be a L_ℓ-Lipschitz loss function. For every measurable $f : \mathbb{R}^d \to \mathbb{R}^D$ and every $w \in \Delta_N$, we consider its associated *weighted* empirical $\hat{\mathcal{R}}_w^{\mathcal{D}}(f)$ and true risk $\mathcal{R}(f)$

$$\mathcal{R}(f) \stackrel{\text{def.}}{=} \mathbb{E}_{(X,Y)\sim\mu_1}\big[\ell(f(X), Y)\big] \quad \text{and} \quad \hat{\mathcal{R}}_w^{\mathcal{D}}(f) \stackrel{\text{def.}}{=} \sum_{i=1}^N \frac{w_i}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \ell(f(X_j^i), Y_j^i).$$

Note that if $w = e_1 \stackrel{\text{def.}}{=} (1, 0, \dots, 0)$, then $\hat{\mathcal{R}}_w^{\mathcal{D}}(f)$ is precisely the definition of the usual empirical risk of f computed *only* on the focal dataset $\mathcal{D}_1$.

For $i = 1, \dots, N$, define the *true similarity score* of the i^{th} dataset

$$s_i \stackrel{\text{def.}}{=} \mathcal{W}_1(\mu_1, \mu_i). \tag{6}$$

Theorem 3.1 (Non-Asymptotic Transfer Learning Guarantee (General Lipschitz Learners))**.** *Fix $L \geq 0$, and let $\mathcal{F}$ be a non-empty family of L-Lipschitz functions mapping $\mathbb{R}^d$ to $\mathbb{R}^D$. There exists a constant $C \geq 1$ (depending only on $d + D$ and*

on $\mathcal{Z}$) such that for every $0 < \delta \leq 1$, each $\eta \geq 0$, $\gamma > 0$, and every $w \in \Delta_N$, the following holds with probability at least $1-\delta$:

$$\sup_{f\in\mathcal{F}_L} \left|\mathcal{R}(f) - \hat{\mathcal{R}}_w^{\mathcal{D}}(f)\right| \leq \bar{L}C \underbrace{\overbrace{\sum_{i=1}^N w_i \left(s_i + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}}\right)}^{\text{Power of } i^{th} \text{ Dataset}} + \frac{\bar{L}C}{\gamma} \overbrace{\sum_{i=1}^N w_i \log\left(C_\eta \frac{w_i}{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}\right)}^{\text{Entropic Penalty: KL}\left(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta\right)}}_{\text{Bayesian Objective (I)}} + \underbrace{\frac{\sqrt{\bar{L}\ln(2N/\delta)}}{\sqrt{2N^\star}}}_{\text{PAC Term (II)}} \tag{7}$$

where $\bar{L} \overset{\text{def.}}{=} L_\ell \max\{1, L\}$, $\mathrm{KL}(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta) = \sum_{i=1}^N \frac{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}{C_\eta}\delta_i$, $C_\eta \overset{\text{def.}}{=} \sum_{u=1}^N I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\leq\eta}$, *and* $N^\star \overset{\text{def.}}{=} \min_{i=1,\dots,N}|\mathcal{D}_i|$.

If w and η are chosen such that $\mathbb{P}_w \ll \bar{\mathbb{P}}^\eta$ and $|w-e_1|, \eta\,\mathrm{KL}\left(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta\right) \in \mathcal{O}\left(\frac{1}{N^{\star\,1/(d+D)}}\right)$, then

$$\sup_{f\in\mathcal{F}_L} \left|\mathcal{R}(f) - \hat{\mathcal{R}}_w^{\mathcal{D}}(f)\right| \in \mathcal{O}\left(1/N^{\star\,1/(d+D)}\right).$$

In Theorem 3.1, term (I) could be optimized, whereas (II) did not depend on the weights $w \in \Delta_N$, and thus it could not. Now, even if the scores $\{s_i\}_{i=1}^N$ are not directly observable, then term (I) is not directly computable from the data, and the main result of [52] shows that in our setting the *empirical scores* $\{\hat{s}_i\}_{i=1}^N$ converge to the actual scores as the number of data points in each dataset becomes arbitrarily large, where for each $i \in [N]$,

$$\hat{s}_i \overset{\text{def.}}{=} \mathcal{W}_1\left(\hat{\mu}_1, \hat{\mu}_i\right).$$

Moreover, the main result of [52] further guarantees that each $\hat{s}_i$ converges to s_i at least at the optimal rate of $\mathcal{O}(1/\max\{|\mathcal{D}_1|, |\mathcal{D}_i|\}^{1/(d+D)})$. Consequentially, minimizing the following objective will minimize (I), and thus it will minimize the upper-bound in Theorem 3.1 for the gap between the true risk of any L-Lipschitz model (such as any aggregated model) and the weighted empirical risk. This motivates our performance objective

$$\min_{w\in\Delta_N} \left\{ \underbrace{\sum_{i=1}^N w_i \left(\mathcal{W}_1\left(\hat{\mu}_1, \hat{\mu}_i\right) + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}}\right)}_{\text{(III)}} + \underbrace{\frac{1}{\gamma}\,\mathrm{KL}\left(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta\right)}_{\text{(IV)}} \right\} \tag{8}$$

where we treat $C, \eta > 0$ as hyperparameters.

Let $\mathcal{P}([N])$ consist of all probability measures on the set $[N] \overset{\text{def.}}{=} \{1, \dots, N\}$. Any probability measure $\mathbb{P}_w$ in $\mathcal{P}([N])$ is uniquely determined by a weight w in the N-simplex, consisting of all $w \in [0,1]^N$ whose entries sum to 1. Let I_w be a random dataset index in $[N]$ distributed according to $\mathbb{P}_w \overset{\text{def.}}{=} \sum_{i=1}^N w_i\delta_i$. The objective function in (8) is the following *Bayesian optimization problem*:

$$\min_{w\in\Delta_N} \left\{ \underbrace{\mathbb{E}_{I_w\sim\mathbb{P}_w}\left[\left(\mathcal{W}_1\left(\hat{\mu}_1, \hat{\mu}_{I_w}\right) + \frac{1}{|\mathcal{D}_{I_w}|^{1/(d+D)}}\right)\right]}_{\text{(V)}} + \underbrace{\frac{1}{\gamma}\,\mathrm{KL}\left(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta\right)}_{\text{(VI)}} \right\}. \tag{9}$$

A direct application of [65, Proposition 1] shows that there exists a unique optimizer $w^\star \in \Delta_N$ of the strictly convex problem (9), and for each $i = 1, \dots, N$, it is given by

$$w_i^\star = \frac{e^{-\gamma\,\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i) - \frac{\gamma}{|\mathcal{D}_i|^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}{\sum_{u=1}^N e^{-\gamma\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u) - \frac{\gamma}{N_u^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\leq\eta}}.$$

We emphasize that Theorem 3.1 is fully general and equally holds for complicated learners, such as deep neural networks optimizing arbitrary loss functions, just as much as it holds for our specialized case of kernel ridge regressor case optimizing the MSE loss.

3.2. **Discussion: Optimal weighting.** Notice that as $\eta \to 0$, $w_i^\star = I_{1=i}$, and as $\eta \to \infty$, w_i tends to the "unthresholded soft*min* function" applied to the vector of "dataset relevances" $\big(\mathcal{W}_1(\hat{\mu}_1, \hat{\mu}_i) + \mathcal{D}_i^{-1/(d+D)}\big)_{i=1}^N$ with temperature parameter γ. Let us note that term (V) in the Bayesian objective (9) improves by a factor of

$$\text{T-Gain}_{\eta,\gamma} \overset{\text{def.}}{=} \sum_{i=1}^N \left[\frac{e^{-\gamma\, \mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i) - \frac{\gamma}{|\mathcal{D}_i|^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\le\eta}}{\sum_{u=1}^N e^{-\gamma \mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u) - \frac{\gamma}{N_u^{1/(d+D)}}}\, I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\le\eta}} - I_{i=1} \right] \times \left(\mathcal{W}_1(\hat{\mu_1}, \hat{\mu}_i) + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}} \right) \tag{10}$$

when performing transfer learning using the optimal weights $w^\star$ rather than performing no transfer learning (i.e. weights $w = e_1$). The quantity $\text{T-Gain}_{\eta,\gamma}$ in the above expression thus quantifies an upper bound on the gain from performing transfer learning using our optimal weighting scheme versus not performing any transfer learning.

Setting $\gamma = 0$ yields the equal weighting scheme $w_i^\star = 1/N$ for each i *included in the transfer learning problem*. In this case, the gain reduces to

$$\text{T-Gain}_{\eta,0} = \sum_{i=1}^N \left(\frac{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\le\eta}}{\sum_{u=1}^N I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\le\eta}} - I_{i=1} \right) \left(\mathcal{W}_1(\hat{\mu_1}, \hat{\mu}_i) + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}} \right) \tag{11}$$

and we see that η controls how many datasets are included based on their similarity to the focal dataset $\mathcal{D}_1$ (encoded into the empirical measure $\hat{\mu}_1$). Thus, any dataset $\mathcal{D}_i$ is deemed to be too dissimilar to the focal dataset $\mathcal{D}_1$ if its associated empirical measure $\hat{\mu}_i$ is at a distance of η or greater from $\hat{\mu}_1$. In this case, it is *not included* in the transfer learning problem.

For instance, in the extreme case where $\eta = 0$, (11) reduces to 0 and no gain is made. Conversely, if η is large ($\eta \ge \eta^\star \overset{\text{def.}}{=} \max_{i=1,\dots,N} \mathcal{W}_1(\hat{\mu}_1, \hat{\mu}_i)$, then (11) simplifies to

$$\text{T-Gain}_{\eta^\star,0} = \sum_{i=1}^N \left(\frac{1}{N} - I_{i=1} \right) \left(\mathcal{W}_1(\hat{\mu_1}, \hat{\mu}_i) + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}} \right).$$

Finally, we remark that sending γ to infinity in (9) implies that $w^\star$ tends to e_1 and $\lim_{\gamma\to 0} w_i^* = \frac{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\le\eta}}{\sum_u I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\le\eta}}$. In this case, no transfer can happen, and thus $\lim\limits_{\gamma\to\infty} \text{T-Gain}_{\eta,\gamma} = 0$.

3.3. **Optimization guarantees - Algorithm 1 and 2.** This section contains the description of the *unique* regret-optimal algorithm and an analysis of its theoretical properties. Throughout our manuscript, we maintain the following two assumptions.

Assumption 3.1 ($\mathcal{D}$-Boundedness)**.** *There are constants $K_x, K_y > 0$ such that all $(x_j^i, y_j^i) \in \mathcal{D}$ satisfy $\|u_j^i\|_2 \le K_x$ and $|y_j^i|^2 \le K_y$, where $u_j^i = \phi(x_j^i)$.*

With the notation introduced before, the object $[w_1^\star I_p, \cdots, w_N^\star I_p]$ above is a $p \times Np$ matrix. Our next result shows that Algorithm 2 and its iterates are regret-optimal.

Theorem 3.2 (Algorithm 2 Computes the Unique Regret-Optimal Algorithm)**.** *Suppose that Assumption 3.1 holds and fix* $\Theta^\star \in \mathbb{R}^{N\,p}$ *as in* (26)*. Then, the systemic regret functional* $\mathcal{R}$ *defined in* (2) *admits a unique minimizer* $\hat{\Theta} \overset{\text{def.}}{=} (\hat{\Theta}(t))_{t=0}^T$ *of the form* (27) *with* $\hat{\Theta}(0) = \Theta^\star$ *and increments*

$$\begin{aligned}&\Delta\hat{\Theta}(t)\\ &= -[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}\big[(\lambda I_{Np} + P(t+1))\hat{\Theta}(t) - \lambda\Theta^\star + S(t+1)\big],\end{aligned} \tag{12}$$

where, for $t = 1, \dots, T$*, the matrices* $P(t)$ *and* $S(t)$ *in* (12) *are determined by*

$$\begin{cases} P(T) = [w_1^\star I_p, \cdots, w_N^\star I_p]^\top \big(\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}\big)[w_1^\star I_p, \cdots, w_N^\star I_p], \\ P(t) = \beta I_{Np} - \beta^2[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}, \end{cases} \tag{13}$$

$$\begin{cases} S(T) = -[w_1^\star I_p, \cdots, w_N^\star I_p]^\top \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i, \\ S(t) = \beta\big[(\lambda+\beta)I_{Np} + P(t+1)\big]^{-1}(S(t+1) - \lambda\Theta^\star). \end{cases} \tag{14}$$

In particular, for each $t = 0, \dots, T-1$*, we have* $\Delta\hat{\Theta}(t) = \boldsymbol{\alpha}(t)$*, where* $\boldsymbol{\alpha}$ *is as in Algorithm 2.*

We evaluate the computational complexity of Algorithm 1 for computing the optimal weights $w^\star$.

Theorem 3.3 (Computational Complexity of Algorithm 1)**.** *The complexity of computing the optimal weights* $w^\star$ *by Algorithm 1 is*

$$\mathcal{O}\big(N(|\mathcal{D}_1|D_{-1}^\star)^{3.5}L + N\log_2 d_{prec}\big),$$

where $D_{-1}^\star = \max_{i=2,\dots,N}|\mathcal{D}_i|$*,* L *denotes the bit-length of each data point in* $\{\mathcal{D}_i\}_{i=1}^N$*, and* d_{prec} *is the number of digits of precision for computing exponential functions.*

Next, we evaluate the computational complexity of Algorithm 2 in the regime where the number of datasets is much larger than the typical number of instances/data per dataset.

Theorem 3.4 (Computational Complexity of Algorithm 2)**.** *If* $\frac{1}{N}\sum_{i=1}^N |\mathcal{D}_i| \le N$*, then the computational complexity of computing the sequence* $(\hat{\Theta}(t))_{t=0}^T$ *in the regret-optimal algorithm is* $\mathcal{O}\big(N^2p^3 + T\,(Np)^{2.373}\big)$.

3.3.1. *Robustness guarantees.* We quantify the sensitivity of Algorithm 2 to adversarial attacks by a fictitious *malicious adversary*. Our objective is to understand how *sensitive* our algorithm is to noise in the training data. We study the robustness of Algorithm 2 on a family of perturbations that switch a given percentage $0 \le q \le 1$ of all training data by selecting fake data points at a distance of at-most $\varepsilon \ge 0$ for any perturbed datapoint. Thus, for a fixed set of datasets $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^N$, and q, ε as above, we define the class $\mathbb{D}^{q,\varepsilon}$ of *adversarially generated datasets* of "magnitude" (q,ε) as all sets of datasets $\{\widetilde{\mathcal{D}}_i\}_{i=1}^N$, $\widetilde{\mathcal{D}}_i = \{\{(\tilde{x}_j^i, \tilde{y}_j^i)\,|\,1 \le j \le |\mathcal{D}_i|\} \subseteq \mathbb{R}^d \times \mathbb{R}$ satisfying Assumption 3.1, as well as

$$\underbrace{\frac{\#\{(i,j) : u_j^i \ne \tilde{u}_j^i \text{ or } y_j^i \ne \tilde{y}_j^i\}}{\sum_{i=1}^N |\mathcal{D}_i|} \le q}_{\text{Attack Persistence}} \quad \text{and} \quad \underbrace{\max_{i,j}\big\{\max\big(\|u_j^i - \tilde{u}_j^i\|_2, |y_j^i - \tilde{y}_j^i|^2\big)\big\} \le \varepsilon}_{\text{Attack Severity}},$$

where the indices (i,j) run over $\{(i,j)\,|\,1 \le i \le N,\ 1 \le j \le |\mathcal{D}_i|\}$ and $\tilde{u}_j^i \overset{\text{def.}}{=} \phi(\tilde{x}_j^i)$.

The updates of the unique regret-optimal algorithm $\hat{\Theta}^{0\cdots T}$ are determined by the sequence $\boldsymbol{\alpha}_\cdot \overset{\text{def.}}{=} (\boldsymbol{\alpha}(t))_{t=0}^{T}$ computed by Algorithm 2. It is convenient to emphasize this connection on $\boldsymbol{\alpha}_\cdot$ and the connection with the dataset $\mathcal{D}$ when expressing the following theorem, with the energy (4) determining the value of the systematic regret functional $\mathcal{R}$. Therefore, we write $\mathcal{L}(\boldsymbol{\alpha}_\cdot; \mathcal{D}) \overset{\text{def.}}{=} \mathcal{L}(\hat{\Theta}^{0\cdots T})$.

Theorem 3.5 (Adversarial Robustness)**.** *Fix a $\mathcal{D}$ satisfying Assumption 3.1. For any level of "attack persistence" $0 \leq q \leq 1$ and any degree of "attack severity" $\varepsilon \geq 0$, we have that*

$$\left|\mathcal{L}(\boldsymbol{\alpha}_\cdot; \mathcal{D}) - \mathcal{L}(\widetilde{\boldsymbol{\alpha}}_\cdot; \widetilde{\mathcal{D}})\right| \leq \mathcal{O}\Big(\varepsilon\sqrt{\bar{N}\, q}\Big),$$

where $\widetilde{\boldsymbol{\alpha}}_\cdot$ is the output of Algorithm 2 for any given adversarially-generated set of datasets $\{\widetilde{\mathcal{D}}_i\}_{i=1}^{N} \in \mathbb{D}^{q,\varepsilon}$ and $\bar{N} = \sum_{i=1}^{N} |\mathcal{D}_i|$. Here, $\mathcal{O}$ hides a constant depending only on Assumption 3.1 and on $\sqrt{p}$.

Before moving on to our experimental section, we ask: "Is it possible further to reduce the computational complexity of our meta-optimization algorithm while maintaining most of its performance?" The next, and final, theoretical section, considers precisely this; namely, we study a situation in which the computational complexity of the regret-optimal algorithm can be reduced. This is achieved by exploiting symmetries in the optimal policy computed by Algorithm 2.

3.4. **An accelerated heuristic.** Examining the explicit update rules, we notice that *if* the weights w defining the energy (4) were to be $\overline{1/N} \overset{\text{def.}}{=} (1/N, \ldots, 1/N)$, then the matrices defining the optimal policy/control's updates would become highly symmetric Toeplitz matrices, depending on exactly three p-dimensional block sub-matrices. This allows us to parameterize the entire regret-optimal algorithm for the sub-optimal weights $\overline{1/N}$ using only three low-dimensional systems whose dimension is *independent* of the number of datasets N. In line with this, Algorithm 3 implements the regret-optimal algorithm for the weight specification $\overline{1/N}$ by only keeping track of these three optimal low-dimensional systems, and then efficiently recombining them by leveraging the Toeplitz structure defining the optimal policy/control. These low-dimensional systems are defined using the *helper functions* listed in equations (15)-(17).

The approximate near-optimality of Algorithm 3 only depends on the stability of the optimal policy/control, defining the algorithm's updates as a function of the weights w. If the optimal weights defining the systemic regret functional (2) are close to $\overline{1/N}$, then Algorithm 3 is nearly regret-optimal. Otherwise, the sub-optimality it incurs in favor of computational efficiency is explicitly quantifiable and depends only on the difference between the optimal weights computed in Algorithm 1 and the symmetric weights $\overline{1/N}$.

The matrix-valued functions $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ in Algorithm 3 are defined as

$$\gamma_1(\cdot) \overset{\text{def.}}{=} \big\{(\lambda+\beta)I_p + \pi_1(\cdot)(N-1)\pi_2(\cdot)\big[(\lambda+\beta)I_p + \pi_1(\cdot) + (N-2)\pi_2(\cdot)\big]^{-1}\pi_2(\cdot)\big\}^{-1}, \tag{15}$$

$$\gamma_2(\cdot) \overset{\text{def.}}{=} -\gamma_1(\cdot)\pi_2(\cdot)\big[(\lambda+\beta)I_p + \pi_1(\cdot) + (N-2)\pi_2(\cdot)\big]^{-1}, \tag{16}$$

where

$$\pi_1(T) = \pi_2(T) = \frac{1}{N^3}\sum_{i=1}^{N}\sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \qquad \text{and} \qquad \pi_3(T) = -\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i$$

and, for $t = T-1, \dots, 0$,

$$\begin{aligned}
\pi_1(t) &= \beta I_p - \beta^2 \gamma_1(t+1) \\
\pi_2(t) &= -\beta^2 \gamma_2(t+1) \\
\pi_3(t) &= \beta\big[\gamma_1(t+1) + (N-1)\gamma_2(t+1)\big]\big(\pi_3(t+1) - \lambda\theta^\star_{(N)}\big)
\end{aligned}$$

where $\theta^\star_{(N)} = \frac{1}{N}\sum_{i=1}^N \theta^\star_i$.

The updates of the low-dimensional systems which Algorithm 3 exploits are implemented via the helper function $\hat{\alpha}$, defined as

$$\begin{aligned}
\widehat{\alpha}\big(\theta_i, ((\theta_j)_{j=1,j\neq i}^N), (\gamma_i)_{i=1}^2, (\pi_i)_{i=1}^3\big) \overset{\text{def.}}{=} &-(\gamma_1-\gamma_2)(\lambda I_p + \pi_1 - \pi_2)\theta_i \\
&- \big\{\gamma_1\pi_2 + \gamma_2[\lambda I_p + \pi_1 + (N-2)\pi_2]\big\}\sum_{j=1}^N \theta_j \\
&- \big[\gamma_1 + (N-1)\gamma_2\big](\pi_3 - \lambda\theta^\star_{(N)}). \qquad (17)
\end{aligned}$$

We record the matrix-valued functions $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ used to define the updates in Algorithm 3.

Algorithm 3: Accelerated (Nearly) Regret-Optimal Algorithm

Require: Datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$, N. Iterations $T \in \mathbb{N}_+$, finite-rank kernel ϕ, hyperparameters $\lambda, \beta, \kappa, \eta > 0$.

// Get Initialize Weights and Locally-Optimal fRKR Parameters

$\theta^\star_1, \dots, \theta^\star_N, w^\star \leftarrow$ Run: Algorithm 1 with $\mathcal{D}_1, \dots, \mathcal{D}_N$, ϕ, and κ, η.

// Initialize Updates

$\pi_1(T) = \pi_2(T) = (1/N^3)\sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}$

$\pi_3(T) = -(1/N^2)\sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i$

$\theta_1(0) = \theta^\star_1$, $\theta_2(0) = \theta^\star_2$, $\dots$, $\theta_N(0) = \theta^\star_N$

$\theta^\star_{(N)} = \frac{1}{N}\sum_{i=1}^N \theta^\star_i$.

// Generate Iterates

for $t = T-1, \dots, 0$ **do**

  // Update Low-Dimensional Driving Parameters

  $\pi_1(t) = \beta I_p - \beta^2\gamma_1(t+1)$ // $\gamma_1(\cdot)$ is defined by (15)

  $\pi_2(t) = -\beta^2\gamma_2(t+1)$ // $\gamma_2(\cdot)$ is defined by (16)

  $\pi_3(t) = \beta\big[\gamma_1(t+1) + (N-1)\gamma_2(t+1)\big](\pi_3(t+1) - \lambda\theta^\star_{(N)})$

end for

for $t = 0, ..., T-1$ **do**

  // Update Low-Dimensional Control

  for $i = 1, ..., N$ **do**

    $\widehat{\alpha}_i(t) = \widehat{\alpha}\big(\theta_i(t),\ (\theta_j(t))_{j=1,j\neq i}^N,\ (\gamma_i(t+1))_{i=1}^2,\ (\pi_i(t+1))_{i=1}^3\big)$

    $\theta_i(t+1) = \theta_i(t) + \widehat{\alpha}_i(t)$

  end for

end for

return Synchronized fKRR $f_{\theta^{w^\star}}$

The degree to which Algorithm 3 is nearly regret-optimal depends on how far the weights w determined by Algorithm 1 are from the equal weighting $\overline{1/N}$. This is precisely quantified by the following results.

Theorem 3.6 (Near-Regret Optimality)**.** *Suppose that $\mathcal{D}$ satisfies Assumption 3.1, and let $\hat{\Theta}^{0\cdots T}$ denote the unique regret-optimal algorithm[3] computed according to Algorithm 2. Then, the output $\vartheta^{0\cdots T}$ of Algorithm 3 satisfies.*

(i) ***Near-Optimality:*** *The difference $\left|\mathcal{L}(\Theta^{0\cdots T}) - \mathcal{L}(\vartheta^{0\cdots T})\right|$ is of the order*

$$\mathcal{O}\Big(\bar{N}N^3 \max_{1\le i,k,l\le N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| + \bar{N}N^{3/2}\max_{1\le k,l\le N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| + \bar{N}\max_{1\le i\le N}\Big|\frac{1}{N} - w_i^\star\Big| + N\big\|\Theta^\star - \Theta^\star_{(N)}\big\|_2\Big).$$

(ii) ***Near-Optimal Increments:*** *The difference $\big\|\Delta\Theta^{0\cdots T-1} - \Delta\vartheta^{0\cdots T-1}\big\|_{\ell^1}$ is of the order*

$$\mathcal{O}\Big(\bar{N}N^{3/2}\max_{1\le i,k,l\le N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| + \bar{N}N^{1/2}\max_{1\le k,l\le N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| + \big\|\Theta^\star - \Theta^\star_{(N)}\big\|_2\Big).$$

Here, $\mathcal{O}$ hides a constant depending only on λ, β, $\Theta^\star$, $(w_1^\star, ..., w_N^\star)$, T, K_x, and in K_y.

The next result shows that Algorithm 3 accelerates the regret-optimal algorithm by a linear factor in N and by a quadratic factor in p when N is large.

Theorem 3.7 ($\mathcal{O}(Np^2)$-Acceleration over Regret-Optimal Algorithm)**.** *With the convention that floating-point arithmetic operations are $\mathcal{O}(1)$, we have that:*

(i) ***Complexity:*** *Algorithm 3 has a complexity of $\mathcal{O}(TNp\max\{N,p\}))$.*

(ii) ***Acceleration over Regret-Optimal Algorithm:*** *The regret-optimal algorithm has a complexity of $\mathcal{O}(TN^3p^3)$.*

When $N \ge p$, Algorithm 3 requires $\mathcal{O}(Np^2)$ fewer operations than the regret-optimal algorithm.

Finally, we establish that Algorithm 3 is the regret-optimal algorithm, optimizing the systemic regret functional (4) in the "symmetric" case where the optimal weights $w^\star$ are all $(1/N, \ldots, 1/N)$.

Theorem 3.8 (Regret-Optimality in the Symmetric Case)**.** *Suppose that $w_i^\star = 1/N$ for $i = 1, \ldots, N$. Then, $\hat{\Theta}^{0\cdots T} = \vartheta^{0\cdots T}$.*

The computational complexity of the matrix-valued function $P(\cdot)$, as determined by (47), does not primarily stem from the high dimensionality of the output matrices, but rather from their inherent lack of symmetries. As we will see later, such symmetry will allow us to efficiently encode $P(\cdot)$ into low-dimensional structures, therefore completely mitigating the high-dimensionality effect.

We therefore introduce the following control, which is optimal for a "symmetrized version" of the control problem (45), where $w^\star = (1/N, \ldots, 1/N)$. This advantages this surrogate problem in that the associated P function, which we denote by $\widehat{P}$, outputs highly symmetric[4] Toeplitz matrices which are determined by exactly three factors. This allows us to encode the entire control problem into a control problem on $\mathbb{R}^{\bar{O}(P)}$ whose dimension is independent of N. Furthermore, the optimal controls of both problems can be related by stability estimates, depending only on the deviation

[3] That is, the unique optimizer of (4).

[4] As formalized in Proposition F.3.

of $(1/N,\dots,1/N)$ from $w^\star$, which allows us to infer the degree of sub-optimality of our $\widehat{P}$. Therefore, we consider the control

$$\widehat{\boldsymbol{\alpha}}(t) = -\left[(\lambda+\beta)I_{Np} + \widehat{P}(t+1)\right]^{-1}\left[(\lambda I_{Np} + \widehat{P}(t+1))\Theta(t) - \lambda\Theta^\star_{(N)} + \widehat{S}(t+1)\right]. \tag{18}$$

We will show that the algorithm under control (18) enjoys lower computational complexity in comparison with the regret-optimal algorithm (45), whose computational complexity is summarized in Theorem 3.4. Furthermore, we will then establish the near-regret optimality of (18) stated in Theorem 3.6.

The associated matrix-valued functions $\widehat{P}$ and $\widehat{S}$ are uniquely determined by

$$\begin{cases} \widehat{P}(t) = \beta I_{Np} - \beta^2[(\lambda+\beta)I_{Np} + \widehat{P}(t+1)]^{-1}, \\ \widehat{P}(T) = [(1/N)I_p,\cdots,(1/N)I_p]^\top\big(\sum_{i=1}^N (1/N)\sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}\big) \\ \qquad\qquad \times[(1/N)I_p,\cdots,(1/N)I_p], \end{cases} \tag{19}$$

$$\begin{cases} \widehat{S}(t) = \beta[(\lambda+\beta)I_{Np} + \widehat{P}(t+1)]^{-1}\big(\widehat{S}(t+1) - \lambda\Theta^\star_{(N)}\big), \\ \widehat{S}(T) = -[(1/N)I_p,\cdots,(1/N)I_p]^\top \sum_{i=1}^N (1/N)\sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i, \end{cases} \tag{20}$$

where $[I_p,\cdots,I_p]$ denotes the $Np\times p$-dimensional matrix formed by concatenating N copies of the $p\times p$-dimensional identity matrix $\frac{1}{N}I_p$. The parameter $\Theta^\star_{(N)}$ is defined as

$$\Theta^\star_{(N)} \overset{\text{def.}}{=} [\theta^{\star\top}_{(N)},\cdots,\theta^{\star\top}_{(N)}]^\top, \quad \theta^\star_{(N)} \overset{\text{def.}}{=} \frac{1}{N}\sum_{i=1}^N \theta^\star_i.$$

4. Experiments.

We experimentally ablate our theory. We first focus on *American option pricing* in quantitative finance, although our method could equivalently be used for any other transfer learning problem; see, for example, all the works surveyed in [68]. We consider random kernels whose feature map ϕ in (3) is a randomized neural network with ReLU activation. Then, we study the convergence of the regret-optimal algorithm compared to standard gradient descent. Since there are no available benchmarks for the regret-optimal meta-optimization problem, we only benchmark against the standard gradient descent in our first set of experiments.

4.1. Transfer learning in American option pricing.

We test our algorithm in the context of the optimal stopping problem for American option pricing, a computationally highly challenging problem which has recently become tractable in higher-dimensional settings through deep learning [9, 11, 32, 36]. This is a useful example since American option pricing (see Appendix E) is computationally expensive, and one often does not have access to large datasets, but does have access to several small and similar datasets arising from comparable market regimes.

We generate our synthetic "true market dynamics" $\mathcal{D}_1$ and several more or less similar datasets $\mathcal{D}_2,\dots,\mathcal{D}_N$. We always use the randomized least squares Monte Carlo (RLSM) algorithm of [32] (see Appendix E) for American option pricing, and we only vary the optimization method used to train the randomized neural networks in the RLSM. We compare RO against the following baselines. *Local optimizer (LO-i):* The local optimizers $\theta^\star_i$ solf the individual optimization problem on the datasets $\mathcal{D}_i$ without considering the other datasets. *Mean local optimizer (MLO):* The mean of the local optimizers $\bar{\theta}^\star = \frac{1}{N}\sum_{i=1}^N \theta^\star_i$ is the bagging aggregation. *Joint optimizer (JO):* The joint optimizer $\theta^\star_{\mathcal{N}}$ with $\mathcal{N} = \{1,\dots,N\}$ solves the optimisation

problem of the pooled regime $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$. *Joint subset optimizer (JSO-$\mathcal{I}$):* For any subset $\mathcal{I} \subset \mathcal{N}$, this is the joint optimizer of the dataset $\mathcal{D}_\mathcal{I} = \bigcup_{i\in\mathcal{I}} \mathcal{D}_i$.

In this application, we are only interested in the terminal parameters $\Theta(T)$, thus a stable convergence behavior is not important. Hence, we choose $T = 1$ to fully focus on the terminal loss under the transfer learning constraint (cf. Appendix C.2.2). As with fKRR, we use the same randomized neural networks as in Section 4.2. The random weights of the randomized neural networks as well as the randomly sampled paths both introduce some randomness into the pricing problem. Therefore, we follow [32] by always running the experiments $n_{runs} = 100$ times with different seeds and taking means over the prices. For easier comparison, we then consider the relative performance (RP) on our main dataset, computed as the mean price of any given method divided by the mean price of the local optimizer on the main dataset. Additionally, we compute asymptotically valid (by the central limit theorem) 95%-confidence-intervals as $RP \pm z_{0.975}\hat{\sigma}/\sqrt{n_{runs}}$, where $z_{0.975}$ is the 0.975-quantile of the standard normal distribution and $\hat{\sigma}$ is the standard deviation of RP (computed as the sample standard deviation of the prices over the runs divided by the mean price of the local optimizer on the main dataset). To make the results comparable between the optimization methods, we use the same seeds for all of them such that run i always uses the exact same paths and the exact same random weights in the randomized neural network, independently of the chosen method. Moreover, we always use $\nu_2 \geq 50,000$ evaluation paths such that the noise of the Monte Carlo approximation of the expectations is relatively small when evaluating the prices, even if we use much smaller training set sizes ν_1.

Experiment 4.1. We study the ability of our method to transfer knowledge of a large number of small equally-sized similar and dissimilar datasets to the main task. We generate 13 different datasets (the main one, 6 similar, and 6 dissimilar ones) with $\nu_1 = 100$ training samples each, all from the well-studied *Heston model*, defined by

$$\begin{aligned} dX_t &= (r-\delta)X_t dt + \sqrt{v_t} X_t dW_t, \\ dv_t &= -k(v_t - v_\infty)dt + \sigma\sqrt{v_t} dB_t. \end{aligned} \tag{21}$$

The main dataset $\mathcal{D}_1$ has the parameters $r = 0.05$, $\delta = 0.1$, $k = 2$, $v_\infty = 0.01$, $\sigma = 0.2$, and $\rho = -0.3$ (the correlation between the two Brownian motions W and B) and starting values $X_0 = 100$ and $v_0 = v_\infty$. The other datasets are defined as all combinations of $r \in \{0.05, 0.5\}$, $\sigma \in \{0.15, 0.2, 0.25\}$, and $v_\infty \in \{0.005, 0.015\}$, and otherwise the same parameters as the main dataset. The biggest difference between the datasets constitutes the drift r, while the other parameters only lead to small gaps in the dataset distributions. In line with this, the 6 datasets with $r = 0.05$ are similar to $\mathcal{D}_1$, while the other 6 datasets with $r = 0.05$ are very different from $\mathcal{D}_1$. For all datasets, we use a max call option with strike $K = 100$ on $d = 2$ i.i.d. stocks of the given model with $M = 9$ equidistant exercise dates until the maturity $T_{mat} = 3$. The discounting factor is $e^{-r\,T_{mat}\,t/M}$ for $0 \leq t \leq M$. Moreover, we use a randomized neural network with one hidden layer with 300 nodes, and all methods use the ridge coefficient $\kappa = 2$, and, where applicable, regret coefficients $\lambda = 2, \beta = 1$.

Results are reported in Table 1a. For our regret-optimal method (RO), we use information sharing levels $\eta \in \{10, 100, 500\}$, where $\eta = 100$ leads to the best result outperforming the local optimizer of the main dataset (LO-1) by 9%. It also outperforms all other local optimizers (LO-i), the mean local optimizer (MLO),

and the joint optimizer (JO). For $\eta = 10$, we have similar performance as LO-1, and for $\eta = 500$ we also have significant outperformance, though smaller than for $\eta = 100$. The local optimizers on the 6 datasets similar to the main one (LO-2, ..., LO-7) have similar performance as LO-1, while the other 6 (LO-8, ..., LO-13) are significantly worse. Therefore, we claim that our algorithm should mainly transfer knowledge from those 6 similar datasets, while avoiding to learn from the others. In line with this, we provide results of the joint optimizer on the similar datasets $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_7$, which performs better than the other baselines and outperforms LO-1 by 6.2%. In particular, this confirms that knowledge can be transferred from the datasets $\mathcal{D}_2, \ldots, \mathcal{D}_7$. This joint subset optimizer is significantly outperformed by our regret-optimal algorithm RO ($\eta = 100$) by about 3%, which shows that the transfer learning abilities of our method are better than simply selecting the datasets most similar to $\mathcal{D}_1$.

Additionally, we compare our method to the local optimizer for the main dataset, which uses either 700 (the number of training samples in the 7 similar datasets) or $50,000$ training samples. Importantly, both are just references that would not be available in a real-world setting where the datasets are limited. The first one only leads to a slight increase in RP of about 1%, which shows that our method extracts nearly as much knowledge out of the (same amount of) samples of the similar (but not equally distributed) datasets. The second one (which is considered a good proxy for the true American option price on the main dataset) has about 10% better performance. This also shows the limitations of our method; in particular, it can only transfer as much knowledge as available in the other datasets.

TABLE 1. **Ablation Results: American Option Pricing.** Relative performance (RP) and 95%-confidence-intervals for different optimization methods. We compare the local optimizers on the different datasets (LO-n) with the mean local optimizer (MLO), the joint optimizer (JO), and our regret-optimal method (RO). The *oracle* local optimizer on the main dataset with additional training samples (standard: 100 samples per dataset) is included. See Table 2 for additional details.

(A) **Experiment** 1 (compressed)

method	RP	95%-CI
LO-1	1.000	[0.991; 1.009]
LO-2, ..., LO-7	≥ 0.966	$[\geq 0.957; \leq 0.994]$
LO-8, ..., LO-13	≤ 0.725	$[\geq 0.696; \leq 0.736]$
MLO	0.823	[0.813; 0.833]
JO	0.886	[0.878; 0.894]
JSO-$\{1, \ldots, 7\}$	1.062	[1.057; 1.067]
RO ($\eta = 100$)	**1.090**	**[1.086; 1.095]**
LO-1 (700 tr. samp.)	1.100	[1.097; 1.104]
LO-1 (50K tr. samp.)	1.194	[1.192; 1.195]

(B) **Experiment** 2

method	RP	95%-CI
LO-1	1.000	[0.993; 1.007]
LO-2	0.773	[0.765; 0.782]
LO-3	1.003	[0.994; 1.012]
MLO	0.932	[0.920; 0.944]
JO	0.763	[0.754; 0.772]
RO ($\eta = 10$)	0.993	[0.985; 1.000]
RO ($\eta = 50$)	1.040	[1.034; 1.046]
RO ($\eta = 100$)	**1.041**	**[1.036; 1.047]**

Experiment 4.2. We study the adversarial robustness of our model when having a dominating dataset that should be avoided due to its dissimilarity. In particular,

TABLE 2. Relative performance (RP) and 95%-confidence-intervals for different optimization methods in Experiment 4.1. We compare the local optimizers on the different datasets (LO-n) with the mean local optimizer (MLO), the joint optimizer (JO), and our regret-optimal method (RO). The *oracle* local optimizer on the main dataset with additional training samples (standard: 100 samples per dataset) is included. When the information sharing parameter η is set to 100, the regret-optimal (RO) algorithm outperforms all "non-oracle" baselines.

method	RP	95%-CI
LO-1	1.000	[0.991; 1.009]
LO-2	0.985	[0.977; 0.994]
LO-3	0.966	[0.958; 0.973]
LO-4	0.975	[0.966; 0.984]
LO-5	0.966	[0.957; 0.975]
LO-6	0.979	[0.971; 0.988]
LO-7	0.978	[0.971; 0.986]
LO-8	0.721	[0.711; 0.730]
LO-9	0.720	[0.711; 0.729]
LO-10	0.706	[0.696; 0.716]
LO-11	0.725	[0.715; 0.736]
LO-12	0.707	[0.697; 0.716]
LO-13	0.718	[0.708; 0.727]
MLO	0.823	[0.813; 0.833]
JO	0.886	[0.878; 0.894]
JO (datasets 1-7)	1.062	[1.057; 1.067]
RO ($\eta = 10$)	1.000	[0.990; 1.010]
RO ($\eta = 100$)	**1.091**	**[1.087; 1.095]**
RO ($\eta = 500$)	1.050	[1.043; 1.057]
LO-1 (700 tr. samp.)	1.100	[1.097; 1.104]
LO-1 (50K tr. samp.)	1.194	[1.192; 1.195]

we generate 3 datasets, a small main dataset, another small and similar dataset to transfer knowledge from, and a dominating much larger dissimilar dataset. The main dataset is generated from a rough Heston model

$$dX_t = (r-\delta)X_t dt + \sqrt{v_t}X_t dW_t,$$
$$v_t = v_0 + \int_0^t \frac{(t-s)^{H-1/2}}{\Gamma(H+1/2)}\kappa(v_\infty - v_s)ds + \int_0^t \frac{(t-s)^{H-1/2}}{\Gamma(H+1/2)}\sigma\sqrt{v_s}dB_s,$$

which is similar to the Heston model except that the volatility is a rough process with Hurst parameter $H \in (0, 1/2]$ instead of $H = 1/2$. Datasets $\mathcal{D}_2$ and $\mathcal{D}_3$ are from

a standard Heston model, where $\mathcal{D}_2$ is dissimilar from $\mathcal{D}_1$ and $\mathcal{D}_3$ is similar to it. For $\mathcal{D}_1$ and $\mathcal{D}_3$, we use $\nu_1 = 100$ training samples, while $\mathcal{D}_2$ has $\nu_1 = 50,000$ samples, such that it dominates the others in its size. The second and third datasets are generated from a standard Heston model (i.e. $H = 1/2$) with the same parameters, except that the drift is set to $r = 0.5$ for the second dataset. In particular, dataset 3 is the closest non-rough dataset to the main dataset, while dataset 2 has a very different distribution. All the other choices are the same as in Experiment 4.1.

The results are reported in Table 1b. The baselines MLO and JO perform significantly worse than the LO on the main dataset. As expected, the JO suffers much more from the dominating $\mathcal{D}_2$ than the MLO, since JO weights all samples equally. Indeed, the MLO would suffer more from a large number of datasets that are quite different from the main dataset, no matter their sample size. In contrast to this, our regret optimal method with $\eta \in \{50, 100\}$ outperforms the LO-1 by 4%, showing also that a quite different dataset dominating in sample size is not a problem.

Both of these experiments show that our regret optimal method is well suited for automatic (up to the hyper-parameter η) dataset selection in transfer learning tasks that have potentially large numbers of datasets and do not allow a (manual) pre-selection. In particular, neither a large number of "bad" datasets (of which the MLO suffers) nor some "bad" datasets that dominate in sample size (of which the JO suffers) constitute a problem for our regret-optimal method. Moreover, the sample efficiency when transferring knowledge from similar datasets to the main task is very high (outperforming the JO on the respective subset of similar datasets and nearly being on par with the LO on the main dataset using a larger training sample), as we saw in Experiment 4.1.

Relation to random-feature pipelines for optimal stopping. Randomized-feature networks (RFNs) provide an efficient and scalable approach to American option pricing, cf. [32], since they amortize the computational cost of training the network's hidden layers by effectively linearizing the learning problem via randomization of all hidden layers; this design unlocks proven scalability to high-dimensional option pricing problems. In our American option pricing examples, the training pipeline is built on top of these RFN methods, and therefore necessarily exhibits a similar training-time profile and similar sensitivities to random initializations. A more in-depth, statistically focused analysis of these aspects, akin to [28], is a natural direction for further investigation.

4.2. **Convergence of the regret-optimal algorithm.** We study the evolution of the energy functional (4) and loss achieved in each iteration by our regret-optimal Algorithm 2 (RO) with weights initialized by Algorithm 1 by computing $\mathcal{L}(\Theta^{0\cdots t})$ as defined in (4) but for intermediate iterations t, as well as the loss $l(\Theta(t), w; \mathcal{D})$. We compare the regret-optimal algorithm with the results when using gradient descent (GD) to optimize the loss directly with the version of the RO algorithm where each w_i is fixed to $1/N$ (ARO).

We use $\Theta(0) = \Theta^\star_{(N)}$ and replace $\Theta^\star$ by $\Theta^\star_{(N)}$ in (4). For all algorithms, we use a ridge coefficient $\kappa = 0$, regret coefficients λ as specified, $\beta = 1$, and Algorithm 1 with information sharing level $\eta = 10$ to get the corresponding optimal weights $w^\star$ and to initialize $\Theta(0)$ with the locally optimal parameters. For our convergence analysis we use $N = 5$ randomly generated datasets. In particular, each dataset $\mathcal{D}_i$ corresponds to a neural network g_{ω_i} with fixed randomly-generated hidden weights and biases.

Here, ω_i denotes the vector of all these hidden weights and bias parameters. The networks have $d = 5$ dimensional input, one hidden layer with 10 hidden nodes, ReLU activation function, and map to a 1-dimensional output. The dataset is generated by sampling $S_i = 100$ inputs x_j^i from a d-dimensional standard normal distribution, and the corresponding target values are given by $g_{\omega_i}(x_j^i)$, i.e.,

$$\mathcal{D}_i = \{(x_j^i, g_{\omega_i}(x_j^i)) \,|\, 1 \le i \le S_i, x_j^i \sim N(0, I_d)\}.$$

Our feature map ϕ in (3) is a randomized neural network[5] with one hidden layer, 500 hidden nodes, and ReLU activation. In particular, the weights of the hidden layers are randomly initialized and fixed, and only the weights of the last layer, corresponding to θ in (3), are optimized. We note that by our choice for the number of hidden nodes to equal the total number of samples, the algorithms can in principle learn to have perfect replication on the training set, i.e., loss $l(\Theta(t), w; \mathcal{D}) = 0$. For a fair comparison of the algorithms, we use the same randomly generated datasets and randomized neural networks for each method.

Remark 4.1. The comparison of our algorithms to gradient descent optimizing the loss is most meaningful in the $\lambda = 0$ regime since, otherwise, the objectives, and, therefore, the behavior are different. On the other hand, using $\beta > 0$ does not change the objective, but only forces our algorithm to make equally sized steps $\Delta\Theta(t)$, where the step size depends on β and T.

Figure 1 shows the averaged results for 10 runs using $\lambda = 0$ each with a different set of randomly generated datasets and a different randomized neural network as fKRR. For the regret-optimal algorithm and for its accelerated version, we use $T = 10^3$ iterations, while we use 10^5 steps for gradient descent with learning rate $7 \cdot 10^{-5}$. We note that for learning rates $\ge 8 \cdot 10^{-5}$ the training becomes unstable such that the loss explodes. The gradient descent method gradually, but slowly, reduces the loss. Since the gradient descent method makes small parameter updates (and since $\lambda = 0$), the regret is nearly 0, and the energy $\mathcal{L}$ nearly equal to the loss. Even though it runs 100 times the number of iterations of our algorithms, it still needs much more training to achieve the same loss.

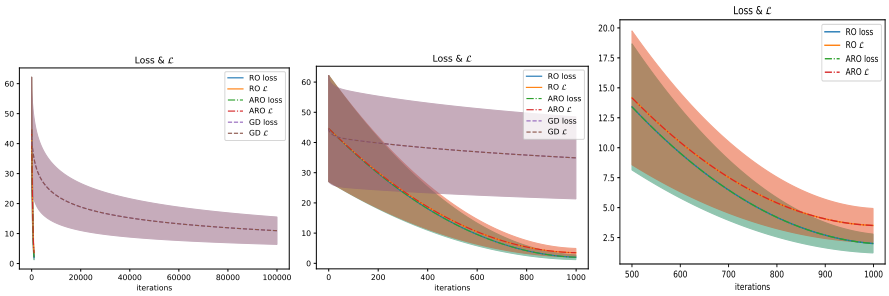


FIGURE 1. Loss and energy $\mathcal{L}$ for gradient descent (GD), the regret-optimal algorithm (RO), and the accelerated regret-optimal algorithm (ARO). Means and standard deviations over 10 runs are shown. Left: all iterations; middle: first 1000 iterations; right: iterations 500 to 1000 for our algorithms.

[5]Also called a random feature model or an extreme learning machine.

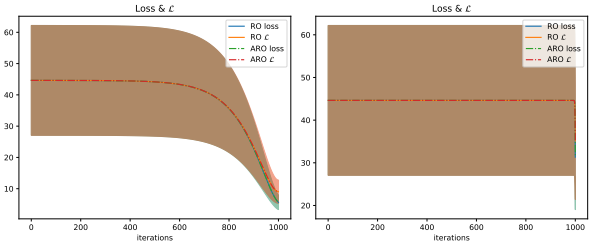


FIGURE 2. Loss and energy $\mathcal{L}$ for the regret-optimal algorithm (RO) and the accelerated regret-optimal algorithm (ARO) with means and standard deviations over 10 runs. Left: $\lambda = 10^{-4}$; right: $\lambda = 2$.

In the regime where $\lambda > 0$, the behavior of the algorithm changes drastically. In particular, since deviating from $\Theta(0) = \Theta^\star$ is penalized, the algorithm remains at the starting point in the initial steps and only starts to converge when approaching T. The exact behavior depends on the choice of λ (and is additionally influenced by the choice of β). In Figure 2, we compare the behavior for $\lambda = 10^{-4}$ and $\lambda = 2$, while otherwise using the same setup as before. For the smaller value of λ, the algorithm starts the descent earlier and decreases the loss by more compared to the larger value for λ, where the algorithm deviates from $\Theta^\star$ only in the very last steps.

5. **Conclusion.** In this work we presented a regret-optimal algorithm for federated transfer learning based on pre-trained models where only the last linear layer is fine-tuned. Besides the theoretical properties of this algorithm, we also provided experiments demonstrating the transfer learning capabilities of our method in the context of American option pricing. From the theoretical side, we would like to extend our method in future work to regression models that do not necessarily permit a closed-form solution.

6. **Limitations and future work.** Our results specialized on kernel ridge regressors. In future work, we would like to extend our pipeline to deep learning models.

Appendix A. **Game theoretic interpretation of regret-optimal algorithm.** We now offer a game-theoretic interpretation of the regret functional, as defined in (4). This interpretation views the user as a central planner whose objective is to organize a system of individual agents, representing the pretrained KRR models, to maximize the singular goal of identifying a parameter optimizing (2). We now interpret the roles and interactions of the central planner and each agent as defined by our regret-optimization problem.

The central planner: The purpose of the central planner is to avoid the situation whereby the structure of the different datasets is ignored in the model selection problem by merging them into a single dataset $\mathcal{D} \overset{\text{def.}}{=} \bigcup_{i=1}^{N} \mathcal{D}_i$, thereby reducing the problem to standard kernel ride regression. Though intuitively simple, such merging approach can be particularly disadvantageous in the presence of heterogeneity among the datasets since a dataset that is potentially very different from the focal dataset $\mathcal{D}_1$ would be equally influential in the regression problem. Instead, the user, acting

as a central planner, organizes the model selection problem through a cooperative objective

$$l(\Theta, w; \mathcal{D}) \stackrel{\text{def.}}{=} \sum_{i=1}^{N} w_i \sum_{j=1}^{|\mathcal{D}_i|} (f_{\theta^w}(x_j^i) - y_j^i)^2 \;\text{ with }\; \theta^w \stackrel{\text{def.}}{=} \sum_{i=1}^{N} w_i\, \theta_i, \tag{22}$$

where $\Theta \stackrel{\text{def.}}{=} (\theta_1^\top, \cdots, \theta_N^\top)^\top \in \mathbb{R}^{Np\times 1}$. Here, the central planner *ascribes* the influence of each agent on the model selection problem through the weight vector $w = (w_1, \cdots, w_N) \in [0,1]^N$ with $\sum_{i=1}^N w_i = 1$. As can be seen in (22), this influence is two-fold: On the one hand, it impacts the cooperative objective l in the aggregation of the SSE of each player; on the other hand, it affects the proportion of each player's preferred parameter choice entering into the collectively selected parameter θ^w. The choice of w is approached in different ways in the literature, e.g. by bagging reminiscent of mean-field games [14], by data-driven weighting procedures [8], Bayesian aggregation [56, 59, 69], or with dictionaries [5].

The agents: When acting in isolation, each agent's preferred model is determined by optimizing the kernel ridge regression objective using only their dataset. Under mild conditions[6] on f, this corresponds to an *individual* parameter selection

$$\theta_i^\star \in \operatorname{argmin}_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{D}_i} (f_\theta(x) - y)^2 + \kappa \|\theta\|_2, \tag{23}$$

for $i = 1, \ldots, N$, which we assume to be fixed and known to all agents prior to their *collective* parameter selection. Taking into account the other agents, each agent *acts* by specifying an iterative deterministic algorithm of the form with the intent of maximizing their influence on the joint model selection problem (22). Thus, the i^{th} player wants the jointly selected model, specified by $\theta^w(T) = \sum_{n=1}^N w_i\, \theta_i(T)$, to be as close to $\theta_i^\star$ as possible, thereby maximally encoding the characteristics of the i^{th} dataset into the selected model $f_{\theta^w(T)}$. The accumulated *regret* which the i^{th} player incurs by deploying an algorithm $\theta_i^{0\cdots T}$ that deviates from their preferred selection θ_i^* is measured by

$$\sum_{t=0}^{T-1} \underbrace{\lambda\, \|\theta_i(t+1) - \theta_i^\star\|_2^2}_{\text{Preference Strength}} + \underbrace{\beta\, \|\Delta\theta_i(t)\|_2^2}_{\text{Algorithm Stability}}, \tag{24}$$

where $\lambda > 0$ is a hyperparameter quantifying the player's *attachment* toward their preferred model choice $\theta_i^\star$, and the hyperparameter $\beta > 0$ quantifies the the stability of the algorithm during the iterations, where $\Delta\theta_i(t) \stackrel{\text{def.}}{=} \theta_i(t+1) - \theta_i(t)$.

The central planner then organizes the action $\Theta^{0\cdots T} \stackrel{\text{def.}}{=} \{\Theta(t)\}_{t=0}^T$ of the system of N players to reach the objective (22) while encoding their individual regrets (24), where

$$\Theta(t) = (\theta_1(t)^\top, \ldots, \theta_N(t)^\top)^\top \in \mathbb{R}^{Np\times 1}.$$

This is achieved by coupling the individual agent's regret functionals (24) through the *systemic regret functional*

$$\mathcal{R}(\Theta^{0\cdots T}) \stackrel{\text{def.}}{=} \underbrace{l(\Theta(T), w; \mathcal{D})}_{\text{Cooperative Objective}} - l^\star + \sum_{t=0}^{T-1} \Big(\underbrace{\lambda\, \|\Theta(t+1) - \Theta^\star\|_2^2}_{\text{Preference Strength}} + \underbrace{\beta\, \|\Delta\Theta(t)\|_2^2}_{\text{Algo. Stability}} \Big), \tag{25}$$

[6] E.g. coercivity and lower semi-continuity in θ.

where

$$\Theta^\star \overset{\text{def.}}{=} (\theta_1^{\star\top}, \cdots, \theta_N^{\star\top})^\top \in \mathbb{R}^{Np\times 1} \tag{26}$$

encodes the individual preferences of the players,

$$\Delta\Theta(t) \overset{\text{def.}}{=} \Theta(t+1) - \Theta(t) \tag{27}$$

quantifies the integrative updates of any candidate optimizing sequence $\Theta^{0\cdots T}$, and the *ideal terminal loss* is

$$l^\star \overset{\text{def.}}{=} \underbrace{\min_{\Xi\in\mathbb{R}^{Np\times 1}} l(\Xi, w; \mathcal{D})}_{\text{Cooperative Sub-optimality}} .$$

The term $(l(\cdot,\cdot;\cdot) - l^\star)$ measures the central planner's regret in failing to optimize (22). The algorithm stability term in (25) appears in forward-backward proximal splitting algorithms with quadratic objectives (e.g. [6, 19]) and several standard proximal algorithms (see [12, Section 5.1]).

The systemic regret functional (25) acts as a performance criterion for any optimization algorithm, played by N players, initialized at an arbitrary $\Theta(0)$, and running for T iterations. In what follows, we take the initial condition of the algorithm as $\Theta(0) = \Theta^\star$ for simplicity, but other choices are possible. In other words, the regret of each player is initialized at zero as they all start from their individually preferred parameter $\theta_i^\star$, and starts to accumulate as the system moves away from $\Theta^\star$ toward the optimizer of (25).

Appendix B. Proofs and technical results.

Proof of Theorem 3.1. For every $L(>0)$-Lipschitz $f \in \mathcal{F}$ and each $w \in \Delta_N$, we have that

$$\begin{aligned}
\left|\mathcal{R}(f) - \hat{\mathcal{R}}_w^{\mathcal{D}}(f)\right| &= \left|\mathbb{E}_{(X,Y)\sim\mu_1}\left[\ell(f(X),Y)\right] - \sum_{i=1}^N w_i \mathbb{E}_{(X,Y)\sim\hat{\mu}_i}\left[\ell(f(X),Y)\right]\right| && (28)\\
&= \left|\sum_{i=1}^N w_i\, \mathbb{E}_{(X,Y)\sim\mu_1}\left[\ell(f(X),Y)\right] - \sum_{i=1}^N w_i \mathbb{E}_{(X,Y)\sim\hat{\mu}_i}\left[\ell(f(X),Y)\right]\right| && (29)\\
&= \left|\sum_{i=1}^N w_i \left(\mathbb{E}_{(X,Y)\sim\mu_1}\left[\ell(f(X),Y)\right] - \mathbb{E}_{(X,Y)\sim\hat{\mu}_i}\left[\ell(f(X),Y)\right]\right)\right| \\
&\le \sum_{i=1}^N w_i \left|\mathbb{E}_{(X,Y)\sim\mu_1}\left[\ell(f(X),Y)\right] - \mathbb{E}_{(X,Y)\sim\hat{\mu}_i}\left[\ell(f(X),Y)\right]\right| \\
&\le \sum_{i=1}^N w_i\, \bar{L}\, \mathcal{W}_1(\mu_1, \hat{\mu}_i) && (30)\\
&\le \sum_{i=1}^N w_i\, \bar{L}\, \left(\mathcal{W}_1(\mu_1, \mu_i) + \mathcal{W}_1(\mu_i, \hat{\mu}_i)\right) \\
&= \sum_{i=1}^N w_i\, \bar{L}\, \left(s_i + \mathcal{W}_1(\mu_i, \hat{\mu}_i)\right) && (31)
\end{aligned}$$

$$= \bar{L} \sum_{i=1}^{N} w_i \, s_i + \bar{L} \sum_{i=1}^{N} w_i \, \mathcal{W}_1(\mu_i, \hat{\mu}_i), \tag{32}$$

where (29) held since w belongs to the N-simplex Δ_N, (30) held by Kantorovich duality (since $\mathcal{Z}$ is compact every probability thereon belongs to the 1-Wasserstein space over $(\mathcal{Z}, \|\cdot\|_2)$) and since the map $\mathcal{Z} \in (x, y) \mapsto \ell(f(x), y) \in \mathbb{R}$ is $\bar{L} \overset{\text{def.}}{=} \max\{1, L\}$-Lipschitz, see e.g. [64, Theorem 5.10 as simplified in Equation (5.11)], and (31) held by definition of the scores $\{s_i\}_{i=1}^N$ (see (6)). Since the upper bound in (32) held independently of the choice of L-Lipschitz $f \in \mathcal{F}$, then taking the supremum over all L-Lipschitz functions $f \in \mathcal{F}$ across (28)-(32) yields

$$\sup_{f \in \mathcal{F}_L} \left| \mathcal{R}(f) - \hat{\mathcal{R}}_w^{\mathcal{D}}(f) \right| \leq \bar{L} \sum_{i=1}^{N} w_i \, s_i + \bar{L} \sum_{i=1}^{N} w_i \, \mathcal{W}_1(\mu_i, \hat{\mu}_i) \tag{33}$$

where $\mathcal{F}_L \overset{\text{def.}}{=} \{f \in \mathcal{F} : \operatorname{Lip}(f) \leq L\}$ (note that we can include the case where $\operatorname{Lip}(f) = 0$ trivially).

Let $[N] \overset{\text{def.}}{=} \{1, \ldots, N\}$. Next, for each $t > 0$, we bound the following probability beginning with a simple union bound and then applying [34, Lemma 16] N times (once for each $\mathcal{W}_1(\mu_i, \hat{\mu}_i)$),

$$\begin{aligned}
&\mathbb{P}\big(\big|\mathcal{W}_1(\mu_i, \hat{\mu}_i) - \mathbb{E}\big[\mathcal{W}_1(\mu_i, \hat{\mu}_i)\big]\big| < t, \, \forall i \in [N]\big) && (34)\\
&\geq 1 - \sum_{i=1}^{N} \mathbb{P}\big(\big|\mathcal{W}_1(\mu_i, \hat{\mu}_i) - \mathbb{E}\big[\mathcal{W}_1(\mu_i, \hat{\mu}_i)\big]\big| \geq t\big) \\
&\geq 1 - \sum_{i=1}^{N} 2e^{-2\mathcal{D}_i \, t^2} \\
&\geq 1 - 2Ne^{-2N^\star \, t^2} && (35)
\end{aligned}$$

(we have used the fact that $\operatorname{diam}(\mathcal{X}) = 1$) where $N^\star = \min_{i=1,\ldots,N} |\mathcal{D}_i|$, as defined in the statement of the theorem. Let $\delta \in (0, 1]$ and solve the following for t:

$$\delta = 2Ne^{-2N^\star \, t^2}$$

yields

$$t = \left(\frac{\ln(2N/\delta)}{2N^\star} \right)^{1/2}. \tag{36}$$

Thus, (36) and our union bound in (34)-(35) imply that with probability at least $1 - \delta$, the following holds:

$$\max_{i \in [N]} \mathcal{W}_1(\mu_i, \hat{\mu}_i) - \mathbb{E}\big[\mathcal{W}_1(\mu_i, \hat{\mu}_i)\big] \leq \left(\frac{\ln(2N/\delta)}{2N^\star} \right)^{1/2}. \tag{37}$$

Applying the bound on the expected 1-Wasserstein distance between the empirical measure $\hat{\mu}_i$ and the true data-generating measure for the i^{th} dataset once for each $i \in [N]$ [34, Lemma 16 and Table 2] to (37) implies that there exists some constant $C \geq 1$ (depending only on $d + D$ and on $\mathcal{Z}$) such that the following holds with probability at least $1 - \delta$:

$$\max_{i \in [N]} \mathcal{W}_1(\mu_i, \hat{\mu}_i) - \frac{C}{|\mathcal{D}_i|^{1/(d+D)}} \leq \left(\frac{\ln(2N/\delta)}{2N^\star} \right)^{1/2}, \tag{38}$$

where we note that we have used the fact that $d+D \geq 3$ since either $d>1$ or $D>1$ (to avoid the critical regime where $d=D$ which gives a an additional $\log(N)$ factor to the second term in (38)). Merging (38) with the bounds in (33) implies that with probability at least $1-\delta$, we have

$$\begin{aligned}
&\sup_{f\in\mathcal{F}_L} \left|\mathcal{R}(f)-\hat{\mathcal{R}}_w^{\mathcal{D}}(f)\right| && (39)\\
\leq& \bar{L}\sum_{i=1}^N w_i\, s_i + \bar{L}\sum_{i=1}^N w_i \left(\frac{C}{|\mathcal{D}_i|^{1/(d+D)}} + \left(\frac{\ln(2N/\delta)}{2N^\star}\right)^{1/2}\right)\\
=& \bar{L}\sum_{i=1}^N w_i\, s_i + \bar{L}\sum_{i=1}^N w_i \frac{C}{|\mathcal{D}_i|^{1/(d+D)}} + \bar{L}\sum_{i=1}^N w_i \left(\frac{\ln(2N/\delta)}{2N^\star}\right)^{1/2}\\
=& \bar{L}\sum_{i=1}^N w_i\, s_i + C\,\bar{L}\sum_{i=1}^N \frac{w_i}{|\mathcal{D}_i|^{1/(d+D)}} + \bar{L}\left(\frac{\ln(2N/\delta)}{2N^\star}\right)^{1/2} && (40)\\
=& \bar{L}\sum_{i=1}^N w_i \left(s_i + \frac{C}{|\mathcal{D}_i|^{1/(d+D)}}\right) + \bar{L}\,\frac{\sqrt{\ln(2N/\delta)}}{\sqrt{2N^\star}}\\
\leq& \bar{L}\,C\sum_{i=1}^N w_i \left(s_i + \frac{1}{|\mathcal{D}_i|^{1/(d+D)}}\right) + \bar{L}\,\frac{\sqrt{\ln(2N/\delta)}}{\sqrt{2N^\star}} && (41)
\end{aligned}$$

where (40) again follows from the fact that $w\in\Delta_N$. In particular, if $|w-e_1|\in \mathcal{O}\big(\frac{1}{N^{\star\,1/(d+D)}}\big)$, then for every $\delta>0$, the right-hand side of (41) converges to 0 and is of the order of $\mathcal{O}\big(\frac{1}{N^{\star\,1/(d+D)}}\big)$.

Finally, for each $\eta\geq 0$, define the probability measure $\bar{\mathbb{P}}^\eta$ on $[N]$ by

$$\bar{\mathbb{P}}^\eta \overset{\text{def.}}{=} \sum_{i=1}^N \frac{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}{\sum_{u=1}^N I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\leq\eta}}\,\delta_i \overset{\text{def.}}{=} \sum_{i=1}^N C_\eta^{-1} I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}\,\delta_i,$$

where $C_\eta \overset{\text{def.}}{=} \sum_{u=1}^N I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_u)\leq\eta}$. Then, for each $w\in\Delta_N$, consider the probability measure $\mathbb{P}_w \overset{\text{def.}}{=} \sum_{i=1}^N w_i\delta_i$ on $[N]$. The KL-divergence between $\mathbb{P}_w$ and $\bar{\mathbb{P}}^\eta$ is non-negative and given by

$$\mathrm{KL}(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta) = \sum_{i=1}^N w_i \log\left(C_\eta\,\frac{w_i}{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}\right).$$

Let $w\in\Delta_N$ be such that $\mathbb{P}_w \ll \bar{\mathbb{P}}^\eta$ so that $\mathrm{KL}(\mathbb{P}_w\|\bar{\mathbb{P}}^\eta)$ is finite. Then, for every $\eta\geq 0$ and $\gamma>0$, the right-hand side of (41) can be further bounded above by the following *finite* value: For every $0<\delta\leq 1$, the following holds with probability at least $1-\delta$

$$\begin{aligned}
\sup_{f\in\mathcal{F}_L} \left|\mathcal{R}(f)-\hat{\mathcal{R}}_w^{\mathcal{D}}(f)\right| \leq& \bar{L}C\sum_{i=1}^N w_i\left(s_i+\frac{1}{|\mathcal{D}_i|^{1/(d+D)}}\right) && (42)\\
&+\frac{\bar{L}C}{\gamma}\sum_{i=1}^N w_i\log\left(C_\eta\,\frac{w_i}{I_{\mathcal{W}_1(\hat{\mu}_1,\hat{\mu}_i)\leq\eta}}\right)\\
&+\bar{L}\,\frac{\sqrt{\ln(2N/\delta)}}{\sqrt{2N^\star}}.
\end{aligned}$$

If $\mathbb{P}_w \ll \bar{\mathbb{P}}^\eta$, and, further, w and η are chosen such that $|w - e_1|, \eta$ $\mathrm{KL}\left(\mathbb{P}_w \| \bar{\mathbb{P}}^\eta\right) \in \mathcal{O}\left(\frac{1}{N^{\star\,1/(d+D)}}\right)$, then the right-hand side of (42) is of the order of $\mathcal{O}(1/N^{1/(d+D)})$. □

Assumption B.1 ($(w^\star, \mathcal{D})$-Compatibility)**.** *The data $\mathcal{D} = \cup_{i=1}^N \mathcal{D}_i$ and the weights $w^\star$ satisfy*

$$\sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \geq 0, \tag{43}$$

$$[w_1^\star I_p, \cdots, w_N^\star I_p]^\top \left(\sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \right) [w_1^\star I_p, \cdots, w_N^\star I_p] \geq 0. \tag{44}$$

Remark B.1. We keep our notation light and use $u^\top v$ to implement the Euclidean inner-product between any two vectors $u, v \in \mathbb{R}^k$ for any $k \in \mathbb{N}_+$. We use $|\cdot|$ to denote the Euclidean norm of a vector or Fröbenius norm of a matrix, whichever is applicable. For two square matrices A and B, $A \leq B$ ($A < B$, resp.) means that $B - A$ is positive semi-definite (positive definite, resp.); see page 146 - Equation (7) [46] for details.

Lemmas B.1 and B.2 give some elementary properties of matrix algebra, which will be used repeatedly in the subsequent proofs.

Lemma B.1. *The Euclidean norm $|\cdot|$ of matrices has the following properties:*
i). *For any matrices $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{n_2 \times n_3}$, we have that $|AB| \leq |A| \cdot |B|$;*
ii). *For any matrices A, $B \in \mathbb{R}^{n_1 \times n_2}$, we have $|A + B| \leq |A| + |B|$.*

Proof. Denote $A = (A_{ij})_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$ and $B = (B_{jk})_{1 \leq j \leq n_2, 1 \leq k \leq n_3}$. By the Cauchy-Schwarz inequality, we have for each column vector $B_{\cdot k}$ of B that

$$\begin{aligned}
|AB_{\cdot k}|^2 &= \sum_{i=1}^{n_1} \big(\sum_{j=1}^{n_2} A_{ij} B_{jk} \big)^2 \\
&\leq \sum_{i=1}^{n_1} \big(\sum_{j=1}^{n_2} A_{ij}^2 \sum_{j=1}^{n_2} B_{jk}^2 \big) \\
&= \big(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A_{ij}^2 \big) \sum_{j=1}^{n_2} B_{jk}^2 \\
&= |A|^2 \sum_{j=1}^{n_2} B_{jk}^2.
\end{aligned}$$

It then follows that

$$\begin{aligned}
|AB|^2 &= |(AB_{\cdot 1}, AB_{\cdot 2}, ..., AB_{\cdot n_3})|^2 \\
&= |AB_{\cdot 1}|^2 + |AB_{\cdot 2}|^2 + \cdots + |AB_{\cdot n_3}|^2 \\
&\leq |A|^2 \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} B_{jk}^2 \\
&= |A|^2 |B|^2,
\end{aligned}$$

and assertion i) is proved.

For A, $B \in \mathbb{R}^{n_1 \times n_1}$, we have

$$\begin{aligned}|A+B|^2 =& \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}(A_{ij}+B_{ij})^2 \\ =& \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}A_{ij}^2 + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}B_{ij}^2 + 2\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}A_{ij}B_{ij} \\ \leq& |A|^2 + |B|^2 + 2|A|^2|B|^2 \\ =& (|A|+|B|)^2,\end{aligned}$$

which proves assertion ii). □

Lemma B.2. *Let w_1, w_2, ..., w_n be positive numbers such that $\sum_{k=1}^n w_k = 1$. For any n matrices A_1, ..., $A_n \in \mathbb{R}^{n_1 \times n_2}$, it satisfies that $\left|\sum_{k=1}^n w_k A_k\right|^2 \leq \sum_{k=1}^n w_k |A_k|^2$; particularly, when $w_1 = w_2 = \cdots = w_n = 1/n$, we have $\left|\sum_{k=1}^n A_k\right|^2 \leq n\sum_{k=1}^n |A_k|^2$.*

Proof. Denote $A_k = (A_{k,ij})_{1\leq i\leq n_1, 1\leq j\leq n_2}$, $k = 1$, ..., n. By Jensen's inequality, we have

$$(w_1 A_{1,ij} + \cdots + w_n A_{n,ij})^2 \leq w_1 A_{1,ij}^2 + \cdots + w_n A_{n,ij}^2.$$

Then, it follows that

$$\begin{aligned}|w_1A_1 + \cdots + w_nA_n|^2 =& \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}(w_1A_{1,ij} + \cdots + w_nA_{n,ij})^2 \\ \leq& \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} w_1A_{1,ij}^2 + \cdots + w_nA_{n,ij}^2 \\ =& w_1\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}A_{1,ij}^2 + \cdots + w_n\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}A_{n,ij}^2 \\ =& w_1|A_1|^2 + \cdots + w_n|A_n|^2.\end{aligned}$$

□

Next, we prove our main result, namely Theorem 3.2. The result was first rewritten in the optimal control-theoretic notation introduced in Section 2.

Theorem B.1 (The Regret-Optimal Algorithm – Control Theoretic Form)**.** *Suppose that Assumption 3.1 holds. Then, the optimal control problem (4) and (5) admits a unique solution for $t = 0, 1, ..., T-1$ such that*

$$\boldsymbol{\alpha}(t) = -\left[(\lambda+\beta)I_{Np} + P(t+1)\right]^{-1}\left[(\lambda I_{Np} + P(t+1))\Theta(t) - \lambda\Theta^\star + S(t+1)\right], \tag{45}$$

and the resulting cost is

$$L^\star(\boldsymbol{\alpha}; \mathcal{D}) = \Theta^{\star\top}P(0)\Theta^\star + 2S^\top(0)\Theta^\star + r(0). \tag{46}$$

The $P(t)$, $S(t)$, and $r(t)$ in (45) and (46) are determined by the backward equation system on $t = 1, \ldots, T$:

$$\begin{cases} P(t) = \beta I_{Np} - \beta^2[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}, \\ P(T) = [w_1^\star I_p, \cdots, w_N^\star I_p]^\top\left(\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}\right)[w_1^\star I_p, \cdots, w_N^\star I_p], \end{cases} \tag{47}$$

$$\begin{cases} S(t) = \beta\big[(\lambda+\beta)I_{Np} + P(t+1)\big]^{-1}(S(t+1) - \lambda\Theta^\star), \\ S(T) = -[w_1^\star I_p, \cdots, w_N^\star I_p]^\top \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i, \end{cases} \tag{48}$$

$$\begin{cases} r(t) = -(S(t+1) - \lambda\Theta^\star)^\top [(\lambda+\beta)I_{Np} + P(t+1)]^{-1} \\ \qquad\quad \times (S(t+1) - \lambda\Theta^\star) + \lambda\|\Theta^\star\|_2^2 + r(t+1), \\ r(T) = \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i|^2. \end{cases} \tag{49}$$

Proof of Theorem B.1 (and therefore of Theorem 3.2). Let $V(t,\Theta)$ be the value function associated with the optimal control problem (4) and (5). By the dynamic programming principle, $V(t,\Theta)$ satisfies the equation

$$V(t,\Theta) = \min_{\boldsymbol{\alpha}} \big[\lambda\,|\Theta + \boldsymbol{\alpha} - \Theta^\star|^2 + \beta\,|\boldsymbol{\alpha}|^2 + V(t+1, \Theta+\boldsymbol{\alpha})\big], \quad t = 0, 1, ..., T-1, \tag{50}$$

$$V(T,\Theta) = l(\theta^{w^\star}; \mathcal{D}). \tag{51}$$

The loss function can be written as a linear quadratic function of $\Theta = [\theta_1^\top, \cdots, \theta_N^\top]^\top$ such that

$$\begin{aligned} l(\theta^{w^\star}; \mathcal{D}) =& \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} \Big[u_j^{i\top} \sum_{k=1}^N w_k^\star \theta_k - y_j^i\Big]^2 \\ =& \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} \Big[\sum_{k=1}^N w_k^{\star 2} \theta_k^\top u_j^i u_j^{i\top} \theta_k - 2\sum_{k=1}^N w_k^\star y_j^i u_j^{i\top} \theta_k + |y_j^i|^2\Big] \\ =& \Big(\sum_{k=1}^N w_k^\star \theta_k^\top\Big)\Big(\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}\Big)\Big(\sum_{k=1}^N w_k^\star \theta_k\Big) \\ & - 2\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} y_j^i u_j^{i\top} \sum_{k=1}^N w_k^\star \theta_k + \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i|^2 \\ =& \Theta^\top [w_1^\star I_p, \cdots, w_N^\star I_p]^\top \Big(\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}\Big)[w_1^\star I_p, \cdots, w_N^\star I_p]\Theta \\ & + \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i|^2 - 2\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} y_j^i u_j^{i\top} [w_1^\star I_p, \cdots, w_N^\star I_p]\Theta. \end{aligned} \tag{52}$$

We claim that $V(t,\Theta)$ takes the linear quadratic form

$$V(t,\Theta) = \Theta^\top P(t)\Theta + 2S^\top(t)\Theta + r(t). \tag{53}$$

To see this, note from (51) and (52) that $V(T,\Theta)$ takes the linear quadratic form of (53). By solving the dynamic programming equation (50) at $t = T-1$ for the optimal $\boldsymbol{\alpha}$ and substituting it back into the equation, we obtain $V(T-1,\Theta)$ also taking the linear quadratic form of (53). By backward induction we have that $V(t,\Theta)$ takes the form of (53) for all $t = 0, ..., T$.

Remark B.2. Here, $r(t)$ is a constant term independent of the "state variable" Θ, which we note exists because the cost $L^\star$ defined by (4) contains terms independent of $\Theta(t)$.

Combining (51), (52), and (53) verifies that $P(T)$ and $S(T)$ satisfy the terminal condition as given by (47) and (48). We substitute (53) into (50) and reorganize the

terms to get

$$\begin{aligned}
&\Theta^\top P(t)\Theta + 2S^\top(t)\Theta + r(t)\\
&= \min_{\boldsymbol{\alpha}} \Big\{ \boldsymbol{\alpha}^\top[(\lambda+\beta)I_{Np} + P(t+1)]\boldsymbol{\alpha} + 2[\Theta^\top P(t+1) + S^\top(t+1)\\
&\quad + \lambda(\Theta^\top - \Theta^{\star\top})]\boldsymbol{\alpha}\\
&\quad + \Theta^\top(\lambda I_{Np} + P(t+1))\Theta + 2(S^\top(t+1) - \lambda\Theta^\star)\Theta + \lambda|\Theta^\star|^2 + r(t+1)\Big\}. \quad (54)
\end{aligned}$$

By the first-order condition[7] for the right-hand side of (54) with respect to $\boldsymbol{\alpha}$, we have

$$0 = [(\lambda+\beta)I_{Np} + P(t+1)]\boldsymbol{\alpha} + [(\lambda I_{Np} + P(t+1))\Theta + S(t+1) - \lambda\Theta^\star]. \quad (55)$$

By Assumption B.1, we have $P(T) + (\lambda+1)I_{Np} > 0$, and therefore, from (55) we obtain the optimal $\boldsymbol{\alpha}$ taking the form of (45).

We further substitute $\boldsymbol{\alpha}$ of the form (45) into the right side of (54) to obtain

$$\begin{aligned}
&\Theta^\top P(t)\Theta + 2S^\top(t)\Theta + r(t)\\
&= -\left[\Theta^\top(\lambda I_{Np} + P(t+1)) + S^\top(t+1) - \lambda\Theta^{\star\top}\right]\cdot\left[(\lambda+\beta)I_{Np} + P(t+1)\right]^{-1}\\
&\quad \times \left[(\lambda I_{Np} + P(t+1))\Theta + S(t+1) - \lambda\Theta^\star\right] + \Theta^\top(\lambda I_{Np} + P(t+1))\Theta\\
&\quad + 2[S^\top(t+1) - \lambda\Theta^{\star\top}]\Theta + \lambda|\Theta^\star|^2 + r(t+1). \quad (56)
\end{aligned}$$

Reorganizing the terms on right side of (56), we have

$$\begin{aligned}
&\Theta^\top P(t)\Theta + 2S^\top(t)\Theta + r(t)\\
&= -\Theta^\top(\lambda I_{Np} + P(t+1))[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\Theta\\
&\quad + \Theta^\top(\lambda I_{Np} + P(t+1))\Theta + 2[S^\top(t+1) - \lambda\Theta^{\star\top}]\Theta\\
&\quad - 2[S^\top(t+1) - \lambda\Theta^{\star\top}][(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\Theta\\
&\quad - (S^\top(t+1) - \lambda\Theta^{\star\top})[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star)\\
&\quad + \lambda|\Theta^\star|^2 + r(t+1). \quad (57)
\end{aligned}$$

Matching the coefficients of the quadratic terms of Θ on both sides of (57), we obtain

$$\begin{aligned}
P(t) = &-(\lambda I_{Np} + P(t+1))[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\\
&+ \lambda I_{Np} + P(t+1). \quad (58)
\end{aligned}$$

The right side of (58) further simplifies to

$$\begin{aligned}
&-(\lambda I_{Np} + P(t+1))[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1)) + \lambda I_{Np} + P(t+1)\\
&= -[(\lambda+\beta)I_{Np} + P(t+1) - \beta I_{Np}][(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\\
&\quad + \lambda I_{Np} + P(t+1)\\
&= -(\lambda I_{Np} + P(t+1)) + \beta[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\\
&\quad + \lambda I_{Np} + P(t+1)\\
&= \beta[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(\lambda I_{Np} + P(t+1))\\
&= \beta[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}[(\lambda+\beta)I_{Np} + P(t+1) - \beta I_{Np}]
\end{aligned}$$

[7] Note that we can argue this way since the problem is convexified upon fixing $w^\star$. In fact, this is the control-theoretic motivation for using Algorithm 1 to decouple the optimization of w and of θ from one another, and not treating them as a single (non-convex if left coupled) control problem.

$$=\beta I_{Np} - \beta^2[(\lambda+1)I_{Np} + P(t+1)]^{-1}. \tag{59}$$

Combining (58) and (59) gives (47) for $t = T-1$.

Matching the coefficients of the linear terms of Θ on both sides of (57) and taking the transpose, we obtain

$$\begin{aligned} S(t) =& -(\lambda I_{Np} + P(t+1))[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star) \\ &+ S(t+1) - \lambda\Theta^\star. \end{aligned} \tag{60}$$

The right side of (60) further simplifies to

$$\begin{aligned} &-(\lambda I_{Np} + P(t+1))[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star) + S(t+1) - \lambda\Theta^\star \\ &= -[(\lambda+\beta)I_{Np} + P(t+1) - \beta I_{Np}][(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star) \\ &\quad + S(t+1) - \lambda\Theta^\star \\ &= -(S(t+1) - \lambda\Theta^\star) + \beta[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star) \\ &\quad + S(t+1) - \lambda\Theta^\star \\ &= \beta[(\lambda+\beta)I_{Np} + P(t+1)]^{-1}(S(t+1) - \lambda\Theta^\star). \end{aligned} \tag{61}$$

Combining (60) and (61) gives (48) for $t = T-1$.

Matching the terms independent of Θ on both sides of (57) gives

$$\begin{aligned} r(t) =& -(S(t+1) - \lambda\Theta^\star)^\top[(\lambda+\beta)I_{Np} + P(t+1)]^{-1} \cdot (S(t+1) - \lambda\Theta^\star) \\ &+ \lambda|\Theta^\star|^2 + r(t+1), \end{aligned}$$

which is (49) for $t = T-1$.

Since $P(T) + (\lambda+\beta)I_{Np} > \beta I_{Np}$, we have

$$P(T-1) = \beta I_{Np} - \beta^2[(\lambda+\beta)I_{Np} + P(T)]^{-1} > 0.$$

By backward induction on t, we can show that $\boldsymbol{\alpha}$ takes the form of (45) for all $t = 0, 1, ..., T-1$, and P and S satisfy the system (47) and (48). □

Corollary B.1. *Under Assumption 3.1, the solution P and S of the system (47)-(48) satisfies that for all $t = 1, 2, ..., T$,*

$$\left|[(\lambda+\beta)I_{Np} + P(t)]^{-1}\right| \le \sqrt{Np}(\lambda+\beta)^{-1}, \tag{62}$$

$$|S(t)| \le C \cdot N^{2+(T-t)/2}, \tag{63}$$

where $C > 0$ is a constant depending on λ, β, $(w_1^\star, ..., w_N^\star)$, $\Theta^\star$, K_x, K_y, and T.

Proof of Corollary B.1. Let $\|\cdot\|$ be the spectral norm of a matrix such that $\|A\| \overset{\text{def.}}{=} \sup_{x\neq 0} |Ax|/|x|$. Since $P(t)$ is positive semi-definite, we have $(\lambda+\beta)I_{Np} + P(t) \ge (\lambda+\beta)I_{Np}$ and

$$0 \le [(\lambda+\beta)I_{Np} + P(t)]^{-1} \le (\lambda+\beta)^{-1}I_{Np},$$

which, by the Courant-Fisher theorem (see [33]), implies that $\|[(\lambda+\beta)I_{Np} + P(t)]^{-1}\| \le \|(\lambda+\beta)^{-1}I_{Np}\| = (\lambda+\beta)^{-1}$. Since all norms of finite-dimensional normed spaces are equivalent, and, in particular, $|\cdot| \le \sqrt{Np}\|\cdot\|$ on the space of $Np \times Np$-matrices, then we find that

$$\left|[(\lambda+\beta)I_{Np} + P(t)]^{-1}\right| \le \sqrt{Np}\left\|[(\lambda+\beta)I_{Np} + P(t)]^{-1}\right\|,$$

and thus $\left|[(\lambda+\beta)I_{Np} + P(t)]^{-1}\right| \le \sqrt{Np}(\lambda+\beta)^{-1}$, which proves (62).

Next, we prove (63). By Lemmas B.1 and B.2, and by the assumption that

$$\frac{1}{N}\sum_{i=1}^{N}|\mathcal{D}_i| \leq N, \tag{64}$$

we obtain

$$\begin{aligned}
|S(T)|^2 =& \Big(\sum_{i=1}^{N} w_i^{\star 2}\Big)\Big|\sum_{i=1}^{N} w_i^{\star}\sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i\Big|^2 \\
\leq& \Big(\sum_{i=1}^{N} w_i^{\star 2}\Big)\sum_{i=1}^{N} w_i^{\star}\Big|\sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i\Big|^2 \\
\leq& \Big(\sum_{i=1}^{N} w_i^{\star 2}\Big)\sum_{i=1}^{N} w_i^{\star}|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i y_j^i|^2 \\
\leq& \Big(\sum_{i=1}^{N} w_i^{\star 2}\Big)\sum_{i=1}^{N} w_i^{\star}|\mathcal{D}_i|^2 K_x^2 K_y^2 \\
\leq& \Big(\sum_{i=1}^{N} w_i^{\star 2}\Big)\sum_{i=1}^{N} w_i^{\star} N^3 K_x^2 K_y^2,
\end{aligned} \tag{65}$$

and therefore the estimate (63) holds for $t = T$.

Assume by induction that (63) holds for $t = u$. By (48), Lemma B.1, and the triangle inequality, we have

$$\begin{aligned}
|S(u-1)| \leq& \beta\big|\big[(\lambda+\beta)I_{Np} + P(u)\big]^{-1}\big| \cdot \big|S(u) - \lambda\Theta^{\star}\big| \\
\leq& \beta\big|\big[(\lambda+\beta)I_{Np} + P(u)\big]^{-1}\big| \cdot (|S(u)| + \lambda|\Theta^{\star}|).
\end{aligned}$$

The estimate (63) for $t = u-1$ then follows from (62) and the induction hypothesis. Thus, by induction, we have shown that (62) holds for all $t = 1, 2, ..., T$. □

We first prove Theorem 3.3 for the complexity of computing the optimal information sharing weights.

Proof of Theorem 3.3. **Complexity of Computing $\{\mathcal{W}_1(\hat{\mu}_1, \hat{\mu}_i)\}_{i=2}^N$ is $\mathcal{O}(N \times (|\mathcal{D}_1|D_{-1}^{\star})^{3.5})$.**

According to [55, Chapter 6], the complexity of computing $\mathcal{W}_1(\hat{\mu}_1, \hat{\mu}_i)$ as a linear programming problem is $\mathcal{O}\big((|\mathcal{D}_1||\mathcal{D}_i|)^{3.5}L\big)$ for each $i = 1, \ldots, N$. Therefore, the complexity of computing the scores $\{s^i\}_{i=1}^N$ and furthermore $\{T^i\}_{i=1}^N$ is $\mathcal{O}\big(N(|\mathcal{D}_1|D_{-1}^{\star})^{3.5}L\big)$, where $D_{-1}^{\star} = \max_{i=2,\ldots,N}|\mathcal{D}_i|$.

Complexity of computing $\mathrm{Softmin}(s)$ **is** $\mathcal{O}(Nd_{prec})$**.**

The complexity of computing the exponential function $\exp(-s^i)$ up to d_{prec} digits of accuracy is $\mathcal{O}(\log_2 d_{prec})$ by [1]. Then, the complexity of computing $\{\exp(-s^i)\}_{i=1}^N$ is $\mathcal{O}(N\log_2 d_{prec})$. By Lemma B.3, the complexity of summing up $\sum_{i=1}^N \exp(-s^i)$ is $\mathcal{O}(N)$. Therefore, the complexity of computing $\mathrm{Softmin}(s)$ is $\mathcal{O}(N\log_2 d_{prec})$.

Complexity of computing $\mathrm{Softmin}(s)^{\top}T$ **is** $\mathcal{O}(N)$**.**

The Hadamard product of two N-dimensional vectors $\mathrm{Softmin}(s)^{\top}$ and T has a complexity of $\mathcal{O}(N)$ by Lemma B.3.

Tallying up complexities: The complexity of computing the optimal weights $w^\star$ is

$$\mathcal{O}\big(N(|\mathcal{D}_1|D^\star_{-1})^{3.5}L + N\log_2 d_{prec}\big).$$

□

We proceed to prove Theorem 3.4 about the complexity of the regret-optimal algorithm. To facilitate the proof, we introduce the following Lemma B.3 that summarizes complexities of matrix operations.

Lemma B.3. *Suppose all elementary arithmetic operations (e.g. addition, subtraction, multiplication) of two real numbers are of constant (i.e. $\mathcal{O}(1)$) complexity. Then, elementary matrix computations have the following complexities:*

(i) Adding two $n_1 \times n_2$ matrices has a complexity of $\mathcal{O}(n_1 n_2)$.
(ii) Multiplying an $n_1 \times n_2$ matrices by a scalar has a complexity of $\mathcal{O}(n_1 n_2)$.
(iii) Multiplying an $n_1 \times n_2$ matrix with an $n_2 \times n_3$ matrix has a complexity of $\mathcal{O}(n_1 n_2 n_3)$. In particular, multiplying two $n \times n$ matrices has a complexity of $\mathcal{O}(n^{2.373})$.

Proof. Adding two $n_1 \times n_2$ matrices involves adding each entry of one matrix with the corresponding entry of the other matrix, and each entry addition has a complexity of $\mathcal{O}(1)$. Hence the total complexity is $\mathcal{O}(n_1 n_2)$.

Multiplying an $n_1 \times n_2$ matrix by a scalar involves multiplying each entry by the scalar, and the scalar multiplication of each entry has a complexity of $\mathcal{O}(1)$. Hence, the total complexity is $O(n_1 n_2)$.

Multiplication of an $n_1 \times n_2$ matrix and an $n_2 \times n_3$ matrix requires multiplying the n_1 row vectors of the first matrix by the n_3 column vectors of the second matrix, which needs a total of $n_1 n_3$ multiplications. Multiplication of two n_2-dimensional vectors has a complexity of $O(n_2)$. Therefore the total complexity is $O(n_1 n_2 n_3)$. When multiplying two matrices of $n \times n$-dimension with an optimized CW-like algorithm the complexity is $\mathcal{O}(n^{2.373})$. □

Proof of Theorem 3.4. **Complexity of computing $P(T)$ is $\mathcal{O}\big(N^2p^3\big)$.**

Since $u^i_j \in \mathbb{R}^{p\times 1}$, computing each $u^i_j u^{i\top}_j$ has $\mathcal{O}(p^2)$ complexity by Lemma B.3-(iii). Multiplying the $p \times p$ matrix $u^i_j u^{i\top}_j$ by the scalar $w^\star_i$ has a complexity of $\mathcal{O}(p^2)$ by Lemma B.3-(iii). Then, computing the $p \times p$ matrix $\sum_{i=1}^N w^\star_i \sum_{j=1}^{S_i} u^i_j u^{i\top}_j$ by adding the $\bar{N}$ matrices of dimension $p \times p$ has complexity $\mathcal{O}(\bar{N}\,p^2)$ by Lemma B.3-(i). Since $[w^\star_1 I_p, \cdots, w^\star_N I_p]^\top$ is $Np \times p$-dimensional, then computing the $np \times p$-dimensional product $[w^\star_1 I_p, \cdots, w^\star_N I_p]^\top \sum_{i=1}^N w^\star_i \sum_{j=1}^{S_i} u^i_j u^{i\top}_j$ has $\mathcal{O}(Np^3)$ complexity by Lemma B.3-(iii). Likewise, the product

$$\Big([w^\star_1 I_p, \cdots, w^\star_N I_p]^\top \sum_{i=1}^N w^\star_i \sum_{j=1}^{S_i} u^i_j u^{i\top}_j\Big)[w^\star_1 I_p, \cdots, w^\star_N I_p]$$

has a complexity of $\mathcal{O}(N^2p^3)$. Therefore, computing $P(T)$ itself is of $\mathcal{O}\big(\bar{N}p^2 + N^2p^3\big)$ complexity. Assumption (64) implies that $\bar{N} \le N^2$; hence, the complexity of computing $P(T)$ is $\mathcal{O}\big(N^2p^3\big)$.

Complexity of computing $S(T)$ is $\mathcal{O}\big(Np\max\{N,p\}\big)$.

Since $u^i_j \in \mathbb{R}^{p\times 1}$ and $y^i_j \in \mathbb{R}$, computing the product $u^i_j y^i_j$ has complexity $\mathcal{O}(p)$ by Lemma B.3-(iii). And, multiplying the $p \times 1$ dimensional vector $u^i_j y^i_j$ by the scalar

$w_i^\star$ has complexity $\mathcal{O}(p)$ by Lemma B.3-(ii). Then, the complexity of computing $\sum_{i=1}^N w_i^\star \sum_{j=1}^{S_i} u_j^i y_j^i$ by adding $\bar{N}$ vectors of dimension p is $\mathcal{O}\big(\bar{N}p\big)$ by Lemma B.3-(i). Since $-[w_1^\star I_p, \cdots, w_N^\star I_p]^\top$ is an $Np\times p$-dimensional matrix and $\sum_{i=1}^N w_i^\star \sum_{j=1}^{S_i} u_j^i y_j^i$ is a $p\times 1$-dimensional vector, then computing their $Np\times 1$ dimensional product has $\mathcal{O}(Np^2)$ complexity according to Lemma B.3-(iii). Therefore, computing $S(T)$ has a complexity of $\mathcal{O}(\bar{N}p + NP^2)$. Since (64) implies that $\bar{N}\leq N^2$, then computing $S(T)$ has a computational cost of $\mathcal{O}\big(Np\max\{N,p\}\big)$.

Complexity of computing $P(t)$ and $S(t)$ for $t=T-1,\dots,1$ is $\mathcal{O}\big(T(Np)^{2.373}\big)$.

The complexity of computing $(\lambda+\beta)I_{Np}+P(t+1)$ for each t is $\mathcal{O}(Np)$, since we only need to add scalars on the matrix's *diagonal*. By [24], computing the matrix inverse $[(\lambda+\beta)I_{Np}+P(t+1)]^{-1}$ with a CW-like algorithm has a complexity of $\mathcal{O}\big((Np)^{2.373}\big)$. Since the addition $\beta I_{Np}-\beta^2[(\lambda+\beta)I_{Np}+P(t+1)]^{-1}$ only requires us to multiply all entries of $[(\lambda+\beta)I_{Np}+P(t+1)]^{-1}$ by $-\beta^2$ and add along the diagonal, it has a complexity of $\mathcal{O}\big((Np)^2+Np\big)$ by Lemma B.3-(i) and (ii). Performing the matrix addition $S(t+1)-\lambda\Theta^\star$ has complexity $\mathcal{O}(Np)$ by Lemma B.3-(i). Computing the $Np\times 1$-dimensional product $[(\lambda+1)I_{Np}+P(t+1)]^{-1}(S(t+1)-\lambda\Theta^\star)$ by multiplying an $Np\times Np$ matrix by an $Np\times 1$ matrix has complexity $\mathcal{O}(N^2p^2)$ according to Lemma B.3-(iii). Therefore, computing each $P(t)$ and $S(t)$ for $t=T-1,\dots,1$ has complexity $\mathcal{O}\big((Np)^{2.373}\big)$. Since there are $T-1$ such matrices to compute, then the total cost of computing every $P(T-1),\dots,P(1)$ and $S(T-1)$, ..., $S(1)$ is $\mathcal{O}\big(T(Np)^{2.373}\big)$.

The cost of computing each $\Theta(t)$ is $\mathcal{O}(T\,N^2p^2)$.

To compute $\Theta(t)$ for all $t=1,\dots,T$, starting from $\Theta(0)=\Theta^\star$, we need to compute $\mathbf{a}(t)$ given by (45) for $t=0,1,\dots,T-1$. Hence, we need to quantify the complexity of computing $\mathbf{a}(t)$. By (47), we can obtain $-[(\lambda+\beta)I_{Np}+P(t+1)]^{-1}$ by computing $\beta^{-2}(P(t)-\beta I_{Np})$, which involves adding a scalar to the diagonal of an $Np\times Np$ matrix and then multiplying the matrix by a scalar. The cost of computing the product $(\lambda I_{Np}+P(t+1))\Theta(t)$ is $\mathcal{O}(N^2p^2)$ and the cost of computing all sums in $(\lambda I_{Np}+P(t+1))\Theta(t)-\lambda\Theta^\star+S(t+1)$ are of lower-order complexity. Finally, the cost of computing the matrix-product between $-[(\lambda+\beta)I_{Np}+P(t+1)]^{-1}$ and $\big[(\lambda I_{Np}+P(t+1))\Theta(t)-\lambda\Theta^\star+S(t+1)\big]$ is $\mathcal{O}(N^2p^2)$ by Lemma B.3-(iii). Therefore, the cost of computing each $\mathbf{a}(t)$ is $\mathcal{O}(N^2p^2)$, and therefore the cost of computing $\Theta(t)$ for all $t=1$, ..., $t=T$ from $\Theta(0)=\Theta^\star$ is $\mathcal{O}(TN^2p^2)$.

Tallying up complexities: The complexity of computing $(\Theta(t))_{t=0}^T$ is $\mathcal{O}(N^2p^3+T\,(Np)^{2.373})$.

Looking over all computations involved, we deduce that the cost of computing the sequence $(\Theta(t))_{t=1}^T$ is (in order) dominated by the computational complexity of computing the sequence $(P(t))_{t=1}^{T-1}$ and $P(T)$, whence it is $\mathcal{O}(N^2p^3+T\,(Np)^{2.373})$. □

The following Lemma B.4, Proposition B.1, and Corollary B.2 are devoted to proving Theorem 3.5, the adversarial robustness of Algorithm 2.

We introduce the linear map $M(\cdot):\mathbb{R}^{Np\times Np}\to\mathbb{R}^{Np\times Np}$ such that $M(Q)\overset{\text{def.}}{=}(\lambda+\beta)I_{Np}+Q$ for any $Q\in\mathbb{R}^{Np\times Np}$.

Lemma B.4. *Given any two data sets $\mathcal{D}=\bigcup_{i=1}^N\mathcal{D}_i=\bigcup_{i=1}^N\bigcup_{j=1}^{|\mathcal{D}_i|}\{(x_j^i,y_j^i)\}$ and $\widetilde{\mathcal{D}}=\bigcup_{i=1}^N\widetilde{\mathcal{D}}_i=\bigcup_{i=1}^N\bigcup_{j=1}^{|\mathcal{D}_i|}\{(\tilde{x}_j^i,\tilde{y}_j^i)\}$ satisfying Assumption 3.1, let (P,S) and $(\widetilde{P},\widetilde{S})$ be the solutions of (47)-(48) corresponding to $\mathcal{D}$ and $\widetilde{\mathcal{D}}$, respectively. Then, for all*

$t = 1, ..., T$, we have

$$\left|P(t) - \widetilde{P}(t)\right| \leq C \cdot \Big[\sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i - \tilde{u}_j^i|^2\Big]^{1/2}, \tag{66}$$

$$\left|[M(P(t))]^{-1} - [M(\widetilde{P}(t))]^{-1}\right| \leq C \cdot \Big[\sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i - \tilde{u}_j^i|^2\Big]^{1/2}, \tag{67}$$

$$|S(t) - \widetilde{S}(t)| \leq C \cdot \Big[\sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i - \tilde{u}_j^i|^2 + |y_j^i - \tilde{y}_j^i|^2\Big]^{1/2}, \tag{68}$$

where $C > 0$ is a constant depending on λ, β, N, $\bar{N}$, $(w_1^\star, ..., w_N^\star)$, $\Theta^\star$, K_x, K_y, and T.

Proof. We let C be a constant depending on λ, β, N, $\bar{N}$, $\Theta^\star$, $(w_1^\star, ..., w_N^\star)$, K_x, K_y, and T, and allow C to vary from place to place throughout the proof. We employ backward induction starting from $t = T$ to prove (67), (66), and (68).

From (47), the terminal condition $|P(T) - \widetilde{P}(T)|^2$ can be written as

$$\begin{aligned}
&\left|P(T) - \widetilde{P}(T)\right|^2 \\
=&\Big|[w_1^\star I_p, \cdots, w_N^\star I_p]^\top \Big[\sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} (u_j^i u_j^{i\top} - \tilde{u}_j^i \tilde{u}_j^{i\top})\Big] [w_1^\star I_p, \cdots, w_N^\star I_p]\Big|^2 \\
=&\sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \Big|\sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} (u_j^i u_j^{i\top} - \tilde{u}_j^i \tilde{u}_j^{i\top})\Big|^2.
\end{aligned}$$

By Lemmas B.1 and B.2, we have

$$\begin{aligned}
\left|P(T) - \widetilde{P}(T)\right|^2 &\leq \sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \sum_{i=1}^{N} w_i^\star \Big|\sum_{j=1}^{|\mathcal{D}_i|} (u_j^i u_j^{i\top} - \tilde{u}_j^i \tilde{u}_j^{i\top})\Big|^2 \\
&\leq \sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} \left|u_j^i u_j^{i\top} - \tilde{u}_j^i \tilde{u}_j^{i\top}\right|^2 \\
&= \sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} \left|(u_j^i - \tilde{u}_j^i) u_j^{i\top} + \tilde{u}_j^i (u_j^{i\top} - \tilde{u}_j^{i\top})\right|^2 \\
&\leq \sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} 2\big(\left|u_j^i - \tilde{u}_j^i\right|^2 \left|u_j^i\right|^2 + \left|\tilde{u}_j^i\right|^2 \left|u_j^i - \tilde{u}_j^i\right|^2\big) \\
&\leq \sum_{k,l=1}^{N} w_k^{\star 2} w_l^{\star 2} \sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} 4K_x^2 \left|u_j^i - \tilde{u}_j^i\right|^2.
\end{aligned} \tag{69}$$

Therefore, (66) holds for $t = T$.

Assume by induction that (66) holds for $t = u$. We show that (66) also holds for $t = u - 1$. From (47), we have

$$P(u-1) - \widetilde{P}(u-1) = [M(\widetilde{P}(u))]^{-1} - [M(P(u))]^{-1}$$

$$=[M(P(u))]^{-1}(P(u)-\widetilde{P}(u))[M(\widetilde{P}(u))]^{-1},$$

and by Lemma B.1, we further have

$$\left|P(u-1)-\widetilde{P}(u-1)\right| \leq \left|[M(P(u))]^{-1}\right| \cdot \left|P(u)-\widetilde{P}(u)\right| \cdot \left|[M(\widetilde{P}(u))]^{-1}\right|. \tag{70}$$

By Corollary B.1 and the induction hypothesis for $t=u$, (70) implies that (66) holds for $t=u-1$. We have shown by induction that (66) holds for all $t=1$, ..., T.

Since

$$[M(P(t))]^{-1}-[M(\widetilde{P}(t))]^{-1}=[M(P(t))]^{-1}(\widetilde{P}(t)-P(t))[M(\widetilde{P}(t))]^{-1},$$

by Lemma B.1 we have

$$\left|[M(P(t))]^{-1}-[M(\widetilde{P}(t))]^{-1}\right| \leq \left|[M(P(t))]^{-1}\right| \cdot \left|\widetilde{P}(t)-P(t)\right| \cdot \left|[M(\widetilde{P}(t))]^{-1}\right|. \tag{71}$$

Estimate (67) then follows from (71), (66), and Corollary B.1.

Now we establish estimate (68). By (48), the terminal condition $S(T)-\widetilde{S}(T)$ satisfies

$$\begin{aligned}|S(T)-\widetilde{S}(T)|^2 =& \Big|[w_1^\star,\cdots,w_N^\star]^\top \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}(u_j^i y_j^i-\tilde{u}_j^i\tilde{y}_j^i)\Big|^2\\ =&\Big(\sum_{i=1}^N w_i^{\star 2}\Big)\Big|\sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}(u_j^i y_j^i-\tilde{u}_j^i\tilde{y}_j^i)\Big|^2.\end{aligned}$$

By Lemmas B.1 and B.2, we obtain

$$\begin{aligned}|S(T)-\widetilde{S}(T)|^2 \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star \Big|\sum_{j=1}^{|\mathcal{D}_i|}(u_j^i y_j^i-\tilde{u}_j^i\tilde{y}_j^i)\Big|^2\\ \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}\left|u_j^i y_j^i-\tilde{u}_j^i\tilde{y}_j^i\right|^2\\ =& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}\left|(u_j^i-\tilde{u}_j^i)y_j^i+\tilde{u}_j^i(y_j^i-\tilde{y}_j^i)\right|^2\\ \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\cdot 2\sum_{j=1}^{|\mathcal{D}_i|}\left[\left|(u_j^i-\tilde{u}_j^i)y_j^i\right|^2+\left|\tilde{u}_j^i(y_j^i-\tilde{y}_j^i)\right|^2\right]\\ \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\cdot 2\sum_{j=1}^{|\mathcal{D}_i|}\left[|u_j^i-\tilde{u}_j^i|^2|y_j^i|^2+|\tilde{u}_j^i|^2|y_j^i-\tilde{y}_j^i|^2\right]\\ \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\cdot 2\sum_{j=1}^{|\mathcal{D}_i|}\left[K_y^2|u_j^i-\tilde{u}_j^i|^2+K_x^2|y_j^i-\tilde{y}_j^i|^2\right]\\ \leq& \Big(\sum_{i=1}^N w_i^{\star 2}\Big)2(K_x^2+K_y^2)\sum_{i=1}^N w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}\left[|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\right],\end{aligned} \tag{72}$$

and thus (68) holds for $t=T$. Assuming by induction that (68) holds for $t=u$, we show that (68) also holds for $t=u-1$. From (48), the difference $S(t)-\widetilde{S}(t)$ can be

written as

$$\begin{aligned} S(u-1)-\widetilde{S}(u-1) =&[M(\widetilde{P}(u))]^{-1}[S(u)-\widetilde{S}(u)] \\ &+\big[(M(P(u)))^{-1}-(M(\widetilde{P}(u)))^{-1}\big](S(u)-\lambda\Theta^\star), \end{aligned}$$

and satisfies, by Lemma B.1 and Corollary B.1, that

$$\begin{aligned} |S(u-1)-\widetilde{S}(u-1)| \le&\big|[M(\widetilde{P}(u))]^{-1}\big|\cdot|S(u)-\widetilde{S}(u)| \\ &+\big|(M(P(u)))^{-1}-(M(\widetilde{P}(u)))^{-1}\big|\cdot(|S(u)|+\lambda|\Theta^\star|). \end{aligned} \tag{73}$$

By Corollary B.1, (67), and the induction hypothesis, (73) implies that (68) holds for $t=u-1$. We have shown by induction that (68) holds for all $t=1,...,T$. □

Proposition B.1. *For two arbitrary data sets $\mathcal{D}=\bigcup_{i=1}^N\mathcal{D}_i=\bigcup_{i=1}^N\bigcup_{j=1}^{|\mathcal{D}_i|}\{(x_j^i,y_j^i)\}$ and $\widetilde{\mathcal{D}}=\bigcup_{i=1}^N\widetilde{\mathcal{D}}_i=\bigcup_{i=1}^N\bigcup_{j=1}^{|\mathcal{D}_i|}\{(\tilde{x}_j^i,\tilde{y}_j^i)\}$ satisfying Assumption 3.1, let $\boldsymbol{\alpha}=(\alpha_1^\top,\cdots,\alpha_N^\top)^\top$ and $\widetilde{\boldsymbol{\alpha}}=(\widetilde{\alpha}_1^\top,\cdots,\widetilde{\alpha}_N^\top)^\top$ be the optimal controls for (4) and (5) corresponding to $\mathcal{D}$ and $\widetilde{\mathcal{D}}$, respectively. Then, we have*

$$\sum_{t=0}^{T-1}|\boldsymbol{\alpha}(t)-\widetilde{\boldsymbol{\alpha}}(t)|\le C\Big[\sum_{i=1}^N w_i^{\star 2}|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}, \tag{74}$$

where $C>0$ is a constant depending on λ, β, N, $\bar{N}$, $\Theta^\star$, $(w_1^\star,\cdots,w_N^\star)$, K_x, K_y, and T.

Proof of Proposition B.1. Let C be a constant depending on λ, β, N, $\bar{N}$, $\Theta^\star$, $(w_1^\star,\cdots,w_N^\star)$, K_x, K_y, and T, and C is allowed to vary from place to place throughout the proof. By (45), the optimal control $\boldsymbol{\alpha}(t)$ along the optimal trajectory can be written as

$$\begin{aligned} \boldsymbol{\alpha}(t)=&\big[(M(P(t+1)))^{-1}-I_{Np}\big]\Big[\Theta^\star+\sum_{u=0}^{t-1}\boldsymbol{\alpha}(u)\Big] \\ &+\big[M(P(t+1))\big]^{-1}(\lambda\Theta^\star-S(t+1)), \end{aligned}$$

and the difference $\boldsymbol{\alpha}(t)-\widetilde{\boldsymbol{\alpha}}(t)$ can be written as

$$\begin{aligned} &\boldsymbol{\alpha}(t)-\widetilde{\boldsymbol{\alpha}}(t) \\ &=\big[(M(P(t+1)))^{-1}-(M(\widetilde{P}(t+1)))^{-1}\big]\Big[\Theta^\star+\sum_{u=0}^{t-1}\boldsymbol{\alpha}(u)+\lambda\Theta^\star-S(t+1)\Big] \\ &\quad+\big[(M(\widetilde{P}(t+1)))^{-1}-I_{Np}\big]\sum_{u=0}^{t-1}(\boldsymbol{\alpha}(u)-\widetilde{\boldsymbol{\alpha}}(u)) \\ &\quad+(M(\widetilde{P}(t+1)))^{-1}(S(t+1)-\widetilde{S}(t+1)). \end{aligned} \tag{75}$$

For $t=0$, the difference $\boldsymbol{\alpha}(0)-\widetilde{\boldsymbol{\alpha}}(0)$ can be written as

$$\begin{aligned} \boldsymbol{\alpha}(0)-\widetilde{\boldsymbol{\alpha}}(0)=&\big[(M(P(1)))^{-1}-(M(\widetilde{P}(1)))^{-1}\big]\big[(\lambda+1)\Theta^\star-S(1)\big] \\ &-[M(\widetilde{P}(1))]^{-1}(S(1)-\widetilde{S}(1)), \end{aligned}$$

and satisfies by Lemma B.1 that

$$\begin{aligned} \big|\boldsymbol{\alpha}(0)-\widetilde{\boldsymbol{\alpha}}(0)\big|\le&\big|(M(P(1)))^{-1}-(M(\widetilde{P}(1)))^{-1}\big|\cdot\big[(\lambda+1)|\Theta^\star|+|S(1)|\big] \\ &+\big|[M(\widetilde{P}(1))]^{-1}\big|\cdot\big|S(1)-\widetilde{S}(1)\big|. \end{aligned}$$

It then follows from Corollary B.1, (67), and (68) that

$$\left|\boldsymbol{\alpha}(0)-\widetilde{\boldsymbol{\alpha}}(0)\right| \leq C\cdot\Big[\sum_{i=1}^{N} w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}.$$

Assume by induction that

$$\sum_{u=0}^{t-1}\left|\boldsymbol{\alpha}(u)-\widetilde{\boldsymbol{\alpha}}(u)\right| \leq C\cdot\Big[\sum_{i=1}^{N} w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}.$$

By Lemma B.1, (75) implies that

$$\begin{aligned}
&|\boldsymbol{\alpha}(t)-\widetilde{\boldsymbol{\alpha}}(t)|\\
\leq&\left|(M(P(t+1)))^{-1}-(M(\widetilde{P}(t+1)))^{-1}\right|\cdot\Big[|\Theta^\star|+\sum_{u=0}^{t-1}|\boldsymbol{\alpha}(u)|+\lambda|\Theta^\star|+|S(t+1)|\Big]\\
&+\left|(M(\widetilde{P}(t+1)))^{-1}-I_{Np}\right|\sum_{u=0}^{t-1}|\boldsymbol{\alpha}(u)-\widetilde{\boldsymbol{\alpha}}(u)|\\
&+\left|(M(\widetilde{P}(t+1)))^{-1}\right|\cdot\left|S(t+1)-\widetilde{S}(t+1)\right|.
\end{aligned}$$

By (67), (68), and the induction hypothesis, the above inequality implies that

$$|\boldsymbol{\alpha}(t)-\widetilde{\boldsymbol{\alpha}}(t)| \leq C\cdot\Big[\sum_{i=1}^{N} w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2},$$

and therefore

$$\sum_{u=0}^{t}\left|\boldsymbol{\alpha}(u)-\widetilde{\boldsymbol{\alpha}}(u)\right| \leq C\cdot\Big[\sum_{i=1}^{N} w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{u}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}.$$

Estimate (74) is then proved by induction. □

Corollary B.2. *Under the hypothesis of Proposition B.1, the costs (4) under the optimal controls* $\mathbf{a}$ *and* $\widetilde{\mathbf{a}}$ *satisfy the estimate*

$$\left|L^\star(\boldsymbol{\alpha};\mathcal{D})-L^\star(\widetilde{\boldsymbol{\alpha}};\widetilde{\mathcal{D}})\right| \leq C\cdot\Big[\sum_{i=1}^{N} w_i^\star|\mathcal{D}_i|\sum_{j=1}^{|\mathcal{D}_i|}|u_j^i-\tilde{x}_j^i|^2+|y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}. \tag{76}$$

where $C>0$ *is a constant depending on* λ, β, N, $\bar{N}$, $\Theta^\star$, $(w_1^\star,\cdots,w_N^\star)$, K_x, K_y, *and* T.

Proof of Corollary B.2. Throughout the proof, C is a constant depending on λ, β, $\bar{N}$, $\Theta^\star$, $(w_1^\star,\cdots,w_N^\star)$, K_x, K_y, and T, and may vary from place to place. By (53) and (5), the difference of the costs can be written as

$$\begin{aligned}
&L^\star(\boldsymbol{\alpha};\mathcal{D})-L^\star(\widetilde{\boldsymbol{\alpha}};\widetilde{\mathcal{D}})\\
=&\Theta^\top(0)(P(0)-\widetilde{P}(0))\Theta(0)+2(S(0)-\widetilde{S}(0))^\top\Theta(0)+r(0)-\widetilde{r}(0),\\
=&\Theta^{\star\top}(P(0)-\widetilde{P}(0))\Theta^\star+2(S(0)-\widetilde{S}(0))^\top\Theta^\star+r(0)-\widetilde{r}(0).
\end{aligned}$$

By the triangular inequality, the difference satisfies

$$\begin{aligned}
&\left|L^\star(\boldsymbol{\alpha};\mathcal{D})-L^\star(\widetilde{\boldsymbol{\alpha}};\widetilde{\mathcal{D}})\right|\\
&\leq|P(0)-\widetilde{P}(0)|\cdot|\Theta^\star|^2+2|S(0)-\widetilde{S}(0)|\cdot|\Theta^\star|+|r(0)-\widetilde{r}(0)|.
\end{aligned}\tag{77}$$

We claim that $r(0)-\widetilde{r}(0)$ satisfies the estimate

$$|r(0)-\widetilde{r}(0)| \le C\cdot\Big[\sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i-\tilde{u}_j^i|^2 + |y_j^i-\tilde{y}_j^i|^2\Big]^{1/2}. \tag{78}$$

Then, the desired estimate (76) is established by (77), (66), (68), and (78).

Now we prove the claim (78). From (49), the difference $r-\widetilde{r}$ satisfies

$$\begin{aligned}
r(t)-\widetilde{r}(t) &= r(t+1)-\widetilde{r}(t+1)\\
&\quad -(S(t+1)-\widetilde{S}(t+1))\big\{(M(P(t+1)))^{-1}\big[S(t+1)-2\lambda\Theta^\star\big]\\
&\quad +(M(\widetilde{P}(t+1)))^{-1}\widetilde{S}(t+1)\big\}\\
&\quad -\widetilde{S}^\top(t+1)\big[(M(P(t+1)))^{-1}-(M(\widetilde{P}(t+1)))^{-1}\big]\big[S(t+1)-2\lambda\Theta^\star\big]\\
&\quad -\Theta^{\star\top}\lambda\big[(M(P(t+1)))^{-1}-(M(\widetilde{P}(t+1)))^{-1}\big]\lambda\Theta^\star,\\
r(T)-\widetilde{r}(T) &= \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}(|y_j^i|^2-|\tilde{y}_j^i|^2).
\end{aligned} \tag{79}$$

Inductively, $r(0)-\widetilde{r}(0)$ can be written in terms of $r(T)-\widetilde{r}(T)$ as

$$\begin{aligned}
&r(0)-\widetilde{r}(0)\\
&=r(T)-\widetilde{r}(T)\\
&\quad -\sum_{t=1}^{T}\Big\{(S(t)-\widetilde{S}(t))\big[(M(P(t)))^{-1}\big(S(t)-2\lambda\Theta^\star\big)+(M(\widetilde{P}(t)))^{-1}\widetilde{S}(t)\big]\\
&\quad +\widetilde{S}^\top(t)\big[(M(P(t)))^{-1}\\
&\quad -(M(\widetilde{P}(t)))^{-1}\big]\big[S(t)-2\lambda\Theta^\star\big]+\Theta^{\star\top}\lambda\big[(M(P(t)))^{-1}-(M(\widetilde{P}(t)))^{-1}\big]\lambda\Theta^\star\Big\}.
\end{aligned}$$

By Lemma B.1, we have

$$\begin{aligned}
&|r(0)-\widetilde{r}(0)|\\
&\le |r(T)-\widetilde{r}(T)|+\sum_{t=1}^{T}\Big\{\big|S(t)-\widetilde{S}(t)\big|\\
&\quad\times\Big[\big|(M(P(t)))^{-1}\big|\cdot\big(|S(t)|+2\lambda|\Theta^\star|\big)+\big|(M(\widetilde{P}(t)))^{-1}\big|\cdot\big|\widetilde{S}(t)\big|\Big]\\
&\quad+\big|(M(P(t)))^{-1}-(M(\widetilde{P}(t)))^{-1}\big|\cdot\big(\big|\widetilde{S}(t)\big|\cdot\big|S(t)-2\lambda\Theta^\star\big|+\lambda|\Theta^\star|^2\big)\Big\}.
\end{aligned} \tag{80}$$

By (79), Lemma B.2, and the Cauchy-Schwarz inequality, the terminal condition satisfies

$$\begin{aligned}
|r(T)-\widetilde{r}(T)| &= \Big|\sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}(y_j^i-\tilde{y}_j^i)(y_j^i+\tilde{y}_j^i)\Big|\\
&\le \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}\big|(y_j^i-\tilde{y}_j^i)(y_j^i+\tilde{y}_j^i)\big|\\
&\le \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|}|y_j^i+\tilde{y}_j^i|\cdot|y_j^i-\tilde{y}_j^i|
\end{aligned}$$

$$\leq 2K_y \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i - \widetilde{y}_j^i|$$

$$\leq 2K_y \Big(\sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i - \widetilde{y}_j^i|^2 \Big)^{1/2}. \tag{81}$$

Estimate (78) then follows from (80), (81), (67), (68), and (63). □

We may now derive the proof of Theorem 3.5 as a direct consequence of Corollary B.2.

Proof of Theorem 3.5. Fix $\widetilde{\mathcal{D}} \in \mathbb{D}^{p,\varepsilon}$, and let $I \subseteq \{(i,j) : i = 1,\dots,N,\, j = 1,\dots,|\mathcal{D}_i|\}$ consist of all indices (i,j) for which $\{(i,j) : u_j^i \neq \widetilde{u}_j^i \text{ and } y_j^i \neq \widetilde{y}_j^i\}$. By Corollary B.2, there is a constant $C > 0$ which is independent of $\mathcal{D}$ such that

$$\begin{aligned}
\big|L^\star(\boldsymbol{\alpha};\mathcal{D}) - L^\star(\widetilde{\boldsymbol{\alpha}};\widetilde{\mathcal{D}})\big|^2 \leq & \widetilde{C} \sum_{i=1}^{N} w_i^\star |\mathcal{D}_i| \sum_{j=1}^{|\mathcal{D}_i|} (|u_j^i - \widetilde{u}_j^i|^2 + |y_j^i - \widetilde{y}_j^i|^2) \\
\leq & \widetilde{C} \cdot \sum_{(i,j)\in I} w_i^\star |\mathcal{D}_i| (|u_j^i - \widetilde{u}_j^i|^2 + |y_j^i - \widetilde{y}_j^i|^2) \\
\leq & \widetilde{C} \cdot \sum_{(i,j)\in I} w_i^\star |\mathcal{D}_i| \cdot \max\{|u_j^i - \widetilde{u}_j^i|^2, |y_j^i - \widetilde{y}_j^i|^2\} \\
= & \widetilde{C} \cdot \sum_{(i,j)\in I} w_i^\star |\mathcal{D}_i| \cdot \max\{|u_j^i - \widetilde{u}_j^i|, |y_j^i - \widetilde{y}_j^i|\}^2 \\
\leq & \widetilde{C} \cdot \sum_{(i,j)\in I} w_i^\star |\mathcal{D}_i| \cdot 2\varepsilon^2 \\
\leq & \widetilde{C} \# I \, p \, 2\varepsilon^2,
\end{aligned}$$

where we have set $\widetilde{C} \overset{\text{def.}}{=} C^{1/2}$ and used the identity $\# I = \bar{N} q$. □

Appendix C. Intuition for our transfer learning algorithm. In this section, we give some intuition on our algorithm, which, at the same time, can also be understood as a motivation for the way it is defined. But first, we elaborate on our method actually achieving transfer learning and not multi-task learning.

C.1. **Is it really transfer learning?** The terminal cost in (4) uses a weighted combination of the loss on all datasets, leading to the question of whether our method performs multi-task learning. This is not the case as explained below.

Algorithm 1 evaluates the predictive performance of each $\theta_i^\star$, $i \in [N]$ on the focal dataset $\mathcal{D}_1$ to determine the weights $w_i^\star$, $i \in [N]$, which rank the similarity of the datasets $\mathcal{D}_i$ to the focal dataset $\mathcal{D}_1$. In particular, they quantify how much knowledge $\mathcal{D}_i$ provides for the task on $\mathcal{D}_1$. Based on those weights, the ensemble parameters are biased towards maximizing performance on the primary dataset $\mathcal{D}_1$. Contrarily, in multi-task learning, the weights $w^\star$ should rather be selected to all be equal (adjusting for dataset imbalances) so as to ensure that the aggregated learner is not biased toward any individual dataset.

The extreme cases best explain this. If all datasets are i.i.d. copies of $\mathcal{D}_1$, then they are equally weighted through $w^\star$, and the model is simply fine-tuned on the

union of all samples as terminal cost (as it should be). On the other hand, if all datasets are completely orthogonal to each other, then the full weight is on $\mathcal{D}_1$, and no "orthogonal" knowledge is transferred (again as it should be). Indeed, in this case, the kernel regressors pre-trained on datasets $\mathcal{D}_i$ for $i > 1$ will not exhibit any predictive power on the focal dataset $\mathcal{D}_1$ and visa versa, so Algorithm 1 will set $w^\star = (1, 0, 0, \dots, 0)$. Therefore, Algorithm 2 returns

$$\sum_i w_i \theta_i^\star = 1\theta_1^\star + 0 \sum_{i>1} \theta_i^\star = \theta_1^\star.$$

The aggregated model will have no predictive performance on datasets $\mathcal{D}_i$ for $i \neq 1$ (since the datasets were assumed to be orthogonal to one another).

Thus, Algorithm 2 does not perform multi-task learning but transfer learning, as can also be seen in our experiments in Section 4.1. Our algorithm can, however, be modified to perform multi-task learning. Though we are not interested in this for this research paper, we do describe how it can be done in our main rebuttal above.

C.2. **What the hyper-parameters λ, β, and T control and how to choose them.**

C.2.1. *The transfer learning factor λ.* The hyper-parameter λ controls the level of transfer learning by controlling how much the parameters θ_i are fine-tuned, starting from $\theta_i^\star$. In particular, Algorithm 1 first determines how useful each of the datasets is for the focal task, and the terminal cost l can be interpreted as a loss on the "joint dataset weighted by $w^\star$". To understand how the λ influences the result, it is best to consider the extreme cases and why they can be suboptimal in certain data regimes. For simplicity, we assume that $\beta = 0$.

The choice $\lambda \approx \infty$ means that no deviation from $\theta^\star$ is allowed. Hence, this simply leads to the parameter combination $\sum_i w_i^\star \theta_i^\star$, so no additional fine-tuning on the "joint weighted" dataset happens. This is suboptimal if there is one very large and very useful dataset and otherwise small datasets. For simplicity, consider $N = 2$ datasets with the focal dataset $\mathcal{D}_1$ being comparatively small to $\mathcal{D}_2$. Then, the pre-trained parameter $\theta_1^\star$ would be less informative, but would be weighted as equally important to the more informative parameter $\theta_2^\star$. Note here that $\theta_2^\star$ evaluated on $\mathcal{D}_1$ cannot lead to a higher weight than for $\theta_1^\star$, since this is the optimal weight for $\mathcal{D}_1$. This is a natural limitation of Algorithm 1 since it cannot know that $\mathcal{D}_2$ is actually a larger i.i.d. copy of $\mathcal{D}_1$. This shows that for $\lambda \approx \infty$, the inability to fine-tune will clearly produce a suboptimal aggregation.

On the other hand, setting $\lambda = 0$, then $\theta^{w^\star} = \sum_i w_i \theta_i$ purely minimizes the weighted loss

$$\sum_i w_i \sum_j (f_{\theta^w}(x_j^i) - y_j^i)^2,$$

completely forgetting about the pre-trained parameters $\theta_1^\star, \dots, \theta_N^\star$. This is suboptimal if one dataset is much larger than the others, and only slightly useful for $\mathcal{D}_1$. Indeed, this would lead (in the limit where the size of this large dataset goes to ∞) to an optimizer of the large dataset simply because it dominates the loss as long as its weight $w_i^\star > 0$.

In either case, we see that intermediate values of λ yield the most desirable outputs of Algorithm 2.

C.2.2. *Convergence behaviour controlled by β and T.* First, we note that β controls the stability of the convergence by penalizing large steps in the sequence $\Theta^{0\cdots T}$. This is only needed when intermediate steps $\Theta(t)$ are of importance, i.e., if we want to obtain stable iterates, since otherwise we can simply focus on the terminal cost under the transfer learning constraint by setting $\beta = 0$.

In the regime $\beta = 0$, only the final $\Theta(T)$ is of importance since our algorithm admits a closed form solution. In particular, it is optimal that all prior parameters for $t < T$ satisfy $\Theta(t) = \Theta^\star$. Hence, $T = 1$ is sufficient, and at the same time the most efficient and therefore optimal choice. Hence, we used $T = 1$ in Section 4.1, where only the final $\Theta(T)$ is of interest.

Contrarily, we note that for $T = 1$, the stability term coincides with the transfer learning term, hence they can be accumulated such that $\lambda + \beta$ acts as the factor for transfer learning.

This leaves us with the question of whether there is a benefit of the regime $\beta > 0, T > 1$ at all. The answer is affirmative, since this regime "spreads out the convergence in time", which allows for the option of implicit regularization through an early stopping procedure. Though the investigation of the statistical properties of early stopping of our algorithm is beyond the scope of this paper, one would expect that using this regime as a vehicle toward early stopping regularization would likely improve the statistical properties of our aggregated learning, as the benefits to model generalization of early stopping on classical learning algorithms are well documented [31, 67, 70, 73]. In this regime, the total iteration number T indeed represents a trade-off between the computing budget and the convergence of the loss function. When computational resources permit, choosing T large enough will ensure that the output $\sum_i w_i^\star \theta_i$ of the algorithm will converge to its $T = \infty$ limit by a similar argument as in the proof of [17, Theorem 4].

Appendix D. Background.

D.1. **Finite-rank kernel ridge regression.** In this paper, we consider regret-optimal memoryless training dynamics for ***f**inite-rank **k**ernel **r**idge **r**egressors* (fKRR) trained from N finite datasets drawn from N (possibly) different data-generating distributions. We consider a scenario in which these datasets can originate from N distinct tasks or be adversarially generated perturbations of the initial dataset, and we aim for our fKRR to exhibit robustness toward them.

Any continuous *feature map* $\phi : \mathbb{R}^d \to \mathbb{R}^p$ induces a hypothesis of fKRR $f_\theta : \mathbb{R}^d \to \mathbb{R}$, each of which is uniquely determined by a *trainable parameter* $\theta \in \mathbb{R}^p$ via

$$f_\theta(x) = \phi(x)^\top \theta; \tag{82}$$

where all vectors are represented as column-vectors and not row vectors.

The hypothesis class $\mathcal{H}$ possesses a natural reproducing kernel Hilbert space (RKHS) structure with inner-product given for $f_\theta, f_{\tilde\theta} \in \mathcal{H}$ by $\langle f_\theta, f_{\tilde\theta}\rangle_\phi \overset{\text{def.}}{=} \sum_{k=1}^p \theta_k \tilde\theta_k$. This RKHS structure suggests an optimal selection criterion to decide which hypothesis in $\mathcal{H}$ best describes a *single* training set $\mathcal{D}_1 \overset{\text{def.}}{=} \{(x_j^1, y_j^1)\}_{j=1}^{S^1}$ of $S^1 \in \mathbb{N}_+$ by solving the penalized least-squares problem

$$\theta_1^\star \in \underset{\theta\in\mathbb{R}^p}{\arg\min} \left\|\Phi(x^1)\theta - y^1\right\|_2^2 + \kappa\left\|\theta\right\|_2^2, \tag{83}$$

where

$$\Phi(x^1) \overset{\text{def.}}{=} \left(\phi(x_1^1),\ \phi(x_2^1),\ \cdots,\ \phi(x_{S^1}^1)\right)^\top \in \mathbb{R}^{S^1\times p},\ y^1 \overset{\text{def.}}{=} (y_1^1, y_2^1, \cdots, y_{S^1}^1)^\top \in \mathbb{R}^{S^1},$$

and $\kappa > 0$ is a fixed hyperparameter. In this classical case, the representer theorem [41, 42] states that the optimal choice of f_θ to the convex optimization problem (83) is parameterized by

$$\theta_1^\star = (\Phi^\top(x^1)\Phi(x^1) + \kappa I_p)^{-1}\Phi^\top(x^1)y^1. \tag{84}$$

Alternatively, the optimal parameter choice $\theta_1^\star$ can be arrived at via *gradient descent* since (83) has a strictly convex loss with a Lipschitz gradient. The training of fKRR on a single dataset is very well understood. Likewise, other aspects of fKRR are well understood; namely, their statistical properties [18], their in-sample behavior in [4], and even the approximation power and risk of randomly generated fKRR are respectively known in [28] and [27].

Using the fKRR (82), the joint terminal time objective simplifies to

$$\begin{aligned} l(\theta_1, \dots, \theta_N, w; \mathcal{D}) &\stackrel{\text{def.}}{=} \sum_{i=1}^N w_i \left(\sum_{j=1}^{S_i} \left| \langle u_j^i, \theta^w \rangle - y_j^i \right|^2 \right) \\ \text{s.t. } \theta^w &\stackrel{\text{def.}}{=} \sum_{i=1}^N \theta_i w_i, \end{aligned} \tag{85}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner-product, $u_j^i \stackrel{\text{def.}}{=} \phi(x_j^i)$, and $w = (w_1, \cdots, w_N) \in [0,1]^N$ with $\sum_{i=1}^N w_i = 1$.

Appendix E. **Background: American option pricing and optimal stopping.** This section contains a brief recap of the optimal stopping problem in American option pricing.

An American option is a financial derivative that gives its owner the right to execute a certain predetermined trade at any time between the start date and the maturity of the option. For example, a call (put) option gives its owner the right to buy (sell) a certain stock from the writer of the option for a fixed price K, called the strike. If the current stock price X_t is higher (lower) than K, then the option owner can make a profit of $(X_t - K)_+$ (or $(K - X_t)_+$ respectively) by directly selling (buying) the stock again in the open market after having exercised their right to buy (sell) from the option writer. While American options allow for continuous-in-time execution, it is common practice to approximate them via Bermudan options, which can be executed on a predetermined time grid $t_0 < \cdots t_M$. We briefly overview the American option pricing problem, details of which can be found in [57, Chapter 25].

In order to capture the computational challenges posed by the American option pricing problem in higher dimensions, we consider a stochastic process $(X_t)_{t=0}^M \in (\mathbb{R}^d)^{(M+1)}$ describing the price of d stocks on the time grid, and let $(Z_t)_{t=0}^M \in \mathbb{R}^{(M+1)}$ be the corresponding discounted payoff process of an American option written on a one-dimensional process obtained from X_t (e.g. the largest price among the d stocks). We define the filtration $\mathbb{F} = (\mathcal{F}_t)_{t=0}^M$ generated by X via $\mathcal{F}_t = \sigma(X_s | s \le t)$. Then, the price of the American option is given by the starting value U_0 of the *Snell envelope* [62], given by

$$U_m = \sup_{\tau \in \mathcal{T}_m} \mathbb{E}[Z_\tau \,|\, \mathcal{F}_m], \tag{86}$$

where $\mathcal{T}_m$ is the set of all stopping times $\tau \ge m$. The smallest optimal stopping time is

$$\tau_M := M, \tag{87}$$

$$\tau_m := \begin{cases} m, & \text{if } Z_m \geq \mathbb{E}[U_{m+1} \,|\, \mathcal{F}_m], \\ \tau_{m+1}, & \text{otherwise,} \end{cases}$$

hence, to compute the price of the American option, it suffices to compute the *continuation values* $\mathbb{E}[U_{m+1} \,|\, \mathcal{F}_m]$ for all m. In the following, we will use the *randomized least squares Monte Carlo* (RLSM) algorithm of [32], which utilizes randomized neural networks to approximate the continuation values via backward induction (note the backward recursive scheme that (86) and (87) imply). In particular, this leads to M backward-recursively defined regression problems that need to be solved sequentially. Since (86) is a maximization problem, and since we only approximate the optimal stopping time, we compute lower bounds of the price U_0. The standard procedure is to fix some law for the process X and to sample a dataset of $\nu = \nu_1 + \nu_2$ i.i.d. paths of X, where ν_1 of them are used for training the randomized neural networks (approximating the continuation value functions) and ν_2 of them are used to evaluate the resulting price U_0 (using these trained networks). Through the independence of the two parts of the dataset, it is clear that any over-fitting bias the networks might learn will *not* lead to prices that are too large. This is important, since the quality of a method can therefore be evaluated by the resulting price, with a higher price indicating a better result.

Appendix F. Proofs for the accelerated heuristic.

Remark F.1. According to (43) in Assumption B.1, $\widehat{P}(T) \geq 0$, and therefore (19) admits a unique solution.

Remark F.2. Similar to Corollary B.1, we have $\left|[(\lambda+\beta)I_{Np} + \widehat{P}(t)]^{-1}\right| \leq (\lambda+\beta)^{-1}$. By replacing $(w_1^\star, ..., w_N^\star)$ in the proof of Corollary B.1 with $(1/N, ..., 1/N)$, we obtain $|\widehat{S}(t)| \leq C \cdot N^{1/2}$ for all $t = 1, ..., T$, where $C > 0$ is a constant depending on λ, β, $\Theta^\star$, K_x, K_y, and T.

Lemma F.1. *Let P and $\widehat{P}$ be the solutions of systems (47) and (19), respectively. Then, they satisfy that for all $t = 1, 2, ..., T$,*

$$\left|P(t) - \widehat{P}(t)\right| \leq C \cdot \bar{N} N \max_{1 \leq i,k,l \leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right|, \tag{88}$$

$$\left|(M(P(t)))^{-1} - (M(\widehat{P}(t)))^{-1}\right| \leq C \cdot \bar{N} N \max_{1 \leq i,k,l \leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right|, \tag{89}$$

where $C > 0$ is a constant depending on λ, β, and K_x.

Proof of Lemma F.1. By (19), (47), and Lemmas B.1 and B.2, the difference $\widehat{P}(T) - P(T)$ satisfies

$$\begin{aligned} |\widehat{P}(T) - P(T)|^2 &= \sum_{k,l=1}^{N} \left| \frac{1}{N^2} \sum_{k=1}^{N} \frac{1}{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} - w_k^\star w_l^\star \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \right|^2 \\ &\leq N^2 \cdot \max_{1 \leq k,l,i \leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right|^2 \cdot \left| \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \right|^2 \\ &\leq N^2 \cdot \max_{1 \leq k,l,i \leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right|^2 \cdot \bar{N} \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i|^4 \end{aligned}$$

and

$$|\widehat{P}(T) - P(T)| \leq \bar{N} N K_x^2 \cdot \max_{1\leq k,l,i\leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right|. \tag{90}$$

Therefore, (88) holds for $t = T$.

Assume by induction that (88) holds for $t = u$. We show that (88) also holds for $t = u - 1$. From (19) and (47), we have

$$\begin{aligned} \widehat{P}(u-1) - P(u-1) =& \big[M(P(u))\big]^{-1} - \big[M(\widehat{P}(u))\big]^{-1} \\ =& \big[M(P(u))\big]^{-1} (\widehat{P}(u) - P(u)) \big[M(\widehat{P}(u))\big]^{-1}. \end{aligned} \tag{91}$$

By Lemmas B.1 and B.2, we further have

$$\begin{aligned} \big|\widehat{P}(u-1) - P(u-1)\big| =& \big|[M(P(u))]^{-1} - [M(\widehat{P}(u))]^{-1}\big| \\ \leq& \big|[M(P(u))]^{-1}\big| \cdot \big|\widehat{P}(u) - P(u)\big| \cdot \big|[M(\widehat{P}(u))]^{-1}\big|. \end{aligned} \tag{92}$$

By (92), Corollary B.1, Remark F.2, and the induction hypothesis, we have that (88) holds for $t = u - 1$. By induction, we have shown that (88) holds for all $1 \leq t \leq T$.

From (91), we have

$$\big|(M(P(t)))^{-1} - (M(\widehat{P}(t)))^{-1}\big| \leq \big|(M(P(t)))^{-1}\big| \cdot \big|\widehat{P}(t) - P(t)\big| \cdot \big|(M(\widehat{P}(t)))^{-1}\big|,$$

which, together with (88), Corollary B.1, and Remark F.2, establishes (89). □

Lemma F.2. *Let S and $\widehat{S}$ be the solutions of (48) and (20), respectively. Then, they satisfy that for all $t = 1, \ldots, T$,*

$$\begin{aligned} |S(t) - \widehat{S}(t)| \leq& C \cdot \Big\{ \bar{N} N^{3/2} \max_{1\leq i,k,l\leq N} \left| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \right| \\ &+ \bar{N} N^{1/2} \max_{1\leq k,l\leq N} \left| \frac{1}{N^2} - w_k^\star w_l^\star \right| + |\Theta^\star - \Theta^\star_{(N)}| \Big\}, \end{aligned} \tag{93}$$

where $C > 0$ is a constant depending on λ, β, $\Theta^\star$, K_x, K_y, and T.

Proof of Lemma F.2. By (48), (20), and Lemmas B.1 and B.2, the terminal condition $S(T) - \widehat{S}(T)$ satisfies

$$\begin{aligned} \left|S(T) - \widehat{S}(T)\right|^2 =& \sum_{k=1}^N \left| \frac{1}{N} \sum_{i=1}^N \frac{1}{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i - w_k^\star \sum_{i=1}^N w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i \right|^2 \\ \leq& N \cdot \max_{1\leq k,l\leq N} \left| \frac{1}{N^2} - w_k^\star w_l^\star \right|^2 \cdot \left| \sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i \right|^2 \\ \leq& N \cdot \max_{1\leq k,l\leq N} \left| \frac{1}{N^2} - w_k^\star w_l^\star \right|^2 \cdot \bar{N} \sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} |u_j^i|^2 \cdot |y_j^i|^2, \end{aligned}$$

which implies

$$\left|S(T) - \widehat{S}(T)\right| \leq \bar{N} N^{1/2} \max_{1\leq k,l\leq N} \left| \frac{1}{N^2} - w_k^\star w_l^\star \right| \cdot K_x K_y. \tag{94}$$

Therefore, (93) holds for $t = T$.

Assume by induction that (93) holds for $t = u$. We show that (93) holds for $t = u - 1$. From (48) and (20), the difference $S(u-1) - \widehat{S}(u-1)$ can be written as

$$\begin{aligned} S(u-1) - \widehat{S}(u-1) =& [M(P(u))]^{-1}[S(u) - \widehat{S}(u) - \lambda(\Theta^\star - \Theta^\star_{(N)})] \\ & - \big[(M(\widehat{P}(u)))^{-1} - (M(P(u)))^{-1}\big](\widehat{S}(u) - \lambda\Theta^\star_{(N)}), \end{aligned} \tag{95}$$

and satisfies by Lemma B.1 that

$$\begin{aligned} \big|S(u-1) - \widehat{S}(u-1)\big| \leq& \big|[M(P(u))]^{-1}\big| \cdot \big[|S(u) - \widehat{S}(u)| + \lambda|\Theta^\star - \Theta^\star_{(N)}|\big] \\ & + \big|(M(\widehat{P}(u)))^{-1} - (M(P(u)))^{-1}\big| \cdot \big(|\widehat{S}(u)| + \lambda|\Theta^\star_{(N)}|\big). \end{aligned} \tag{96}$$

By (96), (89), Remark F.2, and the induction hypothesis, we have that (93) holds for $t = u - 1$. By induction, we have shown that (93) holds for all $t = 1, ..., T$. □

We now obtain Theorem 3.6, which is a direct consequence of the following Propositions F.1 and F.2.

Proposition F.1. *The controls* (45) *and* (18) *satisfy* $|\widehat{\boldsymbol{\alpha}}(t)| \leq C \cdot N^{1/2}$ *for all* $t = 0, 1, ..., T-1$, *and*

$$\begin{aligned} \sum_{t=0}^{T-1} |\boldsymbol{\alpha}(t) - \widehat{\boldsymbol{\alpha}}(t)| \leq& C \cdot \Big\{ \bar{N} N^{3/2} \max_{1\leq i,k,l\leq N} \Big| \frac{1}{N^3} - w^\star_k w^\star_l w^\star_i \Big| \\ & + \bar{N} N^{1/2} \cdot \max_{1\leq k,l\leq N} \Big| \frac{1}{N^2} - w^\star_k w^\star_l \Big| + |\Theta^\star - \Theta^\star_{(N)}| \Big\}, \end{aligned} \tag{97}$$

where $C > 0$ *is a constant depending on* λ, β, $\Theta^\star$, K_x, K_y, *and* T.

Proof of Proposition F.1. We let C be a constant depending on λ, β, $\Theta^\star$, K_x, K_y, and T, and C is allowed to vary from place to place throughout the proof. By (45), (18), and the representation of $(\Theta(t))_{t=0}^{T-1}$, we can write $\boldsymbol{\alpha}$ and $\widehat{\boldsymbol{\alpha}}$ as

$$\begin{aligned} \boldsymbol{\alpha}(t) =& [(M(P(t+1)))^{-1} - I_{Np}]\Big[\Theta^\star + \sum_{u=0}^{t-1} \boldsymbol{\alpha}(u)\Big] \\ & + [M(P(t+1))]^{-1}(\lambda\Theta^\star - S(t+1)), \\ \widehat{\boldsymbol{\alpha}}(t) =& [(M(\widehat{P}(t+1)))^{-1} - I_{Np}]\Big[\Theta^\star + \sum_{u=0}^{t-1} \widehat{\boldsymbol{\alpha}}(u)\Big] \\ & + [M(\widehat{P}(t+1))]^{-1}(\lambda\Theta^\star_{(N)} - \widehat{S}(t+1)). \end{aligned}$$

For $t = 0$, we have

$$|\widehat{\boldsymbol{\alpha}}(0)| \leq |(M(\widehat{P}(1)))^{-1} - I_{Np}| \cdot |\Theta^\star| + \big|[M(\widehat{P}(1))]^{-1}\big|(\lambda|\Theta^\star_{(N)}| + |\widehat{S}(1)|),$$

which implies $|\widehat{\boldsymbol{\alpha}}(0)| \leq CN^{1/2}$. Assume by induction that $|\widehat{\boldsymbol{\alpha}}(u)| \leq CN^{1/2}$ for $u = 0, ..., t-1$. Then,

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}(t) \leq& \big|(M(\widehat{P}(t+1)))^{-1} - I_{Np}\big|\Big[|\Theta^\star| + \sum_{u=0}^{t-1} |\widehat{\boldsymbol{\alpha}}(u)|\Big] \\ & + \big|[M(\widehat{P}(t+1))]^{-1}\big|(\lambda|\Theta^\star_{(N)}| + |\widehat{S}(t+1)|) \end{aligned}$$

gives that $|\widehat{\boldsymbol{\alpha}}(t)| \le CN^{1/2}$. By induction, we have shown that $|\widehat{\boldsymbol{\alpha}}(t)| \le CN^{1/2}$ for all $t = 0, 1, ..., T-1$.

Further, we write the difference $\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}$ as

$$\begin{aligned}
&\boldsymbol{\alpha}(t) - \widehat{\boldsymbol{\alpha}}(t) \\
=&\big[(M(P(t+1)))^{-1} - (M(\widehat{P}(t+1)))^{-1}\big]\Big[(\lambda+1)\Theta^\star + \sum_{u=0}^{t-1}\boldsymbol{\alpha}(u) - S(t+1)\Big] \\
&+ \big[(M(\widehat{P}(t+1)))^{-1} - I_{Np}\big]\sum_{u=0}^{t-1}(\boldsymbol{\alpha}(u) - \widehat{\boldsymbol{\alpha}}(u)) \\
&+ [M(\widehat{P}(t+1))]^{-1}(\widehat{S}(t+1) - S(t+1)) \\
&+ \lambda[M(\widehat{P}(t+1))]^{-1}(\Theta^\star - \Theta^\star_{(N)}).
\end{aligned} \tag{98}$$

For $t = 0$, the difference

$$\begin{aligned}
\boldsymbol{\alpha}(0) - \widehat{\boldsymbol{\alpha}}(0) =&\big[(M(P(1)))^{-1} - (M(\widehat{P}(1)))^{-1}\big]\big[(\lambda+1)\Theta^\star_{(N)} - \widehat{S}(1)\big] \\
&+ [M(P(1))]^{-1}[(\lambda+1)(\Theta^\star - \Theta^\star_{(N)}) + \widehat{S}(1) - S(1)]
\end{aligned}$$

satisfies

$$\begin{aligned}
\big|\boldsymbol{\alpha}(0) - \widehat{\boldsymbol{\alpha}}(0)\big| \le& C\cdot\Big\{\bar{N}N^{3/2}\max_{1\le i,k,l\le N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| \\
&+ \bar{N}N^{1/2}\max_{1\le k,l\le N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| + |\Theta^\star - \Theta^\star_{(N)}|\Big\},
\end{aligned}$$

due to (89), (93), Lemma B.1, Corollary B.1, and Remark F.2. Assume by induction for all $0 \le u \le t-1$, $\boldsymbol{\alpha}(u-1) - \widehat{\boldsymbol{\alpha}}(u-1)$ satisfies

$$\begin{aligned}
\big|\boldsymbol{\alpha}(u-1) - \widehat{\boldsymbol{\alpha}}(u-1)\big| \le& C\cdot\Big\{\bar{N}N^{3/2}\max_{1\le i,k,l\le N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| \\
&+ \bar{N}N^{1/2}\max_{1\le k,l\le N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| + |\Theta^\star - \Theta^\star_{(N)}|\Big\}.
\end{aligned}$$

By (98), we have

$$\begin{aligned}
&|\boldsymbol{\alpha}(t) - \widehat{\boldsymbol{\alpha}}(t)| \\
\le&\big|(M(P(t+1)))^{-1} - (M(\widehat{P}(t+1)))^{-1}\big|\cdot\Big[\sum_{u=0}^{t-1}|\boldsymbol{\alpha}(u)| + |(\lambda+1)\Theta^\star| + |S(t+1)|\Big] \\
&+ \big|(M(\widehat{P}(t+1)))^{-1} - I_{Np}\big|\sum_{u=0}^{t-1}|\boldsymbol{\alpha}(u) - \widehat{\boldsymbol{\alpha}}(u)| \\
&+ |(M(\widehat{P}(t+1)))^{-1}|\cdot|S(t+1) - \widehat{S}(t+1)|, \\
&+ \big|[M(\widehat{P}(t+1))]^{-1}\big|\cdot|\lambda|\cdot\big|\Theta^\star - \Theta^\star_{(N)}\big|.
\end{aligned}$$

By the induction hypothesis, (89), (93), and Remark F.2, the above inequality implies that

$$\begin{aligned}
|\boldsymbol{\alpha}(t) - \widehat{\boldsymbol{\alpha}}(t)| \le& C\cdot\Big\{\bar{N}N^{3/2}\max_{1\le i,k,l\le N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| \\
&+ \bar{N}N^{1/2}\max_{1\le k,l\le N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| + |\Theta^\star - \Theta^\star_{(N)}|\Big\}.
\end{aligned} \tag{99}$$

We have shown by induction that (99) holds for all $t = 0, 1, ..., T-1$, and therefore (97) follows. □

Proposition F.2. *Under the hypothesis of Proposition F.1, the costs (4) under the controls (45) and (18), respectively, satisfy the estimate*

$$\begin{aligned}&\left|L^\star(\boldsymbol{\alpha};\mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}};\mathcal{D})\right| \\ &\leq C\cdot\Big\{\bar{N}N^3 \max_{1\leq i,k,l\leq N}\Big|\frac{1}{N^3} - w_k^\star w_l^\star w_i^\star\Big| + \bar{N}N^{3/2}\max_{1\leq k,l\leq N}\Big|\frac{1}{N^2} - w_k^\star w_l^\star\Big| \\ &\quad + \bar{N}\max_{1\leq i\leq N}\Big|\frac{1}{N} - w_i^\star\Big| + N|\Theta^\star - \Theta^\star_{(N)}|\Big\},\end{aligned} \tag{100}$$

where $C > 0$ is a constant depending on λ, β, $\Theta^\star$, $(w_1^\star, ..., w_N^\star)$, K_x, K_y, and T.

Proof of Proposition F.2. Let $L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})$, $L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D})$, and $L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D})$ be as defined in Lemmas F.3 and F.4. The difference $L^\star(\boldsymbol{\alpha};\mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}};\mathcal{D})$ can be decomposed as

$$\begin{aligned}&L^\star(\boldsymbol{\alpha};\mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}};\mathcal{D}) \\ &= L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) + L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D}),\end{aligned}$$

and satisfies

$$\begin{aligned}&|L^\star(\boldsymbol{\alpha};\mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}};\mathcal{D})| \\ &\leq |L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})| + |L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D})|.\end{aligned}$$

The desired estimate (100) then follows from the above inequality and (101) and (103). □

Lemma F.3. *Let $L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D})$ be (4) with information sharing weights $w^\star = (w_1^\star, ..., w_N^\star)$ and $\Theta(t+1) - \Theta(t) = \widehat{\boldsymbol{\alpha}}(t)$, and let $L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})$ be (4) with information sharing weights $\overline{1/N} = (1/N, ..., 1/N)$ and $\Theta(t+1) - \Theta(t) = \widehat{\boldsymbol{\alpha}}(t)$. Then, it satisfies*

$$|L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D})| \leq C\cdot\Big\{\bar{N}\max_{1\leq i\leq N}\Big|\frac{1}{N} - w_i^\star\Big| + N^{1/2}|\Theta^\star - \Theta^\star_{(N)}|\Big\}, \tag{101}$$

where $C > 0$ is a constant depending on λ, β, $\Theta^\star$, $(w_1^\star, \cdots, w_N^\star)$, K_x, K_y, and T.

Proof. By (4) and (52), we have

$$\begin{aligned}L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; w^\star, \mathcal{D}) = &\sum_{t=0}^{T-1}\left(|\Theta(t+1) - \Theta^*|^2 - |\Theta(t+1) - \Theta^*_{(N)}|^2\right) \\ &+ l(\Theta(T), \overline{1/N}; \mathcal{D}) - l(\Theta(T), w^\star; \mathcal{D}).\end{aligned}$$

We have that

$$\begin{aligned}&\Big|\sum_{t=0}^{T-1}\left(|\Theta(t+1) - \Theta^*|^2 - |\Theta(t+1) - \Theta^*_{(N)}|^2\right)\Big| \\ =&\Big|\sum_{t=0}^{T-1}(\Theta^\star_{(N)} - \Theta^\star)^\top(2\Theta(t+1) - \Theta^\star - \Theta^\star_{(N)})\Big| \\ \leq&\sum_{t=0}^{T-1}|\Theta^\star_{(N)} - \Theta^\star|\cdot|2\Theta(t+1) - \Theta^\star - \Theta^\star_{(N)}| \leq CN^{1/2}\cdot|\Theta^\star_{(N)} - \Theta^\star|,\end{aligned}$$

and

$$
\begin{aligned}
&l(\Theta(T), \overline{1/N}; \mathcal{D}) - l(\Theta(T), w^\star; \mathcal{D}) \\
&= \sum_{i=1}^{N} w_i^\star \sum_{j=1}^{|\mathcal{D}_i|} \Big[u_j^{i\top} \sum_{k=1}^{N} w_k^\star \theta_k - y_j^i \Big]^2 - \sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{|\mathcal{D}_i|} \Big[u_j^{i\top} \sum_{k=1}^{N} \frac{1}{N} \theta_k - y_j^i \Big]^2 \\
&= \sum_{i=1}^{N} (w_i^\star - 1/N) \sum_{j=1}^{|\mathcal{D}_i|} \Big[u_j^{i\top} \sum_{k=1}^{N} w_k^\star \theta_k - y_j^i \Big]^2 \\
&\quad + \sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{|\mathcal{D}_i|} \Big[u_j^{i\top} \sum_{k=1}^{N} (w_k^\star - 1/N) \theta_k \Big] \Big[u_j^{i\top} \sum_{k=1}^{N} (w_k^\star + 1/N) \theta_k - 2 y_j^i \Big]^2,
\end{aligned}
$$

which implies that

$$
|l(\Theta(T), \overline{1/N}; \mathcal{D}) - l(\Theta(T), w^\star; \mathcal{D})| \le C \bar{N} \max_{1 \le i \le N} |1/N - w_i^\star|. \tag{102}
$$

Estimate (101) then follows from (102) and (102). □

Lemma F.4. *Let $L^\star(\alpha; w^\star, \mathcal{D})$ be (4) with information sharing weights $w^\star = (w_1^\star, ..., w_N^\star)$ and $\Theta(t+1) - \Theta(t) = \boldsymbol{\alpha}(t)$, and let $L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})$ be (4) with information sharing weights $\overline{1/N} = (1/N, ..., 1/N)$ and $\Theta(t+1) - \Theta(t) = \widehat{\boldsymbol{\alpha}}(t)$. Then, it satisfies*

$$
\begin{aligned}
&|L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})| \\
&\le C \cdot \Big\{ \bar{N} N^3 \cdot \max_{1 \le i,k,l \le N} \Big| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \Big| \\
&\quad + \bar{N} N^{3/2} \max_{1 \le k,l \le N} \Big| \frac{1}{N^2} - w_k^\star w_l^\star \Big| + \bar{N} \cdot \max_{1 \le i \le N} \Big| \frac{1}{N} - w_i^\star \Big| + N |\Theta^\star - \Theta^\star_{(N)}| \Big\},
\end{aligned} \tag{103}
$$

where C is a constant depending on λ, β, $\Theta^\star$, $(w_1^\star, \cdots, w_N^\star)$, K_x, K_y, and T.

Proof. By the similar argument for (53), we have

$$
L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) = \Theta^{\star\top} \widehat{P}(0) \Theta^\star + 2 \widehat{S}(0)^\top \Theta^\star + \widehat{r}(0). \tag{104}
$$

By (53) and (104), the difference $L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D})$ can be written as

$$
\begin{aligned}
&L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) \\
&= \Theta^{\star\top} (P(0) - \widehat{P}(0)) \Theta^\star + 2 (S(0) - \widehat{S}(0))^\top \Theta^\star + r(0) - \widehat{r}(0),
\end{aligned}
$$

and by Lemma B.1, the difference satisfies

$$
\begin{aligned}
\big| L^\star(\boldsymbol{\alpha}; w^\star, \mathcal{D}) - L^\star(\widehat{\boldsymbol{\alpha}}; \overline{1/N}, \mathcal{D}) \big| \le & |\Theta^\star|^2 \cdot |P(0) - \widehat{P}(0)| + 2 |S(0) - \widehat{S}(0)| \cdot |\Theta^\star| \\
& + |r(0) - \widehat{r}(0)|.
\end{aligned} \tag{105}
$$

We claim that $r(0) - \widehat{r}(0)$ has the estimate

$$
\begin{aligned}
&|r(0) - \widehat{r}(0)| \\
&\le C \cdot \Big\{ \bar{N} N^3 \cdot \max_{1 \le i,k,l \le N} \Big| \frac{1}{N^3} - w_k^\star w_l^\star w_i^\star \Big| + \bar{N} N^{3/2} \max_{1 \le k,l \le N} \Big| \frac{1}{N^2} - w_k^\star w_l^\star \Big| \\
&\quad + \bar{N} \cdot \max_{1 \le i \le N} \Big| \frac{1}{N} - w_i^\star \Big| + N \cdot |\Theta^\star - \Theta^\star_{(N)}| \Big\},
\end{aligned} \tag{106}
$$

where C is a constant depending on λ, β, $\Theta^\star$, $(w_1^\star, \cdots, w_N^\star)$, K_x, K_y, and T. Then, the desired estimate (103) is established by (105), (88), (93), and (106).

Now we prove the claim (106). From (56) and (52), we obtain the equation that $\widehat{r}$ satisfies on $0 \le t \le T$:

$$\begin{cases} \widehat{r}(t) = -(\widehat{S}(t+1) - \lambda\Theta^\star_{(N)})^\top [(\lambda+\beta) I_{Np} + \widehat{P}(t+1)]^{-1} \cdot (\widehat{S}(t+1) - \lambda\Theta^\star_{(N)}) \\ \qquad\qquad +\lambda \|\Theta^\star_{(N)}\|_2^2 + \widehat{r}(t+1), \\ \widehat{r}(T) = \sum_{i=1}^N (1/N) \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i|^2. \end{cases} \tag{107}$$

Then, $r - \widehat{r}$ satisfies the equation on $0 \le t \le T$,

$$\begin{aligned} & r(t) - \widehat{r}(t) \\ =& r(t+1) - \widehat{r}(t+1) + \lambda(\Theta^\star - \Theta^\star_{(N)})(\Theta^\star + \Theta^\star_{(N)}) \\ & - (S(t+1) - \lambda\Theta^\star)^\top \left[(M(P(t+1)))^{-1} - (M(\widehat{P}(t+1)))^{-1} \right] (S(t+1) - \lambda\Theta^\star) \\ & - \left[S(t+1) - \widehat{S}(t+1) - \lambda(\Theta^\star - \Theta^\star_{(N)}) \right] (M(\widehat{P}(t+1)))^{-1} \\ & \times \left[S(t+1) + \widehat{S}(t+1) - \lambda(\Theta^\star + \Theta^\star_{(N)}) \right], \end{aligned} \tag{108}$$

$$r(T) - \widehat{r}(T) = \sum_{i=1}^N (w_i^\star - 1/N) \sum_{j=1}^{|\mathcal{D}_i|} |y_j^i|^2.$$

Inductively, we have

$$\begin{aligned} & r(0) - \widehat{r}(0) \\ &= r(T) - \widehat{r}(T) + T\lambda(\Theta^\star - \Theta^\star_{(N)})(\Theta^\star + \Theta^\star_{(N)}) \\ & - \sum_{t=1}^T (S(t) - \lambda\Theta^\star)^\top \left[(M(P(t)))^{-1} - (M(\widehat{P}(t)))^{-1} \right] (S(t) - \lambda\Theta^\star) \\ & - \sum_{t=1}^T \left[S(t) - \widehat{S}(t) - \lambda(\Theta^\star - \Theta^\star_{(N)}) \right] (M(\widehat{P}(t)))^{-1} \left[S(t) + \widehat{S}(t) - \lambda(\Theta^\star + \Theta^\star_{(N)}) \right]. \end{aligned}$$

By Lemma B.1, we have

$$\begin{aligned} & |r(0) - \widehat{r}(0)| \\ & \le |r(T) - \widehat{r}(T)| + T\lambda |\Theta^\star - \Theta^\star_{(N)}| \cdot |\Theta^\star + \Theta^\star_{(N)}| \\ & \quad + \sum_{t=1}^T \left| (M(P(t)))^{-1} - (M(\widehat{P}(t)))^{-1} \right| \cdot \left| S(t) - \lambda\Theta^\star \right|^2 \\ & \quad + \sum_{t=1}^T \left[|S(t) - \widehat{S}(t)| + \lambda |\Theta^\star - \Theta^\star_{(N)}| \right] \cdot \left| (M(\widehat{P}(t)))^{-1} \right| \\ & \quad \times \left| S(t) + \widehat{S}(t) - \lambda(\Theta^\star + \Theta^\star_{(N)}) \right|. \end{aligned} \tag{109}$$

Then, estimate (106) follows from (109), (89), (93), and Remark F.2. □

Proof of Theorem 3.6. Under Assumption 3.1, the existence of the regret-optimal algorithm $\theta^\star$ is implied by Theorem B.1. The first claim follows from Proposition F.1 upon noting that the constant "C_w" therein is bounded as a function of w, whence we take the constant C hidden by $\mathcal{O}$ to be $C \overset{\text{def.}}{=} \sup_{w \in [0,1]^N \, \sum_{i=1}^N w_i = 1} C_w$. Likewise,

the second statement follows directly from Corollary B.2 upon defining the constant hidden by $\mathcal{O}$ as before. □

Further investigation of the "symmetrized system" (19) reveals a symmetric form of $\widehat{P}$, which suggests that $\widehat{P}$ can be decomposed into homogeneous submatrices of lower dimensions. Specifically, if we decompose the high-dimensional $Np \times Np$ matrix $\widehat{P}$ into $N \times N$ submatrices of dimensions $p \times p$ each, and either exchange two submatrices symmetrically positioned with respect to the diagonal of $\widehat{P}$ or exchange two submatrices on the diagonal, the system (19) is invariant. This suggests that all the submatrices on the diagonal are homogeneous, and all the off-diagonal submatrices are homogeneous.

Proposition F.3 (Symmetries in $\widehat{P}$)**.** *The solution of* (19) *has the submatrix decomposition*

$$\widehat{P} = \begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_2 \\ \pi_2 & \pi_1 & \cdots & \pi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \pi_2 & \pi_2 & \cdots & \pi_1 \end{bmatrix} \in \mathbb{R}^{Np \times Np}, \quad \pi_1 \in \mathbb{R}^{p \times p}, \quad \pi_2 \in \mathbb{R}^{p \times p}, \tag{110}$$

and the submatrices π_1 and π_2 are uniquely determined by the system

$$\begin{cases} \pi_1(t) = \beta I_p - \beta^2 \gamma_1(t+1), \\ \pi_2(t) = -\beta^2 \gamma_2(t+1), \quad 1 \le t \le T-1, \\ \pi_1(T) = \pi_2(T) = (1/N^3) \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top}, \end{cases} \tag{111}$$

where $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ are defined as (15) *and* (16).

We argue as in [37, Lemmata 1] in deriving the following submatrix decomposition of $\widehat{P}$.

Proof of Proposition F.3. We decompose the $Np \times Np$ identity matrix into $N \times N$ submatrices $I = (I_{ij})_{1 \le i,j \le N}$ with each $I_{ij} \in \mathbb{R}^{p \times p}$. For each $1 \le i, j \le N$, by E_{ij}we denote the $Np \times Np$ elementary matrix by exchanging the ith and jth rows of submatrices of the $Np \times Np$ identity matrix $I = (I_{ij})_{1 \le i,j \le N}$. Note that $E_{ij} = E_{ij}^{-1} = E_{ij}^T$. Multiplying both sides of (47) from the left by E_{ij} and from the right by E_{ij}, we obtain

$$\begin{cases} E_{ij}\widehat{P}(t)E_{ij} = \beta I_{Np} - \beta^2 [(\lambda+\beta) I_{Np} + E_{ij}\widehat{P}(t+1)E_{ij}]^{-1}, \\ E_{ij}\widehat{P}(T)E_{ij} \\ = [(1/N)I_p, \cdots, (1/N)I_p]^\top \Big(\sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{|\mathcal{D}_i|} u_j^i u_j^{i\top} \Big) [(1/N)I_p, \cdots, (1/N)I_p]. \end{cases} \tag{112}$$

Comparing (112) and (19) reveals that $\widehat{P}(t)$ and $E_{ij}\widehat{P}(t)E_{ij}$ satisfy the same ODE for all $1 \le i, j \le N$. This observation implies that

$$\widehat{P}_{ii} = \widehat{P}_{jj}, \quad \widehat{P}_{ij} = \widehat{P}_{ji}, \quad \widehat{P}_{ik} = \widehat{P}_{jk}, \quad \widehat{P}_{ki} = \widehat{P}_{kj}, \quad \forall 1 \le i, j \le N, \quad \forall k \ne i, j,$$

and the first assertion of the proposition is proved.

Since $\widehat{P}$ takes the form (110), it is straightforward to verify that $[(\lambda+\beta)I_{Np}+\widehat{P}]^{-1}$ takes the form

$$[(\lambda+\beta)I_{Np}+\widehat{P}]^{-1}=\begin{bmatrix}\gamma_1 & \gamma_2 & \cdots & \gamma_2\\ \gamma_2 & \gamma_1 & \cdots & \gamma_2\\ \vdots & \vdots & \ddots & \vdots\\ \gamma_2 & \gamma_2 & \cdots & \gamma_1\end{bmatrix}, \tag{113}$$

with γ_1 and γ_2 given by (15) and (16).

We substitute (110) and (113) into (19) to obtain equations (111). □

Upon substituting $\widehat{P}$ into (20), we find that the $Np\times 1$ matrix $\widehat{S}$ can be decomposed into homogeneous submatrices of dimensions $p\times 1$ each.

Proposition F.4. *The solution of* (20) *can be decomposed into submatrices*

$$\widehat{S}=(\pi_3^\top,\cdots,\pi_3^\top)^\top\in\mathbb{R}^{Np\times 1},\quad \pi_3\in\mathbb{R}^{p\times 1}, \tag{114}$$

and π_3 *satisfies*

$$\begin{cases}\pi_3(t)=\beta\big[\gamma_1(t+1)+(N-1)\gamma_2(t+1)\big](\pi_3(t+1)-\lambda\theta^\star_{(N)}),\\ \pi_3(T)=-(1/N^2)\sum_{i=1}^N\sum_{j=1}^{|\mathcal{D}_i|}u_j^iy_j^i.\end{cases} \tag{115}$$

Proof of Proposition F.4. We multiply both sides of (20) from the left by E_{ij} defined in the proof of Proposition F.3,

$$\begin{aligned}E_{ij}\widehat{S}(t)=&E_{ij}\beta[(\lambda+\beta)I_{Np}+\widehat{P}(t+1)]^{-1}(\widehat{S}(t+1)-\lambda\Theta^\star_{(N)})\\ =&E_{ij}\beta[(\lambda+\beta)I_{Np}+\widehat{P}(t+1)]^{-1}E_{ij}E_{ij}(\widehat{S}(t+1)-\lambda\Theta^\star_{(N)})\\ =&\beta[(\lambda+\beta)I_{Np}+E_{ij}\widehat{P}(t+1)E_{ij}]^{-1}(E_{ij}\widehat{S}(t+1)-\lambda\Theta^\star_{(N)})\end{aligned} \tag{116}$$

and

$$\begin{aligned}E_{ij}\widehat{S}(T)=&-E_{ij}[(1/N)I_p,\cdots,(1/N)I_p]^\top\sum_{i=1}^N\frac{1}{N}\sum_{j=1}^{|\mathcal{D}_i|}u_j^iy_j^i\\ =&-[(1/N)I_p,\cdots,(1/N)I_p]^\top\sum_{i=1}^N\frac{1}{N}\sum_{j=1}^{|\mathcal{D}_i|}u_j^iy_j^i.\end{aligned} \tag{117}$$

From (116) and (117), we have that $E_{ij}\widehat{S}$ also satisfies (20) for all $1\le i,j\le N$. This proves that all submatrices of $\widehat{S}=(\widehat{S}_i)_{1\le i\le N}$ are equal on $0\le t\le T$. We further substitute (110), (113), and $\widehat{S}=[\pi_3^\top,\cdots,\pi_3^\top]^\top$ into (20) to obtain (115). □

Corollary F.1. *Fix a dataset* $\mathcal{D}$*, and assume that* (64) *holds. Then, the computational complexity of computing the sequence* $\big(\theta(t)\big)_{t=0}^T$ *by* (18) *is* $\mathcal{O}(Tp^2\max\{N, p^{0.373}\})$.

Proof of Corollary F.1. **Complexity of computing** $\pi_1(T)$**,** $\pi_2(T)$**, and** $\pi_3(T)$ **is** $O(N^2p^2)$**.**

Computing $u_j^iu_j^{i\top}$ has complexity $O(p^2)$ by Lemma B.3-(iii), and computing the sum $\sum_{i=1}^N\sum_{j=1}^{|\mathcal{D}_i|}u_j^iu_j^{i\top}$ has a complexity of $O(N^2p^2)$ by Lemma B.3-(i). Moreover, multiplying the $p\times p$ matrix $\sum_{i=1}^N\sum_{j=1}^{|\mathcal{D}_i|}u_j^iu_j^{i\top}$ by the scalar $1/N^3$ has complexity $O(p^2)$. Hence, computing $\pi_1(T)$ and $\pi_2(T)$ has complexity $\mathcal{O}(N^2p^2)$. Computing

the product $u_j^i y_j^i$ has a complexity of $O(p)$ by Lemma B.3-(iii), and summing $\sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i$ has complexity $O(N^2 p)$ by Lemma B.3-(i). Multiplying the $p \times 1$ vector $\sum_{i=1}^N \sum_{j=1}^{|\mathcal{D}_i|} u_j^i y_j^i$ by the scalar $1/N^2$ has a complexity of $O(p)$. Therefore, computing $\pi_3(T)$ has complexity $O(N^2 p)$, and the complexity of computing $\pi_1(T)$, $\pi_2(T)$, and $\pi_3(T)$ is $\mathcal{O}(N^2 p)$.

Complexity of computing $\gamma_1(T)$ and $\gamma_2(T)$ is $O(p^{2.373})$.

With $\pi_1(T)$ and $\pi_2(T)$ obtained, we can compute $\gamma_1(T)$ and $\gamma_2(T)$ by (15) and (16). From (15) and (16), computing $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ from $\pi_1(\cdot)$ and $\pi_2(\cdot)$ involves multiplication and addition of $p \times p$ matrices, scalar multiplication of $p \times p$-dimensional matrices, multiplication of a $p \times p$ matrix and a $p \times 1$ vector, and inversion of a $p \times p$ matrix. By Lemma B.3, multiplication of two $p \times p$ matrices has the dominant complexity of $\mathcal{O}(p^{2.373})$ among the above matrix operations. Moreover, inverting a $p \times p$ matrix has complexity $\mathcal{O}(p^{2,373})$. Hence, computing $\gamma_1(T)$ and $\gamma_2(T)$ has a complexity of $O(p^{2.373})$.

Complexity of computing $\pi_1(t)$, $\pi_2(t)$, $\pi_3(t)$, $\gamma_1(t)$, and $\gamma_2(t)$ for $t = T-1$, $T-2$, ..., 1 has complexity $\mathcal{O}(Tp^{2.373})$.

Similar to computing $\gamma_1(T)$ and $\gamma_2(T)$, computing $\gamma_1(t)$ and $\gamma_2(t)$ from $\pi_1(t+1)$ and $\pi_2(t+1)$ has complexity $\mathcal{O}(p^{2.373})$. According to (111), computing $\pi_1(t)$ and $\pi_2(t)$ from $\gamma_1(t+1)$ and $\gamma_2(t+1)$ involves scalar multiplication and addition along the diagonal of $p \times p$ matrices, and therefore has complexity $O(p^2)$ by Lemma B.3-(i) and (ii). According to (115), computing $\pi_3(t)$ from $\gamma_1(t+1)$, $\gamma_2(t+1)$, and $\pi_3(t+1)$ involves scalar multiplication and addition of $p \times p$ matrices, scalar multiplication and addition of $p \times 1$ matrices, and multiplication of a $p \times p$ matrix with a $p \times 1$ matrix, respectively. By Lemma B.3, the dominant complexity of these operations is $\mathcal{O}(p^2)$. Summing up the above argument, computing $\pi_1(t)$, $\pi_2(t)$, $\pi_3(t)$, $\gamma_1(t)$, and $\gamma_2(t)$ for each t has a complexity of $\mathcal{O}(p^{2.373})$, and therefore has a complexity of $\mathcal{O}(Tp^{2.373})$ for all $t = T-1$, $T-2$,..., 1.

The cost of computing each $\Theta(t)$ is $O(Tp^2 \max\{N, p^{0.373}\})$.

By (5), the cost of updating from $\Theta(t)$ to $\Theta(t+1)$ is equal to the complexity of computing $\widehat{\boldsymbol{\alpha}}(t) = (\widehat{\alpha}_1^\top(t), \cdots \widehat{\alpha}_N^\top(t))^\top$ for $t = 0, 1, ..., T-1$.

By (17), $\widehat{\alpha}_i(t)$ for all $i = 1$, ..., N share the common term

$$
\begin{aligned}
&- \big\{\gamma_2(t+1)[\lambda I_p + \pi_1(t+1) + (N-2)\pi_2(t+1)] \\
&+ \gamma_1(t+1)\pi_2(t+1)\big\} \sum_{k=1}^N \theta_k(t) \\
&- \big[\gamma_1(t+1) + (N-1)\gamma_2(t+1)\big](\pi_3(t+1) - \lambda\theta^\star_{(N)}),
\end{aligned}
$$

and computing the common term involves matrix multiplication, addition, and scalar multiplication of $p \times p$ matrices, matrix addition and scalar multiplication of $p \times 1$ matrices, and multiplication of a $p \times p$ matrix and a $p \times 1$ matrix, respectively. By Lemma B.3, these operations have complexity $O(Np + p^{2.373})$, where Np is due to adding the N p-dimensional vectors $\sum_{k=1}^N \theta_k(t)$, and $p^{2.373}$ is due to multiplication of two $p \times p$ matrices by a CW-like algorithm.

Computing the individual terms

$$
-(\gamma_1(t+1) - \gamma_2(t+1))(\lambda I_p + \pi_1(t+1) - \pi_2(t+1))\theta_i(t)
$$

for $\widehat{\alpha}_i(t)$, $i = 1, ..., N$ involves addition, scalar multiplication, matrix multiplication of $p \times p$ matrices, and multiplication of a $p \times p$ matrix and a $p \times 1$ matrix. By Lemma B.3, these operations have complexity of $\mathcal{O}(p^{2.373} + Np)$, where $p^{2.373}$ is due to multiplication of two $p \times p$ matrices by a CW-like algorithm, and Np is due to multiplying a $p \times p$ matrix by the p-dimensional vector $\theta_i(t)$ for $i = 1, ..., N$. Hence, the complexity of computing $\widehat{\boldsymbol{\alpha}}(t)$ for $t = 0, 1, ..., T-1$ is $\mathcal{O}(Tp^2 \max\{N, p^{0.373}\})$.

The complexity of computing $(\Theta(t))_{t=0}^{T}$ is $\mathcal{O}(Tp^2 \max\{N, p^{0.373}\})$.

Looking over all computations involved, we deduce that the complexity of computing the sequence $(\Theta(t))_{t=1}^{T}$ starting from $\Theta(0) = \Theta^\star$ has a complexity of $\mathcal{O}(Tp^2 \max\{N, p^{0.373}\})$. □

We may now deduce Theorem 3.7 and Theorem 3.8.

Proof of Theorem 3.7. The result directly follows from Theorem 3.4 and Corollary F.1. □

Proof of Theorem 3.8. Direct consequence of Propositions F.3 and F.4. □

REFERENCES

[1] T. Ahrendt, Fast computations of the exponential function, in *Christoph Meinel and Sophie Tison, editors, STACS 99*, Berlin, Heidelberg, Springer Berlin Heidelberg, (1999), 302-312.

[2] P. Alquier, User-friendly introduction to PAC-Bayes bounds, arXiv preprint, 2025. arXiv:2110.11216.

[3] A. A. Amini, Spectrally-truncated kernel ridge regression and its free lunch, *Electron. J. Stat.*, **15** (2021), 3743-3761.

[4] A. A. Amini, R. Baumgartner and D. Feng, Target alignment in truncated kernel ridge regression, arXiv preprint, 2022. arXiv:2206.14255.

[5] A. Argyriou, T. Evgeniou and M. Pontil, Multi-task feature learning, *Advances in Neural Information Processing Systems*, **19** (2006).

[6] L. M. B. Arias and F. Roldán, Four-operator splitting via a forward-backward-half-forward algorithm with line search, *J. Optim. Theory Appl.*, **195** (2022), 205-225.

[7] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory, 2nd edition*, Society for Industrial and Applied Mathematics, 1999.

[8] J. Baxter, A model of inductive bias learning, *J. Artificial Intelligence Res.*, **12** (2000) 149-198.

[9] S. Becker, P. Cheridito and A. Jentzen, Deep optimal stopping, *Journal of Machine Learning Research*, **20** (2019), Paper No. 74, 25 pp.

[10] S. Becker, P. Cheridito, A. Jentzen and T. Welti, Solving high-dimensional optimal stopping problems using deep learning, *European Journal of Applied Mathematics*, **32** (2021), 470-514.

[11] S. Becker, P. Cheridito, A. Jentzen and T. Welti, Solving high-dimensional optimal stopping problems using deep learning, *European J. Appl. Math.*, **32** (2021), 470-514.

[12] D. P. Bertsekas, *Convex Optimization Algorithms*, Athena Scientific, 2015.

[13] H. Cao, H. Gu, X. Guo and M. Rosenbaum, Transfer learning for portfolio optimization, arXiv preprint, 2023. arXiv:2307.13546.

[14] R. Carmona, F. Delarue, et al., *Probabilistic Theory of Mean Field Games with Applications I-II*, Springer, 2018.

[15] P. Casgrain, A latent variational framework for stochastic optimization, *Advances in Neural Information Processing Systems*, **32** (2019).

[16] P. Casgrain, A latent variational framework for stochastic optimization, in *H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems*, Curran Associates, Inc., **32** (2019).

[17] P. Casgrain and A. Kratsios, Optimizing optimizers: Regret-optimal gradient descent algorithms, In *Conference on Learning Theory*, PMLR, (2021), 883-926.

[18] T. S. Cheng, A. Lucchi, A. Kratsios, D. Belius and I. Dokmanić, A theoretical analysis of the test error of finite-rank kernel ridge regression, *Neural Information Processing Systems*, 2023.

[19] P. L. Combettes and J.-C. Pesquet, Stochastic approximations and perturbations in forward-backward splitting for monotone operators, *Pure Appl. Funct. Anal.*, **1** (2016), 13-37.

[20] E. M. Compagnoni, A. Scampicchio, L. Biggio, A. Orvieto, T. Hofmann and J. Teichmann, On the effectiveness of randomized signatures as reservoir for learning rough dynamics, in *2023 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2023), 1-8.
[21] C. Cuchiero, L. Gonon, L. Grigoryeva, J.-P. Ortega and J. Teichmann, Discrete-time signatures and randomness in reservoir computing, *IEEE Transactions on Neural Networks and Learning Systems*, **33** (2022), 6321-6330.
[22] J. Duchi, E. Hazan and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research*, **12** (2011), 2121-2159.
[23] J. Fliege, A. I. F. Vaz and L. N. Vicente, Complexity of gradient descent for multiobjective optimization, *Optimization Methods and Software*, **34** (2019), 949-959.
[24] F. L. Gall, Powers of tensors and fast matrix multiplication, in *ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ACM, New York, (2014), 296-303.
[25] B. Ghorbani, S. Mei, T. Misiakiewicz and A. Montanari, Linearized two-layers neural networks in high dimension, *The Annals of Statistics*, **49** (2021), 1029-1054.
[26] L. Gonon, Random feature neural networks learn Black-Scholes type PDEs without curse of dimensionality, *Journal of Machine Learning Research*, **24** (2023), Paper No. 189, 51 pp.
[27] L. Gonon, L. Grigoryeva and J.-P. Ortega, Risk bounds for reservoir computing, *Journal of Machine Learning Research*, **21** (2020), Paper No. 240, 61 pp.
[28] L. Gonon, L. Grigoryeva and J.-P. Ortega, Approximation bounds for random neural networks and reservoir systems, *Ann. Appl. Probab.*, **33** (2023), 28-69.
[29] L. Grigoryeva and J.-P. Ortega, Echo state networks are universal, *Neural Networks*, **108** (2018), 495-508.
[30] E. Hazan, et al., Introduction to online convex optimization, *Foundations and Trends in Optimization*, **2** (2016), 157-325.
[31] J. Heiss, J. Teichmann and H. Wutte, How infinitely wide neural networks benefit from multi-task learning-an exact macroscopic characterization, 2022.
[32] C. Herrera, F. Krach, P. Ruyssen and J. Teichmann, Optimal stopping via randomized neural networks, *Front. Math. Finance*, **3** (2024), 31-77.
[33] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2 edition, 2013.
[34] S. Hou, P. Kassraie, A. Kratsios, A. Krausen and J. Rothfuss, Instance-dependent generalization bounds via optimal transport, *Journal of Machine Learning Research*, **24** (2023), 16815-16865.
[35] J. Howard and S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint, 2018. arXiv:1801.06146.
[36] R. Hu, Deep learning for ranking response surfaces with applications to optimal stopping problems, *Quant. Finance*, **20** (2020), 1567-1581.
[37] M. Huang and X. Yang, Linear quadratic mean field social optimization: Asymptotic solvability and decentralized control, *Applied Mathematics & Optimization*, **84** (2021), 1969-2010.
[38] G. Jeong and H. Y. Kim, Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning, *Expert Systems with Applications*, **117** (2019), 125-138.
[39] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo and Y. Wu, Leveraging weakly supervised data to improve end-to-end speech-to-text translation, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, (2019), 7180-7184.
[40] A. Khaled, K. Mishchenko and P. Richtarik, Tighter theory for local sgd on identical and heterogeneous data, in *Silvia Chiappa and Roberto Calandra, editors, Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, *Proceedings of Machine Learning Research*, PMLR, **108** (2020), 4519-4529.
[41] G. Kimeldorf and G. Wahba, Some results on Tchebycheffian spline functions, *J. Math. Anal. Appl.*, **33** (1971), 82-95.
[42] G. S. Kimeldorf and G. Wahba, A correspondence between Bayesian estimation on stochastic processes and smoothing by splines, *Ann. Math. Statist.*, **41** (1970), 495-502.
[43] M. Kraus and S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, *Decision Support Systems*, **104** (2017), 38-48.
[44] I. Kuzborskij and F. Orabona, Stability and hypothesis transfer learning, in *Sanjoy Dasgupta and David McAllester, editors, Proceedings of the 30th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, Atlanta, Georgia, USA, PMLR, **28** (2013), 942-950.

[45] D. Kwon, J. Park and S. Hong, Tighter regret analysis and optimization of online federated learning, arXiv preprint, 2023. arXiv:2205.06491.
[46] P. D. Lax, *Linear Algebra and its Applications*, Pure and Applied Mathematics (Hoboken), Wiley-Interscience, Hoboken, NJ, second edition, 2007.
[47] Q. Li, L. Chen, C. Tai and W. E, Maximum principle based algorithms for deep learning, *Journal of Machine Learning Research*, **18** (2018), Paper No. 165, 29 pp.
[48] Q. Li, C. Tai and W. E, Stochastic modified equations and adaptive stochastic gradient algorithms, in *International Conference on Machine Learning*, PMLR, **70** (2017), 2101-2110.
[49] H. Lin and M. Reimherr, Smoothness adaptive hypothesis transfer learning, in *International Conference on Machine Learning*, PMLR, **235** (2024), 30286-30316.
[50] M. Lukoševičius and H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Computer Science Review*, **3** (2009), 127-149.
[51] O. L. Mangasarian, Parallel gradient distribution in unconstrained optimization, *SIAM J. Control Optim.*, **33** (1995), 1916-1925.
[52] T. Manole and J. Niles-Weed, Sharp convergence rates for empirical optimal transport with smooth costs, *Ann. Appl. Probab.*, **34** (2024), 1108-1135.
[53] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Y. Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, PMLR, **54** (2017), 1273-1282.
[54] S. Mei and A. Montanari, The generalization error of random features regression: Precise asymptotics and the double descent curve, *Comm. Pure Appl. Math.*, **75** (2022), 667-766.
[55] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, 1994.
[56] K. L. Pavasovic, J. Rothfuss and A. Krause, Mars: Meta-learning as score matching in the function space, arXiv preprint, 2022. arXiv:2210.13319.
[57] G. Peskir and A. Shiryaev, *Optimal Stopping and Free-Boundary Problems*, Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2006.
[58] Z. Ren and Y. J. Lee, Cross-domain self-supervised multi-task feature learning using synthetic imagery, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 762-771.
[59] J. Rothfuss, V. Fortuin, M. Josifoski and A. Krause, PACOH: Bayes-optimal meta-learning with PAC-guarantees, in *Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, PMLR, **139** (2021), 9116-9126.
[60] J. Rothfuss, M. Josifoski, V. Fortuin and A. Krause, Scalable PAC-Bayesian meta-learning via the PAC-optimal hyper-posterior: From theory to practice, *Journal of Machine Learning Research*, **24** (2023), Paper No. 386, 62 pp.
[61] O. Sener and V. Koltun, Multi-task learning as multi-objective optimization, in *S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems*, *Curran Associates, Inc.*, **31** (2018).
[62] J. L. Snell, Applications of martingale system theorems, *Trans. Amer. Math. Soc.*, **73** (1952), 293-312.
[63] Y. Tian, Y. Gu and Y. Feng, Learning from similar linear representations: Adaptivity, minimaxity, and robustness, *Journal of Machine Learning Research*, **26** (2025), Paper No. 187, 125 pp.
[64] C. Villani, *Optimal Transport: Old and New*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, Springer-Verlag, Berlin, 2009.
[65] R. Wang, C. Hyndmann and A. Kratsios, The entropic measure transform, *Canad. J. Statist.*, **48** (2020), 97-129.
[66] X. Wang, J. B. Oliva, J. Schneider and B. Póczos, Nonparametric risk and stability analysis for multi-task learning problems, *IJCAI'16, AAAI Press*, (2016), 2146-2152.
[67] Y. Wei, F. Yang and M. J. Wainwright, Early stopping for kernel boosting algorithms: A general analysis with localized complexities, *IEEE Trans. Inform. Theory*, **65** (2019), 6685-6703.
[68] K. Weiss, T. M. Khoshgoftaar and D. Wang, A survey of transfer learning, *Journal of Big Data*, **3** (2016), 1-40.
[69] Y. Xue, X. Liao, L. Carin and B. Krishnapuram, Multi-task learning for classification with Dirichlet process priors, *Journal of Machine Learning Research*, **8** (2007), 35-63.

[70] Y. Yao, L. Rosasco and A. Caponnetto, On early stopping in gradient descent learning, *Constr. Approx.*, **26** (2007), 289-315.

[71] J. Yosinski, J. Clune, Y. Bengio and H. Lipson, How transferable are features in deep neural networks?, *Advances in Neural Information Processing Systems*, **27** (2014).

[72] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, (2014), 818-833.

[73] T. Zhang and B. Yu, Boosting with early stopping: Convergence and consistency, *Ann. Statist.*, **33** (2005), 1538-1579.

[74] J. Zhao, A. Lucchi, F. N. Proske, A. Orvieto and H. Kersting, Batch size selection by stochastic optimal control, in *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022.

Received January 24, 2025; revised November 17, 2025.