SIAM J. MATH. DATA SCI.
Vol. 7, No. 4, pp. 1594–1615

Nonlinear Meta-learning Can Guarantee Faster Rates*

Dimitri Meunier†, Zhu Li‡, Arthur Gretton†, and Samory Kpotufe§

Abstract. Many recent theoretical works on *meta-learning* aim to achieve guarantees in leveraging similar representational structures from related tasks towards simplifying a target task. The main aim of theoretical guarantees on the subject is to establish the extent to which convergence rates—in learning a common representation—*may scale with the number N of tasks* (as well as the number of samples per task). First steps in this setting demonstrate this property when both the shared representation amongst tasks and task-specific regression functions are linear. This linear setting readily reveals the benefits of aggregating tasks, e.g., via averaging arguments. In practice, however, the representation is often highly nonlinear, introducing nontrivial biases in each task that cannot easily be averaged out as in the linear case. In the present work, we derive theoretical guarantees for meta-learning with nonlinear representations. In particular, assuming the shared nonlinearity maps to an infinite dimensional reproducing kernel Hilbert space, we show that additional biases can be mitigated with careful regularization that leverages the smoothness of task-specific regression functions, yielding improved rates that scale with the number of tasks as desired.

Key words. kernel methods, subspace approximation, nonparametric statistics

MSC code. 62G08

DOI. 10.1137/24M1662977

1. Introduction.

Meta-learning refers colloquially to the problem of inferring a deeper internal structure—beyond a specific task at hand, e.g., a regression task—that may be leveraged towards speeding up other similar tasks. This arises for instance in practice with neural networks where, in pre-training, multiple apparently dissimilar tasks may be aggregated to learn a *representation* that enables *faster* training on unseen target tasks (i.e., requiring relatively fewer target data).

Notwithstanding the popularity of meta-learning in practice, the theoretical understanding and proper formalism for this setting is still in its early stages. We consider a common approach in the context of regression, which posits an unknown target-task function of the form $f(x) = g(\Gamma(x))$ and N unknown related task-functions of the form $f_i(x) = g_i(\Gamma(x)), i \in [N]$, i.e., all sharing a common but unknown *representation* $\Gamma(x)$; it is assumed that all *link*

*Received by the editors May 22, 2024; accepted for publication (in revised form) April 30, 2025; published electronically October 3, 2025.
https://doi.org/10.1137/24M1662977
Funding: The first, second, and third authors were supported by the Gatsby Charitable Foundation. The second author is also funded by Imperial College London through the Chapman Fellowship. The fourth author is thankful for support from NSF CNS-2334997, and a Sloan 2021 Fellowship over the bulk of this study.

†Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom. Equal contribution with Zhu Li (dimitri.meunier.21@ucl.ac.uk, arthur.gretton@gmail.com).
‡Department of Mathematics, Imperial College London, London, United Kingdom. Equal contribution with Dimitri Meunier (michael.lzy2013@gmail.com).
§Department of Statistics, Columbia University, New York, NY 10027 USA (skk2175@columbia.edu).

functions g and $\{g_i\}_{i=1}^N$ are *simpler*—for instance linear or at least lower-dimensional—than the corresponding regression functions f and $\{f_i\}_{i=1}^N$. As all these objects are a priori unknown, recent research has aimed to establish how the target regression problem may benefit from the N related tasks. In particular, if $\Gamma(x)$ may be approximated by some $\hat{\Gamma}(x)$ at a rate that scales with N (and the number n of samples per task), then presumably, the target regression function f may be subsequently learned as $\hat{g}(\hat{\Gamma}(x))$ at a faster rate commensurate with the *simplicity* of g.

Recent theoretical results [10, 21, 31, 37] have provided significant new insights in this area by considering an idealized linear setting where $x \in \mathbb{R}^d$, g and $\{g_i\}_{i=1}^N$ are linear functions in $\mathbb{R}^s (s \ll d)$, and $\Gamma(x)$ denotes a linear projection to $\mathbb{R}^s$. These results show that Γ can be learned at a rate of $\tilde{O}(\sqrt{ds/nN})$—under suitable subspace-distance measures, and where $\tilde{O}$ omits log terms—which then allows for the target task to be learned at a rate of $\tilde{O}(\sqrt{s/n}) \ll \tilde{O}(\sqrt{d/n})$. Here, it is emphasized that the representation learning rate of $\tilde{O}(\sqrt{ds/nN})$ scales with the number of tasks N rather than just with n, establishing the benefit of related tasks in improving the target rate.

In practice, however, the representation Γ is in general a nonlinear transformation of x, as when *reproducing kernel Hilbert space* (RKHS) or neural net representations are used. While the importance of the nonlinear setting is well understood, fewer works have so far addressed this more challenging scenario [10, 28].

In the present work, we consider the case where Γ maps x, *nonlinearly,* into an RKHS $\mathcal{H}$, possibly of infinite dimension; more precisely, Γ *projects* the feature maps $K(x, \cdot)$ into an s-dimensional subspace $\mathcal{H}_s$ of $\mathcal{H}$. The link functions g and $\{g_i\}_{i=1}^N$ are assumed to be *simple* in the sense that they are linear in Γ; hence we also have that f and $\{f_i\}_{i=1}^N$ belong to $\mathcal{H}$. In other words, if we knew Γ (or $\mathcal{H}_s = \mathcal{H}_s(\Gamma)$), the target problem would reduce to linear regression in $\mathbb{R}^s$ and therefore would admit (L_2) convergence rates of the form $\tilde{O}(\sqrt{s/n})$, which is significantly faster than usual nonparametric rates for regression over infinite dimensional $\mathcal{H}$ (see discussion after Theorem 4.2 and Corollary 4.5). As in the case of linear Γ discussed above, this improved rate will turn out to require estimating Γ at a fast rate scaling in both N and n.

When moving from linear to nonlinear, nonparametric Γ, a significant new challenge arises due to the bias inherent in the learning procedure. For a high-level intuition, note that a main appeal of meta-learning is that the aggregate of N tasks should help reduce *variance* over using a single task, by carefully combining task-specific statistics computed on each of the N samples; *crucially, such statistics ought to introduce little bias, since bias cannot be averaged out.* Task-specific biases are harder to avoid in nonparametric settings, however, if we wish to avoid overfitting task-specific statistics. This is in contrast to the case of linear projections in $\mathbb{R}^d$, where we have unbiased statistics with no overfitting (one may think, e.g., of ordinary least squares (OLS)).

Fortunately, as we show in this work, nonlinear meta-learning remains possible with rate guarantees improving in both N and n. Our approach relies on the following initial fact: if the links $\{g_i\}_{i=1}^N$ are linear in $\mathcal{H}$, it easily follows that the individual regression functions $\{f_i\}_{i=1}^N$ all live in the span $\mathcal{H}_s \subset \mathcal{H}$ of the shared representation Γ (see set-up in section 3.1). Thus, under a *richness assumption* where $\{f_i\}_{i=1}^N$ span $\mathcal{H}_s$ (extending usual assumptions in the linear case, e.g., of [10]), we may estimate $\mathcal{H}_s$ by estimating the span of regularized estimates $\hat{f}_i$ of f_i. In order to guarantee fast rates that scale with N and n, we need to *under-regularize,* i.e., overfit

task-specific estimates $\{\hat{f}_i\}_{i=1}^N$, to suitably decrease bias, at the cost of increased task-specific (hence overall) variance. Such under-regularization necessarily implies suboptimal regression in each task but improves estimation of the representation defined by Γ.

We demonstrate that these trade-offs may be satisfied, depending on the *smoothness* level of regression functions $\{f_i\}_{i=1}^N$, as captured by complementary regularity conditions on $\{f_i\}_{i=1}^N$ and the interaction between the kernel and data distributions $\{\mu_i\}_{i=1}^N$ defined on $\mathcal{X} \times \mathbb{R}$ (see section 4.1), where we view $\mathcal{X}$ and $\mathbb{R}$ as the input and output spaces, respectively. In the process, some interesting subtleties emerge: meta-learning benefits from *regularity beyond usual saturation points* that were established in traditional RKHS regression (please refer to Remark 4.9). This further illustrates how the meta-learning goal of estimating Γ inherently differs from regression, even when relying on regression estimates. This is discussed in further detail in section 4.

Fast rates scaling in N and n for estimating $\mathcal{H}_s = \mathcal{H}_s(\Gamma)$ from $\text{span}\{\hat{f}_i\}$ are established in Theorem 4.3. This requires, among other tools, a basic variation on Wedin's $\sin-\Theta$ theorem [39] for infinite dimensional operators (Proposition 3). As a consequence, we show that by operating in $\hat{\mathcal{H}}_s$ (the estimation of $\mathcal{H}_s$) for the target regression problem, we can achieve *parametric* target L_2 rates of $\tilde{O}(\sqrt{s/n})$ (see Corollary 4.5), which are much faster than the usual nonparametric rates for $f \in \mathcal{H}$. This last step requires us to establish closeness of projections onto the estimated $\hat{\mathcal{H}}_s$ versus $\mathcal{H}_s$. Moreover, when the feature map $K(x, \cdot)$ is finite dimensional, our results (see Example 1) recover the learning rates obtained in earlier studies (e.g., [10, 37]), where Γ is a linear projection.

Finally, although much of the analysis and involved operations pertain to infinite dimensional $\mathcal{H}$ space, the entire approach can be instantiated in input data space via suitable representation theorems (see section 3.3). This realization supports our theoretical findings with complementary experiments on simulated data, as detailed in section 5.

Related work. Meta-learning is an umbrella term for a rich variety of learning settings, where we are provided with a set of distributions pertaining to relevant training tasks and obtain a functional to speed learning on a target task. In this work, we focus on the case where this functional defines a *representation* Γ *of the data,* and where the target regression function is of the form $f(x) = g(\Gamma(x))$. We begin this section with the closest work to our setting (namely linear and nonlinear projections Γ) and then briefly touch on alternative meta-learning definitions for completeness (although these will be outside the scope of the present study).

We start with works in the *linear setting* that study generalization error where Γ is a learned linear projection $\mathbb{R}^d \to \mathbb{R}^s$, obtained from N training tasks [10, 21, 22, 31, 36, 37, 42]. The authors of [37] study low-dimensional linear representation learning under the assumption of isotropic inputs for all tasks and obtain the learning rate of $\tilde{O}(\sqrt{ds^2/nN} + \sqrt{s/n})$ on the target task. The authors of [10] achieve a similar rate while relaxing the isotropic assumption with a different algorithm. In the linear representation case, they obtain an $\tilde{O}(\sqrt{ds/nN} + \sqrt{s/n})$ rate. The authors of [21] study a somewhat different scenario, where the number of samples per task may differ (and is smaller than the dimension d of the data); the aim is to determine how many tasks must be undertaken in order to achieve consistency. The work of [21] is most closely related to our work, as our procedure, after linearization in $\mathcal{H}$, is quite similar to their procedure in $\mathbb{R}^d$, notably in its reliance on outer-products of regression

estimates. However, many technical issues arise in the infinite dimensional setting considered here, both on the algorithmic and analytical fronts. These are detailed in Remark 3.5 of section 3. The authors of [36] consider an alternate gradient descent algorithm, where they jointly minimize the within task loss and the aggregate loss across all tasks. Under the assumption that the data is Gaussian with the same variance across all tasks, they obtain the learning rate of $\tilde{O}(\sqrt{ds/nN} + \sqrt{s/n})$. The work [22] considers a distribution dependent analysis of meta-learning in the setting of fixed design finite dimensional linear regression, with Gaussian noise and a Gaussian parameter distribution. In the case where the covariance matrix of the parameter is assumed to be known, the authors provide matching upper and lower bounds, which demonstrates a precise characterization of the benefit of meta-learning. While there is no theoretical analysis in the case where the covariance matrix is unknown, the authors provide a detailed description of how the EM algorithm can be employed to solve the meta-learning problem. The works [31, 42] also study the linear representation setting and provide refined theoretical analysis on learning the common representation.

We next consider the case where the representation Γ is nonlinear. The authors of [28] evaluate the performance of a method for learning a nonlinear representation $\Gamma \in \mathcal{F}$ which is s-dimensional, addressing in particular the case of a projection onto a subspace of a reproducing kernel Hilbert space (RKHS). They focus on a *learning to learn* (LTL) scenario, where excess risk is evaluated *in expectation over a distribution of tasks* [28, section 2.2]: we emphasize that this is a fundamentally different objective from the performance on a specific novel test task, as in our setting. The loss they propose to minimize [28, equation (1)] is an average over N training tasks, where each task involves a different linear weighting of the common subspace projection (the work does not propose an algorithm but concerns itself solely with the statistical analysis). Theorem 5 in [28] shows that for an RKHS subspace projection, one can achieve an LTL excess risk for Lipschitz losses (in expectation over the task distribution) that decreases as $\tilde{O}(s/\sqrt{N} + \sqrt{s/n})$. This requires $N \geq n$ in order to approach the parametric rate. Note 2 in [28, p. 8] demonstrates that the factor $1/\sqrt{N}$ is an unavoidable consequence of the LTL setting.

The authors of [10] consider the case of nonlinear representation learning, using the same training loss as equation (1) in [28], but with performance evaluation on a single test task, as in our setting. Again defining $\Gamma \in \mathcal{F}$, they obtain a learning rate of $\tilde{O}(\mathcal{G}(\mathcal{F})/\sqrt{nN} + \sqrt{s/n})$ for the excess risk [10, Theorem 5.1], where $\mathcal{G}(\cdot)$ measures the Gaussian width of $\mathcal{F}$ (a data-dependent complexity measure, and consequently a function of n, N; see, e.g., [27], for further details). The instantiation of $\mathcal{G}(\mathcal{F})$ for specific instances of $\mathcal{F}$ was not pursued further in this work; however, [27] shows that the Gaussian width is of order $\sqrt{nN}$ in n and N, in the case where $\mathcal{F}$ is a projection onto a subspace of an RKHS with Lipschitz kernel.

The problem of learning a "meaningful" low-dimensional representation Γ has also been addressed in the field of sufficient dimension reduction. The works [14, 24, 41] give different criteria for obtaining such Γ and establishing consistency; however, they do not address the risk analysis of downstream learning algorithms that employ Γ. The authors of [23] introduce the so-called principal support vector machine approach for learning both linear and nonlinear Γ. The idea is to learn a set of support vector regression functions, each mapping to different "features" of the output Y (e.g., restrictions to intervals, nonlinear transforms). The estimator $\hat{\Gamma}$ of Γ is then constructed from the principal components of these solutions.

In the linear setting, the authors provide the $\sqrt{n}$-consistency of $\hat{\Gamma}$. The authors of [40] provide a kernelization of sliced inverse regression, which yields a subspace Γ in an RKHS (the so-called effective dimension reduction space). Consistency of the projection by $\hat{\Gamma}$ of an RKHS feature map $\phi(x)$ is established; and an $O(n^{-1/4})$ convergence rate is obtained, under the assumption that all Γ components can be expressed in terms of a finite number of covariance operator eigenfunctions. The learning risk of downstream estimators using $\hat{\Gamma}$ remains to be established, however.

Outside of the regression setting, meta-learning has been studied for classification: [15] investigates the generalization error in this setting, with the representation Γ being a fully connected ReLU neural net of depth Q, common to all tasks. The authors of [1] study the sample complexity per task when the task-specific classifiers are halfspaces in $\mathbb{R}^s$ and the samples per task are extremely low. Finally, there are analyses for other meta-learning schemes such as domain adaption [3, 26], domain generalization [5], and covariate shift [25], as well as alternative gradient-based approaches to refine algorithms on novel test domains, e.g., [9, 11, 12, 20, 29].

2. Background and notation.

Function spaces and basic operators. Let μ be a probability measure on $\mathcal{X} \times \mathbb{R}$, $\mu_{\mathcal{X}}$ denotes the marginal distribution of μ on $\mathcal{X}$, and $\mu(\cdot|x)$ denotes the conditional distribution on $\mathbb{R}$ given $x \in \mathcal{X}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric and positive definite kernel function and $\mathcal{H}$ be a vector space of $\mathcal{X} \to \mathbb{R}$ functions, endowed with a Hilbert space structure via an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. K is a reproducing kernel of $\mathcal{H}$ if and only if 1. $\forall x \in \mathcal{X}, \phi(x) \doteq K(\cdot, x) \in \mathcal{H}$; 2. $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{H}, f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$. A space $\mathcal{H}$ which possesses a reproducing kernel is called a reproducing kernel Hilbert space (RKHS) [4]. $L_2(\mathcal{X}, \mu_{\mathcal{X}})$, abbreviated $L_2(\mu)$, denotes the Hilbert space of square-integrable functions with respect to (w.r.t.) $\mu_{\mathcal{X}}$.[1]

$\|A\|$ and $\|A\|_{HS}$ denote, respectively, the operator and Hilbert–Schmidt norm of a linear operator A on $\mathcal{H}$. For $f, g \in \mathcal{H}$, $g \otimes f \doteq \langle f, \cdot \rangle_{\mathcal{H}} g$ is the generalization of the Euclidean outer product. The covariance operator is defined as $\Sigma \doteq \mathbb{E}_{X \sim \mu}[K(X, \cdot) \otimes K(X, \cdot)]$.

We require some standard technical assumptions on the previously defined RKHS and kernel: 1. $\mathcal{H}$ is separable; this is satisfied if $\mathcal{X}$ is a Polish space and K is continuous [33, Lemma 4.33]; 2. $\phi(x)$ is measurable for all $x \in \mathcal{X}$; 3. $\sup_{x,x' \in \mathcal{X}} K(x, x') \doteq \kappa^2 < \infty$. Note that those assumptions are not restrictive in practice, as well-known kernels such as the Gaussian, Laplacian, and Matérn kernels satisfy all of the above assumptions on $\mathbb{R}^d$ [32].

Matrix notation of basic operators. For a set of vectors $\{u_1, \ldots, u_n\} \in \mathcal{H}$, $U \doteq [u_1, \ldots, u_n]$ denotes the operator with the vectors as "columns," formally $U : \mathbb{R}^n \to \mathcal{H}, \alpha \mapsto \sum_{i=1}^n u_i \alpha_i$. Its adjoint is $U^* : \mathcal{H} \to \mathbb{R}^n, u \mapsto (\langle u_i, u \rangle_{\mathcal{H}})_{i=1}^n$.

Kernel ridge regression and regularization. Given a data set $D = \{(x_i, y_i)\}_{i=1}^n$ independently sampled from μ, kernel ridge regression aims to estimate the *regression function* $f_\mu = \mathbb{E}_\mu[Y \mid X]$, with the following kernel-based regularized least-squares procedure:

$$\hat{f}_\lambda = \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\}, \tag{2.1}$$

[1] To simplify notation, when we integrate over $\mu_{\mathcal{X}}$ a function defined on $\mathcal{X}$, we use $\mathbb{E}_\mu$ instead of $\mathbb{E}_{\mu_{\mathcal{X}}}$.

with $\lambda > 0$ the regularization parameter. $\mathcal{R}_\mu(f) \doteq \mathbb{E}_\mu\left[(Y - f(X))^2\right]$ is the squared expected risk and the excess risk is given by $\mathcal{E}_\mu(f) \doteq \sqrt{\mathcal{R}_\mu(f) - \mathcal{R}_\mu(f_\mu)} = \mathbb{E}_\mu\left[(f(X) - f_\mu(X))^2\right]^{1/2}$. We also introduce the population version of $\hat{f}_\lambda$ as

$$f_\lambda = \operatorname*{argmin}_{f \in \mathcal{H}} \left\{\mathbb{E}_\mu\left[(Y - f(X))^2\right] + \lambda \|f\|_{\mathcal{H}}^2\right\}. \tag{2.2}$$

The normed difference $\hat{f}_\lambda - f_\lambda$ is referred to as the estimation error and is a central object for the study of kernel ridge regression (see, e.g., [13]).

Further notation. For $n, m \in \mathbb{N}^*, n \le m, [n] \doteq \{1, \dots, n\}, [n, m] \doteq \{n, \dots, m\}$. For two real numbers a and b, we denote $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$.

3. Nonlinear meta-learning.

3.1. Population set-up. We consider a setting with N source distributions $\{\mu_i\}_{i \in [N]}$ defined on $\mathcal{X} \times \mathbb{R}$, with corresponding regression functions of the form $f_i(x) = g_i(\Gamma(x))$. We are interested in minimizing the excess risk for a target distribution μ_T, with regression function $f_T(x) = g_T(\Gamma(x))$. In the mostly common linear case, it is assumed that Γ *projects* into a subspace of $\mathbb{R}^d = \mathcal{X}$. However, in this manuscript, we assume that Γ is a projection of nonlinear feature maps in an infinite dimensional space.

Assumption 1. We let $\Gamma : \mathcal{X} \mapsto \mathcal{H}$ be a map from $x \in \mathcal{X}$ to a subspace $\mathcal{H}_s$ of dimension $s \ge 1$ of an RKHS $\mathcal{H}$ as follows: given a projection operator P onto $\mathcal{H}_s$, $\Gamma(x) \doteq PK(x, \cdot)$. Furthermore, all link functions g_T, $\{g_i\}_{i=1}^N$ are assumed linear $\mathcal{H} \mapsto \mathbb{R}$, i.e., $\exists w_T, w_i \in \mathcal{H}_s$ s.t. $g_T(\Gamma(x)) = \langle w_T, \Gamma(x)\rangle_{\mathcal{H}}$, and $g_i(\Gamma(x)) = \langle w_i, \Gamma(x)\rangle_{\mathcal{H}}$.

Remark 3.1. Given an orthonormal basis (ONB) $V = [v_1, \dots, v_s]$ of $\mathcal{H}_s$, we may rewrite $g_T(\Gamma(x)) = \alpha_T^\top V^* K(x, \cdot)$, i.e., for $\alpha_T \in \mathbb{R}^s$, for an s-dimensional (nonlinear) representation $V^*\Gamma(x) = V^* K(x, \cdot)$ of x. The same is true for $\{g_i\}_{i=1}^N$ with respective $\{\alpha_i\}_{i=1}^N$. The representations are nonunique, although their corresponding regression functions and $\mathcal{H}_s$ are unique (see Remark 3.3 below).

Remark 3.2. Since P is self-adjoint, we have $f_T(x) \doteq \langle Pw_T, K(x, \cdot)\rangle_{\mathcal{H}}$; hence, by the reproducing property, $f_T = Pw_T \in \mathcal{H}_s$. Similarly, we have that all $\{f_i\}_{i=1}^N$ are in $\mathcal{H}_s$.

Remark 3.2 indicates that $\operatorname{span}\left(\{f_i\}_{i \in [N]}\right) \subseteq \mathcal{H}_s$. We therefore need the following *richness condition,* similar to previous works on meta-learning in the linear representation case [10], without which we cannot hope to learn $\mathcal{H}_s$.

Assumption 2 (source richness). We have that $\operatorname{span}\left(\{f_i\}_{i \in [N]}\right) = \mathcal{H}_s$.

Remark 3.3. For any projection P onto some complete subspace $\mathcal{H}_s$, $\langle \cdot, PK(x, \cdot)\rangle_{\mathcal{H}}$ evaluates every function in $\mathcal{H}_s$ at x and in fact is well known as the *kernel* of the sub-RKHS defined by $\mathcal{H}_s$. The same fact implies uniqueness of $\mathcal{H}_s$ and in particular that it equals $\overline{\operatorname{span}}\{\Gamma(x) \doteq PK(x, \cdot)\}$.

3.2. Learning set-up. In this section we present the high level ideas of our meta-learning strategy with nonlinear representation. The first step is to learn a subspace approximation $\hat{\mathcal{H}}_s \approx \mathcal{H}_s$ from source tasks. This process aims to find a suitable representation that facilitates

the learning of the target task. We refer to this step as *pre-training*. The second step involves directly learning the target task within the subspace $\hat{\mathcal{H}}_s$. We refer to this step as *inference*.

Source tasks: Pre-training. Our approach to approximate $\mathcal{H}_s$ is inspired by [21], which focused on finite-dimensional linear meta-learning. We extend this strategy to encompass (potentially infinite dimensional) nonlinear meta-learning. Under the source richness assumption (Assumption 2), $\mathcal{H}_s$ is equal to the range of the rank-s operator (see Proposition SM2.1 in the supplementary material)

$$C_N \doteq \frac{1}{N}\sum_{i=1}^{N} f_i \otimes f_i, \qquad \operatorname{ran} C_N = \mathcal{H}_s. \tag{3.1}$$

Therefore, we estimate $\mathcal{H}_s$ via the range of

$$\hat{C}_{N,n,\lambda} \doteq \frac{1}{N}\sum_{i=1}^{N} \hat{f}'_{i,\lambda} \otimes \hat{f}_{i,\lambda}, \tag{3.2}$$

where $\hat{f}'_{i,\lambda}, \hat{f}_{i,\lambda}$ are i.i.d. copies of a ridge regression estimator for source task $i \in [N]$. Here, we use a data-splitting strategy to obtain the following:

$$\mathbb{E}[\hat{C}_{N,n,\lambda}] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[\hat{f}'_{i,\lambda}] \otimes \mathbb{E}[\hat{f}_{i,\lambda}].$$

This property plays a crucial role in deriving approximation rates for $\mathcal{H}_s$. Data-splitting is similarly employed in [21]. Avoiding data-splitting remains an open problem even in the finite-dimensional linear representation setting.

Each source task is learned from a dataset $\mathcal{D}_i = \{(x_{i,j}, y_{i,j})_{j=1}^{2n}\}, i \in [N]$, of i.i.d. observations sampled from μ_i, via regularized kernel regression as in (2.1),

$$\hat{f}_{i,\lambda} = \operatorname*{argmin}_{f\in\mathcal{H}} \sum_{j=1}^{n} (y_{i,j} - f(x_{i,j}))^2 + n\lambda\|f\|_{\mathcal{H}}^2,\ \hat{f}'_{i,\lambda} = \operatorname*{argmin}_{f\in\mathcal{H}} \sum_{j=n+1}^{2n} (y_{i,j} - f(x_{i,j}))^2 + n\lambda\|f\|_{\mathcal{H}}^2. \tag{3.3}$$

For task $i \in [N]$, let $K_i, L_i \in \mathbb{R}^{n\times n}$ be the Gram matrices such that $(K_i)_{j,l} = K(x_{i,j}, x_{i,l})$, $(j,l) \in [n]$, and $(L_i)_{j,l} = K(x_{i,j}, x_{i,l})$, $(j,l) \in [n+1:2n]$. Then for all $x \in \mathcal{X}$,

$$\hat{f}_{i,\lambda}(x) = Y_i^\top (K_i + n\lambda I_n)^{-1} k_{i,x}, \quad \hat{f}'_{i,\lambda}(x) = (Y'_i)^\top (L_i + n\lambda I_n)^{-1} \ell_{i,x}, \tag{3.4}$$

where $k_{i,x} = (K(x_{i,1}, x), \ldots, K(x_{i,n}, x))^\top \in \mathbb{R}^n$, $\ell_{i,x} = (K(x_{i,n+1}, x), \ldots, K(x_{i,2n}, x))^\top \in \mathbb{R}^n$, $Y_i = (y_{i,1}, \ldots, y_{i,n})^\top \in \mathbb{R}^n$, and $Y'_i = (y_{i,n+1}, \ldots, y_{i,2n})^\top \in \mathbb{R}^n$.

After obtaining $\hat{C}_{N,n,\lambda}$, we cannot directly compare $\operatorname{ran} C_N$ to $\operatorname{ran} \hat{C}_{N,n,\lambda}$, since the latter is not guaranteed to be of rank s. We therefore consider the singular value decomposition of $\hat{C}_{N,n,\lambda}$:

$$\hat{C}_{N,n,\lambda} = \sum_{i=1}^{N} \hat{\gamma}_i \hat{u}_i \otimes \hat{v}_i = \hat{U}\hat{D}\hat{V}^*,$$

where $\hat{\gamma}_1 \geq \cdots \geq \hat{\gamma}_N \geq 0$ are the singular values and stored in the diagonal matrix $\hat{D} \in \mathbb{R}^{N\times N}$. The right and left singular vectors are stored as $\hat{V} = [\hat{v}_1, \ldots, \hat{v}_N]$ and $\hat{U} = [\hat{u}_1, \ldots, \hat{u}_N]$, respectively. We use the right singular vectors to construct the approximation of $\mathcal{H}_s$ as follows (note that a similar approach can be applied to the left singular vectors):

$$\hat{\mathcal{H}}_s \doteq \operatorname{span}\{\hat{v}_1, \ldots, \hat{v}_s\}.$$

We define the orthogonal projection onto $\hat{\mathcal{H}}_s$ as $\hat{P}$.

Remark 3.4. In nonparametric regression, as employed in this approach, regularization becomes necessary. This leads to biased estimators since $\mathbb{E}[\hat{f}_{i,\lambda}] \neq f_i$. For subspace approximation, it is crucial to effectively control this bias since it cannot be averaged out.

Target task: Inference. We are given a target task dataset $\mathcal{D}_T = \{(x_{T,j}, y_{T,j})_{j=1}^{n_T}\} \in (\mathcal{X} \times \mathbb{R})^{n_T}$ sampled from μ_T in order to approximate f_T. As mentioned in Remark 3.3, $\hat{\mathcal{H}}_s = \hat{P}(\mathcal{H}) \subseteq \mathcal{H}$ forms an RKHS on $\mathcal{X}$ having the same inner product as $\mathcal{H}$ and with reproducing kernel $\hat{K}(x, y) = \langle \hat{P}\phi(x), \phi(y)\rangle_{\mathcal{H}}, (x, y) \in \mathcal{X}^2$. Consequently, we can estimate f_T via regularized kernel regression within $\hat{\mathcal{H}}_s$, as shown in (2.1). For $\lambda_* > 0$,

$$\hat{f}_{T,\lambda_*} \doteq \operatorname*{arg\,min}_{f\in\hat{\mathcal{H}}_s} \sum_{j=1}^{n_T} (f(x_{T,j}) - y_{T,j})^2 + n_T\lambda_*\|f\|_{\mathcal{H}}^2. \tag{3.5}$$

Since $\hat{\mathcal{H}}_s$ is s-dimensional, it can be treated as a standard regularized regression in $\mathbb{R}^s$ (see section 3.3). The following remark highlights the main technical difficulties compared to the linear case.

Remark 3.5 (differences from linear case). We point out that, while the algorithm used in our meta-learning approach draws inspiration from [21], there are significant differences due to the complexities of the nonlinear setting, as opposed to the linear one, as outlined below.

First, from the algorithmic perspective, proper regularization is crucial in an infinite dimensional space to prevent overfitting. [21] did not employ a regularization scheme but instead relied on OLS regression, which does not directly extend to infinite dimension where some form of regularization is needed to control a learner's capacity. A second algorithmic difference arises in the instantiation of the procedure in input space $\mathbb{R}^d$: while our procedure appears similar to [21]'s when described in the RKHS $\mathcal{H}$, i.e., after embedding, its instantiating in $\mathbb{R}^d$ is nontrivial, as it involves translating operations in $\mathcal{H}$—e.g., projections onto subspaces of $\mathcal{H}$—into operations in $\mathbb{R}^d$. Section 3.3 below addresses such technicality in depth.

Second, many crucial difficulties arise in the analysis of the infinite dimensional setting which are not present in the finite-dimensional case. Importantly, in infinite dimensional space, the analysis effectively concerns two separate spaces: the RKHS $\mathcal{H}$, which encodes the nonlinear representation, and the L_2 regression space. Thus a main technical difficulty is to relate rates of convergence in $\mathcal{H}$ (where all operations are taking place) to rates in L_2, in particular via the *covariance operator,* which links the two norms $\|\cdot\|_{\mathcal{H}}$ and $\|\cdot\|_{L_2}$; this is relatively easy in finite dimension by simply assuming an identity covariance (or bounds on its eigenvalues), as done in [10, 21, 37], but such assumptions do not extend to infinite dimension where concepts such as "identity covariance" are not defined. Namely, an infinite dimensional

covariance operator must be compact, which implies that its eigenvalues decay to zero. Our analysis reveals that the speed of that decay (encoded in Assumptions 3 and 4) determines the rate at which we can learn. Furthermore, unlike in [10, 21, 37], where there was no need to regularize the task-specific regressors, much of our analysis focuses on understanding the bias-variance trade-offs induced by the choice of regularizers. This is nontrivial but is crucial for guaranteeing gains in our nonlinear case, as explained in the paper's introduction. Thus, in the present infinite dimensional setting, as we will see, such crucial trade-offs will depend on specific measures of smoothness—of the RKHS $\mathcal{H}$ and the regression functions therein—as introduced in the main results in section 4.2 (see Assumptions 3, 4, and 5).

3.3. Instantiation in data space. In this section, we describe in detail the steps outlined in section 3.2 to offer a comprehensive understanding of the computational process. In particular, we focus on the computation of the right singular vectors of $\hat{C}_{N,n,\lambda}$, which plays a crucial role in constructing $\hat{\mathcal{H}}_s$. Additionally, we provide insights into the projection of new data points onto $\hat{\mathcal{H}}_s$, which is essential during the inference stage. We emphasize that such instantiations were not provided for kernel classes in the nonlinear settings addressed by [10, 28]; given the nonconvexity of the loss (equation (1) in both papers), this task is nontrivial.

Singular value decomposition of $\hat{C}_{N,n,\lambda}$. We start by explaining how we can compute the SVD of $\hat{C}_{N,n,\lambda}$ in closed form from data. Let $\{\hat{v}_i\}_{i=1}^s$ and $\{\hat{u}_i\}_{i=1}^s$ be the right and left singular vectors corresponding to the largest s singular values, and denote $\hat{V}_s = [\hat{v}_1, \dots, \hat{v}_s]$ and $\hat{U}_s = [\hat{u}_1, \dots, \hat{u}_s]$. The next proposition shows that $(\hat{U}_s, \hat{V}_s)$ can be obtained through the solution of a generalized eigenvalue problem associated to the matrices $J, Q \in \mathbb{R}^{N\times N}$, where for $(i,j) \in [N]^2$

$$J_{i,j} = \langle \hat{f}_i, \hat{f}_j \rangle_{\mathcal{H}} = nY_i^\top (K_i + n\lambda I_n)^{-1} K_{ij} (K_j + n\lambda I_n)^{-1} Y_j,$$
$$Q_{i,j} = \langle \hat{f}'_i, \hat{f}'_j \rangle_{\mathcal{H}} = n(Y'_i)^\top (L_i + n\lambda I_n)^{-1} L_{ij} (L_j + n\lambda I_n)^{-1} Y'_j.$$

Proposition 1. *Consider the generalized eigenvalue problem, which consists of finding generalized eigenvectors $(\alpha^\top, \beta^\top)^\top \in \mathbb{R}^{2N}$ and generalized eigenvalues $\gamma \in \mathbb{R}$ such that*

$$\begin{bmatrix} 0 & QJ \\ JQ & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \gamma \begin{bmatrix} Q & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Define $A \doteq [\hat{f}'_1, \dots, \hat{f}'_N]$ and $B \doteq [\hat{f}_1, \dots, \hat{f}_N]$ and let $\{(\hat{\alpha}_i^\top, \hat{\beta}_i^\top)^\top\}_{i=1}^s$ be the generalized eigenvectors associated to the s-largest generalized eigenvalues of the above problem and re-normalized such that $\alpha_i^\top Q \alpha_i = \beta_i^\top J \beta_i = 1, i \in [s]$. The following two families of vectors $\{\hat{u}_i\}_{i=1}^s$ and $\{\hat{v}_i\}_{i=1}^s$ are orthonormal systems and correspond to top-s left and right singular vectors of $\hat{C}_{N,n,\lambda}$:

$$\hat{u}_i = A\hat{\alpha}_i = \sum_{j=1}^N (\alpha_i)_j \hat{f}'_j, \quad \hat{v}_i = B\hat{\beta}_i = \sum_{j=1}^N (\hat{\beta}_i)_j \hat{f}_j, \quad i \in [s].$$

In other words, we can define the projection onto the subspace $\hat{\mathcal{H}}_s$ via $\{\hat{v}_i\}_{i=1}^s$:

$$\hat{\mathcal{H}}_s \doteq \operatorname{span}\{\hat{v}_1, \dots, \hat{v}_s\} = \operatorname{span}\{B\hat{\beta}_1, \dots, B\hat{\beta}_s\}.$$

Projection onto $\hat{\mathcal{H}}_s$ and inference. Next, we explain how we can project a new point onto $\hat{\mathcal{H}}_s$ and perform inference on such representations. The projection onto $\hat{\mathcal{H}}_s$ satisfies $\hat{P} = \hat{V}_s \hat{V}_s^*$. A new point $x \in \mathcal{X}$ can be projected into $\hat{\mathcal{H}}_s$ as $\hat{P}\phi(x)$ and identified to $\mathbb{R}^s$ via

$$\tilde{x} = \hat{V}_s^* \phi(x) = (\langle \hat{v}_1, \phi(x)\rangle_{\mathcal{H}}, \dots, \langle \hat{v}_s, \phi(x)\rangle_{\mathcal{H}})^\top = (\hat{v}_1(x), \dots, \hat{v}_s(x))^\top \in \mathbb{R}^s. \tag{3.6}$$

By Proposition 1, $\tilde{x}$ can be computed as

$$\tilde{x}_i = \hat{v}_i(x) = \langle \hat{v}_i, \phi(x)\rangle_{\mathcal{H}} = \langle B\hat{\beta}_i, \phi(x)\rangle_{\mathcal{H}} = \hat{\beta}_i^\top B^* \phi(x), \quad i \in [s],$$

where $B^*\phi(x) \doteq (\hat{f}_1(x), \dots, \hat{f}_N(x))^\top \in \mathbb{R}^N$. Recall that after pre-training, at inference, we receive a target task dataset $\mathcal{D}_T = \{(x_{T,j}, y_{T,j})\}_{j=1}^{n_T}$. We denote by $\tilde{x}_{T,j} \in \mathbb{R}^s$ the embedding of the covariate $x_{T,j}$ into $\hat{\mathcal{H}}_s$ according to (3.6), and by $X_T \doteq [\tilde{x}_{T,1}, \dots, \tilde{x}_{T,n_T}] \in \mathbb{R}^{s \times n_T}$ the data matrix that collects the embedded points as columns; $K_T \doteq X_T^\top X_T \in \mathbb{R}^{n_T \times n_T}$ is the associated Gram matrix and $n_T^{-1} X_T X_T^\top \in \mathbb{R}^{s \times s}$ the associated empirical covariance.

Proposition 2. *$\hat{f}_{T,\lambda_*} = \hat{V}_s \beta_{T,\lambda_*}$, where*

$$\hat{\beta}_{T,\lambda_*} \doteq \operatorname*{arg\,min}_{\beta \in \mathbb{R}^s} \sum_{j=1}^{n_T} \left(\beta^\top \tilde{x}_{T,j} - y_{T,j}\right)^2 + n_T \lambda_* \|\beta\|_2^2 = X_T (K_T + n_T \lambda_* I_{n_T})^{-1} Y_T,$$

and $Y_T \doteq (y_{T,1}, \dots, y_{T,n_T})^\top \in \mathbb{R}^{n_T}$. For all $x \in \mathcal{X}$, $\hat{f}_{T,\lambda_*}(x) = \beta_{T,\lambda_*}^\top \tilde{x}$.

4. Main results.

4.1. Regularity assumptions.

Our first two assumptions are related to the eigensystem of the covariance operator. For $i \in [N] \cup \{T\}$, the covariance operator for task i, $\Sigma_i \doteq \mathbb{E}_{\mu_i}[\phi(X) \otimes \phi(X)]$, is positive semidefinite and trace-class and thereby admits an eigenvalue decomposition with eigenvalues $\lambda_{i,1} \geq \lambda_{i,2} \geq \cdots \geq 0$ and eigenvectors $\{\sqrt{\lambda_{i,j}} e_{i,j}\}_{j \geq 1}$ [34, Lemma 2.12].

Assumption 3. For $i \in [N]$, the eigenvalues of the covariance operator Σ_i from the (K, μ_i) pair satisfy a polynomial decay of order $1/p$, i.e., for some constant $c > 0$ and $0 < p \leq 1$, and for all $j \geq 1$, $\lambda_{i,j} \leq c j^{-1/p}$. When the covariance operator has finite rank, we have $p = 0$.

The assumption on the decay rate of the eigenvalues is typical in the risk analysis for kernel ridge regression (see, e.g., [7, 13]).

Assumption 4. There exist $\alpha \in [p, 1]$ and $k_{\alpha,\infty} > 0$, such that, for any task $i \in [N]$ and μ_i-almost all $x \in \mathcal{X}$, $\sum_{j \geq 1} \lambda_{i,j}^\alpha e_{i,j}^2(x) \leq k_{\alpha,\infty}^2$.

This assumption is known as an *embedding property* (into L_∞; see [13]) and is a regularity condition on the pair (K, μ_i). In particular, let $T_{K,i} \doteq \sum_j \lambda_{i,j}\, e_{i,j} \otimes_{L_2(\mu_i)} e_{i,j}$ denote the *integral operator* $L_2(\mu_i) \mapsto L_2(\mu_i)$ induced by K; then the assumption characterizes the smallest α such that the range of $T_{K,i}^{\alpha/2}$ may be continuously embedded into $L_\infty(\mu_i)$. As is well known for continuous kernels, $\operatorname{ran} T_{K,i}^{1/2} \equiv \mathcal{H}$; thus the assumption holds for $\alpha = 1$ whenever K is bounded. Note that the *interpolation spaces* $\operatorname{ran} T_{K,i}^{\alpha/2}$ only get larger as $\alpha \to 0$, eventually coinciding with the closure of $\operatorname{span}\{e_{i,j}\}_{j \geq 1}$ in $L_2(\mu_i)$. Additionally, it can be shown that Assumption 4 implies Assumption 3 with $p = \alpha$ [13, Lemma 10].

As alluded to in the introduction, α has no direct benefit for regression in our *well-specified* setting with $f_i \in \mathcal{H}$ but is beneficial in meta-learning (see Corollary 4.5 and Remark 4.9 thereafter).

Assumption 5. There exist $r \in [0,1]$ and $R \geq 0$, such that for $i \in [N]$, the regression function f_i associated with μ_i is an element of $\mathcal{H}$ and satisfies $\|\Sigma_i^{-r} f_i\|_{\mathcal{H}} \doteq R < \infty$.

This assumption, imposing smoothness on each source task regression function, is standard in the statistical analysis of regularized least-squares algorithms [7].

Remark 4.1. Assumptions 3, 4, and 5 only concern the source tasks towards nonlinear meta-learning. We will see in section 4.2 that they are complementary in ensuring enough *smoothness* of the source regression functions to allow for sufficient *under-regularization* to take advantage of the aggregation of N source tasks. Thus, the main assumption on the target task is simply that it shares the same nonlinear representation as the source tasks.

Finally, to control the noise we assume the following.

Assumption 6. There exists a constant $Y_\infty \geq 0$ such that for all $Y \sim \mu_i, i \in [N] \cup \{T\}$: $|Y| < Y_\infty$.

4.2. Main theorems.

Theorem 4.2. *Under Assumptions 1, 2, and 6 with $s \geq 1$, for $\tau \geq 2.6$, $0 < \lambda_* \leq 1$, and*

$$n_T \geq 6\kappa^2 \lambda_*^{-1} (\tau + \log(s)),$$

with probability not less than $1 - 3e^{-\tau}$ and conditionally on $\{\mathcal{D}_i\}_{i=1}^N$,

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \leq c_0 \left\{ \sqrt{\frac{\tau s}{n_T}} + \frac{\tau}{n_T \sqrt{\lambda_*}} + \sqrt{\lambda_*} + \left\| \hat{P}_\perp P \right\| \right\},$$

where $\hat{P}_\perp \doteq I_{\mathcal{H}} - \hat{P}$ and c_0 is a constant that depends only on $Y_\infty, \|f_T\|_{\mathcal{H}}$, and κ. Hence, treating τ as a constant, if we take $\lambda_* = 12\kappa^2(\log(s) \vee \tau) n_T^{-1}$, conditionally on $\{\mathcal{D}_i\}_{i=1}^N$, for $n_T \geq 12\kappa^2(\log(s) \vee \tau)$, we get that $\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*})$ is of the order

$$\sqrt{\frac{s}{n_T}} + \left\| \hat{P}_\perp P \right\|.$$

Theorem 4.2 reveals that the excess risk for the target task consists of two components: $\sqrt{s/n_T}$ due to the inference stage, and $\|\hat{P}_\perp P\|$ in the pre-training stage. In the upcoming Theorem 4.3, we will see that the pre-training error $\|\hat{P}_\perp P\|$ decays with n and N. In other words, if either N (number of tasks) or n (number of data within each task) is sufficiently large, we can guarantee that the excess risk decays at the parametric rate $O(\sqrt{s/n_T})$, an optimal rate achieved only by performing linear regression in a space of dimension s. $\|\hat{P}_\perp P\|$ is the sin-Θ distance between $\mathcal{H}_s$ and $\hat{\mathcal{H}}_s$ [35]. We can relate this distance to the difference between C_N and $\hat{C}_{N,n,\lambda}$ using classic perturbation theory for singular vectors. Proposition 3 is a basic generalization of Wedin's $\sin-\Theta$ theorem [39].

Proposition 3 (Wedin's $\sin-\Theta$ theorem). *Given C_N and $\hat{C}_{N,n,\lambda}$ defined in (3.1) and (3.2), with γ_s smallest nonzero eigenvalues of C_N, we have*

$$\|\hat{P}_\perp P\| \le 2\gamma_s^{-1}\|\hat{C}_{N,n,\lambda} - C_N\|. \tag{4.1}$$

We refer the reader to section SM1.2 in the supplementary material for the proof. Note that the operator norm $\|\hat{C}_{N,n,\lambda}-C_N\|$ is dominated by the Hilbert–Schmidt norm $\|\hat{C}_{N,n,\lambda}-C_N\|_{HS}$. The following theorem provides high probability bounds on this quantity.

Theorem 4.3. *Let Assumptions 3, 4, 5, and 6 hold with parameters $0 < p \le \alpha \le 1$ and $r \in [0,1]$. Let $\tau \ge \log(2)$, $N \ge \tau$, and $0 < \lambda \le 1 \wedge \min_{i\in[N]} \|\Sigma_i\|$. Define the following terms:*

$$A_\lambda \doteq c\log(Nn)\left(1 + p\log(\lambda^{-1})\right)\lambda^{-\alpha},$$
$$B_\lambda \doteq c\log(Nn)\left(1 + p\log(\lambda^{-1})\right)\lambda^{-(1+p)},$$

where c only depends on $k_{\alpha,\infty}, D, \kappa$. We require $n \ge A_\lambda$ if $r \in (0,1/2]$ or $n \ge B_\lambda$ if $r \in (1/2,1]$. Under both scenarios, with probability greater than $1-2e^{-\tau}-o((nN)^{-10})$ over the randomness in the source tasks we have

$$\|\hat{C}_{N,n,\lambda} - C_N\|_{HS} \le C_1\left(\frac{\log(nN)\sqrt{\tau}}{\sqrt{nN}\lambda^{\frac{1}{2}+\frac{p}{2}}}\sqrt{1+\frac{1}{n\lambda^{\alpha-p}}} + \lambda^r\right), \tag{4.2}$$

where C_1 only depends on Y_∞, R, κ, p, and $k_{\alpha,\infty}$.

We highlight two key aspects of Theorem 4.3. First, the bound is comprised of two terms that come from a bias-variance decomposition (refer to section 6 for details):

$$\|\hat{C}_{N,n,\lambda} - C_N\|_{HS} \le \underbrace{\|\hat{C}_{N,n,\lambda} - \mathbb{E}(\hat{C}_{N,n,\lambda})\|_{HS}}_{\text{Variance}} + \underbrace{\|\mathbb{E}(\hat{C}_{N,n,\lambda}) - C_N\|_{HS}}_{\text{Bias}}.$$

The first and second terms in (4.2) correspond to bounds on the variance and on the bias, respectively. Second, while we obtain the same upper bound in (4.2) for the two distinct scenarios $r \in (0,1/2]$ and $r \in (1/2,1]$, the requirement on the number of training samples per task is different. In particular, $B_\lambda \ge A_\lambda$, since $\lambda \le 1$ and $p+1 \ge \alpha$. This means that we can benefit from further smoothness $r > 1/2$, but at the cost of a higher number of samples per source task. Our analysis in Theorem SM1.7 implies that the difference comes from bounding the bias term. We specifically show that uniformly bounding the bias from each task when $r \in (1/2,1]$ (which requires $n \ge B_\lambda$) is strictly harder than doing so when $r \in (0,1/2]$ (which requires $n \ge A_\lambda$). As such, our results reveal the inherent difficulty of nonlinear meta-learning: analyzing the bias is more involved than analyzing the variance, a fact which cannot be seen in the linear representation case.

Remark 4.4 (further smoothness and the well-specified regime). While in usual analyses, consistency in L_2 norm is ensured for $r = 0$ (implying that the regression function is in $\mathcal{H}$), we require further smoothness on source regression functions (i.e., $r > 0$) to guarantee consistency in our setting. The requirement for additional smoothness stems from the fact that the result depends on convergence of regression estimates in the *stronger RKHS norm* rather than in L_2 norm, as the above $\|\cdot\|_{HS}$ and projections are defined w.r.t. the RKHS itself.

We point out that in kernel learning literature (see, e.g., [7, 13]), one often observes the Tikhonov saturation effect, where the learning rate does not improve for $r > 1/2$. However, we remark that this saturation happens only when the L_2 norm is used. In particular, (4.2) demonstrates that our learning rate can be improved up to $r = 1$. This reflects the fact that, if the RKHS norm is employed, the Tikhonov saturation effect happens for $r > 1$. A similar phenomenon is observed by [6].

Combining Theorem 4.2, Proposition 3, and (4.2) from Theorem 4.3, we obtain the following results on the meta-learning excess risk.

Corollary 4.5. *Under the assumptions of Theorem 4.2 and Theorem 4.3, for $\tau \geq 2.6$ and $\lambda_* = 12\kappa^2(\log(s) \vee \tau)n_T^{-1}$, with probability $1 - 5e^{-\tau} - o((nN)^{-10})$ over the randomness in both the source and target tasks, we have the following regimes of rates for a constant C_3 that only depends on Y_∞, R, κ, γ_1, p, c, $\|f_T\|_{\mathcal{H}}$, and $k_{\alpha,\infty}$.*

A. Small number of tasks. *In this regime, with the number of tasks N being small, the variance is significant compared to the bias. Therefore, we must choose λ to balance the order of the bias with that of the variance. If $N \leq n^{\frac{2r+1+p}{\alpha}-1}$ and $r \in (0, 1/2]$ or $N \leq n^{\frac{2r+1+p}{p+1}-1}$ and $r \in (1/2, 1]$, for a choice of $\lambda = \left(\log^2(nN)/(nN)\right)^{\frac{1}{2r+1+p}}$,*

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \leq C_3\tau \left\{ \sqrt{\frac{s}{n_T}} + \left(\frac{\log^2(nN)}{nN}\right)^{\frac{r}{2r+1+p}} \right\}. \tag{4.3}$$

B. Large number of tasks. *In this regime, we consider larger N (see B.1 and B.2 below), so that the variance term becomes negligible compared to the bias. Therefore, the rates below correspond to the choices of λ that minimize the bias in (4.2) (under the constraints $n \geq A_\lambda, B_\lambda$). In what follows, $\omega > 2$ is a free parameter.*

- B.1. *For $r \in (0, 1/2]$, if $n^{\frac{2r+1+p}{\alpha}-1} \leq N \leq o(e^n)$, for a choice of $\lambda = (\frac{\log^\omega(nN)}{n})^{\frac{1}{\alpha}}$,*

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \leq C_3\tau \left\{ \sqrt{\frac{s}{n_T}} + \log^{\frac{\omega r}{\alpha}}(nN) \cdot n^{-\frac{r}{\alpha}} \right\}.$$

- B.2. *For $r \in (1/2, 1]$, if $n^{\frac{2r+1+p}{p+1}-1} \leq N \leq o(e^n)$, for a choice of $\lambda = (\frac{\log^\omega(nN)}{n})^{\frac{1}{p+1}}$,*

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \leq C_3\tau \left\{ \sqrt{\frac{s}{n_T}} + \log^{\frac{\omega r}{p+1}}(nN) \cdot n^{-\frac{r}{p+1}} \right\}.$$

Remark 4.6 (saturation effect on large N). Corollary 4.5 shows no further improvement from larger N once $N \geq n^{\frac{2(r\wedge 1/2)+1+p}{\alpha}-1}$, since the rates then only depend on n (as outlined in case B). This is due to a saturation effect from the bias-variance trade-off, i.e., N only helps decrease the variance term below the best achievable bias; at that point the bias (within each task) can only be further improved by larger per-task sample size n.

Remark 4.7 (regime $N \gtrsim \exp(n)$). The regimes presented in Corollary 4.5 only cover settings where $N \lesssim \exp(n)$, which is in fact the only regime covered by previous works (see, for instance, [10, 38]). This is due to the constraints $n \geq A_\lambda, B_\lambda$, which prevent $N \gtrsim \exp(n)$. However, at the cost of a less tight rate we can obtain a bound on the pre-training error

that is free of any constraint on n (see section SM1.6). As a corollary of this theorem, when $N \gtrsim \exp(n)$, choosing $\lambda = n^{-\frac{1}{2}}$ results in the nontrivial rate

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \lesssim \sqrt{\frac{s}{n_T}} + n^{-\frac{r}{2}}.$$

Notice that this is a slower rate than shown for smaller N in regime B of Corollary 4.5. Tightening the rates in the regime of $N \gtrsim \exp(n)$ appears difficult and is left as an open problem. We emphasize, as stated earlier, that this regime is in fact not addressed by previous works, even under the stronger assumption of linear representations.

Regimes of gain. We want to contrast our results in the meta-learning setting with the rates obtainable on the target task without the benefits of source tasks. Since no regularity condition is imposed on the target distribution, the best target rate, absent any source tasks, is of the form $O(n_T^{-1/4})$ (see, e.g., [7]);[2] thus we gain from the source tasks whenever $\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) = o(n_T^{-1/4})$.

Our interest, however, is in *regimes where the gain is greatest,* in that the source tasks permit a final meta-learning rate of $\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \lesssim \sqrt{s/n_T}$; Corollary 4.5 displays such regimes according to the number of source samples N and n, and the parameters r, α, and p, denote the difficulty of the source tasks. While it is clear that larger r indicates *smoother* source regression functions f_i as viewed from within the RKHS $\mathcal{H}$, smaller parameters α and p can be understood as a *smoothness level* of the RKHS $\mathcal{H}$ itself—e.g., consider a Sobolev space $\mathcal{H}$ of m-smooth functions; then we may take $\alpha, p \propto 1/m$ (see Example 3). Thus the smoother the source tasks, viewed under r, α, and p, the faster rates we can expect, since our approach aims at reducing the bias in each individual task (which is easiest under smoothness; see Remark 4.8 below).

Focusing on the situation where the number of samples per task is roughly the same across source and target, i.e., $n \propto n_T$, the conditions for meta-learning to provide the greatest gain, i.e., achieving $O(n^{-1/2})$ rate, under various regimes, are listed in Table 1.

Remark 4.8 (under-regularization/overfitting). In order for meta-learning to provide gain, in particular for $n \propto n_T$, we have to *overfit* the regression estimates in each source task, i.e., set

Table 1

Conditions for meta-learning to reach the parametric rate $O(\sqrt{s/n})$; log terms are removed for clarity.

Cases	Range of source tasks	Choice of λ	Regimes of gain
A	$n^{\frac{2r+1+p}{2r}-1} \le N \le n^{\frac{2r+1+p}{\alpha}-1}$	$(nN)^{-\frac{1}{2r+1+p}}$	$\frac{\alpha}{2} \le r \le \frac{1}{2}$
A	$n^{\frac{2r+1+p}{2r}-1} \le N \le n^{\frac{2r+1+p}{p+1}-1}$	$(nN)^{-\frac{1}{2r+1+p}}$	$\frac{p+1}{2} \le r \le 1$
B.1	$n^{\frac{2r+1+p}{\alpha}-1} \le N \le o(e^n)$	$n^{-\frac{r}{\alpha}}$	$\frac{\alpha}{2} \le r \le \frac{1}{2}$
B.2	$n^{\frac{2r+1+p}{p+1}-1} \le N \le o(e^n)$	$n^{-\frac{r}{p+1}}$	$\frac{p+1}{2} \le r \le 1$

[2]Note that the assumption that f_T is in some subspace $\mathcal{H}_s$ is irrelevant for usual kernel ridge regression, since it is always true once we know that f belongs to $\mathcal{H}$.

λ lower than would have been prescribed for optimal regression (choices of λ for the different regimes of gain are summarized in Table 1).

Overfitting is essential because, as highlighted in the introduction, the bias inherent in each task during meta-learning cannot be averaged out. Deliberate under-regularization reduces this bias at the expense of increased variance within each task. However, the variance in the target task may subsequently be mitigated by aggregating across multiple tasks.

More specifically, in the regimes of gain discussed earlier, the choices of λ in Corollary 4.5 are consistently lower than the optimal regression choice of $\lambda_{KRR} \asymp n^{-\frac{1}{2(r\wedge 1/2)+1+p}}$ (see, e.g., Theorem 1 in [13]) in the well-specified regime. This deviation from the optimal regression setting indicates overfitting, which again reveals that understanding nonlinear meta-learning is fundamentally more difficult than the linear setting due to the bias term. This effect is similarly observed in nonparametric kernel regression when splitting the dataset and averaging estimators trained on each split of the dataset [43].

Remark 4.9 (regularity beyond regression). Notice that the choice of the regularization parameter in kernel ridge regression $\lambda_{KRR} \asymp n^{-\frac{1}{2(r\wedge 1/2)+1+p}}$ has no direct dependence on α: lower values of $0 < \alpha \leq 1$ yield no further benefit in regression once we assume $f_i \in \mathcal{H}$, as opposed to the misspecified setting where f_i lies outside $\mathcal{H}$.[3] By contrast, in meta-learning, we do benefit from considering α, as α governs both the threshold level at which the saturation effect on large N kicks in (see Remark 4.6) and the level of smoothness required for meta-learning to provide the greatest gain (See Table 1 and associated discussion). Ultimately, if $\alpha \to 0$, there is no saturation effect, and the rates always match the parametric rate $O(n^{-1/2})$. This indicates that subspace learning is a fundamentally different problem from ridge regression.

Characterizing α, p, and r. As discussed above, smaller parameters α and p and higher parameter r yield faster meta-learning rates. The next examples yield insights on these situations. Throughout, recall that by Lemma 10 in [13], we have $p \leq \alpha$, i.e., $p = \alpha$ is always admissible.

Example 1 (finite-dimensional kernels). Suppose $\mathcal{H}$ is finite dimensional, i.e., the covariance operators Σ_i each admit a finite number of eigenfunctions $e_{i,j}, j = 1, 2, \dots k$, for some $k \geq 1$. Then clearly the eigenfunctions $\{e_{i,j}\}$ are bounded[4] and Assumptions 3 and 4 hold for $\alpha, p = 0$. Furthermore, Assumption 5 holds for any value of r. In this regime,

$$\mathcal{E}_{\mu_T}(\hat{f}_{T,\lambda_*}) \lesssim \sqrt{\frac{s}{n_T}} + \sqrt{\frac{k}{\gamma_s^2 nN}} \log(nN). \tag{4.4}$$

See Remark SM1.8 in the supplementary material for the detailed derivations. As an example, for polynomial kernels $K(x, x') \doteq (x^\top x' + b)^m$ on compact domains $\mathcal{X} \subset \mathbb{R}^d$, we obtain $k = d^m$. Note that, since polynomial regression converges at rate $O(\sqrt{d^m/n_T})$ (see, e.g., [2, 8, 16, 44]), we can gain in meta-learning whenever the representation $\mathcal{H}_s$ is of dimension $s \ll d^m$.

Remark 4.10 (subspace learning guarantees in the linear setting). In the meta-learning model with linear representations, with d the dimension of the input points and s the dimension

[3] Note, however, that $p \leq \alpha$, and therefore a small α implies that we are in the small p regime (and the rates do depend on p).

[4] As we employ a bounded kernel, every function in the RKHS is bounded [33, Lemma 4.23].

of the subspace, [37, Theorem 5] provides an information-theoretic lower bound on the $\sin-\Theta$ distance $\|\hat{P}_\perp P\|$ of the order $\sqrt{\frac{ds}{nN}}$ valid for estimators that are functions of the nN data points. Assuming that the eigenvalues of C_N are well conditioned ($\gamma_s \asymp s^{-1}$), estimators with matching guarantees on the $\sin-\Theta$ distance are obtained in [10, 31]. By the previous example, if we employ a linear kernel on $\mathbb{R}^d$ and under the assumption $\gamma_s \asymp s^{-1}$, we obtain a subspace learning error (up to a log term) of $\sqrt{\frac{ds^2}{nN}}$, recovering the learning rate obtained in [37]. Generalizing the result of [37] to the nonlinear setting with a lower bound depending on the parameters (N, n, s, p, r, α) represents a significant and valuable direction for future research.

Example 2 (Gaussian kernel). Let $\mathcal{X} \subset \mathbb{R}d$ be a bounded set with Lipschitz boundary,[5] μ a distribution supported on $\mathcal{X} \times \mathbb{R}$, with marginal distribution uniform on $\mathcal{X}$, and let K be a Gaussian kernel. Then by Corollary 4.13 in [18], Assumption 4 is satisfied with any $\alpha \in (0, 1]$, implying that Assumption 3 is also satisfied with any $p \in (0, 1]$.

Example 3 (Sobolev spaces and Matérn kernels). Let $\mathcal{X} \subset \mathbb{R}d$, be a nonempty, open, connected, and bounded set with a C_∞-boundary. Let μ be a distribution supported on $\mathcal{X} \times \mathbb{R}$, with marginal equivalent to the Lebesgue measure on $\mathcal{X}$. Choose a kernel which induces a Sobolev space H^m of smoothness $m \in \mathbb{N}$ with $m > d/2$, such as the Matérn kernel (see, e.g., [18, Examples 2.2 and 2.6]). Then by Corollary 5 in [13], Assumption 3 is satisfied with $p = \frac{d}{2m}$, and Assumption 4 is satisfied for every $\alpha \in (\frac{d}{2m}, 1]$. Furthermore, it can be shown that Assumption 5 is satisfied if and only if the $\{f_i\}_{i=1}^N$ belong to a Sobolev space (with fractional smoothness) of smoothness $m(2r + 1)$ (see [13]).

5. Experimental results.

In this section, we report the results of experiments on simulated data to test the two main theoretical predictions of our paper: (1) with the proper number of tasks it is possible to learn at the parametric rate; (2) overfitting is beneficial for meta-learning. Consider the Sobolev space $\mathcal{H} = \{f : [0, 1] \to \mathbb{R}, f \text{ absolutely continuous}, f' \in L^2([0, 1]), f(0) = 0\}$, equipped with the inner product $\langle f, g\rangle_{\mathcal{H}} = \int_0^1 f'(x)g'(x)dx$. $\mathcal{H}$ is the RKHS associated to the kernel $K : [0, 1] \times [0, 1] \to \mathbb{R}, (x, x') \mapsto \min(x, x')$ [17]. For a fixed parameter $s \in \mathbb{N}$, we consider an orthonormal system (with respect to $\langle\cdot,\cdot\rangle_{\mathcal{H}}$) of s splines of degree 2 (i.e., piecewise quadratic functions with continuous derivative) $(\psi_1, \dots, \psi_s)$ as shown in Figure 1. We then take $\mathcal{H}_s = \text{span}\{\psi_1, \dots, \psi_s\}$ and $P = \sum_{j=1}^s \psi_j \otimes \psi_j$, the projection onto $\mathcal{H}_s$. Note that $P = VV^*$ with $V = [\psi_1, \dots, \psi_s]$. Any $\omega \in \mathbb{R}^s$ leads to an element of $\mathcal{H}_s$ as

$$f = \sum_{\ell=1}^s \omega_\ell \psi_\ell(x) = \sum_{\ell=1}^s \omega_\ell \langle \psi_\ell, K(x,\cdot)\rangle_{\mathcal{H}} = \langle g, PK(x,\cdot)\rangle_{\mathcal{H}}, \qquad g \doteq \sum_{\ell=1}^s \omega_\ell \psi_\ell.$$

To generate each task, we proceed as follows. For $i \in [N] \cup \{T\}$, $\omega_i \sim \mathcal{U}(\sqrt{s}\mathbb{S}^{s-1})$, $f_i = \sum_{\ell=1}^s \omega_{i,\ell}\psi_\ell$, for $j = 1, \dots, 2n$ (or $j = 1, \dots, n_T$ for the target task),

$$y_{i,j} = f_i(x_{i,j}) + \epsilon_{i,j}, \qquad x_{i,j} \sim \mathcal{U}(0, 1), \quad \epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2).$$

Throughout the experiments, σ is fixed to 0.1. In Figure 1, we display an example of a generated task for $s = 10$. Given an estimator $\hat{f}$ for the target task, we evaluate its performance

[5] For the definition of Lipschitz boundary see [19, Definition 3].

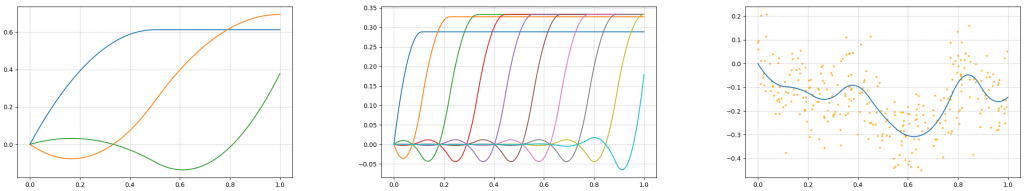

Figure 1. *Left–center: Orthonormal system in $\mathcal{H}$ spanning $\mathcal{H}_s$ for, respectively, $s = 3$ (left) and $s = 10$ (center). Right: Example of sampled task for $s = 10$ with 300 datapoints; the blue solid line represents the ground truth.*

by approximating the squared excess risk $\mathbb{E}_{\mu_T}[(\hat{f}(X)-f_T(X))^2]$ on independent samples, where μ_T is the Lebesgue measure on $[0,1]$.

Parameter values: p, α, and r. As the marginal probability distribution is the uniform measure on $[0,1]$ and K induces a Sobolev space of smoothness $m = 1$, by Example 3, Assumption 3 is satisfied with $p = \frac{1}{2}$ and Assumption 4 is satisfied with every $\alpha \in (\frac{1}{2}, 1]$. Finally, task functions are generated as linear combinations of order 2 splines and therefore belong to $H^m(0,1)$ for every $m < \frac{5}{2}$ (and do not belong to $H^m(0,1)$ for any $m \geq \frac{5}{2}$). By Example 3, Assumption 5 is therefore satisfied for every $r \in [0, \frac{3}{4})$ (and Assumption 5 is not satisfied for any $r \geq \frac{3}{4}$). In the experiments, we set $r = \frac{1}{2}$.

Choice of regularization. We focus on the *small number of tasks regime*, Corollary 4.5(A), where $N \leq n^{\frac{2r+1+p}{\alpha}-1} = n^4$. According to case A, we set $\lambda = (nN)^{-\frac{1}{2r+1+p}} = (nN)^{-\frac{2}{5}}$ and $\lambda_* = n_T^{-1}$. By Corollary 4.5, the excess risk on the target task is upper bounded (up to constants and log terms) by $\sqrt{s/n_T} + (nN)^{-\frac{1}{5}}$.

Learning at the parametric rate. We have shown in Table 1 that given enough source tasks and samples per source task it is possible to learn at the parametric rate $\sqrt{s/n_T}$. To illustrate this fact, we compare our meta-learning approach to an oracle estimator accessing the true subspace. The oracle estimator has access to $(\psi_1, \ldots, \psi_s)$ and is trained with linear ridge regression. For $x \in [0,1]$, define its transform $\tilde{x}^s \doteq (\psi_1(x), \ldots, \psi_s(x))^\top \in \mathbb{R}^s$. Then, $\hat{f}_{\text{oracle}}(x) \doteq \hat{\beta}^\top \tilde{x}^s$, with

$$\hat{\beta} = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^s} \frac{1}{n_T} \sum_{i=1}^{n_T} \left(y_{T,i} - \beta^\top \tilde{x}^s_{T,i} \right)^2 + \lambda_{\text{oracle}} \|\beta\|_2^2.$$

For $\lambda_{\text{oracle}} = n_T^{-1}$, $\mathcal{E}_{\mu_T}(\hat{f}_{\text{oracle}})$ is of the order $\sqrt{s/n_T}$ [30]. In Figure 2 (left), for $s = 25$ and $n = 300$ we show the evolution of the squared excess risk as we vary n_T for the oracle estimator and our meta-learning estimator trained with different values of N. Results are averaged over 100 runs, where for each run we sample new source and target tasks. For $N = 250$, the performance of the meta-learning is identical to that of the oracle. It demonstrates that our meta-learning strategy successfully leverages the source tasks and that given enough source tasks, it learns at a rate similar to that of the oracle estimator, leading to a parametric rate of convergence. We refer the reader to section SM4 for additional results.

Effect of overfitting. To assess the effect of overfitting (see Remark 4.8), we compare our meta-learning approach trained with $\lambda_1 = (nN)^{-\frac{2}{5}}$ and $\lambda_2 = n^{-\frac{2}{5}}$. In Figure 2 (right), for

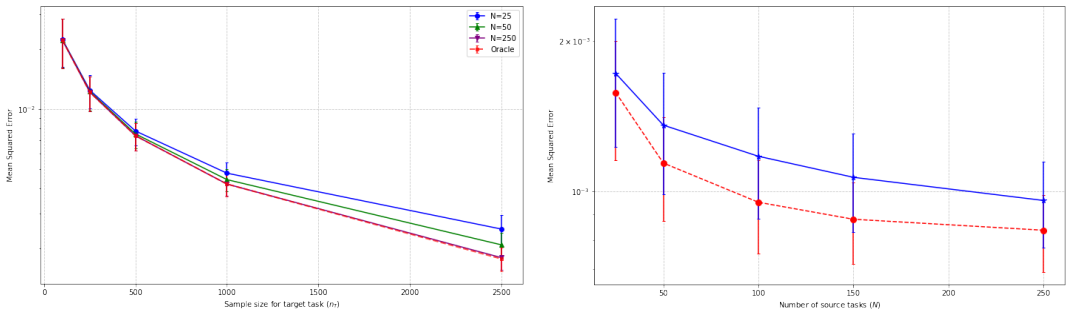

Figure 2. *Left: Meta-learning versus oracle: Comparison of the squared excess risk on the target task for the oracle estimator $\hat{f}_{oracle}$ (dotted red line) and the meta-learning estimator $\hat{f}_{T,\lambda_*}$ trained with different number of tasks N (solid lines). The x-axis represents the size of the dataset for the target task (n_T). Right: Effect of under-regularization: Comparison of the squared excess risk of the meta-learning estimator trained with $\lambda = (nN)^{-\frac{2}{5}}$ (red dotted line) and $\lambda = n^{-\frac{2}{5}}$ (blue solid line). The x-axis represents the number of source tasks (N). For both figures $n = 300$, $s = 25$, and results are averaged over 100 generations of the source and target tasks.*

$s = 25$, $n = 300$, and $n_T = 5000$, we plot the evolution of the squared excess risk as we increase N for λ_1 (red dotted line) and λ_2 (blue solid line). Results are averaged over 100 runs. They confirm the message of Remark 4.8 that overfitting (with respect to the usual regularization of kernel ridge regression) on each source task is beneficial for meta-learning. We refer the reader to section SM4 for additional results.

6. Analysis outline. To prove Theorem 4.3, we proceed with a bias-variance decomposition:

$$\|\hat{C}_{N,n,\lambda} - C_N\|_{HS} \le \underbrace{\|\hat{C}_{N,n,\lambda} - \bar{C}_{N,n,\lambda}\|_{HS}}_{\text{Variance}} + \underbrace{\|\bar{C}_{N,n,\lambda} - C_N\|_{HS}}_{\text{Bias}}, \tag{6.1}$$

where $\bar{C}_{N,n,\lambda} \doteq \frac{1}{N}\sum_i \mathbb{E}(\hat{f}_{i,\lambda}) \otimes \mathbb{E}(\hat{f}_{i,\lambda})$. Next we consider both of these terms separately.

- The variance term can be written as follows:

$$\|\hat{C}_{N,n,\lambda} - \bar{C}_{N,n,\lambda}\|_{HS} = \left\| \frac{1}{N}\sum_i^N \xi_i \right\|_{HS},$$

with $\xi_i \doteq \hat{f}'_{i,\lambda} \otimes \hat{f}_{i,\lambda} - \mathbb{E}(\hat{f}_{i,\lambda}) \otimes \mathbb{E}(\hat{f}_{i,\lambda}), i \in [N]$. Thus, the variance term being an average with mean 0, we would naturally want to bound it via a concentration inequality. However, this requires ξ_i to be well behaved, e.g., bounded or sub-Gaussian. A naive upper bound on $\|\xi_i\|_{HS}$ is of the order $\|\hat{f}'_{i,\lambda}\|_{\mathcal{H}} \cdot \|\hat{f}_{i,\lambda}\|_{\mathcal{H}} \le \lambda^{-1}$ (see Proposition SM2.9); however, this would lead to a loose concentration bound on the variance term; in particular, such a bound would not go down with the per-task sample size n.

Therefore, we first establish a high probability bound on $\|\xi_i\|_{HS}$ in terms of n and λ as follows. First, recall $f_{i,\lambda}$ from (2.2), and let $\eta_i \doteq \hat{f}'_{i,\lambda} \otimes \hat{f}_{i,\lambda} - f_{i,\lambda} \otimes f_{i,\lambda}$, whereby $\xi_i = \eta_i - \mathbb{E}[\eta_i]$. With some algebra we can get

$$\|\eta_i\|_{HS} \le \|\hat{f}'_{i,\lambda} - f_{i,\lambda}\|_{\mathcal{H}} \|\hat{f}_{i,\lambda} - f_{i,\lambda}\|_{\mathcal{H}} + \|f_i\|_{\mathcal{H}} (\|\hat{f}_{i,\lambda} - f_{i,\lambda}\|_{\mathcal{H}} + \|\hat{f}'_{i,\lambda} - f_{i,\lambda}\|_{\mathcal{H}}).$$

From existing results on kernel ridge regression (see, e.g., [13]), we can bound $\|\hat{f}_{i,\lambda} - f_{i,\lambda}\|_{\mathcal{H}}$ in terms of both n and λ, in high probability. This leads to a high probability bound on $\|\xi_i\|_{HS}$ that takes the form $\mathbb{P}(\|\xi_i\|_{HS} \le V(\delta,n,\lambda)) \ge 1 - 2e^{-\delta}$ for all $\delta \ge 0$ and $i \in [N]$ (see Theorem SM1.6 in section SM1.3 for details). Define the event $E_{N,\delta,n,\lambda} = \cap_{i\in[N]} E_{i,\delta,n,\lambda}$, where $E_{i,\delta,n,\lambda} \doteq \{\|\xi_i\|_{HS} \le V(\delta,n,\lambda)\}$. We then have

$$\mathbb{P}\left(\left\|\frac{1}{N}\sum_{i=1}^{N}\xi_i\right\|_{HS} \ge \epsilon\right) \le \mathbb{P}\left(\left\|\frac{1}{N}\sum_{i=1}^{N}\xi_i\right\|_{HS} \ge \epsilon \,\middle|\, E_{N,\delta,n,\lambda}\right) + 2Ne^{-\delta}. \tag{6.2}$$

For the first term on the right-hand side, we can now apply the Hoeffding inequality (Theorem SM3.6) since ξ_i conditionally on $E_{N,\delta,n,\lambda}$ is bounded. However, conditioning on $E_{N,\delta,n,\lambda}$, the variable ξ_i may no longer have zero mean, a requirement for usual concentration arguments. We therefore proceed by first centering ξ_i around $\mathbb{E}(\xi_i \mid E_{N,\delta,n,\lambda}) = \mathbb{E}(\xi_i \mid E_{i,\delta,n,\lambda})$ (by independence of the source tasks) and upper-bounding this expectation as

$$\|\mathbb{E}[\xi_i \mid E_{i,\delta,n,\lambda}]\| = \|\mathbb{E}(\xi_i \mid E_{i,\delta,n,\lambda}) - \mathbb{E}(\xi_i)\| \le 2\mathbb{E}\left[\|\xi_i\| \mid E^c_{i,\delta,n,\lambda}\right]\mathbb{P}\left(E^c_{i,\delta,n,\lambda}\right) \le 4e^{-\delta}\lambda^{-1},$$

where we used the upper bound λ^{-1} on $\|\xi_i\|_{HS}$. Then, applying the Hoeffding inequality to the first term, we obtain with probability greater than $1 - 2e^{-\tau} - 2Ne^{-\delta}$

$$\left\|\frac{1}{N}\sum_{i=1}^{N}\xi_i\right\|_{HS} \le V(\delta,n,\lambda)\sqrt{\frac{\tau}{N}} + \frac{4e^{-\delta}}{\lambda} \le V(\delta,n,\lambda)\sqrt{\frac{\tau}{N}} + \frac{4}{\lambda N^{12} n^{12}},$$

by choosing δ (a free parameter) as $12\log(nN)$. In that way, for our choices of λ (see Corollary 4.5), $(\lambda N^{12} n^{12})^{-1}$ is always of lower order and $2Ne^{-\delta} = o((nN)^{-10})$. Our choice of $V(\delta,n,\lambda)$ is given in Theorem SM2.6 (leading to (4.2)), with the constraint that $n \ge A_\lambda$ (see Theorem 4.3 for the definition of A_λ). For the detailed proof of the variance bound, please refer to Theorem SM1.6 in section SM1.3.

- To bound the bias, we first notice that it can be decomposed in the following way:

$$\|\bar{C}_{N,n,\lambda} - C_N\|_{HS} \lesssim \frac{1}{N}\sum_{i=1}^{N}\left\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\right\|_{\mathcal{H}}.$$

The key is therefore to obtain a good control on $\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\|_{\mathcal{H}}$. We consider two different ways of bounding this term, commensurate with regimes of r.

When $r \in (0, 1/2]$, we proceed as follows:

$$\begin{aligned}\left\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\right\|_{\mathcal{H}} &= \lambda\left\|\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\right) f_i\right\|_{\mathcal{H}} = \lambda\left\|\Sigma_{i,\lambda}^{-1/2}\mathbb{E}\left(I + \Sigma_{i,\lambda}^{-1/2}\left(\hat{\Sigma}_i - \Sigma_i\right)\Sigma_{i,\lambda}^{-1/2}\right)^{-1}\Sigma_{i,\lambda}^{-1/2} f_i\right\|_{\mathcal{H}} \\ &\le \lambda\left\|\Sigma_{i,\lambda}^{-1/2}\right\|\left\|\mathbb{E}\left(I + \Sigma_{i,\lambda}^{-1/2}\left(\hat{\Sigma}_i - \Sigma_i\right)\Sigma_{i,\lambda}^{-1/2}\right)^{-1}\right\|\left\|\Sigma_{i,\lambda}^{-1/2} f_i\right\|_{\mathcal{H}}.\end{aligned}$$

For $r \le 1/2$, we have $\left\|\Sigma_{i,\lambda}^{-1/2} f_i\right\|_{\mathcal{H}} = \left\|\Sigma_{i,\lambda}^{r-1/2}\Sigma_{i,\lambda}^{r} f_i\right\|_{\mathcal{H}} \le \lambda^{r-1/2}$, while $\left\|\Sigma_{i,\lambda}^{-1/2}\right\| \le \lambda^{-1/2}$. We then have

$$\left\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\right\|_{\mathcal{H}} \le \lambda^{r}\left\|\mathbb{E}\left(I + \Sigma_{i,\lambda}^{-1/2}\left(\hat{\Sigma}_i - \Sigma_i\right)\Sigma_{i,\lambda}^{-1/2}\right)^{-1}\right\|.$$

For $n \geq A_\lambda$, with probability over $1 - 2e^{-\delta}$—where δ is chosen as discussed for the variance bound—we can show that $\|(I + \Sigma_{i,\lambda}^{-1/2}(\hat{\Sigma}_i - \Sigma_i\Sigma_{i,\lambda}^{-1/2})^{-1}\| \leq 3$, whereby we get with the same probability $\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\|_{\mathcal{H}} \leq 3\lambda^r$. Thus, conditioning on this event, we get a final bound

$$\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\|_{\mathcal{H}} \leq 3\lambda^r + 2e^{-\delta}\|f_i\|_{\mathcal{H}},$$

using the fact that $\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\|_{\mathcal{H}} = \lambda\|\mathbb{E}(\hat{\Sigma}_{i,\lambda}^{-1})f_i\|_{\mathcal{H}}$ is always at most $\|f_i\|_{\mathcal{H}}$.

When $r \in (1/2, 1]$, we proceed as follows:

$$\begin{aligned}\left\|f_i - \mathbb{E}(\hat{f}_{i,\lambda})\right\|_{\mathcal{H}} &= \lambda\left\|\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\right)f_i\right\|_{\mathcal{H}} = \lambda\left\|\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\Sigma_{i,\lambda}\right)\Sigma_{i,\lambda}^{-1}f_i\right\|_{\mathcal{H}} \\ &\leq \lambda\left\|\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\Sigma_{i,\lambda}\right)\right\|\left\|\Sigma_{i,\lambda}^{r-1}\Sigma_{i,\lambda}^{-r}\Sigma_i^r\Sigma_i^{-r}f_i\right\|_{\mathcal{H}} \\ &\leq \lambda^r\left\|\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\Sigma_{i,\lambda}\right)\right\| = \lambda^r\left\|\Sigma_{i,\lambda}\mathbb{E}\left(\hat{\Sigma}_{i,\lambda}^{-1}\right)\right\|.\end{aligned}$$

We then use the following derivation:

$$\hat{\Sigma}_{i,\lambda}^{-1} = \left(\hat{\Sigma}_i + \lambda\right)^{-1} = \left(\Sigma_i + \lambda - (\Sigma_i - \hat{\Sigma}_i)\right)^{-1} = \Sigma_{i,\lambda}^{-1}\left(I - (\Sigma_i - \hat{\Sigma}_i)\Sigma_{i,\lambda}^{-1}\right)^{-1}.$$

We are left with bounding the term $\mathbb{E}\|(I - (\Sigma_i - \hat{\Sigma}_i)\Sigma_{i,\lambda}^{-1})^{-1}\|$, which can be obtained by using a Neumann series. For a detailed analysis of the bias, see Theorem SM1.7 of section SM1.3

7. Conclusion. We address the problem of meta-learning with nonlinear representations, providing theoretical guarantees for its effectiveness. Our study focuses on the scenario where the shared representation maps inputs nonlinearly into an infinite dimensional RKHS. By leveraging the smoothness of task-specific regression functions and employing careful regularization techniques, the paper demonstrates that biases introduced in the nonlinear representation can be mitigated. Importantly, the derived guarantees show that the convergence rates in learning the common representation can scale with the number of tasks, in addition to the number of samples per task. The analysis extends previous results obtained in the linear setting and highlights the challenges and subtleties specific to the nonlinear case. The findings presented in this work open up several avenues for future research, which include exploration of different types of nonlinear representations beyond RKHS, alternative subspace estimation techniques, and further refinement of trade-offs between bias and variance.

REFERENCES

[1] M. Aliakbarpour, K. Bairaktari, G. Brown, A. Smith, N. Srebro, and J. Ullman, *Metalearning with very few samples per task*, in Conference on Learning Theory, PMLR, 2024, pp. 46–93.

[2] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, *Learning sparse polynomial functions*, in Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 500–510, https://doi.org/10.1137/1.9781611973402.3.

[3] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, *Analysis of representations for domain adaptation*, in Advances in Neural Information Processing Systems 19, 2006.

[4] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer Science & Business Media, 2011.

[5] G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott, *Domain generalization by marginal transfer learning*, J. Mach. Learn. Res., 22 (2021), pp. 46–100.

[6] G. Blanchard and N. Mücke, *Optimal rates for regularization of statistical inverse learning problems*, Found. Comput. Math., 18 (2018), pp. 971–1013, https://doi.org/10.1007/s10208-017-9359-7.
[7] A. Caponnetto and E. De Vito, *Optimal rates for the regularized least-squares algorithm*, Found. Comput. Math., 7 (2007), pp. 331–368, https://doi.org/10.1007/s10208-006-0196-8.
[8] S. Chen and R. Meka, *Learning polynomials in few relevant dimensions*, in Conference on Learning Theory, PMLR, 2020, pp. 1161–1227.
[9] G. Denevi, C. Ciliberto, R. Grazzi, and M. Pontil, *Learning-to-learn stochastic gradient descent with biased regularization*, in International Conference on Machine Learning, PMLR, 2019, pp. 1566–1575.
[10] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei, *Few-shot learning via learning the representation, provably*, in International Conference on Learning Representations, 2021.
[11] C. Finn, P. Abbeel, and S. Levine, *Model-agnostic meta-learning for fast adaptation of deep networks*, in International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
[12] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, *Online meta-learning*, in International Conference on Machine Learning, PMLR, 2019, pp. 1920–1930.
[13] S. Fischer and I. Steinwart, *Sobolev norm learning rates for regularized least-squares algorithms*, J. Mach. Learn. Res., 21 (2020), 205.
[14] K. Fukumizu, F. R. Bach, and M. I. Jordan, *Kernel dimension reduction in regression*, Ann. Statist., 37 (2009), pp. 1871–1905, https://doi.org/10.1214/08-AOS637.
[15] T. Galanti, A. György, and M. Hutter, *Generalization Bounds for Transfer Learning with Pretrained Classifiers*, preprint, arXiv:2212.12532, 2022.
[16] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, *Linearized two-layers neural networks in high dimension*, Ann. Statist., 49 (2021), pp. 1029–1054, https://doi.org/10.1214/20-AOS1990.
[17] C. Gu and C. Gu, *Smoothing spline ANOVA models*, 2nd ed., Springer Ser. Statist. 297, Springer, 2013.
[18] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, *Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences*, preprint, arXiv:1807.02582, 2018.
[19] M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu, *Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings*, Found. Comput. Math., 20 (2020), pp. 155–194, https://doi.org/10.1007/s10208-018-09407-7.
[20] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, *Adaptive gradient-based meta-learning methods*, in Advances in Neural Information Processing Systems, 32, 2019.
[21] W. Kong, R. Somani, Z. Song, S. Kakade, and S. Oh, *Meta-learning for mixed linear regression*, in International Conference on Machine Learning, PMLR, 2020, pp. 5394–5404.
[22] M. Konobeev, I. Kuzborskij, and C. Szepesvári, *A distribution-dependent analysis of meta learning*, in International Conference on Machine Learning, PMLR, 2021, pp. 5697–5706.
[23] B. Li, A. Artemiou, and L. Li, *Principal support vector machines for linear and nonlinear sufficient dimension reduction*, Ann. Statist., 39 (2011), pp. 3182–3210, https://doi.org/10.1214/11-AOS932.
[24] B. Li and Y. Dong, *Dimension reduction for nonelliptically distributed predictors*, Ann. Statist., 37 (2009), pp. 1272–1298, https://doi.org/10.1214/08-AOS598.
[25] C. Ma, R. Pathak, and M. J. Wainwright, *Optimally tackling covariate shift in RKHS-based nonparametric regression*, Ann. Statist., 51 (2023), pp. 738–761, https://doi.org/10.1214/23-AOS2268.
[26] Y. Mansour, M. Mohri, and A. Rostamizadeh, *Domain adaptation: Learning bounds and algorithms*, in Conference on Learning Theory, PMLR, 2009.
[27] A. Maurer, *A chain rule for the expected suprema of Gaussian processes*, in Proceedings of the 25th International Conference on Algorithmic Learning Theory, 2014.
[28] A. Maurer, M. Pontil, and B. Romera-Paredes, *The benefit of multitask representation learning*, J. Mach. Learn. Res., 17 (2016), 81.
[29] D. Meunier and P. Alquier, *Meta-strategy for learning tuning parameters with guarantees*, Entropy, 23 (2021), 1257, https://doi.org/10.3390/e23101257.
[30] J. Mourtada and L. Rosasco, *An elementary analysis of ridge regression with random design*, C. R. Math. Acad. Sci. Paris, 360 (2022), pp. 1055–1063.
[31] X. Niu, L. Su, J. Xu, and P. Yang, *Collaborative Learning with Shared Linear Representations: Statistical Rates and Optimal Algorithms*, preprint, arXiv:2409.04919, 2024.

[32] B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet, *Universality, characteristic kernels and RKHS embedding of measures*, J. Mach. Learn. Res., 12 (2011), pp. 2389–2410.
[33] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer Science & Business Media, 2008.
[34] I. Steinwart and C. Scovel, *Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs*, Constr. Approx., 35 (2012), pp. 363–417, https://doi.org/10.1007/s00365-012-9153-3.
[35] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
[36] K. K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh, *Statistically and computationally efficient linear meta-representation learning*, in Advances in Neural Information Processing Systems 34, 2021, pp. 18487–18500.
[37] N. Tripuraneni, C. Jin, and M. Jordan, *Provable meta-learning of linear representations*, in International Conference on Machine Learning, PMLR, 2021, pp. 10434–10443.
[38] N. Tripuraneni, M. Jordan, and C. Jin, *On the theory of transfer learning: The importance of task diversity*, in Advances in Neural Information Processing Systems 33, 2020, pp. 7852–7862.
[39] P.-Å. Wedin, *Perturbation bounds in connection with singular value decomposition*, BIT, 12 (1972), pp. 99–111, https://doi.org/10.1007/BF01932678.
[40] Q. Wu, F. Liang, and S. Mukherjee, *Regularized Sliced Inverse Regression for Kernel Models*, Technical report, Citeseer, 2007.
[41] X. Yin, B. Li, and R. D. Cook, *Successive direction extraction for estimating the central subspace in a multiple-index regression*, J. Multivariate Anal., 99 (2008), pp. 1733–1757, https://doi.org/10.1016/j.jmva.2008.01.006.
[42] O. K. Yüksel, E. Boursier, and N. Flammarion, *First-order anil provably learns representations despite overparametrisation*, in International Conference on Learning Representations, 2024.
[43] Y. Zhang, J. Duchi, and M. Wainwright, *Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates*, J. Mach. Learn. Res., 16 (2015), pp. 3299–3340.
[44] R. Zippel, *Probabilistic algorithms for sparse polynomials*, in International Symposium on Symbolic and Algebraic Manipulation, Springer, 1979, pp. 216–226.