Implicit Kernel Meta-Learning Using Kernel Integral Forms (Supplementary Material)

John Isak Texas Falk[1, 2] **Carlo Ciliberto**[1] **Massimiliano Pontil**[1, 2]

[1]Dept. of Computer Science, University College London, U.K.
[2]CSML, Italian Institute of Technology, Genoa, Italy

The supplementary material is organized as follows. In Sec. 1 we introduce a glossary of terms used in the main body. In Sec. 2 we derive the closed form of the stochastic kernel of the affine pushforward kernel. In Sec. 3 we derive the detailed bounds presented in Theorem 1. In Sec. 4 we elaborate on the creation of the Air Quality (4.1) and the Gas Sensor (4.2) datasets. In Sec. 5 we include the information on the numerical experiment presented in the main body. Finally, in Sec. 6 we comment on the computational complexity of IKML and compare it against that of R2D2 since they both rely on KRR as the inner algorithm.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

1 GLOSSARY

Notation	Description
$\mathcal{X}$	input space
$\mathcal{Y}$	output space
$\mathcal{Z}$	data space / latent space
$\mathcal{P}(\mathcal{Z})$	set of distributions with support on $\mathcal{Z}$
ℓ	inner loss
$\mathcal{R}_\mu(f)$	risk of estimator f with respect to distribution μ
$\hat{\mathcal{R}}(f, D)$	empirical risk of f on dataset D
D^{tr}	train set
D^{val}	validation set
$A(\theta, D)$	inner algorithm with hyperparameter θ evaluated on dataset D
$L(\theta, D)$	meta-loss of θ on $D = D^{\text{tr}} \cup D^{\text{val}}$
K	kernel
$\mathcal{H}, \mathcal{H}_K$	Hilbert space with corresponding kernel K
$\langle\cdot,\cdot\rangle_K, \langle\cdot,\cdot\rangle_{\mathcal{H}_K}$	RKHS inner product of RKHS $\mathcal{H}_K$
K_τ	Bochner kernel with measure τ
ω	frequency in random feature kernel
ψ_θ	pushforward function parameterized by θ
$\mathcal{N}$	latent distribution
τ_θ	pushforward $\psi_\theta \# \mathcal{N}$
K_{τ_θ}	bochner kernel using using pushforward τ_θ
$K_{\hat{\tau}_{\theta S}}$	random feature kernel using sample $S \sim \tau_\theta^M$
$A_{\text{KRR}}(K, D)$	estimator of KRR on dataset D using kernel K
$L(\theta, S, D)$	meta-loss when using inner algorithm $A_{\text{KRR}}(K_{\hat{\tau}_{\theta S}}, \cdot)$ on the dataset $D = D^{\text{tr}} \cup D^{\text{val}}$
$\tilde{L}(\theta, S, D)$	train error on D when using KRR with random feature kernel $K_{\hat{\tau}_{\theta S}}$
ρ	meta-distribution
$\mathcal{E}(\theta)$	transfer risk of hyperparameter θ
$\mathcal{E}_M(\theta)/\mathcal{E}(\theta, S)$	transfer risk of θ when using an M-sample random feature kernel $K_{\hat{\tau}_{\theta S}}$ and KRR averaged over S
$\hat{\mathcal{E}}(\theta, S)$	estimation error for future task
$\hat{\mathcal{E}}_T(\theta, S)$	average train error (multitask empirical risk) on $(D_t)_{t=1}^T$ when using KRR with random feature kernel $K_{\hat{\tau}_{\theta S}}$
$\hat{\theta}$	multitask empirical risk minimizer when using bochner kernel
n, M, T	dataset size, number of random features, number of datasets
λ	regularization strength in KRR
$R_{n,M,T}$	Rademacher complexity $\mathbb{E}_\epsilon \sup_{\theta\in\Theta} \epsilon_{i,j,t} \langle \psi_\theta(s_j), x_i^t \rangle$
G_n^*	complexity term $\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{D\sim\mu^n} \|(K_{\tau_{\theta^*}}(x_i, x_j))_{i,j=1}^n\|_\infty$ measuring alignment of kernel $K_{\tau_{\theta^*}}$ with ρ

2 KERNEL FOR AFFINE PUSHFORWARD AND GAUSSIAN LATENT

In this section we give the closed form of the kernel when the distribution τ is the affine pushforward of a standard Gaussian.

We use the following trick to find the closed form kernel. We can rewrite the kernel in Bochner's theorem as

$$K(x, x') = \int \cos(\langle \omega, x - x' \rangle) \mathrm{d}\tau(\omega) = \int \Re(\cos(\langle \omega, x - x' \rangle) + i \sin(\langle \omega, x - x' \rangle)) \mathrm{d}\tau(\omega) \tag{1}$$

$$= \int \Re \exp(i \langle \omega, x - x' \rangle) \mathrm{d}\tau(\omega) \tag{2}$$

$$= \Re \int \exp(i \langle \omega, x - x' \rangle) \mathrm{d}\tau(\omega) \tag{3}$$

so finding the kernel is the same as finding the real part of the characteristic function (CF) of τ. For a Gaussian the CF is well-known and we give it below.

Lemma 1. *Let $\omega \sim \tau = \mathcal{N}(\mu, \Sigma)$ where Σ is pd, then for any $\Delta \in \mathbb{R}^d$*

$$\int_{\mathbb{R}^d} \exp(i\omega^\top \Delta) \mathrm{d}\tau(\omega) = \exp(i\mu^\top \Delta - \frac{1}{2}\Delta^\top \Sigma \Delta). \tag{4}$$

Proof. The pdf of ω is $f(\omega) = (2\pi)^{-d/2}|\det(\Sigma)|^{-1/2}\exp(-\frac{1}{2}(\omega-\mu)\Sigma^{-1}(\omega-\mu))$. We make the change of variable $\phi = \Sigma^{-1/2}(\omega - \mu)$ so $\omega = \Sigma^{1/2}\phi + \mu$ where $\Sigma^{1/2}$ and $\Sigma^{-1/2}$ exist due to Σ being pd. This means that $\mathrm{d}\omega = |\det(\Sigma)|^{1/2}\mathrm{d}\phi$ so that we have

$$\begin{aligned}
\int_{\mathbb{R}^d} \exp(i\omega^\top \Delta)\mathrm{d}\tau(\omega) &= \int_{\mathbb{R}^d} \exp(i\omega^\top \Delta) f(\omega)\mathrm{d}\omega \\
&= (2\pi)^{-d/2} \int_{\mathbb{R}^d} \exp(i(\Sigma^{1/2}\phi + \mu)^\top \Delta) \exp(-\frac{1}{2}\phi^\top \phi)\mathrm{d}\phi \\
&= (2\pi)^{-d/2} \exp(i\mu^\top \Delta) \int_{\mathbb{R}^d} \exp(i\phi^\top \Sigma^{1/2}\Delta) \exp(-\frac{1}{2}\phi^\top \phi)\mathrm{d}\phi \\
&= (2\pi)^{-d/2}(2\pi)^{d/2} \exp(i\mu^\top \Delta - \frac{1}{2}\Delta^\top \Sigma \Delta) = \exp(i\mu^\top \Delta - \frac{1}{2}\Delta^\top \Sigma \Delta).
\end{aligned}$$

□

Now we parameterize τ using $S \sim \mathcal{N}$ and $\theta = (Q, b)$ with $Q \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$ so that $\tau = \psi_{(Q,b)}\#\mathcal{N}$. An affine transformation of a Gaussian random variable is again Gaussian, and in this particular case it's easy to show that $\tau \sim \mathcal{N}(b, QQ^\top)$. Combining (3) and Lemma 1 we have

$$\begin{aligned}
K(x, x') &= \Re \int \exp(i\langle \omega, x - x'\rangle)\mathrm{d}\tau(\omega) && (5)\\
&= \Re \exp(ib^\top(x - x') - \frac{1}{2}(x - x')^\top QQ^\top (x - x')) && (6)\\
&= \cos(b^\top(x - x')) \exp(-\frac{1}{2}(x - x')^\top QQ^\top (x - x')) && (7)\\
&= \cos(b^\top(x - x')) \exp(-\frac{1}{2}\|Q^\top(x - x')\|^2). && (8)
\end{aligned}$$

3 ERROR DECOMPOSITION

3.1 SETUP

We follow the notation of [Maurer, 2009] with some modifications and note that this differs at places from the notation used in the main body of the paper. We recall the meta-learning setting. There is some meta-distribution ρ which generates tasks μ, from μ we are given a train set $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \sim \mu^n$, where $(x, y) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times [0, 1]$. Given the kernel ridge regression (KRR) algorithm with a fixed regularization parameter $\lambda > 0$ and an RKHS and corresponding kernel indexed by $\theta \in \Theta$, where $\Theta \subseteq \mathbb{R}^D$ is compact. We write this family as $\mathcal{H}_\Theta$ and the family of kernels as $\mathcal{K}_\Theta$. For a kernel K_θ let $\phi_\theta(x) = K_\theta(x, \cdot)$ be the canonical feature map and $\mathcal{H}_\theta$ the corresponding RKHS. The KRR solution is

$$\omega_\theta(\boldsymbol{z}) = \operatorname*{argmin}_{w \in \mathcal{H}_\theta} \left(\frac{1}{n} \sum_{i=1}^{n} (\langle w, \phi_\theta(x_i)\rangle_\theta - y_i)^2 + \lambda \|w\|_\theta^2 \right), \tag{9}$$

where we use $\langle \cdot, \cdot \rangle_\theta$ and $\|\cdot\|_\theta$ to denote the inner product and norm in RKHS $\mathcal{H}_\theta$. We will drop θ when it's clear what RKHS we are referring to. Given a weight vector $w \in \mathcal{H}_\theta$, a prediction on a new datapoint x is given by $\langle w, \phi_\theta(x)\rangle$.

The transfer risk of the algorithm ω_θ and a loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is defined to be

$$\mathcal{E}(\theta) = \mathbb{E}_{\mu\sim\rho} \mathbb{E}_{\boldsymbol{z}\sim\mu^n} \mathbb{E}_{(x,y)\sim\mu} \ell(\langle \omega_\theta(\boldsymbol{z}), \phi_\theta(x)\rangle, y). \tag{10}$$

We have access to T datasets from tasks by sampling $(\mu_t)_{t=1}^T \sim \rho^T$ which gives rise to datasets $\boldsymbol{z}^t = (\boldsymbol{x}^t, \boldsymbol{y}^t) \sim \mu_t^n$. For the meta-dataset $\boldsymbol{Z} = (\boldsymbol{z}^t)_{t=1}^T$ sampled by first sampling $(\mu_t)_{t=1}^T \sim \rho^T$ and then $\boldsymbol{z}^t \sim \mu^n$, we denote this sampling process by $\boldsymbol{Z} \sim \hat{\rho}^T$. Using the KRR algorithm ω_θ we let

$$\hat{\ell}_\theta(\boldsymbol{z}^t) = \frac{1}{n}\sum_{i=1}^n \ell(\langle \omega_\theta(\boldsymbol{z}^t), \phi_\theta(x_i^t)\rangle, y_i^t), \tag{11}$$

which is the training error of task t using ω_θ. For a sample of latent variables $S = (s_k)_{k=1}^M \sim \mathcal{N}^M$ so that the random features $\psi_\theta(s_k) \sim \tau_\theta$ (that is $\tau_\theta = \psi_\theta \# \mathcal{N}$), in which case we define $K_\theta = K_{\tau_\theta}$ and $K_{\theta,S} = K_{\hat{\tau}_\theta S}$, we let

$$\hat{\ell}_\theta(\boldsymbol{z}^t, S) = \frac{1}{n}\sum_{i=1}^n \ell(\langle \omega_{\theta,S}(\boldsymbol{z}^t), \phi_{\theta,S}(x_i^t)\rangle, y_i^t), \tag{12}$$

where $\omega_{\theta,S}$ is the same as (9) where we replace the kernel K_θ by the random feature kernel $K_{\theta,S}$ and the corresponding RKHS, see Sec. 3.2. When the algorithm ω is clear from context we simply write $\hat{\ell}(\boldsymbol{z})$ and $\hat{\ell}(\boldsymbol{z}, S)$. We opt to select θ using ERM, letting

$$\hat{\theta} = \operatorname*{argmin}_{\theta\in\Theta} \left\{ \hat{\mathcal{E}}_T(\theta) = \frac{1}{T}\sum_{t=1}^T \hat{\ell}_\theta(\boldsymbol{z}^t) \right\}. \tag{13}$$

As the problem of (13) is non-convex we cannot solve it in general. We let $\tilde{\theta}$ be the output of an optimization procedure $\tilde{\theta} = \mathrm{Alg}(\hat{\mathcal{E}}_T)$ and encode this optimization discrepancy through the term $\hat{\mathcal{E}}_T(\tilde{\theta}) - \hat{\mathcal{E}}_T(\hat{\theta})$.

3.2 KERNEL FAMILY

Let $\mathcal{H}$ be an RKHS defined by Bochner's theorem through the kernel defined by any probability measure $\tau \in M_1(\mathcal{X})$,

$$K(x, x') = \int \xi(x; v)\overline{\xi}(x'; v)\mathrm{d}\tau(v). \tag{14}$$

We will assume that $\xi(x; v) = \exp(iv^\top x)$, but the analysis should generalize to the more general setting. For a real-valued kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, it can be shown that any such kernel satisfying (14) can be rewritten as

$$K(x, x') = \int_{(-\pi/2,\pi/2]^d} \cos(\langle v, x - x'\rangle)\mathrm{d}\tau'(v), \tag{15}$$

for some measure τ' with support on $(-\pi/2, \pi]^d$. For IKML we parameterise a class of measures by $\psi_\theta \# \mathcal{N}$ where ψ_θ is an MLP with weights θ and we denote the kernel and RKHS by K_θ and $\mathcal{H}_\theta$.

Given a dataset of inputs $\boldsymbol{x} = (x_i)_{i=1}^n$, denote the kernel matrix $\boldsymbol{G}_\theta(\boldsymbol{x})$ so that $\boldsymbol{G}_\theta(\boldsymbol{x})_{ij} = K_\theta(x_i, x_j)$ and let $\boldsymbol{G}_{\theta,\lambda}(\boldsymbol{x}) = \boldsymbol{G}_\theta(\boldsymbol{x}) + n\lambda I$. Similarly for a set of latents $S \sim \mathcal{N}^M$ we denote the respective matrices $\boldsymbol{G}_\theta(\boldsymbol{x}, S)$ and $\boldsymbol{G}_{\theta,\lambda}(\boldsymbol{x}, S)$ were we replace every instance of

$$K_\theta(x, x') = \int \cos(\langle \psi_\theta(s), x - x'\rangle)\mathrm{d}\mathcal{N}(s) \tag{16}$$

by the empirical mean

$$K_{\theta,S}(x, x') = \frac{1}{M}\sum_{j=1}^M \cos(\langle \psi_\theta(s_j), x - x'\rangle) = \phi_{\theta,S}(x)^\top \phi_{\theta,S}(x'), \tag{17}$$

where $\phi_{\theta,S}(x) = \frac{1}{\sqrt{M}}(\sin(\psi_\theta(s_1)^\top x), \cos(\psi_\theta(s_1)^\top x), \ldots, \sin(\psi_\theta(s_M)^\top x), \cos(\psi_\theta(s_M)^\top x))^\top \in \mathbb{R}^{2M}$. We will omit θ and $\boldsymbol{x}$ from $\boldsymbol{G}_\theta(\boldsymbol{x})$ when clear from context. Similarly we let $\hat{\ell}_\theta(\boldsymbol{x}, \boldsymbol{y}, S)$ be the train loss when trained on $\boldsymbol{x}, \boldsymbol{y}$ with random features induced by S and we omit θ when clear from context.

3.3 AUXILIARY RESULTS

Let $\|\cdot\|_\infty$ be the operator norm and $\|\cdot\|_F$ the Frobenius norm. For an algorithm ω and a dataset $\boldsymbol{z}$, let $\hat{\ell}(\boldsymbol{z})$ be the training error of ω on $\boldsymbol{z}$ using loss ℓ.

Definition 1. *Given any $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ or two input sets $\boldsymbol{x}_1, \boldsymbol{x}_2$ of size n, where $x \in \mathcal{X}$ and $y \in [0,1]$, relative to a fixed loss function ℓ, an algorithm ω taking outputs in an RKHS $\mathcal{H}$ is said to be*

- *β-bounded if $\|\omega(\boldsymbol{z})\| \leq \beta$ and $\hat{\ell}(\boldsymbol{z}) \leq \beta$.*
- *have kernel stability L if*

$$\hat{\ell}(\boldsymbol{x}_1, \boldsymbol{y}) - \hat{\ell}(\boldsymbol{x}_2, \boldsymbol{y}) \leq \frac{L}{n} \|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F \tag{18}$$

- *have random feature stability L if*

$$\hat{\ell}(\boldsymbol{x}, \boldsymbol{y}, S) - \hat{\ell}(\boldsymbol{x}, \boldsymbol{y}) \leq \frac{L}{n} \|\boldsymbol{G}(\boldsymbol{x}, S) - \boldsymbol{G}(\boldsymbol{x})\|_F. \tag{19}$$

Lemma 2 ([Maurer, 2009], Lemma 3)**.** *Let G_1 and G_2 be positive semidefinite operators on any Hilbert space and $\lambda > 0$, then*

1. *$G_i + \lambda I$ is invertible,*
2. *$\|(G_i + \lambda I)^{-1}\|_\infty \leq \frac{1}{\lambda}$ and*
3. *we have*

$$\|(G_1 + \lambda I)^{-1} - (G_2 + \lambda)^{-1}\|_\infty \leq \frac{1}{\lambda^2} \|G_1 - G_2\|_\infty. \tag{20}$$

4. *Let ϕ_1, ϕ_2 satisfy $(G_i + \lambda I)\phi_i = y$. Then*

$$|\|\phi_1\|^2 - \|\phi_2\|^2| \leq 2\lambda^{-3} \|G_1 - G_2\|_\infty \|y\|^2 \tag{21}$$

For any dataset $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ of size n, kernel K with RKHS $\mathcal{H}$ and feature map ϕ, and corresponding KRR algorithm ω, we define the following quantities, following [Maurer, 2009],

$$\omega(\boldsymbol{z}) = \operatorname*{argmin}_{w \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} (\langle w, \phi(x_i) \rangle - y_i)^2 + \lambda \|w\|^2 \right), \tag{22}$$

$$\hat{\ell}_\omega(\boldsymbol{z}) = \frac{1}{n} \sum_{i=1}^{n} (\langle \omega(\boldsymbol{z}), \phi(x_i) \rangle - y_i)^2, \tag{23}$$

$$\xi_\omega(\boldsymbol{z}) = \min_{w \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} (\langle w, \phi(x_i) \rangle - y_i)^2 + \lambda \|w\|^2 \right) = \hat{\ell}_\omega(\boldsymbol{z}) + \lambda \|\omega(\boldsymbol{z})\|^2 \tag{24}$$

Proposition 3. *For any kernel K of the form* (15)*, for any dataset $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ or two input sets $\boldsymbol{x}_1, \boldsymbol{x}_2$, where $x \in \mathcal{X}, y \in [0,1]$, of size n and a sample of random features $S \sim \mathcal{N}^M$ we have that*

1. *$\hat{\ell}_\omega(\boldsymbol{z}) \leq 1$, $\|\omega(\boldsymbol{z})\| \leq \lambda^{-1/2}$, $\xi_\omega(\boldsymbol{z}) \leq 1$,*
2. *$|\hat{\ell}_\omega(\boldsymbol{x}_1, \boldsymbol{y}) - \hat{\ell}_\omega(\boldsymbol{x}_2, \boldsymbol{y})| \leq \frac{2\lambda^{-1}}{n} \|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F$,*
3. *$|\hat{\ell}_\omega(\boldsymbol{x}, \boldsymbol{y}) - \hat{\ell}_{\omega,S}(\boldsymbol{x}, \boldsymbol{y})| \leq \frac{2\lambda^{-1}}{n} \|\boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x}, S)\|_F$,*
4. *$|\xi_\omega(\boldsymbol{x}_1, \boldsymbol{y}) - \xi_\omega(\boldsymbol{x}_2, \boldsymbol{y})| \leq \frac{\lambda^{-1}}{n} \|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F$,*
5. *$|\xi_\omega(\boldsymbol{x}, \boldsymbol{y}) - \xi_{\omega,S}(\boldsymbol{x}, \boldsymbol{y})| \leq \frac{\lambda^{-1}}{n} \|\boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x}, S)\|_F$*

where $\boldsymbol{G}(\boldsymbol{x}, S)$ is the kernel matrix of $\boldsymbol{x}$ using random features induced by S.

Proof. We simply note that

$$\hat{\ell}_\omega(\boldsymbol{z}) + \lambda\|\omega(\boldsymbol{z})\|^2 = \xi_\omega(\boldsymbol{z}) = \min_{w\in\mathcal{H}}\left(\frac{1}{n}\sum_{i=1}^{n}(\langle w, \phi(x_i)\rangle - y_i)^2 + \lambda\|w\|^2\right) \leq \frac{1}{n}\sum_{i=1}^{n}(\langle 0, \phi(x_i)\rangle - y_i)^2 + \lambda\|0\|^2 \leq 1.$$

Since both $\hat{\ell}_\omega(\boldsymbol{z})$ and $\lambda\|\omega(\boldsymbol{z})\|^2$ are positive and the sum is less than 1, we have that $\hat{\ell}_\omega(\boldsymbol{z}) \leq 1$ and $\lambda\|\omega(\boldsymbol{z})\|^2 \leq 1$ which implies that $\|\omega(\boldsymbol{z})\| \leq \lambda^{-1/2}$.

For the second point, using the dual formulation $\boldsymbol{G}_\lambda(\boldsymbol{x})\alpha = \boldsymbol{y}$ and $\langle\omega(\boldsymbol{z}), \phi(x_i)\rangle = (\boldsymbol{G}(\boldsymbol{x})\alpha)_i$,

$$\hat{\ell}_\omega(\boldsymbol{z}) = \frac{1}{n}\|\boldsymbol{G}(\boldsymbol{x})\alpha - \boldsymbol{y}\|^2 = \frac{1}{n}\|\boldsymbol{G}_\lambda(\boldsymbol{x})\alpha - \boldsymbol{y} - \lambda n\alpha\|^2 = \frac{1}{n}\|\lambda n\alpha\|^2 = \lambda^2 n\|\alpha\|^2. \tag{25}$$

Using this and the fact that $\|\omega(\boldsymbol{z})\|^2 = \alpha^\top\boldsymbol{G}(\boldsymbol{x})\alpha$ in ξ_ω,

$$\begin{aligned}
\xi_\omega(\boldsymbol{z}) &= \hat{\ell}_\omega(\boldsymbol{z}) + \lambda\|\omega(\boldsymbol{z})\|^2 && (26)\\
&= \lambda^2 n\|\alpha\|^2 + \lambda\alpha^\top\boldsymbol{G}(\boldsymbol{x})\alpha && (27)\\
&= \lambda(\lambda n\alpha^\top\alpha + \alpha^\top\boldsymbol{G}(\boldsymbol{x})\alpha) = \lambda(\alpha^\top\boldsymbol{G}_\lambda(\boldsymbol{x})\alpha) = \lambda(\boldsymbol{y}^\top\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{y}). && (28)
\end{aligned}$$

Thus

$$\begin{aligned}
|\hat{\ell}_\omega(\boldsymbol{x}_1, \boldsymbol{y}) - \hat{\ell}_\omega(\boldsymbol{x}_2, \boldsymbol{y})| &= \lambda^2 n|\|\boldsymbol{G}_\lambda(\boldsymbol{x}_1)^{-1}\boldsymbol{y}\|^2 - \|\boldsymbol{G}_\lambda(\boldsymbol{x}_2)^{-1}\boldsymbol{y}\|^2| && (29)\\
&\leq (\lambda^2 n)2(\lambda n)^{-3}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_\infty\|\boldsymbol{y}\|^2 && (30)\\
&\leq 2\lambda^{-1}n^{-2}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_\infty\|\boldsymbol{y}\|^2 \leq \frac{2\lambda^{-1}}{n}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F, && (31)
\end{aligned}$$

where we have used point 4 in Lemma 2 and the fact that $\|\boldsymbol{y}\|^2 = \sum_{i=1}^n y_i^2 \leq n$ as $y_i \in [0, 1]$ for any $i \in [n]$. Then

$$\begin{aligned}
|\xi_\omega(\boldsymbol{x}_1, \boldsymbol{y}) - \xi_\omega(\boldsymbol{x}_2, \boldsymbol{y})| &\leq \lambda|\boldsymbol{y}^\top\boldsymbol{G}_\lambda(\boldsymbol{x}_1)^{-1}\boldsymbol{y} - \boldsymbol{y}^\top\boldsymbol{G}_\lambda(\boldsymbol{x}_2)^{-1}\boldsymbol{y}| && (32)\\
&\leq \lambda|\boldsymbol{y}^\top(\boldsymbol{G}_\lambda(\boldsymbol{x}_1)^{-1} - \boldsymbol{G}_\lambda(\boldsymbol{x}_2)^{-1})\boldsymbol{y}| && (33)\\
&\leq \lambda(\lambda n)^{-2}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_\infty\|\boldsymbol{y}\|^2 && (34)\\
&\leq \lambda n(\lambda n)^{-2}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F && (35)\\
&\leq \frac{\lambda^{-1}}{n}\|\boldsymbol{G}(\boldsymbol{x}_1) - \boldsymbol{G}(\boldsymbol{x}_2)\|_F. && (36)
\end{aligned}$$

For the third point and fifth point, the proof is the same as above, replacing $\boldsymbol{G}_\lambda(\boldsymbol{x_1})$ and $\boldsymbol{G}_\lambda(\boldsymbol{x_2})$ with $\boldsymbol{G}_\lambda(\boldsymbol{x})$ and $\boldsymbol{G}_\lambda(\boldsymbol{x}, S)$ respectively. Thus all of the results follows. □

Definition 2 (Complexities)**.** *Let $(\sigma_i)_{i=1}^k$ denote a sequence of independent Rademacher variables (Uniform distribution on $\{-1, 1\}$) independent of each other. For a set $A \subseteq \mathbb{R}^k$, the Rademacher and Gaussian complexities are defined to be*

$$\mathcal{R}(A) = \mathbb{E}_\sigma \sup_{\boldsymbol{x}\in A}\frac{2}{k}\sum_{i=1}^{k}\sigma_i x_i. \tag{37}$$

If $\mathcal{F}$ is a class of real functions on a space $\mathcal{X}$ and $\boldsymbol{x} \in \mathcal{X}^k$, we write

$$\mathcal{F}(\boldsymbol{x}) = \{(f(x_1), \ldots, f(x_k)) \,:\, f \in \mathcal{F}\} \subseteq \mathbb{R}^k. \tag{38}$$

The empirical Rademacher complexities of $\mathcal{F}$ on $\boldsymbol{x}$ is $\mathcal{R}(\mathcal{F}(\boldsymbol{x}))$. If $\mu \in M_1(\mathcal{X})$ is a probability measure on $\mathcal{X}$ then the corresponding expected complexity is $\mathbb{E}_{\boldsymbol{x}\sim\mu^k}\mathcal{R}(\mathcal{F}(\boldsymbol{x}))$.

Theorem 4 ([Maurer, 2009], Thm. 4)**.** *Let $\mathcal{F}$ be a real-valued function class on a space $\mathcal{X}$ and $\mu \in M_1(\mathcal{X})$. For $\boldsymbol{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$ define*

$$\Phi(\boldsymbol{x}) = \sup_{f\in\mathcal{F}}\left(\mathbb{E}_{x\sim\mu}f(x) - \frac{1}{k}\sum_{i=1}^{k}f(x_i)\right). \tag{39}$$

Then

1. $\mathbb{E}_{\boldsymbol{x}\sim\mu^k}\Phi(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{x}\sim\mu^k}\mathcal{R}(\mathcal{F}(\boldsymbol{x}))$,
2. *if $\mathcal{F}$ is $[0,1]$-valued, then for any $\delta > 0$ we have with probability greater than $1-\delta$ in $\boldsymbol{x} \sim \mu^k$ that*

$$\Phi(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{x}\sim\mu^k}\mathcal{R}(\mathcal{F}(\boldsymbol{x})) + \sqrt{\frac{\log(1/\delta)}{2k}}. \tag{40}$$

Corollary 5 ([Maurer, 2009], Corollary 1)**.** *Let $A \subseteq \mathbb{R}^k$ and $\phi_1, \ldots, \phi_k$ be real functions, each with Lipschitz constant L. Denote $\phi \circ A = \{(\phi_1(x_1), \ldots, \phi_k(x_k)) \,:\, (x_1, \ldots, x_k) \in A\}$. Then $\mathcal{R}(\phi \circ A) \leq L\mathcal{R}(A)$.*

3.4 DECOMPOSITION

We want to control the excess meta-risk $\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\hat{\theta}, S) - \mathcal{E}(\theta^*)]$, where $\theta^* = \operatorname{argmin}_{\theta\in\Theta}\mathcal{E}(\theta)$. We introduce the following terms

$$\hat{\mathcal{E}}(\theta) = \mathbb{E}_{\boldsymbol{Z}\sim\rho^T}\hat{\mathcal{E}}_T(\theta) \tag{41}$$

and the corresponding term $\hat{\mathcal{E}}(\theta, S)$ where we replace the kernel K_θ by $K_{\theta,S}$. We decompose the excess meta-risk as follows

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\hat{\theta}, S) - \mathcal{E}(\theta^*)] = \mathbb{E}_{S\sim\mathcal{N}^M}[\underbrace{\mathcal{E}(\hat{\theta}, S) - \hat{\mathcal{E}}(\hat{\theta}, S)}_{(A)} + \underbrace{\hat{\mathcal{E}}(\hat{\theta}, S) - \hat{\mathcal{E}}_T(\hat{\theta}, S)}_{(B)} + \underbrace{(\hat{\mathcal{E}}_T(\hat{\theta}, S) - \hat{\mathcal{E}}_T(\theta^*, S)}_{(C)} \tag{42}$$

$$+ \underbrace{\hat{\mathcal{E}}_T(\theta^*, S) - \hat{\mathcal{E}}(\theta^*, S)}_{(D)} + \underbrace{\hat{\mathcal{E}}(\theta^*, S) - \mathcal{E}(\theta^*, S)}_{(E)} + \underbrace{\mathcal{E}(\theta^*, S) - \mathcal{E}(\theta^*)}_{(F)}] \tag{43}$$

We bound each of the terms.

3.5 BOUNDING THE ESTIMATION ERROR FOR THE FUTURE TASK

This follows [Maurer, 2009, Sec. 4.1], but we present the results in the order that they are needed. This argument bounds both (A) and (E).

Theorem 6 (Upper bound of estimation error for future task)**.** *For any $\theta \in \Theta$, any loss ℓ such that for all $y \in [0,1]$ $\ell(\cdot, y) : [-L, L] \to \mathbb{R}_+$ has Lipschitz constant $\operatorname{Lip}(L)$, with ω being KRR with regularization parameter $\lambda > 0$ and RKHS induced by K_θ*

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\theta, S) - \hat{\mathcal{E}}(\theta, S)] \leq \operatorname{Lip}(\lambda^{-1/2})\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{z}\sim\hat{\rho}}\mathcal{R}(\mathcal{G}(\boldsymbol{z})), \tag{44}$$

where $\mathcal{G} = \{z = (x, y) \mapsto \lambda^{-1/2}\langle v, \phi_{\theta,S}(x)\rangle_{\theta,S} \,:\, \|v\|_{\theta,S} \leq 1\}$. Furthermore, we also have the upper bound

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\theta, S) - \hat{\mathcal{E}}(\theta, S)] \leq \frac{2\lambda^{-1/2}\operatorname{Lip}(\lambda^{-1/2})}{\sqrt{n}}. \tag{45}$$

Proof. We may rewrite term $\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\theta, S) - \hat{\mathcal{E}}(\theta, S)]$ as

$$\mathbb{E}_{S\sim\mathcal{N}^K}\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\langle\omega_{\theta,S}(\boldsymbol{x}, \boldsymbol{y}), \phi_{\theta,S}(x)\rangle, y) - \hat{\ell}_\theta(\boldsymbol{x}, \boldsymbol{y}, S)\right). \tag{46}$$

This is bounded in [Maurer, 2009, Thm. 6], and we follow similarly. For a fixed $\theta \in \Theta$ and any sample S, let $\mathcal{W} = \{w : \|w\|_{\theta,S} \leq \lambda^{-1/2}\}$. By proposition 3, we have that for any dataset $\boldsymbol{z}$ of size n generated according to our assumptions, for

any $\theta \in \Theta$, $\|\omega_{\theta,S}(\boldsymbol{z})\|_{\theta,S} \leq \lambda^{-1/2}$. Thus, for any $\mu \in M_1(\mathcal{X} \times [0,1])$,

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\langle\omega_{\theta,S}(\boldsymbol{x},\boldsymbol{y}),\phi_{\theta,S}(x)\rangle,y) - \hat{\ell}_\theta(\boldsymbol{x},\boldsymbol{y},S)\right) \tag{47}$$

$$\leq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\sup_{w\in\mathcal{W}}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\langle w,\phi_{\theta,S}(x)\rangle,y) - \frac{1}{n}\sum_{i=1}^n \ell(\langle w,\phi_{\theta,S}(x_i)\rangle,y_i)\right) \tag{48}$$

$$= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\sup_{\|w\|_{\theta,S}\leq\lambda^{-1/2}}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\langle w,\phi_{\theta,S}(x)\rangle,y) - \frac{1}{n}\sum_{i=1}^n \ell(\langle w,\phi_{\theta,S}(x_i)\rangle,y_i)\right) \tag{49}$$

$$= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\sup_{v\,:\,\|v\|_{\theta,S}\leq 1}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\lambda^{-1/2}\langle v,\phi_{\theta,S}(x)\rangle,y) - \frac{1}{n}\sum_{i=1}^n \ell(\lambda^{-1/2}\langle v,\phi_{\theta,S}(x_i)\rangle,y_i)\right) \tag{50}$$

$$= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\sup_{f\in\mathcal{F}}\left(\mathbb{E}_{(x,y)\sim\mu}f(z) - \frac{1}{n}\sum_{i=1}^n f(z_i)\right) \tag{51}$$

where we have the family of functions $\mathcal{F} = \{z = (x,y) \mapsto \ell(\lambda^{-1/2}\langle v,\phi_{\theta,S}(x)\rangle,y) \;:\; \|v\|_{\theta,S} \leq 1\}$. By Thm. 4 we can upper bound this by the Rademacher complexity, getting the upper bound

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\sup_{f\in\mathcal{F}}\left(\mathbb{E}_{(x,y)\sim\mu}f(z) - \frac{1}{n}\sum_{i=1}^n f(z_i)\right) \leq \mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\mathcal{F}(\boldsymbol{z})) \tag{52}$$

Furthermore, by assumption $y \in [0,1]$ and $\ell(\cdot,y)$ has a Lipschitz constant upper bounded by $\mathrm{Lip}(L)$ when we consider $\mathrm{dom}(\ell(\cdot,y)) = [-L,L]$. By Cauchy-Schwartz and $\|v\|_{\theta,S} \leq 1$, we have that $\lambda^{-1/2}\langle v,\phi_{\theta,S}(x)\rangle_{\theta,S} \in [-\lambda^{-1/2},\lambda^{1/2}]$ and so $L = \lambda^{-1/2}$. Letting $\phi_i : t \mapsto \ell(t,y_i)$ with domain $[-\lambda^{-1/2},\lambda^{-1/2}]$ then the Lipschitz constant is $\mathrm{Lip}(\lambda^{-1/2})$. We let $\mathcal{G} = \{z = (x,y) \mapsto \lambda^{-1/2}\langle v,\phi_{\theta,S}(x)\rangle_{\theta,S} \;:\; \|v\|_{\theta,S} \leq 1\}$.

Since $\mathcal{F} = \phi \circ \mathcal{G}$ we have by Cor. 5 that

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\left(\mathbb{E}_{(x,y)\sim\mu}\ell(\langle\omega_{\theta,S}(\boldsymbol{x},\boldsymbol{y}),\phi_{\theta,S}(x)\rangle,y) - \hat{\ell}_{\omega_{\theta,S}}(\boldsymbol{x},\boldsymbol{y})\right) \leq \mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\phi\circ\mathcal{G}(\boldsymbol{z})) \tag{53}$$

$$\leq \mathrm{Lip}(\lambda^{-1/2})\mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\mathcal{G}(\boldsymbol{z})). \tag{54}$$

We can further bound $\mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\mathcal{G}(\boldsymbol{z}))$ using a standard RKHS rademacher complexity argument.

By standard arguments of Rademacher complexity of kernels such that $K(x,x) = 1$ we have the bound

$$\mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\mathcal{G}(\boldsymbol{z})) \leq \frac{2\lambda^{-1/2}}{\sqrt{n}} \tag{55}$$

Substituting the upper bounds $\mathrm{Lip}(\lambda^{-1/2})\mathbb{E}_{\boldsymbol{z}\sim\mu^n}\mathcal{R}(\mathcal{G}(\boldsymbol{z}))$ or $\frac{2\lambda^{-1/2}\mathrm{Lip}(\lambda^{-1/2})}{\sqrt{n}}$ of (47) and combining everything we have

$$\mathbb{E}_{S\sim\mathcal{N}^k}[\mathcal{E}(\hat{\theta},S) - \hat{\mathcal{E}}(\hat{\theta},S)] \leq \mathrm{Lip}(\lambda^{-1/2})\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{z}\sim\hat{\rho}}\mathcal{R}(\mathcal{G}(\boldsymbol{z})) \tag{56}$$

$$\mathbb{E}_{S\sim\mathcal{N}^k}[\mathcal{E}(\hat{\theta},S) - \hat{\mathcal{E}}(\hat{\theta},S)] \leq \frac{2\lambda^{-1/2}\mathrm{Lip}(\lambda^{-1/2})}{\sqrt{n}} \tag{57}$$

□

We note the following about the bound above. The bound $\mathbb{E}_{\boldsymbol{z}\sim\hat{\rho}}\mathcal{R}(\mathcal{G}(\boldsymbol{z})) \leq \frac{2\lambda^{-1/2}}{\sqrt{n}}$ is standard and applies to all kernels such that $K(x,x) = 1$. However, in the benign case that $\mathbb{E}_{S\sim\mathcal{N}^M}\mathcal{R}(\mathcal{G}(\boldsymbol{z})) \ll \frac{2\lambda^{-1/2}\mathrm{Lip}(\lambda^{-1/2})}{\sqrt{n}}$, using IKML would lead to a term much smaller than fixing a kernel and learning the tasks independently.

3.6 PREDICTING THE EMPIRICAL ERROR FOR THE FUTURE TASK

In this section we focus on the terms $(B), (D)$, each of the form $\mathbb{E}_{S\sim\mathcal{N}^M}\hat{\mathcal{E}}_T(\theta,S)-\hat{\mathcal{E}}(\theta,S)$ where $\theta\in\Theta$. To control this term we use [Maurer, 2016, Sec. 4.2]. We want to control the term $\hat{\mathcal{E}}_T(\theta,S)-\hat{\mathcal{E}}(\theta,S)$ using a uniform bound of the form

$$\hat{\mathcal{E}}_T(\theta,S)-\hat{\mathcal{E}}(\theta,S)\leq\sup_{\theta\in\Theta}\left\{\hat{\mathcal{E}}_T(\theta,S)-\hat{\mathcal{E}}(\theta,S)=\frac{1}{T}\sum_{t=1}^{T}\hat{\ell}_\theta(\boldsymbol{z}^t,S)-\mathbb{E}_{\boldsymbol{z}\sim\hat{\rho}}\hat{\ell}_\theta(\boldsymbol{z}^t,S)\right\},\tag{58}$$

where $\boldsymbol{z}^t=(\boldsymbol{x}^t,\boldsymbol{y}^t)\sim\mu_t^n$ and we let $\boldsymbol{Z}=(\boldsymbol{z}^t)_{t=1}^T\sim\hat{\rho}^T$. This enables us to control both (B) involving the ERM parameter $\hat{\theta}$ and (D) involving θ^*. We switch the order of the terms to get the standard form $\sup_{f\in\mathcal{F}}\mathbb{E}f(x)-\frac{1}{n}\sum_{i=1}^n f(x_i)$. We define the loss class $\mathcal{F}=\{f:\mathcal{Z}^n\to\mathbb{R}_{\geq 0},\ f(\boldsymbol{z})=\hat{\ell}_\theta(\boldsymbol{z},S),\ \forall\theta\in\Theta\}$ and note that it implicitly depends on the random feature sample S.

Theorem 7. *For any $\theta\in\Theta$, squared error $\ell(y,\hat{y})=(y-\hat{y})^2$ such that for all $y\in[0,1]$, $\ell(\cdot,y):[-L,L]\to\mathbb{R}_+$ has Lipschitz constant $\mathrm{Lip}(L)$, with ω being KRR with regularization parameter $\lambda>0$ and RKHS induced by K_θ, with probability greater than $1-\delta$ over the choice of meta-train set $\overline{\boldsymbol{Z}}\in(\mathcal{Z}^n)^T$ we have*

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\hat{\mathcal{E}}_T(\theta,S)-\hat{\mathcal{E}}(\theta,S)]\leq\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}\mathcal{R}(\mathcal{F}(\boldsymbol{Z}))+\sqrt{\frac{\log(1/\delta)}{2T}}.\tag{59}$$

Proof. Relating our setting to [Maurer, 2016], for some S we let the loss function class be

$$\mathcal{F}=\{f:\mathcal{Z}^n\to\mathbb{R}_{\geq 0},\ f(\boldsymbol{z})=\hat{\ell}_\theta(\boldsymbol{z},S),\ \forall\theta\in\Theta\},\tag{60}$$

and for $\overline{\boldsymbol{Z}}\in(\mathcal{Z}^n)^T$, we denote $\Phi(\overline{\boldsymbol{Z}})=\sup_{f\in\mathcal{F}}\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}f(\boldsymbol{Z})-\frac{1}{T}\sum_{t=1}^T f(\overline{\boldsymbol{Z}}_t)$. By 2. of Thm. 4, since $\hat{\ell}_\theta(\boldsymbol{z},S)\in[0,1]$, for any $\delta>0$ we have with probability greater than $1-\delta$ over $\overline{\boldsymbol{Z}}\sim\hat{\rho}^T$ that

$$\Phi(\overline{\boldsymbol{Z}})\leq\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}\mathcal{R}(\mathcal{F}(\boldsymbol{Z}))+\sqrt{\frac{\log(1/\delta)}{2T}},\tag{61}$$

so we focus on controlling the Rademacher complexity $\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}\mathcal{R}(\mathcal{F}(\boldsymbol{Z}))$. Following [Maurer, 2016, Sec. 3.3], note that our notation differs, the translation is as follows ([Maurer, 2016, Sec. 3. 3] $\to$ this work) $\mathcal{H}\to\{\psi_{\theta,S}(\cdot),\ \forall\theta\in\Theta\}$, $\psi_t(\cdot)\to\hat{\ell}_{(\cdot),S}(\boldsymbol{z}^t)$, $\mathbb{E}_{t\sim\rho}[\psi_t(h)]\to\hat{\mathcal{E}}(\theta)$, $\frac{2}{n}R(\mathcal{H},\bar{\boldsymbol{x}})\to\mathcal{R}(\mathcal{F}(\overline{\boldsymbol{Z}}))$, $\phi_t(h)\to\Phi_{\theta,S}$, we will specify $\Phi_{\theta,S}$ below. To apply [Maurer, 2016, Thm. 2] we need find a "Lipschitz" constant L such that

$$\hat{\ell}_{\theta,S}(\boldsymbol{z}^t)-\hat{\ell}_{\theta',S}(\boldsymbol{z}^t)\leq\frac{L}{\sqrt{n}}\|\Phi_{\theta,S}-\Phi_{\theta',S}\|_F,\tag{62}$$

where $(\Phi_{\theta,S})_{:,j}=\phi_{\theta,S}(x_j)$ and $\Phi_{\theta,S}\in\mathbb{R}^{2M\times n}$, e.g. $\Phi_{\theta,S}$ is the feature matrix of the kernel $\boldsymbol{G}_\theta(\boldsymbol{x},S)$ so that $\boldsymbol{G}_\theta(\boldsymbol{x},S)=\Phi_{\theta,S}^\top\Phi_{\theta,S}$. We proceed as follows; assume that for loss $\ell(\cdot,\cdot)$ we have kernel stability

$$\hat{\ell}_{\theta,S}(\boldsymbol{z})-\hat{\ell}_{\theta',S}(\boldsymbol{z})\leq\frac{L_\ell}{n}\|\boldsymbol{G}_\theta(\boldsymbol{x},S)-\boldsymbol{G}_{\theta'}(\boldsymbol{x},S)\|_F,\tag{63}$$

this holds true for the least squares loss with $L_\ell=2\lambda^{-1}$ following similarly from the proof of proposition 3. Since $\Phi_{\theta,S}^\top\Phi_{\theta,S}=\boldsymbol{G}_\theta(\boldsymbol{x},S)$ for θ and similarly for θ' we can write

$$\|\boldsymbol{G}_\theta(\boldsymbol{x},S)-\boldsymbol{G}_{\theta'}(\boldsymbol{x},S)\|_F=\|\Phi_{\theta,S}^\top\Phi_{\theta,S}-\Phi_{\theta',S}^\top\Phi_{\theta',S}\|_F\tag{64}$$

and letting $M_{\max}=\max(\|\Phi_{\theta,S}\|_F,\|\Phi_{\theta',S}\|_F)$, using the matrix identity $A^\top A-B^\top B=A^\top(A-B)+(A-B)^\top B$, we can upper bound it

$$\|\Phi_{\theta,S}^\top\Phi_{\theta,S}-\Phi_{\theta',S}^\top\Phi_{\theta',S}\|_F\leq\|\Phi_{\theta,S}\|_F\|\Phi_{\theta,S}-\Phi_{\theta',S}\|_F+\|\Phi_{\theta',S}\|_F\|\Phi_{\theta',S}-\Phi_{\theta,S}\|_F\leq 2M_{\max}\|\Phi_{\theta',S}-\Phi_{\theta,S}\|_F.\tag{65}$$

Now

$$\|\Phi_{\theta,S}\|_F^2 = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{n}(\sin(\psi_\theta(s_k)^\top x_j)^2 + \cos(\psi_\theta(s_k)^\top x_j)^2) = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{n} 1 = n, \tag{66}$$

which means that $M_{\max} = \sqrt{n}$ and thus we have the sought Lipschitz property

$$\hat{\ell}_{\theta,S}(\boldsymbol{z}^t) - \hat{\ell}_{\theta',S}(\boldsymbol{z}^t) \leq \frac{L_\ell}{\sqrt{n}}\|\Phi_{\theta',S} - \Phi_{\theta,S}\|_F. \tag{67}$$

From this we have that

$$\mathcal{R}(\mathcal{F}(\overline{\boldsymbol{Z}})) \leq \frac{2L_\ell}{T\sqrt{nM}}\left(\mathbb{E}_\epsilon \sup_{\theta\in\Theta}\sum_{t,k,i}^{T,M,n}\epsilon_{tki}(\sin(\psi_\theta(s_k)^\top x_i^t) + \cos(\psi_\theta(s_k)^\top x_i^t))\right). \tag{68}$$

□

3.7 RANDOM FEATURE ERROR

In this section we show how to control the term (F) in the bound, the term

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\theta^*, S) - \mathcal{E}(\theta^*)]. \tag{69}$$

Remember that for a dataset $\boldsymbol{x} \in \mathcal{X}^n$, random features S and $\theta \in \Theta$, we let $\boldsymbol{g}(x) = (K_\theta(x_i, x))_{i=1}^n$ and similarly $\boldsymbol{g}(x, S) = (K_{\theta,S}(x_i, x))_{i=1}^n$.

We first need some additional results

Theorem 8 ([Tropp, 2019], Theorem 2.1)**.** *Let A be a self-adjoint matrix of size n. Given a iid samples $(R_k)_{k=1}^M$ of self-adjoint matrices such that $\mathbb{E}R_1 = A$ and $\|R_1\|_\infty \leq B$. Let $m_2(R_1) = \|\mathbb{E}R_1^2\|_\infty$ and $\bar{R}_M = \frac{1}{M}\sum_{k=1}^M R_M$, then*

$$\mathbb{E}\|\bar{R}_M - A\| \leq \sqrt{\frac{2m_2(R_1)\log(2n)}{M}} + \frac{2B\log(2n)}{3M}. \tag{70}$$

Lemma 9. *For any $\boldsymbol{x} \in \mathcal{X}^n$, any $\theta \in \Theta$, $\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x) - \boldsymbol{g}(x, S)\|_2 \leq 2n^{1/2}M^{-1/2}$*

Proof. We have that

$$\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x) - \boldsymbol{g}(x, S)\|_2 \leq \sqrt{\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x) - \boldsymbol{g}(x, S)\|_2^2} = \sqrt{\sum_{i=1}^n \mathbb{E}_{S\sim\mathcal{N}^M}(\boldsymbol{g}(x)_i - \frac{1}{M}\sum_{k=1}^M \boldsymbol{g}(x, s_k)_i)^2}. \tag{71}$$

Define $T_{i,k} = K_\theta(x, x_i) - K_{\theta,s_k}(x, x_i)$, then $\mathbb{E}(T_{i,k}) = 0$, for any $i \in [n]$, $(T_{i,k})_{k=1}^M$ is an iid sample of random variables and finally $|T_{i,k}| \leq |K_\theta(x, x_i)| + |K_{\theta,s_k}(x, x_i)| \leq 2$. We can then express

$$\sum_{i=1}^n \mathbb{E}_{S\sim\mathcal{N}^M}(\boldsymbol{g}(x)_i - \frac{1}{M}\sum_{k=1}^M \boldsymbol{g}(x, s_k)_i)^2 = \sum_{i=1}^n \mathbb{E}_{S\sim\mathcal{N}^M}\left(\frac{1}{M}\sum_{k=1}^M T_{i,k}\right)^2 = M^{-2}\sum_{i=1}^n \mathbb{E}_{S\sim\mathcal{N}^M}\left(\sum_{k=1}^M T_{i,k}\right)^2. \tag{72}$$

For arbitrary $i \in [n]$, we see that

$$\mathbb{E}_{S\sim\mathcal{N}^M}\left(\sum_{k=1}^M T_{i,k}\right)^2 \leq \sum_{k=1}^M \mathbb{E}_{s_k\sim\mathcal{N}} T_{i,k}^2 + 2\sum_{k<l}\mathbb{E}_{s_k,s_l\sim\mathcal{N}} T_{i,k}T_{i,l} \tag{73}$$

$$= \sum_{k=1}^M \mathbb{E}_{s_k\sim\mathcal{N}} T_{i,k}^2 + 2\sum_{k<l}\mathbb{E}_{s_k\sim\mathcal{N}} T_{i,k}\mathbb{E}_{s_l\sim\mathcal{N}} T_{i,l} = \sum_{k=1}^M \mathbb{E}_{s_k\sim\mathcal{N}} T_{i,k}^2, \tag{74}$$

where we used the fact that $T_{i,k}$ is zero-mean and for fixed $i \in [n]$, $T_{i,k}, T_{i,l}$ are independent. Since $|T_{i,k}| \leq 2$ we see that $|T_{i,k}|^2 \leq 4$, hence

$$\mathbb{E}_{S\sim\mathcal{N}^M}\left(\sum_{k=1}^{M} T_{i,k}\right)^2 \leq 4M. \tag{75}$$

Thus we see that

$$\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x) - \boldsymbol{g}(x,S)\|_2 \leq \sqrt{\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x) - \boldsymbol{g}(x,S)\|_2^2} \leq 2n^{1/2}M^{-1/2}. \tag{76}$$

□

Following [Tropp, 2019, Section 2.2] we have the following result

Lemma 10. *For any $\boldsymbol{x} \in \mathcal{X}^n$, any $\theta \in \Theta$, $\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x},S)\|_\infty \leq \sqrt{\frac{4\|\boldsymbol{G}(\boldsymbol{x})\|_\infty n\log(2n)}{M}} + \frac{2n\log(2n)}{4M}$*

Proof. Note that due to the identity $\cos(x-y) = \sin(x)\sin(y)+\cos(x)\cos(y)$ we have $K_\theta(x,y) = \mathbb{E}_{s\sim\mathcal{N}}\cos(\langle x, \psi_\theta(s)\rangle - \langle y, \psi_\theta(s)\rangle) = \mathbb{E}_{s\sim\mathcal{N}}\phi_{\theta,s}(x)^\top\phi_{\theta,s}(y)$. For the sample $\boldsymbol{x} = (x_i)_{i=1}^n$ let $Z_{k,i,:} = \phi_{\theta,s_k}(x_i)$ be the matrix of features corresponding to sample s_k, and let Z_k^l be the l'th column of Z_k. Thus we have that $\mathbb{E}Z_kZ_k^\top = \boldsymbol{G}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x},S) = \frac{1}{M}\sum_{k=1}^M Z_kZ_k^\top$.

To put this in the notation of Thm. 8 we let $\bar{R}_M = \boldsymbol{G}(\boldsymbol{x},S)$ and $A = \boldsymbol{G}(\boldsymbol{x})$. To invoke Thm. 8 need to upper bound the quantities $\|Z_kZ_k\|_\infty$ and $m_2(Z_kZ_k^\top)$. For the first term

$$\|Z_kZ_k^\top\|_\infty \leq \|Z_k\|_\infty\|Z_k\|_\infty \leq \|Z_k\|_F^2 = \sum_{i=1}^n \cos(\langle x_i, \psi_\theta(s_k)\rangle)^2 + \sin(\langle x_i, \psi_\theta(s_k)\rangle)^2 = n. \tag{77}$$

For the second term, consider $\mathbb{E}(Z_kZ_k^\top)^2$. We can rewrite this in the form $\mathbb{E}Z_kCZ_k^\top$ where $C = Z_k^\top Z_k$ hence symmetric and psd. We can write this as a sum

$$Z_kCZ_k^\top = C_{11}Z_k^1(Z_k^1)^\top + C_{22}Z_k^2(Z_k^2)^\top + C_{12}(Z_k^1(Z_k^2)^\top + Z_k^2(Z_k^1)^\top). \tag{78}$$

We can bound $0 \leq C_{11} = \|Z_k^1\|_2^2 \leq n, 0 \leq C_{22} = \|Z_k^2\|_2^2 \leq n$ and $|C_{12}| = |\langle Z_k^1, Z_k^2\rangle| \leq n$. Using the identity $ab^\top + ba^\top = \frac{1}{2}((a+b)(a+b)^\top - (a-b)(a-b)^\top)$ we can then express $Z_kZ_k^\top$ as a sum of four psd matrices (with possibly negative coefficients)

$$Z_kZ_k^\top = C_{11}Z_k^1(Z_k^1)^\top + C_{22}Z_k^2(Z_k^2)^\top + \frac{C_{12}}{2}(Z_k^1+Z_k^2)(Z_k^1+Z_k^2)^\top - \frac{C_{12}}{2}(Z_k^1-Z_k^2)(Z_k^1-Z_k^2)^\top, \tag{79}$$

then we see that we can majorize Z_kZ_k by the matrix $nZ_k^1Z_k^1 + nZ_k^2Z_k^2 + \frac{n}{2}(Z_k^1+Z_k^2)(Z_k^1+Z_k^2)^\top + \frac{n}{2}(Z_k^1-Z_k^2)(Z_k^1-Z_k^2)^\top$ in the Loewner order and expanding this majorant we see that $(Z_kZ_k^\top)^2 \preceq 2nZ_kZ_k^\top$ where we let $\preceq$ be the Loewner order on psd matrices. It follows that $m_2(Z_kZ_k^\top) = \|\mathbb{E}(Z_kZ_k^\top)^2\|_\infty \leq 2n\|\mathbb{E}Z_kZ_k^\top\|_\infty = 2n\|\boldsymbol{G}(\boldsymbol{x})\|_\infty$. □

We are now ready to state the theorem of the random feature error

Theorem 11 (Random feature error)**.** *For any $\theta \in \Theta$, any loss ℓ such that for all $y \in [0,1]$, $\ell(\cdot,y) : [-L,L] \to \mathbb{R}_+$ has Lipschitz constant $\mathrm{Lip}(L)$, with ω being KRR with regularization parameter $\lambda > 0$ and RKHS induced by K_θ we have that*

$$\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\theta,S) - \mathcal{E}(\theta)] \leq 2\mathrm{Lip}(\lambda^{-1/2})\lambda^{-1}M^{-1/2} \tag{80}$$

$$+ 2\mathrm{Lip}(\lambda^{-1/2})\lambda^{-2}M^{-1/2}n^{-1}\sqrt{\log(2n)\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{\boldsymbol{x}\sim\mu^n}\|\boldsymbol{G}(\boldsymbol{x})\|_\infty} \tag{81}$$

$$+ \frac{1}{2}\mathrm{Lip}(\lambda^{-1/2})\lambda^{-2}M^{-1}n^{-1}\log(2n) \tag{82}$$

Proof. For any θ we can bound

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{N}^M}\mathcal{E}(\theta,S)-\mathcal{E}(\theta) &= \mathbb{E}_{\mu\sim\rho}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\mathbb{E}_{(x,y)\sim\mu}\mathbb{E}_{S\sim\mathcal{N}^M}(\ell(\langle\omega_{\theta,S}(\boldsymbol{x},\boldsymbol{y}),\phi_{\theta,S}(x)\rangle,y)-\ell(\langle\omega_\theta(\boldsymbol{x},\boldsymbol{y}),\phi_\theta(x)\rangle,y))\\ &\leq \mathrm{Lip}(\lambda^{-1/2})\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu^n}\mathbb{E}_{x\sim\mu_{\mathcal{X}}}\mathbb{E}_{S\sim\mathcal{N}^M}|\langle\omega_{\theta,S}(\boldsymbol{x},\boldsymbol{y}),\phi_{\theta,S}(x)\rangle-\langle\omega_\theta(\boldsymbol{x},\boldsymbol{y}),\phi_\theta(x)\rangle|.\end{aligned}$$

Then

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{N}^M}|\langle\omega_{\theta,S}(\boldsymbol{x},\boldsymbol{y}),\phi_{\theta,S}(x)\rangle-\langle\omega_\theta(\boldsymbol{x},\boldsymbol{y}),\phi_\theta(x)\rangle| = &\quad \mathbb{E}_{S\sim\mathcal{N}^M}|\boldsymbol{y}^\top(\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{g}(x)-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\boldsymbol{g}(x,S))| &(83)\\ \leq &\quad \mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{y}\|\|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{g}(x)-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\boldsymbol{g}(x,S)\| &(84)\\ \leq &\quad \sqrt{n}\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{g}(x)-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\boldsymbol{g}(x,S)\|. &(85)\end{aligned}$$

Using the matrix identity $AB-CD=A(B-D)+(A-C)D$, the triangle inequality, together with Lemma 2 we get

$$\begin{aligned}\|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{g}(x)-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\boldsymbol{g}(x,S)\| &\leq \|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\|_\infty\|\boldsymbol{g}(x)-\boldsymbol{g}(x,S)\|_2+\|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\|_\infty\|\boldsymbol{g}(x,S)\|_2\\ &\leq (n\lambda)^{-1}\|\boldsymbol{g}(x)-\boldsymbol{g}(x,S)\|_2+(n\lambda)^{-2}\|\boldsymbol{G}(\boldsymbol{x})-\boldsymbol{G}(\boldsymbol{x},S)\|_\infty\sqrt{n} &(86)\end{aligned}$$

and so we consider the terms $\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{g}(x)-\boldsymbol{g}(x,S)\|_2$ and $\mathbb{E}_{S\sim\mathcal{N}^M}\|\boldsymbol{G}(\boldsymbol{x})-\boldsymbol{G}(\boldsymbol{x},S)\|_\infty$. Using Lemma 9 and Lemma 10 we can upper bound (86) (together with factor $\sqrt{n}$) as

$$\begin{aligned}\sqrt{n}\|\boldsymbol{G}_\lambda(\boldsymbol{x})^{-1}\boldsymbol{g}(x)-\boldsymbol{G}_\lambda(\boldsymbol{x},S)^{-1}\boldsymbol{g}(x,S)\| &\leq 2\lambda^{-1}M^{-1/2}+2\lambda^{-2}M^{-1/2}n^{-1/2}\sqrt{\log(2n)\|\boldsymbol{G}(\boldsymbol{x})\|_\infty} &(87)\\ &\quad+\frac{1}{2}\lambda^{-2}M^{-1}n^{-1/2}\log(2n) &(88)\end{aligned}$$

The final bound follows by pulling $\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{\boldsymbol{z}\sim\mu^n}$ into the square root using Jensen and multiplying by $\mathrm{Lip}(\lambda^{-1/2})$.

□

Combining the above we have that

Theorem 12 (IKML Excess risk bound)**.** *Assume that $\mathcal{X}\times\mathcal{Y}\subseteq\mathbb{R}^d\times[0,1]$ and $\ell(y,\hat{y})=(y-\hat{y})^2$. Let $\mathcal{G}_{future}=\{z=(x,y)\mapsto\lambda^{-1/2}\langle v,\phi_{\theta,S}(x)\rangle_{\theta,S}\ :\ \|v\|_{\theta,S}\leq 1\}$ and $\mathcal{F}=\{f:\mathcal{Z}^n\to\mathbb{R}_{\geq 0},\ f(\boldsymbol{z})=\hat{\ell}_\theta(\boldsymbol{z},S),\ \forall\theta\in\Theta\}$, and let $\boldsymbol{Z}=(\boldsymbol{x}^t,\boldsymbol{y}^t)_{t=1}^T\sim\hat{\rho}^T$ then with probability greater than $1-\delta$ over the sampling of $\boldsymbol{Z}$*

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{N}^M}[\mathcal{E}(\hat{\theta},S)-\mathcal{E}(\theta^*)] &\leq 2\mathrm{Lip}(\lambda^{-1/2})\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{z}\sim\hat{\rho}}\mathcal{R}(\mathcal{G}_{future}(\boldsymbol{z})) &(89)\\ &\quad+\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}\mathcal{R}(\mathcal{F}(\boldsymbol{Z}))+\sqrt{\frac{\log(1/\delta)}{2T}} &(90)\\ &\quad+\mathbb{E}_{S\sim\mathcal{N}^M}(\hat{\mathcal{E}}_T(\hat{\theta},S)-\hat{\mathcal{E}}_T(\theta^*,S)) &(91)\\ &\quad+2\mathrm{Lip}(\lambda^{-1/2})\lambda^{-1}M^{-1/2} &(92)\\ &\quad+2\mathrm{Lip}(\lambda^{-1/2})\lambda^{-2}M^{-1/2}n^{-1/2}\sqrt{\log(2n)\mathbb{E}_{\mu\sim\rho}\mathbb{E}_{\boldsymbol{x}\sim\mu^n}\|\boldsymbol{G}(\boldsymbol{x})\|_\infty} &(93)\\ &\quad+\frac{1}{2}\mathrm{Lip}(\lambda^{-1/2})\lambda^{-2}M^{-1}n^{-1/2}\log(2n) &(94)\end{aligned}$$

We note the following

- (89) can be replaced by the strictly greater bound $2\mathrm{Lip}(\lambda^{-1/2})\lambda^{-1/2}n^{-1/2}$
- We hypothesise that the term $\mathbb{E}_{S\sim\mathcal{N}^M}\mathbb{E}_{\boldsymbol{Z}\sim\hat{\rho}^T}\mathcal{R}(\mathcal{F}(\boldsymbol{Z}))$ in (90) is $O(1/\sqrt{T})$ due to standard form of rademacher complexities of bounded balls in RKHS
- (91) is the optimization error and we assume that this is negligible.
- Since we are using the squared loss $\mathrm{Lip}(L)=2(L+1)$ and thus all terms $\mathrm{Lip}(\lambda^{-1/2})=2(\lambda^{-1/2}+1)=O(\lambda^{-1/2})$.

4 DATASETS

4.1 BEIJING AIR QUALITY

The Beijing Air Quality dataset [Zhang et al., 2017] is a time-series dataset measuring air-quality and meterological factors at 12 air-quality monitoring sites. The meterological data for each site is matched with the closest of available weather stations. The data was collected hourly and from the period March 1st, 2013 to February 28th, 2017.

Each datum consists of a timestamp, the site name and various features. We use the feature of interest, "PM2.5" for the fine particulate matter (PM) concentration and remove the features "PM10", "wd", "WSPM" since the first one correlates heavily with "PM2.5" and "wd", "WSPM" since "wd" is the direction of the wind and thus categorical and "WSPM" since this is the wind speed of the direction. This leaves us with 9 features and one output feature.

The dataset was created as follows:

1. Remove any rows with missing entries.
2. For each station, split the time-series into 3 sub-series of $64/16/20\%$ starting at the beginning forming the meta-train, validation and test sets.

Tasks are sampled as follows:

1. Sample a station uniformly at random.
2. Given a train and validation size $n = n_{\mathrm{tr}} + n_{\mathrm{val}}$ sample a contiguous sequence of size n at random from the available starting points. We add a temporal feature t which is just an index from 1 to n to encode temporal dependency local to the task.
3. From this contiguous sequence randomly assign n_{tr} datapoints to the train set and n_{val} datapoints to the validation set.

4.2 GAS SENSOR

The Gas Sensor Modulation dataset [Burgués et al., 2018] is a collection of multivariate timeseries collected in a controlled environment using MOX sensors for CO detection. The sensors output voltage recordings sampled at a frequency of 3.5 Hz.

Each timeseries can be chunked up into contiguous subseries corresponding to experiments by looking at the heating cycle, the end of a cycle which marks a new experiment. We let each subsequence correspond to one task distribution from which we sample $n = n_{\mathrm{tr}} + n_{\mathrm{val}}$ datapoints and permute the indices to make the task into a supervised regression task. The output was chosen to be the 2nd feature 3 timesteps into the future as this seen to vary over the tasks and not directly inferable by one of the other features. In total there are 13 csv files with experiment. Each such file has a set number of experiments after preprocessing, we split each files experiment into $64/16/20\%$ meta-train, validation and test splits.

The dataset was created as follows:

1. All subsequences were extracted by locating the start and end of a heating cycle.
2. All extra features which were not gas sensors were dropped.
3. Output feature isolated and lagged.

Tasks are sampled as follows:

1. For a new task we first sample one of the csv files uniformly at random.
2. From the experiments in this csv file we sample a subsequence uniformly at random which is the task-distribution.
3. Add "t" to the features.
4. From this subsequence we sample $n = n_{\mathrm{tr}} + n_{\mathrm{val}}$ points at random which forms out train and validation set.

5 EXPERIMENTAL RESULTS

5.1 HARDWARE

All of the experiments were run on a single computer with specifications

CPU AMD Ryzen 7 3700X 8-Core Processor

GPU NVIDIA GeForce RTX 2060 SUPER

RAM 2x16GB DDR4 Vengeance

5.2 ALGORITHMS

In this section we elaborate on the algorithm used.

MAML [Finn et al., 2017] parameterize a set of functions $f_\theta : \mathcal{X} \to \mathcal{Y}$, typically a family of neural networks. For a new task D it optimizes the objective $\operatorname{argmin}_\theta \hat{\mathcal{E}}(f_\theta, D^{\mathrm{tr}})$ using gradient descent starting from the hyperparameter θ_0 so that the fine-tuned weight vector is a function of θ_0, $\theta(\theta_0)$. In the outer loop it optimizes the objective $\operatorname{argmin}_{\theta_0} \hat{\mathcal{E}}(f_{\theta(\theta_0)}, D^{\mathrm{val}})$ using gradient descent.

R2D2 [Bertinetto et al., 2018] parameterize a feature map $\phi_\theta : \mathcal{X} \to \mathbb{R}^d$ which give rise to a kernel $M_\theta(x, x') = \langle \phi_\theta(x), \phi_\theta(x') \rangle$. The inner algorithm is KRR with K_θ. For a task D it first does KRR in the inner loop giving the KRR estimator $f_\theta = A_{\mathrm{KRR}}(K_\theta, D^{\mathrm{tr}})$ and in the outer loop it optimizes $\operatorname{argmin}_\theta \hat{\mathcal{E}}(f_\theta, D^{\mathrm{val}})$ using gradient descent.

LS Biased Regularization [Denevi et al., 2019] performs biased ridge regression where the functions are given by $f_\theta(x) = \langle \theta, x \rangle$. For the inner algorithm it solves the biased ridge regression problem $\operatorname{argmin}_w \frac{1}{n}\|X\theta - y\|^2 + \lambda\|\theta - \theta_0\|^2$ which has a closed form, see [Denevi et al., 2019]. For a task D the algorithm first finds $\theta(\theta_0)$ using D^{tr} using biased RR and in the outer loop it optimizes $\operatorname{argmin}_{\theta_0} \hat{\mathcal{E}}(f_{\theta(\theta_0)}, D^{\mathrm{val}})$ using gradient descent.

IKML-MLP is the same as IKML but uses the general random features representation of the kernel $K(x, x') = \int_\Omega \varphi(x, \omega)^\top \varphi(x, \omega) \mathrm{d}\tau(\omega)$. In this case, we let $\varphi(\cdot, \omega) : \mathbb{R}^d \to \mathbb{R}^o$ be an MLP with ReLU activation functions, some fixed hidden dimension h and an output dimension o. Let D be the size of ω. In this case the feature map is complicated, so we opt for a simpler pushforward to make it easier to train. In particular, we let the pushforward take on a "variational form" by letting the pushforward $\psi_\theta(s) = \psi_{\mu,\sigma}(s) = \mu + \sigma \odot s$ where $s, \mu, \sigma \in \mathbb{R}^D$ and for two matrices A, B, $(A \odot B)_{ij} = A_{ij} B_{ij}$ is the Hadamard product. We train using the same procedure as in Alg. 1. Of note is that this can be seen as an ensemble method over R2D2 where instead of ensembling over the learned functions we ensemble over kernel functions.

5.3 TOY REGRESSION

Setup We create a synthetic high-dimensional meta-learning regression setting where each task is sampled from an RKHS $\mathcal{H}$ with a "complicated" kernel K^o. In particular, we choose K^o to be the kernel given by Bochner's theorem and a pushforward of a 3-layers Multi-Layer Perceptron (MLP) with 32 hidden units per layer, ReLU activation functions and a 16-dimensional latent Gaussian distribution. The network was initialized with weights given by the PyTorch [Paszke et al., 2019] default initialization scaled by 100. Since this kernel lacks an analytic form, we sample 10000 frequencies and use the random features kernel in its place.

The tasks are generated in $\mathcal{H}$ by independently sampling R points $(x_r)_{r=1}^R$ with $x_r \sim U_{[0,0.2]^d}$ and R coefficients $(\alpha_r)_{r=1}^R$ with $\alpha_j \sim U_{[0,1]}$. Together they model the target regressor as $f(x) = \sum_{r=1}^R \alpha_r K(x, x_r)$. We set $R = 3$. The task datasets is created by independently sampling $n = n_{\mathrm{tr}} + n_{\mathrm{val}} = 50 + 50$ datapoints $(x_i, y_i)_{i=1}^n$ with $x_i \sim U_{[0,0.2]^d}$ and $y_i = f(x_i)$.

Initial and Learned Kernel For the same setup of the environment as in the synthetic experiments we look at how the initial and learned kernels differ from the true kernel. We do this for the algorithms *IKML* and *Gaussian MKL meta-KRR*. These algorithms where chosen since they define translation invariant kernels and are easy to visualize. We let all of the algorithms (R2D2, MAML, IKML) be parameterized by a 3-layer MLP but with varying the dimensionality of x. We also tried using 1 and 2-layer MLPs for the parameterization but the results were almost identical.

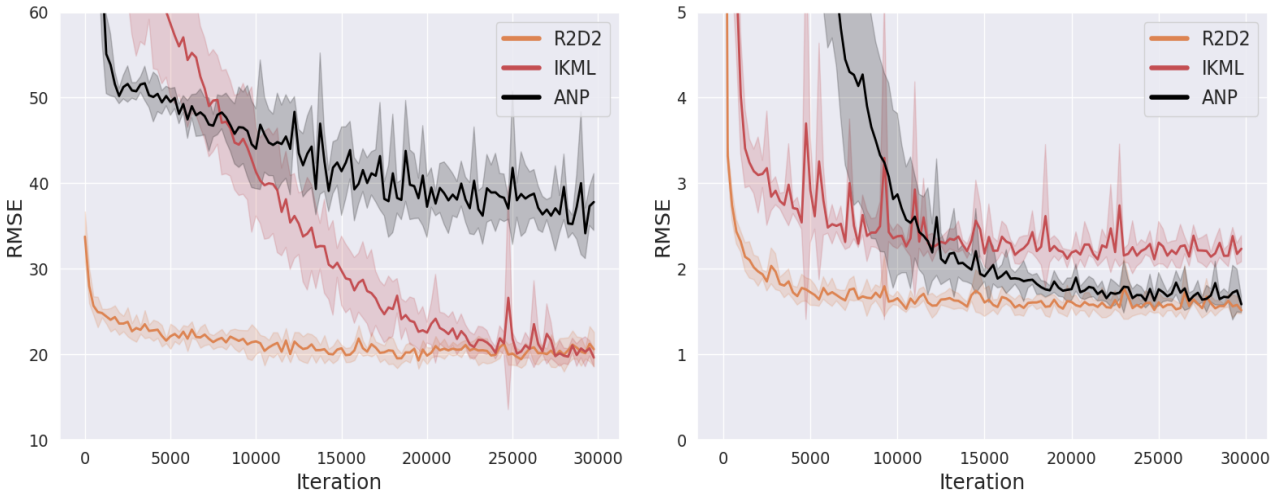
For an experiment with dimension d we visualize the kernels of Gaussian MKL meta-KRR and IKML by sampling 5 directions $(v_i)_{i=1}^5$ from the unit ball in $\mathbb{R}^d$. For a direction v_i we plot the value of the kernel on the line with direction v_i where on the x-axis we have t from -0.4 to 0.4 and on the y-axis the value of $K(0, t \cdot v_i)$. We plot the result of the first run for each experiment, other runs look similar.

We plot the learning curves and kernel for each dimension 1, 5, 10, 20. For each row in Fig. 2 corresponding to a dimension d the i'th column plots kernels in the direction of v_i with the first row of the subplot corresponding to the kernels at

initialization and the second row the kernels after training. The sample $(v_i)_{i=1}^5$ is resampled for each dimension. For IKML we sample 10000 frequencies once and fix them before plotting.

5.4 LEARNING CURVES

We show the behaviour of the optimization trajectory of the algorithms R2D2, IKML and ANP. See Fig. 1a and Fig. 1b.



(a) (b)

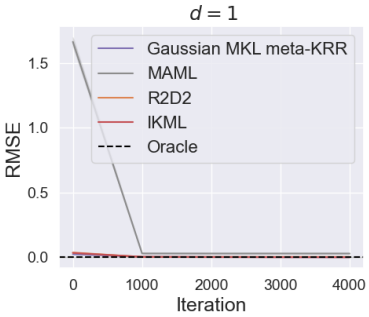
Figure 1: Learning curves of meta-validation RMSE for the algorithms IKML, R2D2 and ANP for (a) Beijing Air Quality (25-shot), (b) Gas Sensor dataset (20-shot) over 5 runs (mean $\pm$ 1 std). R2D2 and ANP were chosen due to their recency and performance as few-shot learning algorithms compared to all other algorithms evaluated.

5.5 CROSS-VALIDATION FOR REAL-WORLD DATASETS

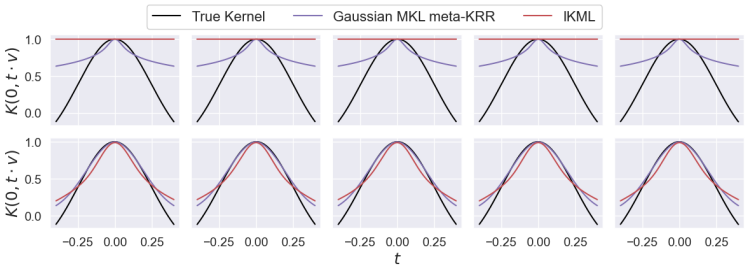
We cross-validated R2D2, IKML and ANP on the 25-shot Air Quality and 20-shot Gas Sensor dataset where we do a grid search over the number of hidden layers in an MLP with ReLU activation function and the meta-learning rate. For IKML and R2D2 the number of hidden layers are in $\{1, 2, 3, 4, 6, 8, 10, 15, 20\}$ while for ANP we use the same architecture for encoder and decoder and use $\{1, 2, 3, 4\}$ layers, the hidden dimension is fixed to 64, the meta-learning rate are in $\{10^{-4}, 10^{-5}\}$. The training setup is the same as in the main body and the metric is the RMSE on a holdout-set sampled from the meta-validation split of the best model from the snapshots during the 30000 iterations. The results can be seen in Tab. 1.

5.6 MORE SHOTS

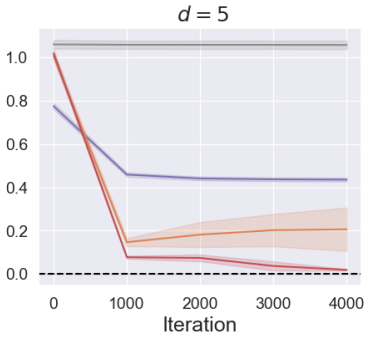
Further test-RMSE for various numbers of shots for Air Quality Tab. 2 and Gas Sensor Tab. 3. We benchmark LS Biased Regularization, IKML, R2D2, ANP for both Air Quality and Gas Sensor and additionally IKML-MLP for Gas Sensor. We reuse the cross-validated models for IKML, R2D2 and ANP and the hyperparameters used for the other models. We get 5 test-RMSE scores for Air Quality and 2 for Gas Sensor and report the mean and standard deviation for Air Quality and mean for Gas Sensor. The low number of shots in Gas Sensor is due to many of the underlying time series from which each task is generated having as few as 40 points.



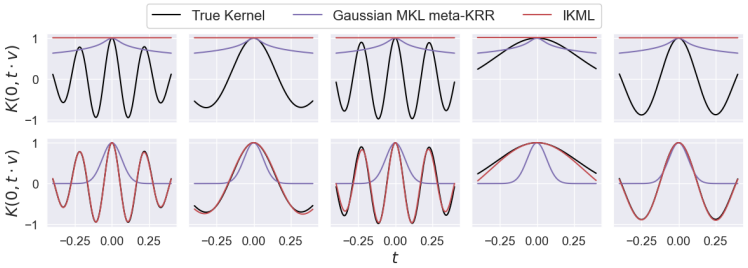
Learning curves for $d = 1$



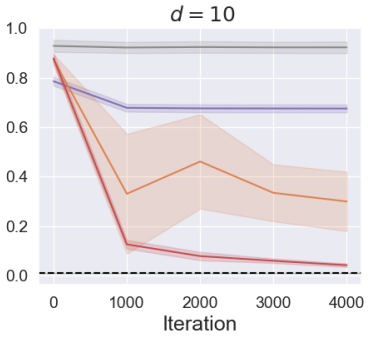
Initial and learned kernels for $d = 1$



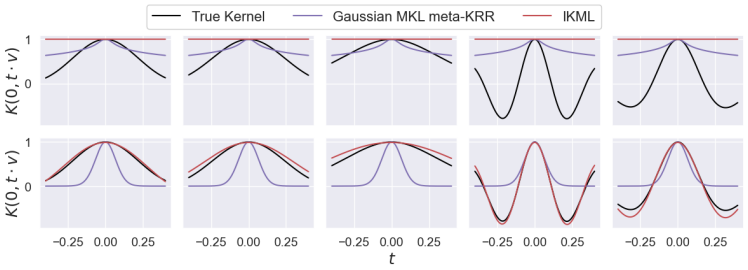
Learning curves for $d = 5$



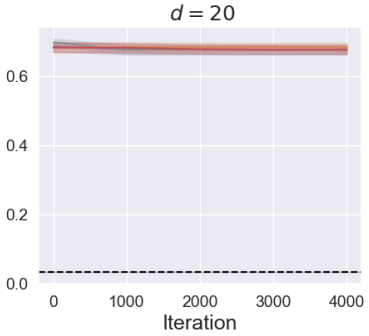
Initial and learned kernels for $d = 5$



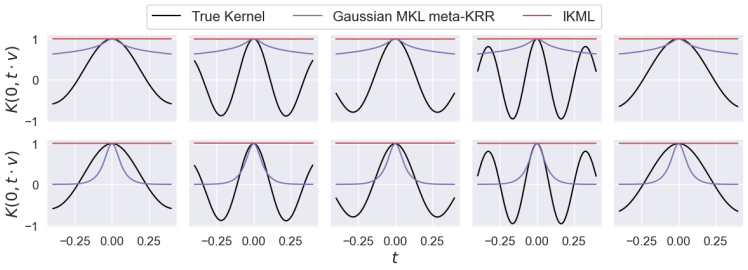
Learning curves for $d = 10$



Initial and learned kernels for $d = 10$



Learning curves for $d = 20$



Initial and learned kernels for $d = 20$

Figure 2: Parameterization using a 3-layer MLP: Learning curves and initial vs learned kernels for different input dimension on synthetic dataset (all algorithms using a 3-layer MLP). **Column 1**: learning curves (meta-test RMSE) over 3 runs. **Column 2**: **Sub-row 1** kernel before training, **Sub-row 2** kernel at test time. Each column plots the kernel in a random direction drawn from the unit ball.

Table 1: Validation results for meta-hyperparameter configurations for IKML, R2D2 [Bertinetto et al., 2018] and ANP [Kim et al., 2019] on 25-shot Air Quality dataset and 20-shot Gas Sensor. Best set of parameters in **bold**. We run the algorithms for 30000 iterations and evaluate it on the validation set at 250 intervals. We get the validation RMSE on a holdout set (3000 tasks) using the model with the lowest evaluation validation error.

		25-shot Air Quality			20-shot Gas Sensor		
Meta-lr	Layers	IKML	R2D2	ANP	IKML	R2D2	ANP
10^{-4}	1	101.65	8861.31	1390.14	2.16	2.64	2.38
	2	98.37	13761.01	38.61	2.14	1.85	1.72
	3	98.07	205.06	**38.37**	2.14	1.65	**1.53**
	4	21.45	508.55	36.32	2.11	1.49	1.57
	6	24.24	21.57	–	2.13	**1.46**	–
	8	23.88	21.96	–	**2.06**	1.53	–
	10	27.30	**21.32**	–	2.06	1.49	–
	15	27.57	22.85	–	2.12	1.48	–
	20	40.57	25.01	–	7.20	1.50	–
10^{-5}	1	125.75	3237.35	76.50	19.53	6.45	7.83
	2	110.01	1233.41	41.75	2.70	3.20	8.43
	3	76.58	431.61	47.24	2.50	2.34	7.42
	4	**19.05**	57.37	43.71	2.41	1.86	6.35
	6	20.52	22.68	–	2.43	1.59	–
	8	23.86	21.98	–	2.35	1.55	–
	10	134.89	22.44	–	2.45	1.53	–
	15	28.40	24.80	–	2.46	1.56	–
	20	135.18	26.62	–	2.45	1.55	–

Table 2: Test-RMSE (mean $\pm$ 1 std) for IKML, R2D2, ANP and LSBR over 5 independent runs on Air Quality for various shots. Same tasks for all algorithms over each run. Best result for each shot in **bold**.

	Air Quality (shots)			
Model	10	25	50	100
IKML	24.32 ± 5.21	$\mathbf{19.14 \pm 0.93}$	$\mathbf{19.36 \pm 1.02}$	$\mathbf{18.88 \pm 0.51}$
R2D2	$\mathbf{21.21 \pm 0.28}$	20.23 ± 0.55	23.42 ± 3.44	20.75 ± 0.79
ANP	31.05 ± 0.90	33.77 ± 0.70	37.30 ± 0.94	41.08 ± 1.07
LSBR	21.49 ± 0.40	21.68 ± 0.29	23.69 ± 0.47	27.32 ± 0.16

Table 3: Test-RMSE (mean, standard deviation left out due to low number of runs) for IKML, R2D2, ANP, LSBR and IKML-MLP over 2 independent runs on Gas Sensor for various shots. Same tasks for all algorithms over each run. Best result for each shot in **bold**.

	Gas Sensor (shots)			
Model	5	10	15	20
IKML	10.04	4.57	3.42	2.80
R2D2	6.00	**2.44**	2.12	1.95
ANP	**2.57**	**2.44**	**2.10**	2.12
LSBR	13.97	12.21	11.12	12.44
IKML-MLP	4.03	2.64	2.23	**1.94**

5.7 SENSITIVITY OF R2D2 AND IKML-MLP

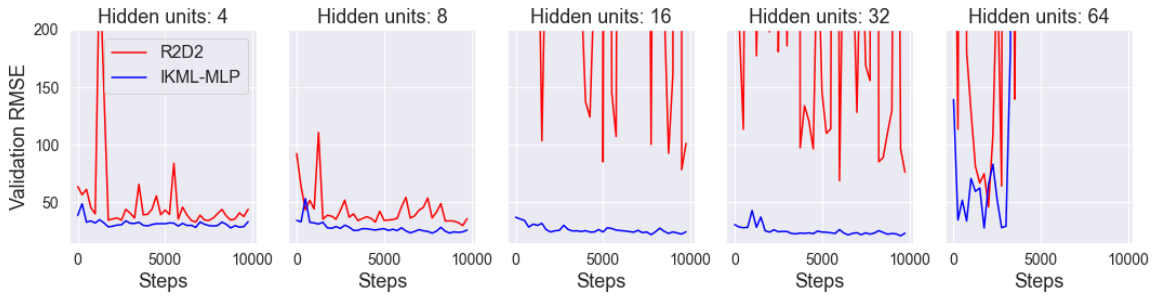
We highlight the qualitative difference between R2D2 and IKML-MLP. We compare the learning curves and holdout meta-valid and test RMSE. Note that the test-split is used to asses out-of-sample few-shot performance since we train and choose best model using train and validation set respectively. In this case we let the feature map $\varphi(x, \omega)$ be an MLP with ReLU activation functions, with number of layers and number of hidden units varied. We perform this analysis on the Air Quality and Gas Sensor datasets with the settings as given in the main body unless specified and compare the results.

Air Quality We run IKML-MLP and R2D2 for 10000 iterations using Adam with meta-learning rate $3 \cdot 10^{-4}$ and vary the number of layers and and the number of hidden units in isolation. Both of the algorithms share the same base feature map $\varphi(\cdot, \omega)$ but IKML-MLP calculates the kernel $K(x, x') = \int_\Omega \varphi(x, \omega)^\top \varphi(x, \omega) \mathrm{d}\tau(\omega)$ by sampling while R2D2 has a fixed feature map yielding the kernel $K(x, x') = \varphi(x, w)^\top \varphi(x', w)$ for a fixed weight w. We use a feature dimension of 8. The only difference to the setup in the main body is that we use 10000 iterations insted of 30000.

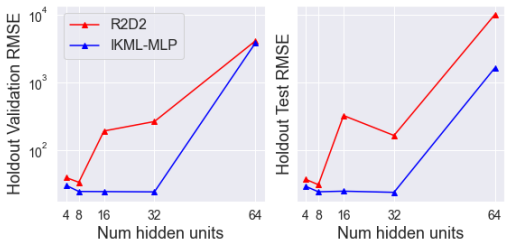
When we fix the number of layers to be 2, we can see from Fig. 3 that IKML-MLP dominates R2D2 in terms of performance both on the validation and test set. In contrast, when we fix the number of hidden units to be 64 and vary the number of layers, we can see from Fig. 4 that R2D2 performs equally well as IKML-MLP. As the network becomes deeper we noticed, for this dataset, that IKML-MLP requires more random features to train well (in contrast to the Gas Sensor case). We hypothesis that for noisy tasks, like in the Air Quality dataset, the number of random features required to get accurate gradients to be able to train deeper networks increase quickly with depth. However, on this dataset we see that the number of layers is not required to be very deep to reach good performance, so in this case it does not pose a problem.

Gas Sensor We run IKML-MLP and R2D2 for 10000 iterations using Adam with meta-learning rate $3 \cdot 10^{-4}$ and vary the number of layers and and the number of hidden units in isolation. Both of the algorithms share the same base feature map $\varphi(\cdot, \omega)$ but IKML-MLP calculates the kernel $K(x, x') = \int_\Omega \varphi(x, \omega)^\top \varphi(x, \omega) \mathrm{d}\tau(\omega)$ by sampling while R2D2 has a fixed feature map yielding the kernel $K(x, x') = \varphi(x, w)^\top \varphi(x', w)$ for a fixed weight w. We use a feature dimension of 4. The only difference to the setup in the main body is that we use 10000 iterations insted of 30000.

Compared to the Air Quality figures Fig. 3 and 4 training is much more stable due to the dataset being much less noisy than that of the Air Quality dataset. R2D2 and IKML-MLP both train well and have good performance, although IKML-MLP overfits less to the meta-split as can be seen on the holdout performance plots in Fig. 5 and 6. In this case 100 random features were enough for the training to be successful for IKML-MLP which supports the hypothesis given in the previous section on Air Quality.

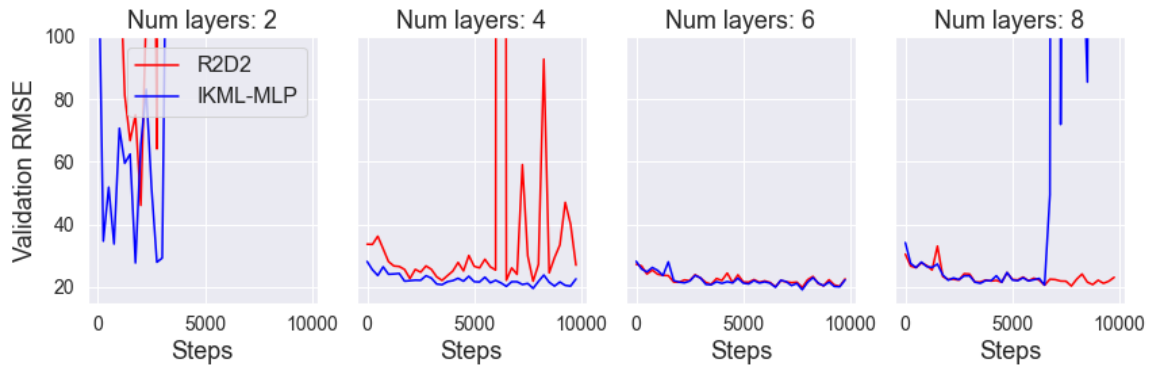


Learning curves

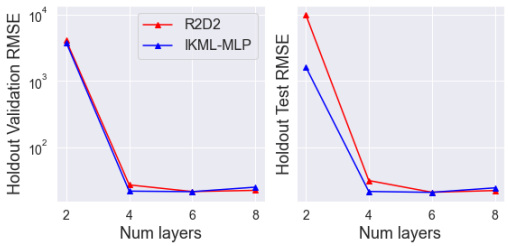


Holdout Performance

Figure 3: Learning curves (above) and performance plots (below) of R2D2 vs IKML-MLP on the Air Quality dataset when varying the number of hidden units. IKML-MLP is more robust to hyperparameter settings than R2D2. We fix the number of hidden layers to 2 and feature dimension to be 8. For IKML-MLP we fix the number of random features to be 100. Note that performance plot is log-scaled due to large range of reported numbers.



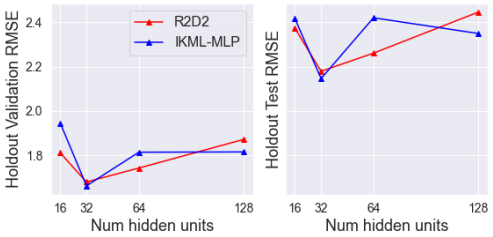
Learning curves: we optimize the models using the train split



Holdout Performance

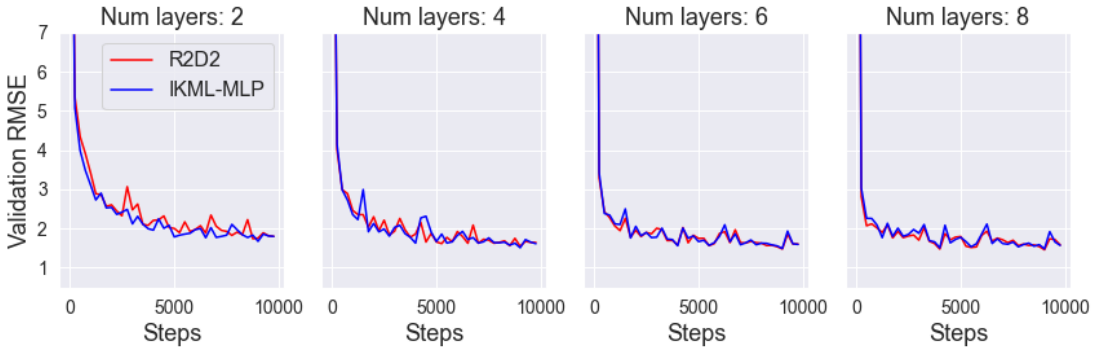
Figure 4: Learning curves (above) and performance plots (below) of R2D2 vs IKML-MLP on the Air Quality dataset when varying the number of layers. IKML-MLP stabilize training up to a point but for deeper networks we found that IKML-MLP requires more random features. We fix the number of hidden units to 64 and feature dimension to be 8. For IKML-MLP we fix the number of random features to be 400.

Learning curves

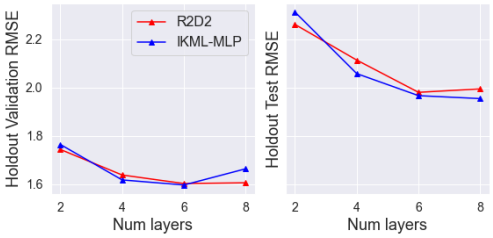


Holdout Performance

Figure 5: Learning curves (above) and performance plots (below) of R2D2 vs IKML-MLP on the Gas Sensor dataset when varying the number of hidden units. IKML-MLP is more robust to hyperparameter settings than R2D2. We fix the number of hidden layers to 2 and feature dimension to be 8. For IKML-MLP we fix the number of random features to be 100.



Learning curves: we optimize the models using the train split



Holdout Performance

Figure 6: Learning curves (above) and performance plots (below) of R2D2 vs IKML-MLP on the Gas Sensor dataset when varying the number of layers. IKML-MLP stabilize training up to a point but for deeper networks we found that IKML-MLP requires more random features. We fix the number of hidden units to 64 and feature dimension to be 8. For IKML-MLP we fix the number of random features to be 100.

Table 4: *Time (seconds) to solve one batch of tasks for IKML, IKML-MLP, R2D2 and MAML.* We measure the time required for solving one batch of tasks: training, calculating the meta-loss, and updating the hyperparameters. We use the Air Quality ($d = 10$) dataset with 25 train and 25 validation points per task, meta-batch size of 4. All algorithms use a 4-layer MLP with 64 hidden units. For IKML we let $M = 2 \cdot 10^4$, while for IKML-MLP we let $M = 100$. We run each algorithm for 5000 steps and normalize the total time by dividing with 5000. We repeat this 3 times and report the mean and standard deviation.

Algorithm	Seconds for one batch (mean $\pm$ 1 std)
IKML	0.017 ± 0.00004
IKML-MLP	0.075 ± 0.002
R2D2	0.031 ± 0.001
MAML	0.022 ± 0.001

6 COMPUTATIONAL COMPLEXITY AND WALLTIME TABLE

In this section we show the computational complexity using big-O notation of IKML / IKML-MLP and compare it against that of R2D2 since they both rely on KRR as the inner algorithm. In addition we measure the performance in practice through wall-time table of IKML, IKML-MLP, MAML and R2D2. We first recall the complexity of training and validation of KRR in the dual form when we have a train set $D^{\text{tr}} = (x_i, y_i)_{i=1}^{n_{\text{tr}}}$ and a validation set $D^{\text{val}} = (x_j, y_j)_{j=1}^{n_{\text{val}}}$. We focus on the dual formulation since generally data set sizes are small in meta-learning while the feature space dimension is large, which means that the dual form is more efficient than the primal form.

Assume that we have a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that can be evaluated in $O(\kappa)$. For training we need to calculate the dual coefficients $\alpha = (G + n_{\text{tr}}\lambda I)^{-1}\mathbf{y}$ where $\mathbf{y}$ is the output vector. This means we first need to calculate the regularized kernel matrix of the train set, $G + n_{\text{tr}}\lambda I \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$, which can be calculated in $O(\kappa n_{\text{tr}}^2 + n_{\text{tr}})$ since $n_{\text{tr}}\lambda I$ is a diagonal matrix, then invert this matrix which can be calculated in $O(n_{\text{tr}}^3)$ and finally perform the matrix-vector multiplication which is $O(n_{\text{tr}}^2)$. Summing all of the steps gives a final complexity of $O(\kappa n_{\text{tr}}^2 + n_{\text{tr}}^3)$. Prediction on the validation set D^{val} means first calculating the matrix $(M_{ls})_{l,s=1}^{n_{\text{val}}, n_{\text{tr}}} = (K(x_l, x_s))_{l,s=1}^{n_{\text{val}}, n_{\text{tr}}}$ between the validation and train set which is done in $O(\kappa n_{\text{tr}} n_{\text{val}})$. After calculating M, calculating $\hat{\mathbf{y}} = M\alpha$ can be done in $O(n_{\text{tr}} n_{\text{val}})$ which means that the total number of operations is $O((\kappa + 1) n_{\text{tr}} n_{\text{val}})$.

The complexity for both training and validation when using KRR depends implicitly on κ which will depend on the meta-learning algorithm. For IKML with Bochner kernel using M random features we first need to sample M features. This can be done in $O(Cm)$ where C is the time it takes to evaluate the pushforward neural network. Note that this is a one-time cost before training and validation. In practice we use batches so that for B tasks we sample M features once, which reduces this cost further by a factor of the number of tasks in a batch. Letting $W \in \mathbb{R}^{M \times d}$ be the matrix of random features stacked horizontally then the feature map $\phi : \mathcal{X} \to \mathbb{R}^M$ is $\phi(x) = \cos(Wx + b)$ where b is a vector of iid entries sampled uniformly from $[0, 2\pi]$, sampled at the same time as W. Evaluating ϕ once is done in $O((d + 1)m)$. For one task, we first calculate the M features in $O((d + 1)m)$ and training This means that training and prediction for IKML costs $O(dmn_{\text{tr}}^2 + n_{\text{tr}}^3)$ and $O(dmn_{\text{tr}} n_{\text{val}})$ respectively, both which are linear in M.

For R2D2 the feature map $\phi : \mathcal{X} \to \mathbb{R}^h$ where h is the dimension of the feature space is a neural network. Assuming that ϕ takes the form of an L-layer MLP with weights and biases $(W_i, b_i)_{i=1}^T$ such that $W_1 \in \mathbb{R}^{h_1 \times d}$, $b_1 \in \mathbb{R}^{h_1}$, and for $1 < l < T$, $W_l \in \mathbb{R}^{h_l \times h_{l-1}}$, b_l and finally $W_L \in \mathbb{R}^{h \times h_l}$, $b_l \in \mathbb{R}^h$ with nonlinearity σ which can be evaluated in constant time A, then evaluating $\phi(x)$ is done in $O(\prod_{l=1}^L h_l h_{l-1} + \sum_{l=1}^{L-1}(1 + A)h_l + h_L) = O(\prod_{l=1}^L h_l h_{l-1})$. Running IKML-MLP, if $h_l = h$ for any l we get $O(dh^{2L-1})$. Except for the extra factor of h^{2L-1} the same conclusion as for Bochner holds in this case.

References

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *CoRR*, 2018.

Javier Burgués, Juan Manuel Jiménez-Soto, and Santiago Marco. Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models. *Analytica Chimica Acta*, 1013:13–25, 2018.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pages 1566–1575, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Andreas Maurer. Transfer bounds for linear feature learning. *Machine Learning*, 75(3):327–350, 2009.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Joel A Tropp. Matrix concentration & computational linear algebra, July 2019.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.