Appendices

A Notation

Table 4: **Notation**

n	number of samples
$x_i \in \mathcal{X}, y_i \in \mathbb{R}$	input, target
$Y \in \mathbb{R}^n$	vector of targets $(y_i) \in \mathbb{R}^n$
$X = (x_1, x_2, \ldots, x_n)$	tuple of inputs
$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	positive definite kernel with $\mathbb{H}$ as its RKHS.
$S : \mathbb{H} \to \mathbb{R}^n$	Sampling operator, defined as $S(f) = f(X) = (f(x_i)) \in \mathbb{R}^n$
$S^* : \mathbb{R}^n \to \mathbb{H}$	Adjoint of sampling operator, defined as $S^*\boldsymbol{\alpha} = \sum_{i=1}^n K(\cdot, x)\alpha_i$
$\mathcal{K}$	empirical covariance operator using n samples, defined as $\frac{1}{n}SS^* : \mathbb{H} \to \mathbb{H}$
$K(X, X)$	kernel matrix. Also equals SS^*
B	mini-batch of indices, subset of $\{1, 2, \ldots, n\}$
$m = \lvert B \rvert$	batch size
$X[B] \in \mathcal{X}^m$	minibatch of size m
$\boldsymbol{v}[B] \in \mathbb{R}^m$	subvector of $\boldsymbol{v} \in \mathbb{R}^n$ corresponding to indices B
$\boldsymbol{H}_B$	selector matrix, rows of I_n corresponding to B, so that $\boldsymbol{v}[B] = \boldsymbol{H}_B\boldsymbol{v}$.
q	level of preconditioner
$(\boldsymbol{E}, \Lambda, \lambda_{q+1})$	top-q eigensystem of $K(X, X)$
$\mathcal{P} : \mathbb{H} \to \mathbb{H}$	preconditioner in $\mathbb{H}$ defined as $\mathcal{I} - \sum_{i=1}^q \left(1 - \frac{\lambda_{q+1}}{\lambda_i}\right) \psi_i \otimes_{\mathbb{H}} \psi_i$
	where $\psi_i = S^*\boldsymbol{e}_i/\sqrt{\lambda_i} \in \mathbb{H}$ are top-q eigenfunction of $\mathcal{K}$ with eigenvalues $\frac{1}{n}\lambda_i$
$\boldsymbol{F}$	rescaled eigenvectors defined as $\boldsymbol{E}\sqrt{I_q - \lambda_{q+1}\Lambda^{-1}}$
$\boldsymbol{F}[B]$	rows of $\boldsymbol{F}$ corresponding to the mini-batch B
J	subset of indices $\{1, 2, \ldots, n\}$ to obtain Nyström approximation of $\mathcal{P}$
$s = \lvert J \rvert$	size of Nyström subset
$X[J]$	Nyström subsample of size s
$(\boldsymbol{D}, \Delta, \delta_{q+1})$	top-q eigensystem of $\frac{1}{n}K(X[J], X[J])$
$\mathcal{Q} : \mathbb{H} \to \mathbb{H}$	Nyström approximation of $\mathcal{P}$ defined as
	$\mathcal{I} - \sum_{i=1}^q \left(1 - \frac{\delta_{q+1}}{\delta_i}\right) \phi_i \otimes_{\mathbb{H}} \phi_i$ where $(\delta_i/s, \phi_i)$ is an eigen-pair of $\frac{1}{s}S_J^*S_J$
$\boldsymbol{G}$	rescaled eigenvectors defined as $\boldsymbol{D}\sqrt{(I_q - \delta_{q+1}\Delta^{-1})\,\Delta^{-1}}$

B EigenPro algorithms

Lemma 9. *Suppose Algorithm 3 is run for t iterations with hyperparameters given by equation* (23)*, then we have,*

$$\|f_t - f^*\|_{\mathbb{H}}^2 \leq \exp\left(-t/\sqrt{\widetilde{\kappa}_m \kappa_m}\right) \|f^*\|_{\mathbb{H}}^2 . \tag{33}$$

Proof. This is an immediate corollary of [20, Theorem 2] and the discussion that ensues therein on the convergence rate of MaSS. Preconditioning only changes the largest eigenvalue from λ_1 to λ_{q+1}. This can via the following reduction commonly applied to analyze spectral preconditioning: (16) is actually (13) with the following modified kernel,

$$\widetilde{K}(x, z) := K(x, z) - K(x, X)\boldsymbol{E}\boldsymbol{Q}\boldsymbol{E}^\top K(X, z), \tag{34}$$

where $\boldsymbol{Q} = \Lambda^{-1}(I_q - \lambda_{q+1}\Lambda^{-1})$, where $(\boldsymbol{E}, \Lambda, \lambda_{q+1})$ is the top-q eigensystem of $K(X, X)$. An interested reader can find this reduction argument in [22, 23]. The largest eigenvalue of $\widetilde{K}(X, X)$ can be verified to be λ_{q+1}.

The other key difference is the quantities L_1 and $\widetilde{\kappa}_m$ defined in [20]. Using elementary properties of our Hilbert space $\mathbb{H}$, we can easily show that our L_1 and $\widetilde{\kappa}_m$ are upper bounds to these quantities in [20]. □

Algorithm 3 AxlePro

Input: positive definite kernel K, batch size m, learning rates $\eta_1, \eta_2 > 0$, damp factor $\gamma \in (0, 1)$.
Output: $f : \mathcal{X} \to \mathbb{R}$ solving (9) approximately.
setup: $\boldsymbol{\alpha} \leftarrow \mathbf{0}_n$, and $\boldsymbol{\beta} \leftarrow \mathbf{0}_n$.
$(\boldsymbol{E}, \Lambda, \lambda_{q+1}) \leftarrow$ top-q eigensystem of $K(X, X)$
$\boldsymbol{F} \leftarrow \boldsymbol{E}\sqrt{I_q - \lambda_{q+1}\Lambda^{-1}} \in \mathbb{R}^{n\times q}$
repeat
  Fetch batch of indices $B \subset \{1, \ldots, n\}$, $|B| = m$
  $\boldsymbol{v} \leftarrow K(X[B], X)\boldsymbol{\beta} - Y[B] \in \mathbb{R}^m$
  $\boldsymbol{w} \leftarrow \boldsymbol{F}\boldsymbol{F}[B]^\top \boldsymbol{v} \in \mathbb{R}^n$
  $\widetilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}$ {copy state for momentum}
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\beta}$
  $\boldsymbol{\alpha}[B] - \leftarrow \eta_1 \boldsymbol{v}$ {gradient step-1}
  $\boldsymbol{\alpha} + \leftarrow \eta_1 \boldsymbol{w}$ {correction-1}
  $\boldsymbol{\beta} \leftarrow \boldsymbol{\alpha} + \gamma(\boldsymbol{\alpha} - \widetilde{\boldsymbol{\alpha}})$ {momentum}
  $\boldsymbol{\beta}[B] + \leftarrow \eta_2 \boldsymbol{v}$ {gradient step-2}
  $\boldsymbol{\beta} - \leftarrow \eta_2 \boldsymbol{w}$ {correction-2}
until Stopping criterion is reached
return $f(x) = \sum_{i=1}^n \alpha_i K(x, x_i)$

Algorithm	FLOPS/batch	memory
EigenPro 2	$nm + sm + 2sq + m + s$	$n + sq$
AxlePro 2	$nm + sm + 2sq + \mathbf{2m + 2s + n}$	$\mathbf{2n} + sq$
EigenPro 3	$2pm + ps + ms + 2sq + m + p$	$p + sq$
AxlePro 3	$2pm + ps + ms + 2sq + \mathbf{2m + 2p}$	$\mathbf{2p} + sq$

Table 5: The bolded quantities indicate overheads due to preconditioning. Here $s = |J|$. We have omitted the lower-order terms of n and $2n$ in the per iteration time since they are dominated by mn. Memory requirement and per iteration complexity of EigenPro and AxlePro. The algorithm AxlePro is based on applying momentum acceleration to EigenPro [22], whereas AxlePro 2 is based on applying momentum acceleration to EigenPro 2 [23]. The overhead of storing the previous parameter is n as seen the last column.

C Numerical Experiments (continued)

See Table 6, Table 7, Table 8, Table 9, Table 10 for experiment comparisons.

D Details on Numerical Experiments

Hardware. Experiments were run on the SDSC Expanse GPU cluster with 32GB RAM, with 1 NVIDIA V100 SMX2 with 32GB VRAM, and 1 Xeon Gold 6248 CPU.

Datasets and Kernels. Our experiments were conducted on the following datasets: Cifar-10($n = 50,000, d = 3072$), Stellar Classification($n = 95,000, d = 13$), and EMNIST Digits($n = 240,000, d = 784$). We test the performance for both Gaussian kernel and Laplacian kernel. And for Cifar-10 dataset, we also compare the performance with Myrtle-5 kernel [32], which is the state-of-the-art kernel for this dataset.

1. EigenPro. We select the precondition level based on the GPU memory so that the batch size at the linear rate regime critical point m_1^* fully exploits GPU memory. This step involves computing eigenpairs of the kernel matrix and we use Nyström approximation with $20k$ subsamples. The optimal learning rate can also be computed using the computed eigenpairs.

2. AxlePro. For the precondition level, we follow the same way as in Eigenpro. The only hyperparameter needs to be tuned is the smallest eigenvalue of the kernel matrix due to the Nyström approximation and we choose this factor according to the datasets.

3. Falkon. According to the GPU memory, we set the number of inducing points to be $20k$ with the uniform

Algorithm 4 AxlePro 3

Input: positive definite kernel K, batch size m, size of approximate preconditioner s, learning rates $\eta_1, \eta_2 > 0$, damping factor $\gamma \in (0,1)$, model centers $Z = \{z_i\}_{i=1}^p$.
Output: $f : \mathcal{X} \to \mathbb{R}$ solving (9) approximately.
setup: Sample $J \subset \{1, 2, \ldots, n\}$ with $|J| = s$.
$(\boldsymbol{D}, \Delta, \delta_{q+1}) \leftarrow$ top-q eigensystem of $K(X[J], X[J])$
$\boldsymbol{G} \leftarrow \boldsymbol{D}\sqrt{\Delta^{-1}(I_q - \delta_{q+1}\Delta^{-1})} \in \mathbb{R}^{s\times q}$
Initialize $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{0}_p$.
repeat
  Fetch batch of indices $B \subset \{1, .., n\}$, $|B| = m$
  $\boldsymbol{v} \leftarrow K(X[B], Z)\boldsymbol{\beta} + K(X[B], X)\boldsymbol{b} + K(X[B], X[J])\boldsymbol{d} - Y[B] \in \mathbb{R}^m$ {gradient}
  $\boldsymbol{w} \leftarrow \boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{v} \in \mathbb{R}^s$ {correction}
  $\widetilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}$
  $\widetilde{\boldsymbol{c}} \leftarrow \boldsymbol{c}$
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\beta}$
  $\boldsymbol{\beta} \leftarrow (1+\gamma)\boldsymbol{\alpha} - \gamma\widetilde{\boldsymbol{\alpha}}$ {momentum-1}
  $A \leftarrow K(Z, X)\boldsymbol{a} + K(Z, X[J])\boldsymbol{c} \in \mathbb{R}^p$
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \texttt{K\_solve}(K, Z, A)$
  $C \leftarrow K(Z, X)\boldsymbol{b} + K(Z, X[J])\boldsymbol{d} \in \mathbb{R}^n$
  $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \texttt{solve}(K, Z, C)$
until Stopping criterion is reached
return $f(x) = \sum_{i=1}^n \alpha_i K(x, z_i)$

Algorithm 5 EigenPro 1 [22]

Input: kernel K, batch size m, learning rate $\eta_0 > 0$.
Output: $f : \mathcal{X} \to \mathbb{R}$ that solves (9) approximately.
setup: $\boldsymbol{\alpha} \leftarrow \boldsymbol{0}_n$.
$(\boldsymbol{E}_q, \Lambda, \lambda_{q+1}) \leftarrow$ top-q eigensystem of $K(X, X)$
$\boldsymbol{F}_q \leftarrow \boldsymbol{E}_q\sqrt{I_q - \lambda_{q+1}\Lambda^{-1}}$
repeat
  Fetch a batch of indices $B \subset \{1, 2, \ldots, n\}$
  $\boldsymbol{v} \leftarrow K(X[B], X)\boldsymbol{\alpha} - Y[B] \in \mathbb{R}^m$
  $\boldsymbol{w} \leftarrow \boldsymbol{F}_q\boldsymbol{F}_q[B]^\top\boldsymbol{v} \in \mathbb{R}^n$
  $\boldsymbol{\alpha}[B] \leftarrow \boldsymbol{\alpha}[B] - \eta_0\boldsymbol{v}$ {gradient step}
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \eta_0\boldsymbol{w}$ {correction}
until Stopping criterion is reached
return $f_t(x) = \sum_{i=1}^n \alpha_i K(x, x_i)$

sampling. We do not use any regularization. We comment here that even in our experiments the MSE does not decrease significantly due to the subsampling, the classification accuracy is indeed increasing during training.

4. PCG. We choose the rank to be 100 for the pivoted Cholesky decomposition.

5. GPyTorch. We use instances of the class `IndependentMultitaskGPModel` in order to deal with multiclass classification problems. The optimizer was set to be 'Adam' with learning rate 0.05. As in the case of Falkon, while it is hard to notice significant decrease in the logarithmic scale plot of MSE, the classification accuracy is improved during optimization.

Unless otherwise specified, we do not tune λ_n for AxlePro and use the following setup for our experiments.

For Cifar-10 dataset, we scale the data by a factor of 0.05 as our bandwidth selection. With Gaussian kernel, we set $(m, q, s) = (2000, 500, 20,000)$, while for Laplacian kernel, we use $q = 300$ instead. For the Myrtle5 kernel experiment, we follow the original paper [32] to use ZCA preprocessing (without Leave-One-Outtilting and ZCA augmentation) and store the kernel matrix before performing regression. For this kernel, we set $(m, q, s) = (2000, 300, 10,000)$.

Algorithm 6 EigenPro 2 [23]

Input: kernel K, batch size m, indices $J = \{j_\ell\}_{\ell=1}^{s} \subset \{1, 2, \ldots, n\}$, learning rate $\eta_0 > 0$.
Output: $f : \mathcal{X} \to \mathbb{R}$ that solves (9) approximately.
setup: $\boldsymbol{\alpha} \leftarrow \mathbf{0}_n$.
$(\boldsymbol{D}, \Delta, \delta_{q+1}) \leftarrow$ top-q eigensystem of $K(X[J], X[J])$
$\boldsymbol{G} \leftarrow \boldsymbol{D}\sqrt{\Delta^{-1}(I_q - \delta_{q+1}\Delta^{-1})}$
repeat
  Fetch a batch of indices $B \subset \{1, 2, .., n\}$
  $\boldsymbol{v} \leftarrow K(X[B], X)\boldsymbol{\alpha} - Y[B] \in \mathbb{R}^m$
  $\boldsymbol{w} \leftarrow \boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{v} \in \mathbb{R}^s$
  $\boldsymbol{\alpha}[B] \leftarrow \boldsymbol{\alpha}[B] - \eta_0 \boldsymbol{v}$ {gradient step}
  $\boldsymbol{\alpha}[J] \leftarrow \boldsymbol{\alpha}[J] + \eta_0 \boldsymbol{w}$ {correction}
until Stopping criterion is reached
return $f_t(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i)$

Algorithm 7 EigenPro 3 [1]

Input: kernel K, batch size m, centers $Z = \{z_i\}$, indices $J = \{j_\ell\}_{\ell=1}^{s} \subset \{1, 2, \ldots, n\}$, learning rate $\eta_0 > 0$.
Output: $f : \mathcal{X} \to \mathbb{R}$ that solves (9) approximately.
setup: $\boldsymbol{\alpha} \leftarrow \mathbf{0}_p$.
$(\boldsymbol{D}, \Delta, \delta_{q+1}) \leftarrow$ top-q eigensystem of $K(X[J], X[J])$
$\boldsymbol{G} \leftarrow \boldsymbol{D}\sqrt{\Delta^{-1}(I_q - \delta_{q+1}\Delta^{-1})}$
repeat
  Fetch a batch of indices $B \subset \{1, 2, .., n\}$
  $\boldsymbol{v} \leftarrow K(X[B], Z)\boldsymbol{\alpha} - Y[B] \in \mathbb{R}^m$ {gradient}
  $\boldsymbol{w} \leftarrow \boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{v} \in \mathbb{R}^s$ {correction}
  $\boldsymbol{h} \leftarrow K(Z, X[B])\boldsymbol{v} - K(Z, X[J])\boldsymbol{w}$
  $\boldsymbol{\theta} \leftarrow \texttt{K\_solve}(K, Z, \boldsymbol{h})$ {projection}
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \eta_0 \cdot \boldsymbol{\theta}$
until Stopping criterion is reached
return $f_t(x) = \sum_{i=1}^{n} \alpha_i K(x, z_i)$

For Star Classification dataset, we use $(m, q, s) = (2000, 150, 20,000)$ for Gaussian kernel (bandwidth=2) and $(m, q, s) = (2000, 600, 20,000)$ for Laplacian kernel (bandwidth=4). We preprocess the data with mean subtraction and standard deviation normalization.

For EMNIST Digits dataset, we preprocess the data by mean subtraction and scale by a factor of 0.001. We set $(m, q, s) = (700, 600, 20,000)$ for Gaussian kernel (bandwidth=1) and $(m, q, s) = (800, 600, 20,000)$ for Laplacian kernel(bandwidth=1).

E Exact preconditioner

Calculating the exact preconditioner requires finding the top-q eigensystem of $K(X, X)$ which can cost $n^2 q$ FLOPs. Instead, we can use a Nyström extension. Let $J = \{j_1, j_2, \ldots, j_s\} \subseteq \{1, 2, \ldots, n\}$. Next, let $(\widetilde{\boldsymbol{E}}, \Lambda, \lambda_{q+1})$ be the top-q eigensystem of $K(X[J], X[J])$, where the columns of $\boldsymbol{E} \in \mathbb{R}^{s \times q}$ are the eigenvectors of $K(X[J], X[J])$. Finally, obtain approximate eigenvectors $\boldsymbol{E} \in \mathbb{R}^{n \times q}$ of $K(X, X)$ as

$$\boldsymbol{E} \leftarrow K(X, X[J])\widetilde{\boldsymbol{E}}$$

Due to the matrix multiplication, the vectors $\boldsymbol{E}$ obtained as such are not orthonormal. Hence we run a thin QR decomposition to refine $\boldsymbol{E}$ as,

$$\boldsymbol{E}, \_ \leftarrow \texttt{thinQR}(\boldsymbol{E}).$$

This only has a total complexity $s^2 q + qsn + nq^2$, where $s^2 q$ is the cost of getting eigenvectors $\widetilde{\boldsymbol{E}}$, and sqn is the cost of Nyström extension, and nq^2 is the cost of thin QR decomposition. Since $q \leq s \leq n$, the term nsq dominates.

Algorithm 8 EigenPro 4 [2]

Input: kernel K, batch size m, centers $Z = \{z_i\}$, indices $J = \{j_\ell\}_{\ell=1}^{s} \subset \{1, 2, \ldots, n\}$, learning rate $\eta_0 > 0$.
Output: $f : \mathcal{X} \to \mathbb{R}$ that solves (9) approximately.
setup: $\boldsymbol{\alpha} \leftarrow \mathbf{0}_p, \boldsymbol{a} \leftarrow \mathbf{0}_n, \boldsymbol{c} \leftarrow \mathbf{0}_{|J|}, \boldsymbol{h} \leftarrow \mathbf{0}_0$
$(\boldsymbol{D}, \Delta, \delta_{q+1}) \leftarrow$ top-q eigensystem of $K(X[J], X[J])$
$\boldsymbol{G} \leftarrow \boldsymbol{D}\sqrt{\Delta^{-1}(I_q - \delta_{q+1}\Delta^{-1})}$
repeat
  Fetch a batch of indices $B \subset \{1, 2, .., n\}$
  $\boldsymbol{v} \leftarrow K(X[B], Z)\boldsymbol{\alpha} + K(X[B], X)\boldsymbol{a} + K(X[B], X[J])\boldsymbol{c} - Y[B] \in \mathbb{R}^m$ {gradient}
  $\boldsymbol{w} \leftarrow \boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{v} \in \mathbb{R}^s$ {correction}
  $\boldsymbol{a}[B] \leftarrow -\eta_0 \boldsymbol{v}$ {update temporary weights-1}
  $\boldsymbol{c} \leftarrow \boldsymbol{c} + \eta_0 \boldsymbol{w}$ {update temporary weights-2}
  $\boldsymbol{h} \leftarrow \boldsymbol{h} - \eta_0 \left(K(Z, X[B])\boldsymbol{v} + K(Z, X[J])\boldsymbol{w}\right)$ {accummulate gradients}
  if projection condition holds **then**
    $\boldsymbol{\theta} \leftarrow$ `K_solve`$(K, Z, \boldsymbol{h})$ {delayed projection}
    $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \eta_0 \cdot \boldsymbol{\theta}$
    Reset $\boldsymbol{a} \leftarrow \mathbf{0}_n, \boldsymbol{c} \leftarrow \mathbf{0}_s, \boldsymbol{h} \leftarrow \mathbf{0}_p$,
  end if
until Stopping criterion is reached
return $f_t(x) = \sum_{i=1}^{n} \alpha_i K(x, z_i)$

F Details on convergence analysis

Proof of Lemma 7(d). We consider the norms restricted to the subspace $\mathsf{span}(\{K(\cdot, x_i)\}_{i=1}^n)$ since we only care about $\|f\|_{\mathbb{H}}$ and $\left\|\widehat{f}\right\|_{\widehat{\mathbb{H}}_q}$ for $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$.

Now, suppose $f = \widehat{f} = \sum_{i=1}^n \alpha_i \widehat{K}_q(\cdot, x_i)$.

Claim: $\mathsf{span}(\{K(\cdot, x_i)\}_{i=1}^n) = \mathsf{span}(\left\{\widehat{K}_q(\cdot, x_i)\right\}_{i=1}^n)$.

W.L.O.G, we assume $J = (1, 2, \ldots, s)$. Denote $\boldsymbol{v}_i := \boldsymbol{G}\boldsymbol{G}^\top K(X[J], x_i) \in \mathbb{R}^s$

$$\begin{aligned}\widehat{K}_q(\cdot, x_i) &= K(\cdot, x_i) - K(\cdot, X[J])\boldsymbol{G}\boldsymbol{G}^\top K(X[J], x_i) \\ &= K(\cdot, X)(e_i - [\boldsymbol{v}_i \ \mathbf{0}_{n-s}]^T)\end{aligned}$$

So $\sum_{i=1}^n \alpha_i \widehat{K}_q(\cdot, x_i) = \widehat{K}_q(\cdot, X)\boldsymbol{\alpha} = K(\cdot, X)A\boldsymbol{\alpha}$, where $A := \mathbf{I}_n - \begin{pmatrix}\boldsymbol{G}\boldsymbol{G}^T K(J, X) \\ \mathbf{0}_{(n-s)\times n}\end{pmatrix} = \begin{pmatrix}\mathbf{I}_s - \boldsymbol{G}\boldsymbol{G}^T K(J, J) & -\boldsymbol{G}\boldsymbol{G}^T K(J, J^C) \\ \mathbf{0}(n-s) \times s & \mathbf{I}_{(n-s)\times(n-s)}\end{pmatrix} = \begin{pmatrix}\mathbf{I}_s - \boldsymbol{D}(\boldsymbol{I}_q - \delta_{q+1}\Delta^{-1})\boldsymbol{D}^T & -\boldsymbol{G}\boldsymbol{G}^T K(J, J^C) \\ \mathbf{0}(n-s) \times s & \mathbf{I}_{(n-s)\times(n-s)}\end{pmatrix}$. Since A is an upper triangular block matrix, we can easily get $\frac{\delta_{q+1}}{\delta_1} = \min(\frac{\delta_{q+1}}{\delta_1}, 1) \le \lambda(A) \le \max(\frac{\delta_{q+1}}{\delta_q}, 1) = 1$. This also implies A is invertible and hence our Claim holds.

As a consequence,

$$\begin{aligned}\|f\|_{\mathbb{H}}^2 &= (A\boldsymbol{\alpha})^T K(X, X)(A\boldsymbol{\alpha}) \\ &\le \lambda_{\max}(K(X, X)) \|A\boldsymbol{\alpha}\|_2^2 \\ &\le \lambda_{\max}(K(X, X))\lambda_{\max}^2(A) \|\boldsymbol{\alpha}\|_2^2 \\ &\le \lambda_{\max}(K(X, X)) \|\boldsymbol{\alpha}\|_2^2 .\end{aligned}$$

Similarly, $\|f\|_{\mathbb{H}}^2 \ge \lambda_{\min}(K(X, X))\lambda_{\min}^2(A) \|\boldsymbol{\alpha}\|_2^2 \ge \lambda_{\min}(K(X, X))\frac{\delta_{q+1}^2}{\delta_1^2} \|\boldsymbol{\alpha}\|_2^2$

On the other hand,

$$\left\|\widehat{f}\right\|_{\widehat{\mathbb{H}}_q}^2 = \boldsymbol{\alpha}^T \widehat{K}_q(X, X)\boldsymbol{\alpha} \le \lambda_{\max}(\widehat{K}_q(X, X)) \|\boldsymbol{\alpha}\|_2^2 .$$

Similarly, $\left\|\widehat{f}\right\|_{\widehat{\mathbb{H}}_q}^2 \geq \lambda_{\min}(\widehat{K}_q(X,X)) \left\|\boldsymbol{\alpha}\right\|_2^2$.

So by proposition 8, with high probability, there exist constants a, a' depending (polynomially) on the eigenvalues of $K(X,X)$ such that $a \left\|f\right\|_{\mathbb{H}} \leq \left\|\widehat{f}\right\|_{\widehat{\mathbb{H}}_q} \leq a' \left\|f\right\|_{\mathbb{H}}$. □

F.1 Condition numbers for analysis of MaSS

Let λ_1 and λ_n be the largest and smallest non-zero eigenvalues of $\mathcal{K}$.
Let a mini-batch of size m be $\{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^m$. Defining mini-batch covariance operator as follows

$$\widetilde{\mathcal{K}}^{(m)} := \frac{1}{m}\sum_{i=1}^{m} K(\widetilde{x}_i, \cdot) \otimes K(\widetilde{x}_i, \cdot) \tag{35}$$

$$\widetilde{\mathcal{K}}_{\mathcal{P}}^{(m)} := \frac{1}{m}\sum_{i=1}^{m} k_{\mathcal{P}}(\widetilde{x}_i, \cdot) \otimes k_{\mathcal{P}}(\widetilde{x}_i, \cdot) \tag{36}$$

where $\widetilde{\mathcal{K}}_{\mathcal{P}}$ is the covariance operator after preconditioning. The kernel function $k_{\mathcal{P}}$ can be defined in the same way as in Lemma 7 with the exact preconditioner.

F.2 Identifying quantities defined by [20]

Suppose we run MaSS to solve $\min_{f\in\mathbb{H}} \frac{1}{n}\sum_{i=1}^n (S_{\{i\}}f - Y_i)^2$, with Hessian $H = \mathcal{K}$. Note that $S_{\{i\}}f = \langle K(x_i,\cdot), f\rangle_{\mathbb{H}}$. The authors define the quantities L, μ to be the largest and smallest eigenvalues of the Hessian $H : \mathbb{H} \mapsto \mathbb{H}$, and $\kappa = L/\mu$ to be the condition number. In our case, $L = \lambda_1/n$, $\mu = \lambda_n/n$ and $\kappa = \frac{\lambda_1}{\lambda_n}$.

They also define the quantity L_1 to be the smallest number such that

$$\mathbb{E}[\|K(\widetilde{x},\cdot)\|_{\mathbb{H}}^2 K(\widetilde{x},\cdot) \otimes_{\mathbb{H}} K(\widetilde{x},\cdot)] \lesssim L_1 H \tag{37}$$

where $\mathbb{E}$ is over the random variable $\widetilde{x}$ from the empirical distribution. Note that in our case $\|K(\widetilde{x},\cdot)\|_{\mathbb{H}}^2 = K(\widetilde{x},\widetilde{x})$ since,

$$\|K(x,\cdot)\|_{\mathbb{H}}^2 = \langle K(x,\cdot), K(x,\cdot)\rangle_{\mathbb{H}} = K(x,x) \leq \beta := \max_i K(x_i,x_i) \tag{38}$$

Then we have $L_1 \leq \beta$, since

$$\mathbb{E}\left[\|K(\widetilde{x},\cdot)\|^2 K(\widetilde{x},\cdot) \otimes K(\widetilde{x},\cdot)\right] \preceq \beta\, \mathbb{E}\left[K(\widetilde{x},\cdot) \otimes K(\widetilde{x},\cdot)\right] = \beta\, \mathcal{K}$$

Thus

$$L_m := \frac{L_1}{m} + \frac{(m-1)L}{m} \leq \frac{\beta + (m-1)\frac{\lambda_1}{n}}{m} \tag{39}$$

Defining $\widetilde{\kappa}$ as the smallest positive real number such that

$$\mathbb{E}\left[\|K(\widetilde{x},\cdot)\|_{\mathcal{K}^{-1}}^2 K(\widetilde{x},\cdot) \otimes K(\widetilde{x},\cdot)\right] \preceq \widetilde{\kappa}\, \mathcal{K} \tag{40}$$

Deriving $\|K(x_i,\cdot)\|_{\mathcal{K}^{-1}}^2$. Recall that $\psi_i = S^*\boldsymbol{e}_i/\sqrt{\lambda_i}$. Also note that $\mathcal{K}^{-1} = \sum_{i=1}^n \frac{n}{\lambda_i}\psi_i \otimes \psi_i$.

$$\begin{aligned}
\|K(x_i,\cdot)\|_{\mathcal{K}^{-1}}^2 &= \left\langle K(x_i,\cdot),\ \mathcal{K}^{-1}K(x_i,\cdot)\right\rangle_{\mathbb{H}} \\
&= \left\langle K(x_i,\cdot), \left(\sum_{j=1}^n \frac{n}{\lambda_j}\psi_j \otimes \psi_j\right) K(x_i,\cdot)\right\rangle_{\mathbb{H}} \\
&= \sum_{j=1}^n \frac{n}{\lambda_j}\langle \psi_j, K(x_i,\cdot)\rangle_{\mathbb{H}}^2 \ = \ \sum_{j=1}^n \frac{n}{\lambda_j}\psi_j^2(x_i) \overset{(a)}{=} \sum_{j=1}^n \frac{1}{\lambda_j} n\lambda_j e_{ji}^2 && \text{(41a)} \\
&= \ n\sum_{j=1}^n e_{ji}^2 = n\left\|\boldsymbol{e}_j\right\|^2 = n && \text{(41b)}
\end{aligned}$$

where (a) follows from the fact that $\psi_j(x_i) = S_{\{i\}}\psi_j = S_{\{i\}}S^*\boldsymbol{e}_j/\sqrt{\lambda_j} = \boldsymbol{H}_{\{i\}}SS^*\boldsymbol{e}_j/\sqrt{\lambda_j} = \boldsymbol{H}_{\{i\}}\boldsymbol{e}_j\sqrt{\lambda_j}$.

Deriving $\widetilde{\kappa}$

$$\mathbb{E}\left[\|K(\widetilde{x},\cdot)\|^2_{\mathcal{K}^{-1}}\, K(\widetilde{x},\cdot)\otimes K(\widetilde{x},\cdot)\right] \overset{(a)}{=} n\mathbb{E}\left[K(\widetilde{x},\cdot)\otimes K(\widetilde{x},\cdot)\right] = n\ \mathbb{E}\left[\widetilde{\mathcal{K}}^{(1)}\right] = n\ \mathcal{K} \overset{(b)}{\implies} \widetilde{\kappa} = n$$

where (a) follows from equation (41b) and (b) from definition of $\widetilde{\kappa}$ in equation (40). This also implies $\widetilde{c}_m = \widetilde{\kappa}_m$ in lemma 7 since (b) holds for any kernel function.

F.3 Formulae for hyperparameters

$L_1 = \beta, L = \frac{\lambda_1}{n}, \widetilde{\kappa} = n$

$$L_m = \frac{L_1}{m} + \frac{(m-1)L}{m} = \frac{\beta + (m-1)\frac{\lambda_1}{n}}{m} \tag{42a}$$

$$\kappa_m = \frac{nL_m}{\lambda_n} \tag{42b}$$

$$\widetilde{\kappa}_m = \frac{\widetilde{\kappa}}{m} + \frac{m-1}{m} = 1 + \frac{n-1}{m} \tag{42c}$$

$$\eta_1(m) = \frac{1}{L_m} = \frac{m}{\beta + (m-1)\frac{\lambda_1}{n}} \tag{42d}$$

$$\eta_2(m) = \eta_1 \frac{\sqrt{\kappa_m\widetilde{\kappa}_m}}{1+\sqrt{\kappa_m\widetilde{\kappa}_m}}\left(1 - \frac{1}{\widetilde{\kappa}_m}\right) \tag{42e}$$

$$\gamma(m) = \frac{\sqrt{\kappa_m\widetilde{\kappa}_m} - 1}{\sqrt{\kappa_m\widetilde{\kappa}_m} + 1} \tag{42f}$$

F.4 Condition numbers after preconditioning

Note that after preconditioning we are operating in Hilbert space $\mathbb{H}_{\mathcal{P}}$. Also, the largest eigenvalue of $\mathcal{K}_{\mathcal{P}}$ is λ_{q+1} i.e., $L = \lambda_{q+1}$

Similar to equation (37), defining L_1 as the smallest positive number such that

$$\mathbb{E}\left[\|k_{\mathcal{P}}(\widetilde{x},\cdot)\|^2\, k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] \preceq L_1\mathcal{K}_{\mathcal{P}} \tag{43}$$

Deriving $\|k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\|^2$

$$\begin{aligned}
\|k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\|^2 &= \langle k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot), k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\rangle_{\mathbb{H}_{\mathcal{P}}} \\
&= k_{\mathcal{P}}(\boldsymbol{x}_i,\boldsymbol{x}_i) \\
&\overset{(a)}{=} K(\boldsymbol{x}_i,\boldsymbol{x}_i) - \sum_{j=1}^{q}\left(1 - \frac{\lambda_{q+1}}{\lambda_j}\right)\lambda_j e_{ji}^2 \\
&= K(\boldsymbol{x}_i,\boldsymbol{x}_i) - \sum_{j=1}^{q}\left(\lambda_j - \lambda_{q+1}\right)e_{ji}^2
\end{aligned} \tag{44}$$

where (a) is from the definition of $k_{\mathcal{P}}$.
Define $\beta_{\mathcal{P}}$ as the maximum norm of $k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)$ among the samples

$$\beta_{\mathcal{P}} := \max_i ||k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)||^2 = \max_i\left\{K(\boldsymbol{x}_i,\boldsymbol{x}_i) - n\sum_{j=1}^{q}\left(\lambda_j - \lambda_{q+1}\right)e_{ji}^2\right\} \tag{45}$$

Deriving L_1

$$\mathbb{E}\left[\|k_{\mathcal{P}}(\widetilde{x},\cdot)\|^2\, k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] \preceq \beta_{\mathcal{P}}\ \mathbb{E}\left[k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] = \beta_{\mathcal{P}}\ \mathbb{E}\left[\widetilde{\mathcal{K}}^{(1)}_{\mathcal{P}}\right] = \beta_{\mathcal{P}}\ \mathcal{K}_{\mathcal{P}}$$

which implies $L_1 \leq \beta_{\mathcal{P}}$ due to the definition of L_1 in equation (43).

Deriving L_m

$$L_m := \frac{L_1}{m} + \frac{(m-1)L}{m} \leq \frac{\beta_{\mathcal{P}} + (m-1)\lambda_{q+1}}{m} \tag{46}$$

Defining $\widetilde{\kappa}$ as the smallest positive real number such that

$$\mathbb{E}\left[\|k_{\mathcal{P}}(\widetilde{x},\cdot)\|^2_{\mathcal{K}_{\mathcal{P}}^{-1}}\, k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] \preceq \widetilde{\kappa}\, \mathcal{K}_{\mathcal{P}} \tag{47}$$

Deriving $\|k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\|^2_{\mathcal{K}_{\mathcal{P}}^{-1}}$

$$\begin{aligned}
&\|k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\|^2_{\mathcal{K}_{\mathcal{P}}^{-1}} = \left\langle k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot),\ \mathcal{K}_{\mathcal{P}}^{-1} k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\right\rangle_{\mathbb{H}_{\mathcal{P}}} \\
&\overset{(a)}{=} \left\langle k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot), \left(\sum_{j=1}^{q}\frac{1}{\lambda_{q+1}}\psi'_j\otimes\psi'_j + \sum_{j=q+1}^{n}\frac{1}{\lambda_j}\psi'_j\otimes\psi'_j\right) k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\right\rangle_{\mathbb{H}_{\mathcal{P}}} \\
&= \sum_{j=1}^{q}\frac{1}{\lambda_{q+1}}\left\langle\psi'_j, k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\right\rangle^2_{\mathbb{H}_{\mathcal{P}}} + \sum_{j=q+1}^{n}\frac{1}{\lambda_j}\left\langle\psi'_j, k_{\mathcal{P}}(\boldsymbol{x}_i,\cdot)\right\rangle^2_{\mathbb{H}_{\mathcal{P}}} \\
&\overset{(b)}{=} n\left\{\sum_{j=1}^{q} e_{ji}^2 + \sum_{j=q+1}^{n} e_{ji}^2\right\} \\
&\overset{(c)}{=} n
\end{aligned} \tag{48}$$

where (a) is from the eigendecomposition of $\mathcal{K}_{\mathcal{P}}$, (b) follows from using eigenfunction evaluation with ψ' in $\mathbb{H}_{\mathcal{P}}$ and (c) follows from the fact that $\boldsymbol{E} = [\boldsymbol{e}_1\ \boldsymbol{e}_2\ \ldots\ \boldsymbol{e}_n]$ is an orthonormal matrix which implies $\boldsymbol{E}^\top\boldsymbol{E} = \boldsymbol{I} = \boldsymbol{E}\boldsymbol{E}^\top$.

Deriving $\widetilde{\kappa}$

$$\mathbb{E}\left[\|k_{\mathcal{P}}(\widetilde{x},\cdot)\|^2_{\mathcal{K}_{\mathcal{P}}^{-1}}\, k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] \tag{49}$$

$$\overset{(a)}{=} n\ \mathbb{E}\left[k_{\mathcal{P}}(\widetilde{x},\cdot)\otimes k_{\mathcal{P}}(\widetilde{x},\cdot)\right] \tag{50}$$

$$= n\ \mathbb{E}\left[\widetilde{\mathcal{K}_{\mathcal{P}}}^{(1)}\right]$$

$$= n\ \mathcal{K}_{\mathcal{P}} \tag{51}$$

$$\overset{(b)}{\Longrightarrow}\ \widetilde{\kappa} = n \tag{52}$$

where (a) follows from equation (48) and (b) from definition of $\widetilde{\kappa}$ in equation (47).

F.5 MaSS parameters after preconditioning

Using (46), η_1 defined in (23a) can be written as

$$\eta_1^*(m) = \frac{m}{\beta_{\mathcal{P}} + (m-1)\lambda_{q+1}}$$

Using (52), (42), we can write η_2, γ defined in (23b), (23c) as

$$\eta_2^*(m) = \frac{\eta_1^*(m)\sqrt{(\beta_{\mathcal{P}} + (m-1)\lambda_{q+1})\,(n+m-1)}}{\sqrt{(\beta_{\mathcal{P}} + (m-1)\lambda_{q+1})\,(n+m-1)} + m\sqrt{\lambda_n}}$$

$$\gamma^*(m) = \frac{\sqrt{(\beta_{\mathcal{P}} + (m-1)\lambda_{q+1})\,(n+m-1)} - m\sqrt{\lambda_n}}{\sqrt{(\beta_{\mathcal{P}} + (m-1)\lambda_{q+1})\,(n+m-1)} + m\sqrt{\lambda_n}}$$

Since the kernel matrix is positive definite we see that $\lambda_{q+1} \geq \beta/n$ and using (52) we get regime critical points

m_1^*, m_2^* defined in [20] as

$$m_1^* = \min\left(\frac{\beta_{\mathcal{P}}}{\lambda_{q+1}}, n\right) = \frac{\beta_{\mathcal{P}}}{\lambda_{q+1}}$$
$$m_2^* = \max\left(\frac{\beta_{\mathcal{P}}}{\lambda_{q+1}}, n\right) = n$$

Note that there is no saturation regime for kernel methods

If optimal mini-batch size for linear regime $m_* := m_1^*$ is used, then optimal MaSS parameters are

$$\eta_1^* = \frac{m_*^2}{\beta_{\mathcal{P}}(2m_* - 1)}$$
$$\eta_2^* = \frac{m_*^2\sqrt{n + m_* - 1}}{\beta_{\mathcal{P}}(2m_* - 1)\sqrt{n + m_* - 1} + m_*\sqrt{m_*\lambda_n\beta_{\mathcal{P}}(2m_* - 1)}}$$
$$\gamma^* = \frac{\sqrt{\beta_{\mathcal{P}}(2m_* - 1)}\sqrt{n + m_* - 1} - m_*\sqrt{m_*\lambda_n}}{\sqrt{\beta_{\mathcal{P}}(2m_* - 1)}\sqrt{n + m_* - 1} + m_*\sqrt{m_*\lambda_n}}$$

Informal: Assume large natural image dataset with gaussian/laplacian kernel. Due to the eigenvalue decay, m_* is reasonably large and $\beta_{\mathcal{P}}$ is usually very close to 1. The MaSS parameters can be approximated as follows

$$\eta_1^* \approx \frac{m_*}{2}$$
$$\eta_2^* \approx \frac{m_*\sqrt{n + m_* - 1}}{2\sqrt{n + m_* - 1} + m_*\sqrt{2\lambda_n}}$$
$$\gamma^* \approx \frac{\sqrt{2}\sqrt{n + m_* - 1} - m_*\sqrt{\lambda_n}}{\sqrt{2}\sqrt{n + m_* - 1} + m_*\sqrt{\lambda_n}}$$

Proposition 10. *Suppose $m \leq \frac{n}{2}$, we have*

$$m\sqrt{\tilde{\kappa}_m\kappa_m} \leq n\sqrt{\kappa}$$

Proof. According to the definitions of $\tilde{\kappa}_m$ and κ_m, the statement is equivalent to

$$m^2(\frac{n}{m} + \frac{m-1}{m})(\frac{L_1 + (m-1)\lambda_{q+1}}{\lambda_n m}) \leq n^2\frac{\lambda_1}{\lambda_n},$$

i.e.

$$(n + m - 1)(L_1 + (m-1)\lambda_{q+1}) \leq n^2\lambda_1.$$

Since $L_1 = \max iK(x_i, x_i) \leq \lambda_1$, we have

$$\begin{aligned}(n + m - 1)(L_1 + (m-1)\lambda_{q+1}) &\leq 2n(\lambda_1 + (m-1)\lambda_1)\\ &\leq 2mn\lambda_1\\ &\leq n^2\lambda_1.\end{aligned}$$

□

G Acceleration of EigenPro 4

Accelerated iteration of EigenPro 3 in RKHS:

$$f_{t+1} \leftarrow \mathrm{proj}_{\mathcal{Z}}(g_t - \eta_1\mathcal{P}_s\widetilde{\nabla}_f\mathsf{L}(g_t))$$
$$g_{t+1} \leftarrow \mathrm{proj}_{\mathcal{Z}}((1+\gamma)f_{t+1} - \gamma f_t + \eta_2\mathcal{P}_s\widetilde{\nabla}_f\mathsf{L}(g_t))$$

Suppose you project after every T iterations.

Consider the AxlePro 2 iterations

$$f_{t+1} = g_t - \eta \mathcal{P}_s \widetilde{\nabla}_f \mathsf{L}(g_t) \tag{53}$$

$$g_{t+1} = (1+\gamma) f_{t+1} - \gamma f_t + \eta_2 \mathcal{P}_s \widetilde{\nabla}_f \mathsf{L}(g_t) \tag{54}$$

for $t = 1, 2, \ldots, T$ and $f_T = \mathrm{proj}_{\mathcal{Z}}(f_T)$ which requires $f_T(Z)$ and $g_T(Z)$ which will be computed iteratively.

Suppose B_t is the batch at step t, define $C_t := C_{t-1} \cup B_t$ and $C_0 = \emptyset$ and let $f_t = S_Z^* \boldsymbol{\alpha}_t^Z + S_{C_t}^* \boldsymbol{\alpha}_t^X + S_J^* \boldsymbol{\alpha}^J$ and $g_t = S_Z^* \boldsymbol{\beta}_t^Z + S_{C_t}^* \boldsymbol{\beta}_t^X + S_J^* \boldsymbol{\beta}^J$.

Then $\widetilde{\nabla}_f L(g_t) \in \mathsf{range}(S_{B_{t+1}}^*)$ given by

$$\begin{aligned}
\widetilde{\nabla}_f \mathsf{L}(g_t) &= S_{B_{t+1}}^* \boldsymbol{v}_t \\
\boldsymbol{v}_t &:= g_t(X[B_{t+1}]) - y_t \\
g_t(X[B_{t+1}]) &= K(X[B_{t+1}], Z)\boldsymbol{\beta}_t^Z + K(X[B_{t+1}], X[C_t])\boldsymbol{\beta}_t^X + K(X[B_{t+1}], X[J])\boldsymbol{\beta}_t^J \\
\boldsymbol{w}_t &:= \boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B_{t+1}])\boldsymbol{v}_t \\
\mathcal{P}_s \widetilde{\nabla}_f \mathsf{L}(g_t) &= S_{B_{t+1}}^* \boldsymbol{v}_t - S_J^* \boldsymbol{w}_t
\end{aligned}$$

$$\begin{aligned}
S_Z^* \boldsymbol{\alpha}_{t+1}^Z + S_{C_{t+1}}^* \boldsymbol{\alpha}_{t+1}^X + S_J^* \boldsymbol{\alpha}_{t+1}^J &\leftarrow S_Z^* \boldsymbol{\beta}_t^Z + S_{C_t}^* \boldsymbol{\beta}_t^X + S_J^* \boldsymbol{\beta}_t^J - \eta_1 (S_{B_{t+1}}^* \boldsymbol{v}_t - S_J^* \boldsymbol{w}_t) \\
S_Z^* \boldsymbol{\beta}_{t+1}^Z + S_{C_{t+1}}^* \boldsymbol{\beta}_{t+1}^X + S_J^* \boldsymbol{\beta}_{t+1}^J &\leftarrow (1+\gamma)(S_Z^* \boldsymbol{\alpha}_{t+1}^Z + S_{C_t}^* \boldsymbol{\alpha}_{t+1}^X + S_J^* \boldsymbol{\alpha}_{t+1}^J) - \gamma (S_Z^* \boldsymbol{\alpha}_t^Z + S_{C_t}^* \boldsymbol{\alpha}_t^X + S_J^* \boldsymbol{\alpha}_t^J) \\
&\quad + \eta_2 (S_{B_{t+1}}^* \boldsymbol{v}_t - S_J^* \boldsymbol{w}_t)
\end{aligned}$$

This gives us

$$\boldsymbol{\alpha}_{t+1}^Z \leftarrow \boldsymbol{\beta}_t^Z \tag{55a}$$

$$\boldsymbol{\beta}_{t+1}^Z \leftarrow (1+\gamma)\boldsymbol{\alpha}_{t+1}^Z - \gamma \boldsymbol{\alpha}_t^Z \tag{55b}$$

$$\boldsymbol{\alpha}_{t+1}^J \leftarrow \boldsymbol{\beta}_t^J + \eta_1 \boldsymbol{w}_t \tag{55c}$$

$$\boldsymbol{\beta}_{t+1}^J \leftarrow (1+\gamma)\boldsymbol{\alpha}_{t+1}^J - \gamma \boldsymbol{\alpha}_t^J - \eta_2 \boldsymbol{w}_t \tag{55d}$$

$$\boldsymbol{\alpha}_{t+1}^X[C_t] \leftarrow \boldsymbol{\beta}_t^X[C_t] \tag{55e}$$

$$\boldsymbol{\alpha}_{t+1}^X[B_{t+1}] \leftarrow -\eta_1 \boldsymbol{v}_t \tag{55f}$$

$$\boldsymbol{\beta}_{t+1}^X[C_t] \leftarrow (1+\gamma)\boldsymbol{\alpha}_{t+1}^X[C_t] - \gamma \boldsymbol{\alpha}_t^X[C_t] \tag{55g}$$

$$\boldsymbol{\beta}_{t+1}^X[B_{t+1}] \leftarrow (1+\gamma)\boldsymbol{\alpha}_{t+1}^X[B_{t+1}] + \eta_2 \boldsymbol{v}_t = (\eta_2 - (1+\gamma)\eta_1)\boldsymbol{v}_t \tag{55h}$$

Claim 1. If we run the extrapolation iteration $a_{t+1} = (1+\gamma)a_t - \gamma a_{t-1}$ initialized at $a_1 \neq a_0$, then

$$a_t = a_0 + \frac{1-\gamma^t}{1-\gamma}(a_1 - a_0). \tag{56}$$

Proof. We will prove this by induction. Observe that the $t = 1$ case hold trivially. Assume that the formula is true for $t \leq \tau$ for some $\tau \geq 1$. If we prove that the formula holds for $t = \tau + 1$, we are done.

$$a_{\tau+1} = (1+\gamma)a_\tau - \gamma a_{\tau-1} = (1+\gamma)\left(a_0 + \frac{1-\gamma^\tau}{1-\gamma}(a_1 - a_0)\right) - \gamma\left(a_0 + \frac{1-\gamma^{\tau-1}}{1-\gamma}(a_1 - a_0)\right) \tag{57}$$

$$= a_0 + \frac{a_1 - a_0}{1-\gamma}\left((1+\gamma)(1-\gamma^\tau) - \gamma(1-\gamma^{\tau-1})\right) \tag{58}$$

$$= a_0 + \frac{a_1 - a_0}{1-\gamma}\left(1 + \gamma - \gamma^\tau - \gamma^{\tau+1} - \gamma + \gamma^\tau\right) = a_0 + \frac{1-\gamma^{\tau+1}}{1-\gamma}(a_1 - a_0). \tag{59}$$

which proves the claim. □

Let t_B be the time step when batch B is chosen for the first time, i.e., $B_t = B$.

$$\beta_t^X[B] = c_t \boldsymbol{v}_{t_B} \qquad \text{where} \quad c_t = \begin{cases} 0 & t < t_B \\ c_1 & t = t_B \\ (c_1 + \frac{1-\gamma^{(t-t_B+1)}}{1-\gamma} c_2) & t > t_B \end{cases} \tag{60}$$

where $c_1 = -((1+\gamma)\eta_1 - \eta_2)$, and $c_2 = \eta_2 - \eta_1$

Finally, after T iterations, we need to project f_T and g_T back onto $\mathsf{span}(S_Z^*)$. To that end, we must calculate $g_T(Z)$. To that end, consider

$$f_T(Z) = K(Z,Z)\boldsymbol{\alpha}_T^Z + K(Z, X[C_T])\boldsymbol{\alpha}_T^X + K(Z, X[J])\boldsymbol{\alpha}_T^J \tag{61}$$
$$g_T(Z) = K(Z,Z)\boldsymbol{\beta}_T^Z + K(Z, X[C_T])\boldsymbol{\beta}_T^X + K(Z, X[J])\boldsymbol{\beta}_T^J \tag{62}$$

We can compute

$$f_{T+1} = \text{proj}_{\mathcal{Z}}(f_T) \tag{63}$$
$$g_{T+1} = \text{proj}_{\mathcal{Z}}(g_T) \tag{64}$$

by

$$f_{T+1} = S_Z^* K(Z,Z)^{-1} f_T(Z) \tag{65}$$
$$g_{T+1} = S_Z^* K(Z,Z)^{-1} g_T(Z) \tag{66}$$

which can be implemented as

$$\boldsymbol{\alpha}_{T+1}^Z = K(Z,Z)^{-1} f_T(Z) = \boldsymbol{\alpha}_T^Z + K(Z,Z)^{-1}(K(Z, X[C_T])\boldsymbol{\alpha}_T^X + K(Z, X[J])\boldsymbol{\alpha}_T^J) \tag{67}$$
$$\boldsymbol{\beta}_{T+1}^Z = K(Z,Z)^{-1} g_T(Z) = \boldsymbol{\beta}_T^Z + K(Z,Z)^{-1}(K(Z, X[C_T])\boldsymbol{\beta}_T^X + K(Z, X[J])\boldsymbol{\beta}_T^J) \tag{68}$$

$$\boldsymbol{v}_t \leftarrow K(X[B], Z)\boldsymbol{\beta}_t - Y[B] \in \mathbb{R}^m \tag{69a}$$
$$\boldsymbol{w}_t \leftarrow K(Z, X[B])\boldsymbol{v}_t - K(Z, X[J])\boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{v}_t \in \mathbb{R}^{|Z|} \tag{69b}$$
$$\boldsymbol{u}_t \leftarrow \text{solve } K(Z,Z)\boldsymbol{u}_t = \boldsymbol{w}_t \tag{69c}$$
$$\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\beta}_t - \eta_1 \boldsymbol{u}_t \tag{69d}$$
$$\boldsymbol{\beta}_{t+1} \leftarrow (1+\gamma)\boldsymbol{\alpha}_{t+1} - \gamma\boldsymbol{\alpha}_t + \eta_2 \boldsymbol{u}_t \tag{69e}$$

emulate the updates Acceleration algorithm of EigenPro 3.

H ADDITIONAL PROOFS

Proof of Proposition 1. For $f_t = S^*\boldsymbol{\alpha}_t$, a stochastic gradient with respect to the mini-batch $(X[B], Y[B])$ is given by

$$\widetilde{\nabla}_f \mathsf{L}(f_t) = \frac{1}{|B|} S_B^*(S_B f_t - Y[B]) \tag{70a}$$
$$= \frac{1}{|B|} S_B^*(S_B S^* \boldsymbol{\alpha}_t - Y[B]) = S_B^* \boldsymbol{v}_t \tag{70b}$$

Observe that $S_B = \boldsymbol{H}_B S$, whereby $S_B^* = S^* \boldsymbol{H}_B^\top$. The claim follows immediately. □

Proof of Proposition 3. We start by showing that

$$\mathcal{P} S_B^* = S_B^* - S^* \boldsymbol{F}\boldsymbol{F}^\top \boldsymbol{H}_B^\top. \tag{71}$$

For a vector $\boldsymbol{u} \in \mathbb{R}^m$, observe that

$$\mathcal{P}S_B^*\boldsymbol{u} = S_B^*\boldsymbol{u} - \sum_{i=1}^{q}\left(1 - \frac{\lambda_{q+1}}{\lambda_i}\right)\psi_i \otimes_{\mathbb{H}} \psi_i S_B^*\boldsymbol{u} \tag{72}$$

Now the term $\psi_i \otimes_{\mathbb{H}} \psi_i S_B^*\boldsymbol{u}$ simplifies as

$$\begin{aligned}
\psi_i \langle \psi_i, S_B^*\boldsymbol{u}\rangle_{\mathbb{H}} &= \tfrac{1}{\lambda_i} S^*\boldsymbol{e}_i \langle S^*\boldsymbol{e}_i, S_B^*\boldsymbol{u}\rangle_{\mathbb{H}} \\
&= \tfrac{1}{\lambda_i} S^*\boldsymbol{e}_i \left\langle \boldsymbol{e}_i, SS^*\boldsymbol{H}_B^\top\boldsymbol{u}\right\rangle_{\mathbb{R}^n} \\
&= \tfrac{1}{\lambda_i} S^*\boldsymbol{e}_i\boldsymbol{e}_i^\top K(X,X)\boldsymbol{H}_B^\top\boldsymbol{u} \\
&= S^*\boldsymbol{e}_i\boldsymbol{e}_i^\top\boldsymbol{H}_B^\top\boldsymbol{u}
\end{aligned}$$

where we have used the defintion of adjoint and the fact that $S_B^* = (\boldsymbol{H}_B S)^* = S^*\boldsymbol{H}_B^\top$. Summing the q terms, and observing that $\boldsymbol{F} = \sum_{i=1}^{q}(1 - \frac{\lambda_{q+1}}{\lambda_i})\boldsymbol{e}_i\boldsymbol{e}_i^\top$, we have $\mathcal{P}S_B^*\boldsymbol{u} = S_B^*\boldsymbol{u} - S^*\boldsymbol{F}\boldsymbol{F}^\top\boldsymbol{H}_B^\top\boldsymbol{u}$ for all $\boldsymbol{u} \in \mathbb{R}^m$. Since $\boldsymbol{u}$ is arbitrary, this proves the claim in equation (71).

Next, observe that $\mathcal{P}\widetilde{\nabla}_f\mathsf{L}(g_t) = \mathcal{P}S_B^*(S_BS^*\boldsymbol{\beta}_t - Y[B]) = \mathcal{P}S_B^*\boldsymbol{v}_t$, and rewriting the updates in equation (27a) using the relation $f_t = S^*\boldsymbol{\alpha}_t$ and $g_t = S^*\boldsymbol{\beta}_t$, we get

$$\begin{aligned}
S^*\boldsymbol{\alpha}_{t+1} &\leftarrow S^*\boldsymbol{\beta}_t - \eta_1\mathcal{P}S_B^*\boldsymbol{v}_t \\
&= S^*\boldsymbol{\beta}_t - \eta_1 S_B^*\boldsymbol{v}_t + \eta_1 S^*\boldsymbol{F}\boldsymbol{F}^\top\boldsymbol{H}_B^\top\boldsymbol{v} \\
&= S^*(\boldsymbol{\beta}_t - \eta_1\boldsymbol{H}_B^\top\boldsymbol{v}_t + \eta_1\boldsymbol{w}_t)
\end{aligned}$$

which is indeed equation (17c). A similar calculation can also show equation (17d) emulates equation (27b). □

Proof of Proposition 4. We start by showing that

$$\mathcal{Q}S_B^* = S_B^* - S_J^*\boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B]). \tag{73}$$

For a vector $\boldsymbol{u} \in \mathbb{R}^m$, observe that

$$\mathcal{Q}S_B^*\boldsymbol{u} = S_B^*\boldsymbol{u} - \sum_{i=1}^{q}\left(1 - \frac{\delta_{q+1}}{\delta_i}\right)\phi_i \otimes_{\mathbb{H}} \phi_i S_B^*\boldsymbol{u} \tag{74}$$

Now the term $\phi_i \otimes_{\mathbb{H}} \phi_i S_B^*\boldsymbol{u}$ simplifies as

$$\begin{aligned}
\phi_i \langle \phi_i, S_B^*\boldsymbol{u}\rangle_{\mathbb{H}} &= \tfrac{1}{\delta_i} S_J^*\boldsymbol{d}_i \langle S_J^*\boldsymbol{d}_i, S_B^*\boldsymbol{u}\rangle_{\mathbb{H}} \\
&= \tfrac{1}{\delta_i} S_J^*\boldsymbol{d}_i \langle \boldsymbol{d}_i, S_JS_I^*\boldsymbol{u}\rangle_{\mathbb{R}^m} \\
&= \tfrac{1}{\delta_i} S_J^*\boldsymbol{d}_i\boldsymbol{d}_i^\top K(X[J], X[B])\boldsymbol{u}.
\end{aligned}$$

Summing the q terms, and observing that $\boldsymbol{G} = \sum_{i=1}^{q}\frac{1}{\delta_i}(1 - \frac{\delta_{q+1}}{\delta_i})\boldsymbol{d}_i\boldsymbol{d}_i^\top$ we have $\mathcal{Q}S_B^*\boldsymbol{u} = S_B^*\boldsymbol{u} - S_J^*\boldsymbol{G}\boldsymbol{G}^\top K(X[J], X[B])\boldsymbol{u}$ for all $\boldsymbol{u} \in \mathbb{R}^m$. Since $\boldsymbol{u}$ is arbitrary, this proves the claim in equation (73). The rest of the proof proceeds similar to the proof of Proposition 3. □

Table 6: Laplacian Kernel

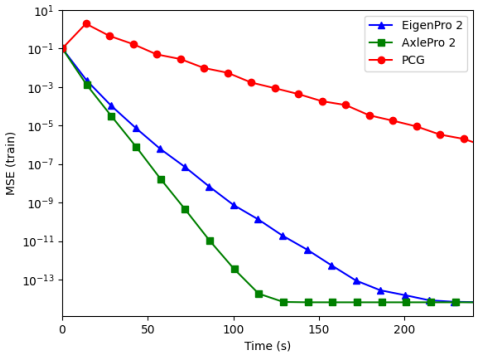
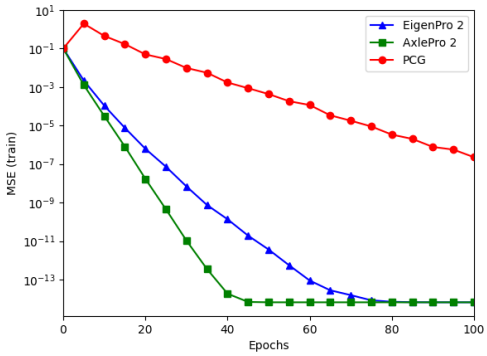
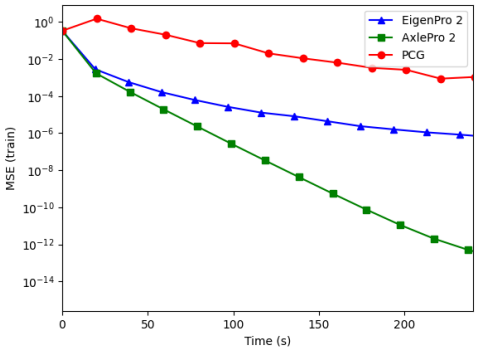
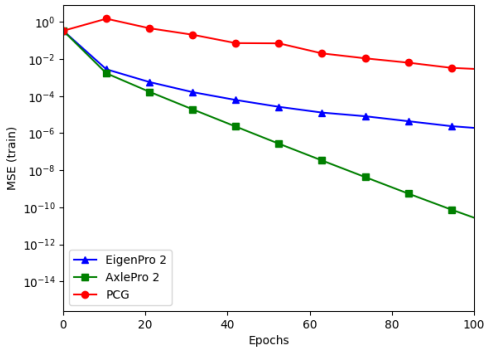
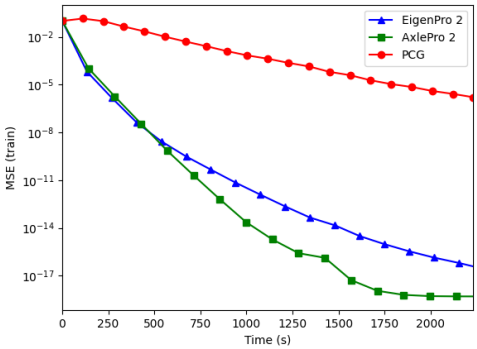
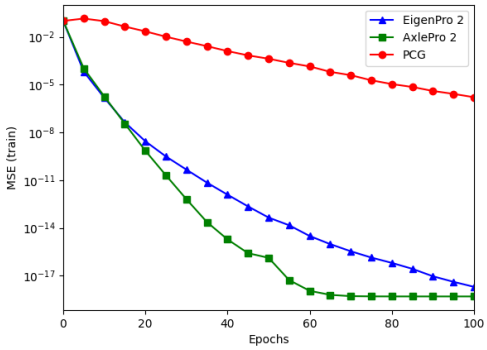
Dataset	MSE v/s Time	MSE v/s Epochs
CIFAR10		
Stellar Classification		
EMNIST Digits		

Table 7: Gaussian Kernel

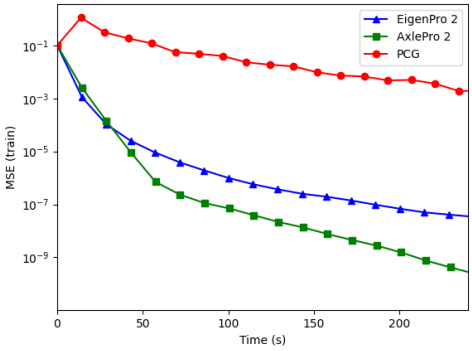
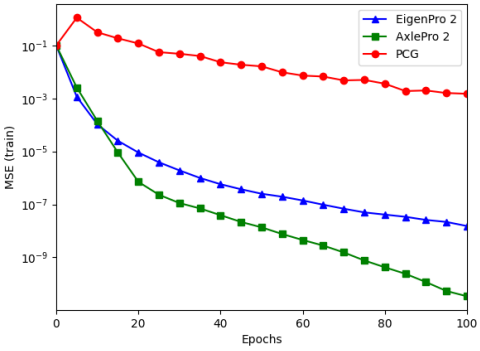
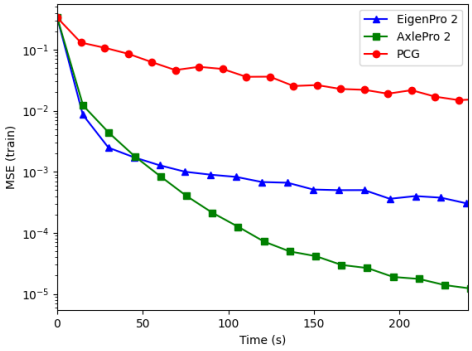
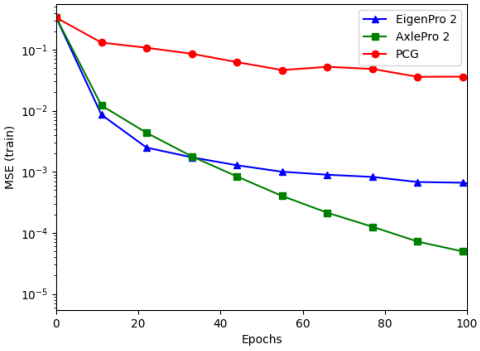
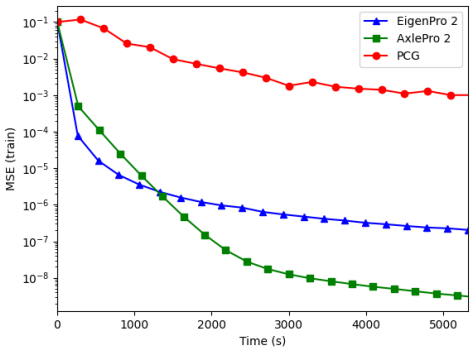
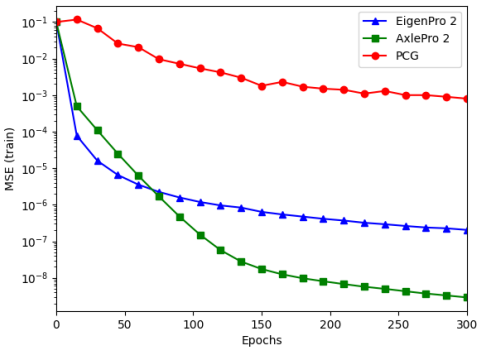
Dataset	MSE v/s Time	MSE v/s Epochs
CIFAR10		
Stellar Classification		
EMNIST Digits		

Table 8: Laplacian Kernel with single precision

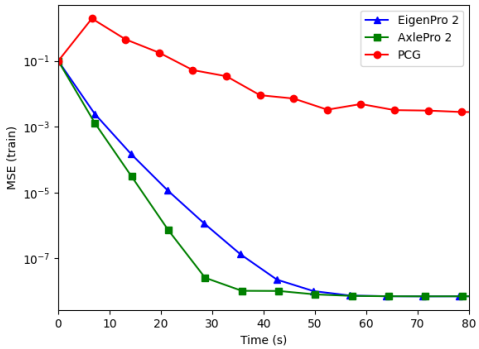
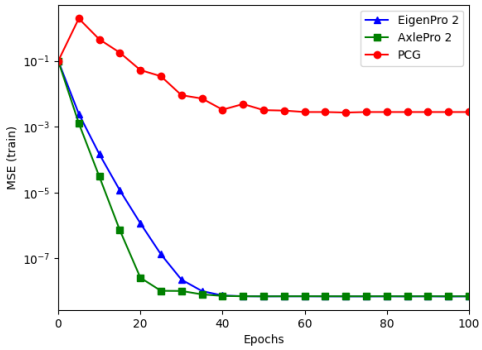
Dataset	MSE v/s Time	MSE v/s Epochs
CIFAR10		

Table 9: Myrtle5 Kernel with stored kernel matrix

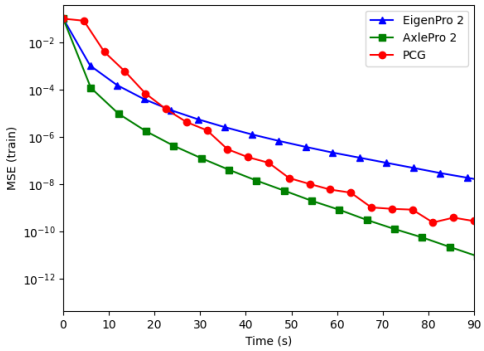
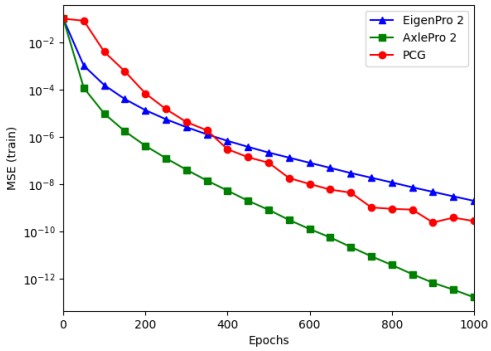
Dataset	MSE v/s Time	MSE v/s Epochs
CIFAR10		

Table 10: Acceleration by Nyström approximation

Dataset	MSE v/s Time	MSE v/s Epochs
$n = 5k$, $s = 800$