Supplementary Material: Posterior and Computational Uncertainty in Gaussian Processes

Jonathan Wenger[1,2] **Geoff Pleiss**[2]

Marvin Pförtner[1] **Philipp Hennig**[1,3] **John P. Cunningham**[2]

[1] University of Tübingen
[2] Columbia University
[3] Max Planck Institute for Intelligent Systems, Tübingen

This supplementary material contains additional results and in particular proofs for all theoretical statements. References referring to sections, equations or theorem-type environments within this document are prefixed with 'S', while references to, or results from, the main paper are stated as is.

S1 Connections to Other GP Approximations

S1.1 Pivoted Cholesky Decomposition

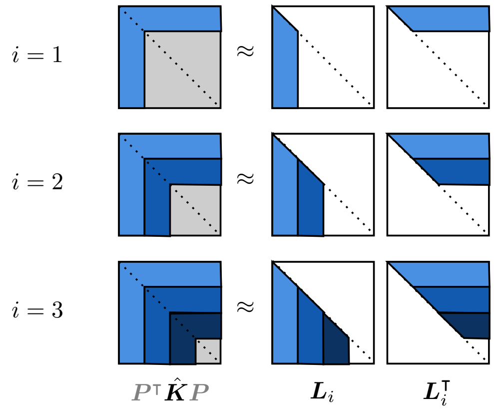
Theorem S3 (Pivoted Cholesky Decomposition)
Let $(j_i)_{i=1}^n$ be a set of indices defining the pivot elements of the pivoted Cholesky decomposition and $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ the corresponding permutation matrix. Assume the actions of Algorithm 1 are given by the standard unit vectors $\boldsymbol{s}_i = \boldsymbol{P}\boldsymbol{e}_i = \boldsymbol{e}_{j_i}$, i.e.

$$(\boldsymbol{s}_i)_j = (\boldsymbol{e}_{j_i})_j = \begin{cases} 1 & \text{if } j = j_i \\ 0 & \text{otherwise} \end{cases}. \tag{S17}$$

Then Algorithm 1 recovers the pivoted Cholesky decomposition, i.e. it holds for all $i \in \{0, \dots, n\}$ that

$$\boldsymbol{P}^\intercal \boldsymbol{Q}_i \boldsymbol{P} = \boldsymbol{L}_i \boldsymbol{L}_i^\intercal, \tag{S18}$$

where $\boldsymbol{L}_i \in \mathbb{R}^{n \times i}$ is the (partial) Cholesky factor of $\boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P}$ as computed by Algorithm S2.



Algorithm S2: (Pivoted) Cholesky Decomposition

Input: kernel matrix $\hat{\boldsymbol{K}}$, permutation matrix $\boldsymbol{P}$
Output: lower triangular $\boldsymbol{L}_i$, s.t. $\boldsymbol{L}_i \boldsymbol{L}_i^\intercal \approx \boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P}$

```
procedure CHOLESKY(K̂, P)
    A ← Pᵀ K̂ P
    for i ∈ {1, ..., n} do
        l_i ← A_{:i} / √(A_ii)
        A ← A − l_i l_iᵀ = Pᵀ K̂ P − L_i L_iᵀ
        L_i = (L_{i−1}   l_i)
    return L_i
```

Figure S1: *Cholesky decomposition.* Every column added to the lower triangular Cholesky factor $\boldsymbol{L}$ defines the ith "right angle ruler"-pattern in $P^\intercal \hat{\boldsymbol{K}} P$. The bottom right matrix in gray given by $P^\intercal \hat{\boldsymbol{K}} P - \boldsymbol{L}_i \boldsymbol{L}_i^\intercal = P^\intercal \hat{\boldsymbol{K}} P - \sum_{j=1}^{i} \boldsymbol{l}_j \boldsymbol{l}_j^\intercal$ changes every iteration.

Proof. We proceed by induction. Assume (S18) holds after i iterations of Algorithm 1. For the base case $i = 0$, it holds by assumption that $\boldsymbol{P}^\intercal \boldsymbol{Q}_0 \boldsymbol{P} = \boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{C}_0 \hat{\boldsymbol{K}} \boldsymbol{P} = \boldsymbol{0}$. Now for the induction step $i \to i + 1$, we have

$$
\begin{aligned}
\frac{1}{\eta_{i+1}} \hat{\boldsymbol{K}} \boldsymbol{d}_i \boldsymbol{d}_i^\intercal \hat{\boldsymbol{K}} &= \frac{1}{\eta_{i+1}} \hat{\boldsymbol{K}} \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \\
&= \frac{1}{\eta_{i+1}} \hat{\boldsymbol{K}} (\boldsymbol{\Sigma}_0 - \boldsymbol{C}_i) \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} (\boldsymbol{\Sigma}_0 - \boldsymbol{C}_i) \hat{\boldsymbol{K}} \\
&= \frac{1}{\eta_{i+1}} (\hat{\boldsymbol{K}} - \boldsymbol{Q}_i) \boldsymbol{s}_{i+1} \boldsymbol{s}_{i+1}^\intercal (\hat{\boldsymbol{K}} - \boldsymbol{Q}_i) \\
&\overset{\text{IH}}{=} \frac{1}{\eta_{i+1}} (\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal) \boldsymbol{s}_{i+1} \boldsymbol{s}_{i+1}^\intercal (\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal) \\
&= \frac{(\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal) \boldsymbol{P} \boldsymbol{e}_{i+1}}{\sqrt{\boldsymbol{e}_{i+1}^\intercal \boldsymbol{P}^\intercal (\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal) \boldsymbol{P} \boldsymbol{e}_{i+1}}} \frac{\boldsymbol{e}_{i+1}^\intercal \boldsymbol{P}^\intercal (\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal)}{\sqrt{\boldsymbol{e}_{i+1}^\intercal \boldsymbol{P}^\intercal (\hat{\boldsymbol{K}} - \boldsymbol{P} \boldsymbol{L}_i \boldsymbol{L}_i^\intercal \boldsymbol{P}^\intercal) \boldsymbol{P} \boldsymbol{e}_{i+1}}} \\
&= \frac{\boldsymbol{P} (\boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P} - \boldsymbol{L}_i \boldsymbol{L}_i^\intercal) \boldsymbol{e}_{i+1}}{\sqrt{\boldsymbol{e}_{i+1}^\intercal (\boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P} - \boldsymbol{L}_i \boldsymbol{L}_i^\intercal) \boldsymbol{e}_{i+1}}} \frac{\boldsymbol{e}_{i+1}^\intercal (\boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P} - \boldsymbol{L}_i \boldsymbol{L}_i^\intercal) \boldsymbol{P}^\intercal}{\sqrt{\boldsymbol{e}_{i+1}^\intercal (\boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P} - \boldsymbol{L}_i \boldsymbol{L}_i^\intercal) \boldsymbol{e}_{i+1}}} \\
&= \boldsymbol{P} \boldsymbol{l}_{i+1} \boldsymbol{l}_{i+1}^\intercal \boldsymbol{P}^\intercal.
\end{aligned}
$$

where $\boldsymbol{l}_{i+1}$ is given by Algorithm S2. Combining this with one more use of the induction hypothesis we obtain

$$
\begin{aligned}
\boldsymbol{P}^\intercal \boldsymbol{Q}_{i+1} \boldsymbol{P} &= \boldsymbol{P}^\intercal \boldsymbol{Q}_i \boldsymbol{P} + \frac{1}{\eta_{i+1}} \boldsymbol{P}^\intercal \hat{\boldsymbol{K}} \boldsymbol{d}_{i+1} \boldsymbol{d}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{P} \\
&= \boldsymbol{L}_i \boldsymbol{L}_i^\intercal + \boldsymbol{l}_{i+1} \boldsymbol{l}_{i+1}^\intercal = (\boldsymbol{L}_i \quad \boldsymbol{l}_{i+1}) \begin{pmatrix} \boldsymbol{L}_i^\intercal \\ \boldsymbol{l}_{i+1}^\intercal \end{pmatrix} = \boldsymbol{L}_{i+1} \boldsymbol{L}_{i+1}^\intercal
\end{aligned}
$$

This proves the claim.

□

S1.2 Singular / Eigenvalue Decomposition

Theorem S4 (Singular / Eigenvalue Decomposition)
Let the actions $\boldsymbol{s}_i = \mathbf{u}_i$ of Algorithm 1 be given by the eigenvectors $\mathbf{u}_i$ of $\hat{\boldsymbol{K}}$ in arbitrary order. Then

for $i \in \{1, \dots, n\}$ it holds that

$$\boldsymbol{C}_i = \boldsymbol{U}_i \boldsymbol{\Lambda}_i^{-1} \boldsymbol{U}_i^\intercal = \mathrm{SVD}_i(\hat{\boldsymbol{K}}^{-1})$$
$$\boldsymbol{Q}_i = \boldsymbol{U}_i \boldsymbol{\Lambda}_i \boldsymbol{U}_i^\intercal = \mathrm{SVD}_i(\hat{\boldsymbol{K}}),$$

where $\boldsymbol{U} = (\mathbf{u}_1, \dots, \mathbf{u}_i) \in \mathbb{R}^{n \times i}$ and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_i) \in \mathbb{R}^{i \times i}$ is the diagonal matrix of eigenvalues of $\hat{\boldsymbol{K}}$ with the order given by the order of the actions.

Proof. It holds by assumption and eq. (S37), that

$$\boldsymbol{C}_i = \boldsymbol{S}_i(\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal = \boldsymbol{U}_i(\boldsymbol{U}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{U}_i)^{-1} \boldsymbol{U}_i^\intercal = \boldsymbol{U}_i \boldsymbol{\Lambda}_i^{-1} \boldsymbol{U}_i^\intercal,$$

as well as

$$\boldsymbol{Q}_i = \hat{\boldsymbol{K}} \boldsymbol{C}_i \hat{\boldsymbol{K}} = \hat{\boldsymbol{K}} \boldsymbol{U}_i \boldsymbol{\Lambda}_i^{-1} \boldsymbol{U}_i^\intercal \hat{\boldsymbol{K}} = \boldsymbol{U}_i \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\Lambda}_i \boldsymbol{U}_i^\intercal = \boldsymbol{U}_i \boldsymbol{\Lambda}_i \boldsymbol{U}_i^\intercal$$

This proves the claim. □

S1.3 Conjugate Gradient Method

Algorithm S3: Preconditioned Conjugate Gradient Method [38]

Input: kernel matrix $\hat{\boldsymbol{K}}$, labels $\boldsymbol{y}$, prior mean $\boldsymbol{\mu}$, preconditioner $\hat{\boldsymbol{P}}$
Output: representer weights $\boldsymbol{v}_i \approx \hat{\boldsymbol{K}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$

procedure CG($\hat{\boldsymbol{K}}, \boldsymbol{y} - \boldsymbol{\mu}, \hat{\boldsymbol{P}}$)
  $\boldsymbol{v}_0 \leftarrow \boldsymbol{0}$
  $\boldsymbol{s}_0 \leftarrow \boldsymbol{0}$
  while $\|\boldsymbol{r}_i\|_2 > \max(\delta_{\text{rtol}}\|\boldsymbol{y}\|_2, \delta_{\text{atol}})$ **and** $i < i_{\max}$ **do**
    $\boldsymbol{r}_{i-1} \leftarrow (\boldsymbol{y} - \boldsymbol{\mu}) - \hat{\boldsymbol{K}} \boldsymbol{v}_{i-1}$
    $\boldsymbol{s}_i \leftarrow \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1} - \frac{(\hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1})^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i-1}}{\boldsymbol{s}_{i-1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i-1}} \boldsymbol{s}_{i-1}$
    $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_{i-1} + \frac{(\hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1})^\intercal \boldsymbol{r}_{i-1}}{\boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i} \boldsymbol{s}_i$
  return $\boldsymbol{v}$

Theorem S5 (Preconditioned Conjugate Gradient Method)
Let $\hat{\boldsymbol{P}} \in \mathbb{R}^{n \times n}$ be a symmetric positive definite preconditioner. Assume the actions of Algorithm 1 are given by

$$\begin{aligned} \boldsymbol{s}_1^{\mathrm{CG}} &= \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_0 \\ \boldsymbol{s}_i^{\mathrm{CG}} &= \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1} - \frac{(\hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1})^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i-1}}{\boldsymbol{s}_{i-1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i-1}} \boldsymbol{s}_{i-1} \end{aligned} \tag{S19}$$

the preconditioned conjugate gradient method. Then Algorithm 1 recovers preconditioned CG initialized at $\boldsymbol{v}_0^{\mathrm{CG}} = \boldsymbol{0}$, i.e. it holds for $i \in \{1, \dots, n\}$ that

$$\boldsymbol{s}_i = \boldsymbol{d}_i = \boldsymbol{s}_i^{\mathrm{CG}} \tag{S20}$$
$$\boldsymbol{v}_i = \boldsymbol{v}_i^{\mathrm{CG}} \tag{S21}$$
$$\boldsymbol{r}_{i-1} = \boldsymbol{r}_{i-1}^{\mathrm{CG}}. \tag{S22}$$

Proof. First note that by assumption $\boldsymbol{s}_i = \boldsymbol{s}_i^{\mathrm{CG}}$ for all i. We prove the remaining claims by induction. For the base case we have by assumption $\boldsymbol{d}_0 = \boldsymbol{\Sigma}_0 \hat{\boldsymbol{K}} \boldsymbol{s}_0 = \boldsymbol{s}_0 = \boldsymbol{s}_0^{\mathrm{CG}}$ and $\boldsymbol{v}_0 = \boldsymbol{0} = \boldsymbol{v}_0^{\mathrm{CG}}$. Now for the induction step $i \to i+1$ assume the hypotheses (S20), (S21) and (S22) hold $\forall j \leq i$. Using the properties of CG it holds for $j' < i$ that

$$\boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{j'} = 0 \tag{S23}$$
$$(\hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_i)^\intercal \boldsymbol{s}_{j'} = 0 \tag{S24}$$
$$(\hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_i)^\intercal \boldsymbol{r}_{j'} = 0 \tag{S25}$$
$$\langle \boldsymbol{s}_1, \dots, \boldsymbol{s}_i \rangle = \langle \boldsymbol{r}_0, \hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}} \boldsymbol{r}_0, \dots, (\hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}})^{i-1} \boldsymbol{r}_0 \rangle = \langle \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_0, \dots, \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1} \rangle \tag{S26}$$

We now first show $\hat{\boldsymbol{K}}$-conjugacy of the actions in iteration $i+1$. We have for $j \leq i$ that

$$\begin{aligned}\boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j &= \big(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i - \frac{(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i}{\boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i} \boldsymbol{s}_i\big)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j \\ &= (\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j - \frac{(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i}{\boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i} \boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j\end{aligned}$$

Now if $j = i$, clearly $\boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j = \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i = 0$. If $j < i$, we have using (S26), that

$$\hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}} \boldsymbol{s}_j \in \langle \hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}} \boldsymbol{r}_0, \dots, (\hat{\boldsymbol{P}}^{-1}\hat{\boldsymbol{K}})^j \boldsymbol{r}_0 \rangle \subset \langle \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_0, \dots, \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_j \rangle. \tag{S27}$$

Therefore we obtain for $j < i$, that

$$\boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j \overset{\text{(S23)}}{=} \boldsymbol{r}_i^\intercal \hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}} \boldsymbol{s}_j \overset{\text{(S27)}}{=} \boldsymbol{r}_i^\intercal \left(\sum_{\ell=1}^{j} \gamma_\ell \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_\ell \right) \overset{\text{(S25)}}{=} 0. \tag{S28}$$

Thus in combination we have

$$\forall j \in \{1, \dots, i\}: \quad \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_j = 0. \tag{S29}$$

Now for the search direction we have

$$\begin{aligned}\boldsymbol{d}_{i+1} = \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} &= \left(\boldsymbol{\Sigma}_0 - \sum_{j=1}^{i} \frac{\boldsymbol{d}_j \boldsymbol{d}_j^\intercal}{\eta_j} \right) \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \\ &= \boldsymbol{s}_{i+1} - \sum_{j=1}^{i} \frac{\boldsymbol{d}_j^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1}}{\eta_j} \boldsymbol{d}_j \overset{\text{(S20)}}{=} \boldsymbol{s}_{i+1} - \sum_{j=1}^{i} \frac{\boldsymbol{s}_j^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1}}{\eta_j} \boldsymbol{d}_j \\ &\overset{\text{(S29)}}{=} \boldsymbol{s}_{i+1}.\end{aligned} \tag{S30}$$

Further, we have for the solution estimate, that $\boldsymbol{v}_{i+1} = \boldsymbol{v}_i + \boldsymbol{d}_{i+1} \frac{\alpha_{i+1}}{\eta_{i+1}}$. It holds that

$$\begin{aligned}\alpha_{i+1} = \boldsymbol{s}_{i+1}^\intercal \boldsymbol{r}_i &= \big(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i - \frac{(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i}{\boldsymbol{s}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i} \boldsymbol{s}_i\big)^\intercal \boldsymbol{r}_i \\ &= (\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \boldsymbol{r}_i - \sum_{j=}^{i} c_j (\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_{j-1})^\intercal \boldsymbol{r}_i \overset{\text{(S25)}}{=} (\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \boldsymbol{r}_i\end{aligned}$$

as well as

$$\eta_{i+1} = \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} = \boldsymbol{d}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \overset{\text{(S30)}}{=} \boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1}$$

Combining the above and recalling Algorithm S3, we obtain

$$\boldsymbol{v}_{i+1} = \boldsymbol{v}_i + \boldsymbol{d}_{i+1} \frac{\alpha_{i+1}}{\eta_{i+1}} = \boldsymbol{v}_i + \boldsymbol{d}_{i+1} \frac{(\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i)^\intercal \boldsymbol{r}_i}{\boldsymbol{s}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1}} = \boldsymbol{v}_{i+1}^{\text{CG}}.$$

Finally, the residual is computed identically in Algorithm 1 as in Algorithm S3, giving

$$\boldsymbol{r}_i = (\boldsymbol{y} - \boldsymbol{\mu}) - \hat{\boldsymbol{K}} \boldsymbol{v}_i = (\boldsymbol{y} - \boldsymbol{\mu}) - \hat{\boldsymbol{K}} \boldsymbol{v}_i^{\text{CG}} = \boldsymbol{r}_i^{\text{CG}}.$$

This proves the claims. □

Corollary S2 (Preconditioned Gradient Actions as CG Actions)
Choosing actions

$$\boldsymbol{s}_i = \hat{\boldsymbol{P}}^{-1} \boldsymbol{r}_{i-1} \tag{S31}$$

in Theorem S5 instead also reproduces the preconditioned conjugate gradient method, i.e. it holds for $i \in \{1, \dots, n\}$ *that*

$$\boldsymbol{d}_i = \boldsymbol{s}_i^{\text{CG}} \tag{S32}$$

$$\boldsymbol{v}_i = \boldsymbol{v}_i^{\text{CG}} \tag{S33}$$

$$\boldsymbol{r}_{i-1} = \boldsymbol{r}_{i-1}^{\text{CG}}. \tag{S34}$$

Proof. It suffices to show that $\boldsymbol{d}_i = \boldsymbol{s}_i^{\mathrm{CG}}$. The rest of the argument is then identical to the proof of Theorem S5. We prove the claim by induction. For the base case by assumption $\boldsymbol{s}_1 = \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_0 = \boldsymbol{s}_1^{\mathrm{CG}}$. Now for the induction step $i \to i+1$, assume that $\boldsymbol{d}_j = \boldsymbol{s}_j$ for all $j \leq i$, then

$$\begin{aligned}
\boldsymbol{d}_{i+1} &= \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i \\
&= (\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}})\hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i \\
&= \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i - \boldsymbol{D}_i(\boldsymbol{D}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{D}_i)^{-1}\boldsymbol{D}_i^\intercal \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i && \text{By eq. (S37).} \\
&\overset{\text{IH}}{=} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i - \boldsymbol{S}_i^{\mathrm{CG}}((\boldsymbol{S}_i^{\mathrm{CG}})^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i^{\mathrm{CG}})^{-1}(\boldsymbol{S}_i^{\mathrm{CG}})^\intercal \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i
\end{aligned}$$

Now by the same argument as in eq. (S28) in the proof of Theorem S5 we have for all $j < i$ that $\boldsymbol{r}_i^\intercal \hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{K}} \boldsymbol{s}_j^{\mathrm{CG}} = 0$. Therefore

$$\begin{aligned}
&= \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i - \boldsymbol{s}_i^{\mathrm{CG}}((\boldsymbol{s}_i^{\mathrm{CG}})^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_i^{\mathrm{CG}})^{-1}(\boldsymbol{s}_i^{\mathrm{CG}})^\intercal \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i \\
&= \boldsymbol{s}_{i+1}^{\mathrm{CG}} && \text{By eq. (S19).}
\end{aligned}$$

This proves the claim. □

Corollary S3 (Deflated Conjugate Gradient Method)
Let the first $0 < \ell < n$ actions $(\boldsymbol{s}_i)_{i=1}^{\ell}$ of Algorithm 1 be linearly independent and the remaining ones be given by $\boldsymbol{s}_i = \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i$, where $\hat{\boldsymbol{P}} \approx \hat{\boldsymbol{K}}$ is a preconditioner. Then Algorithm 1 is equivalent to the preconditioned deflated CG algorithm [63, Alg. 3.6] with deflation subspace $\operatorname{span}\{\boldsymbol{S}_\ell\}$.

Proof. By the form of preconditioned deflated CG given in Algorithm 3.6 of Saad et al. [63] and Corollary S2, it suffices to show that the residual $\boldsymbol{r}_\ell$ satisfies $\boldsymbol{S}_\ell^\intercal \boldsymbol{r}_\ell = \boldsymbol{0}$ and that for all $i > \ell$, it holds that

$$\boldsymbol{s}_i^{\mathrm{defCG}} = \boldsymbol{d}_i = (\boldsymbol{I} - \boldsymbol{C}_{i-1}\hat{\boldsymbol{K}})\boldsymbol{s}_i.$$

Now it holds by Lemma S2 and eq. (S37), that

$$\boldsymbol{S}_\ell^\intercal \boldsymbol{r}_\ell = \boldsymbol{S}_\ell^\intercal(\boldsymbol{I} - \hat{\boldsymbol{K}}\boldsymbol{C}_\ell)(\boldsymbol{y} - \boldsymbol{\mu}) = \underbrace{\boldsymbol{S}_\ell^\intercal(\boldsymbol{I} - \hat{\boldsymbol{K}}\boldsymbol{S}_\ell(\boldsymbol{S}_\ell^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_\ell)^{-1}\boldsymbol{S}_\ell^\intercal)}_{=\boldsymbol{0}}(\boldsymbol{y} - \boldsymbol{\mu}) = \boldsymbol{0}.$$

This proves the first claim. Now, by Saad et al. [63, Alg. 3.6], the search directions $(\boldsymbol{s}_i^{\mathrm{defCG}})_{i=\ell+1}^n$ of preconditioned deflated CG are given by

$$\begin{aligned}
\boldsymbol{s}_i^{\mathrm{defCG}} &= \boldsymbol{s}_i^{\mathrm{CG}} - \boldsymbol{S}_\ell(\boldsymbol{S}_\ell^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_\ell)^{-1}\boldsymbol{S}_\ell^\intercal \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i \\
&= (\boldsymbol{I} - \boldsymbol{C}_{\ell+1:(i-1)}\hat{\boldsymbol{K}})\boldsymbol{s}_i - \boldsymbol{S}_\ell(\boldsymbol{S}_\ell^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_\ell)^{-1}\boldsymbol{S}_\ell^\intercal \hat{\boldsymbol{K}} \hat{\boldsymbol{P}}^{-1}\boldsymbol{r}_i && \text{Corollary S2} \\
&= (\boldsymbol{I} - \boldsymbol{C}_{\ell+1:(i-1)}\hat{\boldsymbol{K}})\boldsymbol{s}_i - \boldsymbol{C}_\ell \hat{\boldsymbol{K}} \boldsymbol{s}_i \\
&= (\boldsymbol{I} - (\boldsymbol{C}_{\ell+1:(i-1)} - \boldsymbol{C}_\ell)\hat{\boldsymbol{K}})\boldsymbol{s}_i \\
&= (\boldsymbol{I} - \boldsymbol{C}_{i-1}\hat{\boldsymbol{K}})\boldsymbol{s}_i \\
&= \boldsymbol{d}_i
\end{aligned}$$

This proves the claim. □

Remark S1 (Preconditioning and Algorithm 1)
Iterative methods typically have convergence rates depending on the condition number of the system matrix. One successful strategy in practice to accelerate convergence is to use a preconditioner $\hat{\boldsymbol{P}} \approx \hat{\boldsymbol{K}}$ [64]. A preconditioner needs to be cheap to compute and allow efficient matrix-vector multiplication $\boldsymbol{v} \mapsto \hat{\boldsymbol{P}}^{-1}\boldsymbol{v}$. Now, Algorithm 1 implicitly constructs and applies a *deflation-based preconditioner*, which are defined via a deflation subspace to be projected out [65]. In Algorithm 1 this is precisely the already explored space $\operatorname{span}\{\boldsymbol{S}_i\} = \operatorname{span}\{\boldsymbol{D}_i\}$ spanned by the actions. Therefore, if we run a mixed strategy, meaning first choosing actions that define a certain subspace and then choose residual actions, we recover the *deflated conjugate gradient method* [63] (see Corollary S3 for a proof). Alternatively, one can also use byproducts of the iteration of Algorithm 1 to construct a diagonal-plus-low-rank preconditioner of the form $\hat{\boldsymbol{P}} = \sigma^2\boldsymbol{I} + \boldsymbol{U}\boldsymbol{U}^\intercal \approx \hat{\boldsymbol{K}}$ where $\boldsymbol{U} = \boldsymbol{K}\boldsymbol{D}_i \operatorname{diag}(\eta_1, \dots, \eta_i) \in \mathbb{R}^{n \times i}$. Therefore, again if running a mixed strategy, one can first construct a preconditioner and then accelerate the convergence of subsequent CG iterations. In this sense one can double-dip in terms of preconditioning (conjugate) gradient iterations by combining these two techniques *at essentially no overhead*.

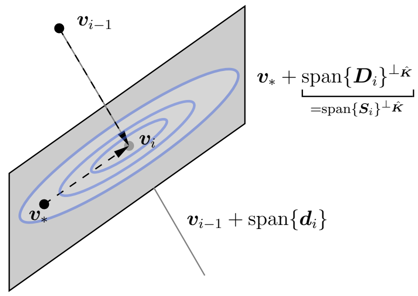


Figure S2: *Geometric perspective on the probabilistic linear solver learning representer weights* $\boldsymbol{v}_*$.

S1.4 Inducing Point Methods

Theorem S6 (Approximate Posterior Mean of Nyström, SoR, DTC and SVGP)
Let $\boldsymbol{Z} \in \mathbb{R}^{n \times m}$ *be a set of distinct inducing inputs such that* $\text{rank}(\boldsymbol{K}_{\boldsymbol{XZ}}) = m \leq n$. *Then the posterior mean of the Nyström variants subset of regressors (SoR) and deterministic training conditional (DTC) is identical to the one of SVGP and given by*

$$\begin{aligned}\mu(\cdot) &= k(\cdot, \boldsymbol{Z})(\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} + \sigma^2 \boldsymbol{K}_{\boldsymbol{ZZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu}) \\ &= q(\cdot, \boldsymbol{X})\boldsymbol{K}_{\boldsymbol{XZ}}(\boldsymbol{K}_{\boldsymbol{ZX}}(q(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I})\boldsymbol{K}_{\boldsymbol{XZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu})\end{aligned} \tag{S35}$$

Proof. First, note that by eqns. (16b) and (20b) of Quiñonero-Candela and Rasmussen [20] the posterior mean of SoR and DTC is identical and given by

$$\mu(\cdot) = k(\cdot, \boldsymbol{Z})(\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} + \sigma^2 \boldsymbol{K}_{\boldsymbol{ZZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu})$$

Now, by Theorem 5 of Wild et al. [43] the posterior mean of SVGP for a fixed set of inducing points is equivalent to the Nyström approximation, which takes the form above. Recognizing that $\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} \in \mathbb{R}^{m \times m}$ is invertible, it holds that

$$\begin{aligned}\mu(\cdot) &= k(\cdot, \boldsymbol{Z})(\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} + \sigma^2 \boldsymbol{K}_{\boldsymbol{ZZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu}) \\ &= k(\cdot, \boldsymbol{Z})(\boldsymbol{K}_{\boldsymbol{ZZ}}(\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} + \sigma^2 \boldsymbol{I}))^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu}) \\ &= k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}((\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}})^{-1}(\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}} + \sigma^2 \boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}}))^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu}) \\ &= k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{XZ}}(\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{K}_{\boldsymbol{XZ}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{K}_{\boldsymbol{ZX}} + \sigma^2 \boldsymbol{I})\boldsymbol{K}_{\boldsymbol{XZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu}) \\ &= q(\cdot, \boldsymbol{X})\boldsymbol{K}_{\boldsymbol{XZ}}(\boldsymbol{K}_{\boldsymbol{ZX}}(q(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I})\boldsymbol{K}_{\boldsymbol{XZ}})^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}(\boldsymbol{y} - \boldsymbol{\mu})\end{aligned}$$

This proves the claim. □

S2 Theoretical Results and Proofs

S2.1 Properties of Algorithm 1

Lemma S1 (Geometric Properties of Algorithm 1)
Let $i \in \{1, \ldots, n\}$, *and assume* $\boldsymbol{\Sigma}_0$ *is chosen such that* $\boldsymbol{\Sigma}_0 \hat{\boldsymbol{K}} \boldsymbol{s}_j = \boldsymbol{s}_j$ *for all* $j \leq i$ *(e.g.* $\boldsymbol{\Sigma}_0 = \hat{\boldsymbol{K}}^{-1}$*). Then it holds for the quantities computed by Algorithm 1 that*

$$\text{span}\{\boldsymbol{S}_i\} = \text{span}\{\boldsymbol{D}_i\} \tag{S36}$$

$$\boldsymbol{C}_i = \boldsymbol{D}_i(\boldsymbol{D}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{D}_i)^{-1}\boldsymbol{D}_i^\intercal = \boldsymbol{S}_i(\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal \tag{S37}$$

$$\boldsymbol{C}_i \hat{\boldsymbol{K}} \textit{ is the } \hat{\boldsymbol{K}}\textit{-orthogonal projection onto } \text{span}\{\boldsymbol{D}_i\} \tag{S38}$$

$$\boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \textit{ is the } \hat{\boldsymbol{K}}\textit{-orthogonal projection onto } \text{span}\{\boldsymbol{D}_i\}^{\perp_{\hat{\boldsymbol{K}}}} \tag{S39}$$

$$\boldsymbol{d}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{d}_j = 0 \qquad \textit{for all } j < i \tag{S40}$$

where $\boldsymbol{S}_i = (\boldsymbol{s}_1 \cdots \boldsymbol{s}_i) \in \mathbb{R}^{n \times i}$ *and* $\boldsymbol{D}_i = (\boldsymbol{d}_1 \cdots \boldsymbol{d}_i) \in \mathbb{R}^{n \times i}$.

Proof. We prove the claims by induction. We begin with the base case $i = 1$.

By assumption it holds that $\boldsymbol{S}_1 = \boldsymbol{s}_1 = \boldsymbol{\Sigma}_0 \hat{\boldsymbol{K}} \boldsymbol{s}_1 = \boldsymbol{d}_1 = \boldsymbol{D}_1$. Now by Algorithm 1, we have $\boldsymbol{C}_1 = \frac{1}{\eta_1} \boldsymbol{d}_1 \boldsymbol{d}_1^\intercal$, which with the above proves (S37). By the batched form (S37) of $\boldsymbol{C}_i$, the statements (S38) and (S39) follow immediately. Finally, $\hat{\boldsymbol{K}}$-orthogonality for a single search direction holds trivially.

Now for the induction step $i \to i+1$. Assume that eqs. (S36) to (S40) hold for iteration i. Then we have that

$$\boldsymbol{d}_{i+1} = \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} = \boldsymbol{s}_{i+1} - \boldsymbol{C}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \overset{\text{(S37)}}{=} \boldsymbol{s}_{i+1} - \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \in \operatorname{span}\{\boldsymbol{S}_{i+1}\}$$

By the induction hypothesis the above also implies $\operatorname{span}\{\boldsymbol{S}_{i+1}\} = \operatorname{span}\{\boldsymbol{D}_{i+1}\}$. This proves eq. (S36). Next, we have by the induction hypotheses (S37) and (S40) that

$$\begin{aligned}
\boldsymbol{C}_{i+1} &= \boldsymbol{C}_i + \frac{1}{\eta} \boldsymbol{d}_{i+1} \boldsymbol{d}_{i+1}^\intercal \\
&= \boldsymbol{D}_i (\boldsymbol{D}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{D}_i)^{-1} \boldsymbol{D}_i^\intercal + \frac{1}{\eta_{i+1}} \boldsymbol{d}_{i+1} \boldsymbol{d}_{i+1}^\intercal \\
&= \sum_{k=1}^{i+1} \frac{1}{\eta_k} \boldsymbol{d}_k \boldsymbol{d}_k^\intercal \\
&= \boldsymbol{D}_{i+1} (\boldsymbol{D}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{D}_{i+1})^{-1} \boldsymbol{D}_{i+1}^\intercal
\end{aligned}$$

This proves the first equality of eq. (S37). For the second, first recognize that an orthogonal projection onto a linear subspace $\operatorname{span}\{\boldsymbol{A}\}$ with respect to the $\boldsymbol{B}$-inner product is given by $\boldsymbol{P}_{\boldsymbol{A}} = \boldsymbol{A}(\boldsymbol{A}^\intercal \boldsymbol{B} \boldsymbol{A})^{-1} \boldsymbol{A}^\intercal \boldsymbol{B}$. The projection onto its $\boldsymbol{B}$-orthogonal subspace is given by $\boldsymbol{P}_{\boldsymbol{A}^{\perp_{\boldsymbol{B}}}} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{A}}$. Therefore eqs. (S38) and (S39) follow directly from the above argument. Now since projection onto a subspace is unique and independent of the choice of basis, we have by $\operatorname{span}\{\boldsymbol{D}_{i+1}\} = \operatorname{span}\{\boldsymbol{S}_{i+1}\}$ that

$$\boldsymbol{C}_i \hat{\boldsymbol{K}} = \boldsymbol{P}_{\boldsymbol{D}_{i+1}} = \boldsymbol{P}_{\boldsymbol{S}_{i+1}} = \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}}$$

Now since $\hat{\boldsymbol{K}}$ is non-singular, the second equality of eq. (S37) follows. Finally, we will prove $\hat{\boldsymbol{K}}$-orthogonality of the search directions. Let $j < i+1$, then it holds that

$$\boldsymbol{d}_{i+1}^\intercal \hat{\boldsymbol{K}} \boldsymbol{d}_j = (\underbrace{\boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1}}_{\in \operatorname{span}\{\boldsymbol{S}_i\}^{\perp \hat{\boldsymbol{K}}}})^\intercal \hat{\boldsymbol{K}} \underbrace{\boldsymbol{d}_j}_{\in \operatorname{span}\{\boldsymbol{S}_i\}} = 0$$

by eqs. (S36) and (S39). This completes the proof. □

Corollary S4
Let $i \in \{1, \dots, n\}$. It holds for $\boldsymbol{C}_i \hat{\boldsymbol{K}}$, the $\hat{\boldsymbol{K}}$-orthogonal projection onto $\boldsymbol{S}_i$, that

$$(\boldsymbol{C}_i \hat{\boldsymbol{K}})^2 = \boldsymbol{C}_i \hat{\boldsymbol{K}} \tag{S41}$$
$$\boldsymbol{C}_i \hat{\boldsymbol{K}} \boldsymbol{C}_i = \boldsymbol{C}_i \tag{S42}$$

Further for $\boldsymbol{H}_i = \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} = \boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}$ the $\hat{\boldsymbol{K}}$-orthogonal projection onto $\boldsymbol{S}_i^{\perp \hat{\boldsymbol{K}}}$, we have

$$\boldsymbol{H}_i^2 = \boldsymbol{H}_i \tag{S43}$$
$$\boldsymbol{H}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{H}_i = \boldsymbol{H}_i^\intercal \hat{\boldsymbol{K}} = \hat{\boldsymbol{K}} \boldsymbol{H}_i \tag{S44}$$

Proof. By Lemma S1, it holds that $\boldsymbol{C}_i = \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal$. Therefore

$$\boldsymbol{C}_i \hat{\boldsymbol{K}} \boldsymbol{C}_i = \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^{-1} \boldsymbol{S}_i^\intercal = \boldsymbol{C}_i.$$

This proves (S42) and (S41). Define $\boldsymbol{H}_i = \boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}$, then

$$\boldsymbol{H}_i \boldsymbol{H}_i = (\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}})(\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}) = \boldsymbol{I} - 2\boldsymbol{C}_i \hat{\boldsymbol{K}} + (\boldsymbol{C}_i \hat{\boldsymbol{K}})^2 = \boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}} = \boldsymbol{H}_i$$

as well as

$$\begin{aligned}
\boldsymbol{H}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{H}_i &= (\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}})^\intercal \hat{\boldsymbol{K}} (\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}) = (\hat{\boldsymbol{K}} - \hat{\boldsymbol{K}} \boldsymbol{C}_i \hat{\boldsymbol{K}})(\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}) \\
&= \hat{\boldsymbol{K}} - 2\hat{\boldsymbol{K}} \boldsymbol{C}_i \hat{\boldsymbol{K}} + \hat{\boldsymbol{K}} (\boldsymbol{C}_i \hat{\boldsymbol{K}})^2 \\
&= \hat{\boldsymbol{K}} - \hat{\boldsymbol{K}} \boldsymbol{C}_i \hat{\boldsymbol{K}} = \boldsymbol{H}_i^\intercal \hat{\boldsymbol{K}} = \hat{\boldsymbol{K}} \boldsymbol{H}_i.
\end{aligned}$$

□

Lemma S2
Let $\boldsymbol{\Sigma}_0 = \hat{\boldsymbol{K}}^{-1}$, then it holds that

$$\boldsymbol{C}_i(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{v}_i, \tag{S45}$$
$$\boldsymbol{\Sigma}_i(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{v}_* - \boldsymbol{v}_i. \tag{S46}$$

Proof. We prove the statement by induction. By assumption $\boldsymbol{C}_0(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{v}_0$. Now assume (S45) holds. Then for $i \to i+1$, we have

$$\boldsymbol{C}_{i+1}(\boldsymbol{y}-\boldsymbol{\mu}) = (\boldsymbol{C}_i + \frac{1}{\eta_{i+1}}\boldsymbol{d}_{i+1}\boldsymbol{d}_{i+1}^\intercal)(\boldsymbol{y}-\boldsymbol{\mu}) \overset{\text{IH}}{=} \boldsymbol{v}_i + \frac{\boldsymbol{d}_{i+1}^\intercal(\boldsymbol{y}-\boldsymbol{\mu})}{\eta_{i+1}}\boldsymbol{d}_{i+1}$$

Now by the update to the representer weights in Algorithm 1 it suffices to show that $\alpha_{i+1} = \boldsymbol{d}_{i+1}^\intercal(\boldsymbol{y}-\boldsymbol{\mu})$. We have

$$\begin{aligned}\boldsymbol{d}_{i+1}^\intercal(\boldsymbol{y}-\boldsymbol{\mu}) &= (\boldsymbol{\Sigma}_i\hat{\boldsymbol{K}}\boldsymbol{s}_{i+1})^\intercal(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{s}_{i+1}^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_i(\boldsymbol{y}-\boldsymbol{\mu})\\ &= \boldsymbol{s}_{i+1}^\intercal\hat{\boldsymbol{K}}(\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_i)(\boldsymbol{y}-\boldsymbol{\mu}) \overset{\text{IH}}{=} \boldsymbol{s}_{i+1}^\intercal((\boldsymbol{y}-\boldsymbol{\mu}) - \hat{\boldsymbol{K}}\boldsymbol{v}_i) = \boldsymbol{s}_{i+1}^\intercal\boldsymbol{r}_i = \alpha_i.\end{aligned}$$

□

Lemma S3
Let $\boldsymbol{\Sigma}_0 = \hat{\boldsymbol{K}}^{-1}$, $\boldsymbol{C}_0 = \boldsymbol{0}$ and consequently $\boldsymbol{v}_0 = \boldsymbol{0}$, then it holds for the residual at iteration $i \in \{1, \dots, n\}$ that

$$\boldsymbol{r}_{i-1} = \hat{\boldsymbol{K}}(\boldsymbol{v}_* - \boldsymbol{v}_{i-1}) \tag{S47}$$
$$= \hat{\boldsymbol{K}}\boldsymbol{\Sigma}_{i-1}\hat{\boldsymbol{K}}\boldsymbol{v}_* \tag{S48}$$
$$= (\hat{\boldsymbol{K}} - \boldsymbol{Q}_{i-1})\boldsymbol{v}_*. \tag{S49}$$

Proof. It holds by definition, that

$$\boldsymbol{r}_{i-1} = (\boldsymbol{y}-\boldsymbol{\mu}) - \hat{\boldsymbol{K}}\boldsymbol{v}_{i-1} = \hat{\boldsymbol{K}}\boldsymbol{v}_* - \hat{\boldsymbol{K}}\boldsymbol{v}_{i-1} = \hat{\boldsymbol{K}}(\boldsymbol{v}_* - \boldsymbol{v}_{i-1}).$$

Further we have by eq. (S46), that

$$= \hat{\boldsymbol{K}}\boldsymbol{\Sigma}_{i-1}(\boldsymbol{y}-\boldsymbol{\mu}) = \hat{\boldsymbol{K}}\boldsymbol{\Sigma}_{i-1}\hat{\boldsymbol{K}}\boldsymbol{v}_*,$$

and finally, by the definition of the kernel matrix approximation in Algorithm 1, we obtain

$$= \hat{\boldsymbol{K}}(\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_{i-1})\hat{\boldsymbol{K}}\boldsymbol{v}_* = (\hat{\boldsymbol{K}} - \boldsymbol{Q}_{i-1})\boldsymbol{v}_*.$$

□

Proposition S3 (Batch of Observations)
Let $\boldsymbol{\Sigma}_0$ such that $\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{s}_j = \boldsymbol{s}_j$ for all $j \in \{1, \dots, i\}$. Then after i iterations the posterior over the representer weights in (4) *is equivalent to the one computed for a batch of observations, i.e.*

$$\boldsymbol{v}_i = \boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal(\boldsymbol{y}-\boldsymbol{\mu})$$
$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0$$

Proof. This can be seen as a direct consequence of recursively applying Bayes' theorem

$$p(\boldsymbol{v}_* \mid \{\alpha_i\}_{i=1}^m, \{\boldsymbol{s}_i\}_{i=1}^m) = \frac{p(\alpha_m \mid \boldsymbol{s}_m, \boldsymbol{v}_*)p(\boldsymbol{v}_* \mid \{\alpha_i\}_{i=1}^{m-1}, \{\boldsymbol{s}_i\}_{i=1}^{m-1})}{\int p(\alpha_m \mid \boldsymbol{s}_m, \boldsymbol{v}_*)p(\boldsymbol{v}_* \mid \{\alpha_i\}_{i=1}^{m-1}, \{\boldsymbol{s}_i\}_{i=1}^{m-1})d\boldsymbol{v}_*}.$$

However, here we also give a geometric proof based on the projection property of the precision matrix approximation $\boldsymbol{C}_i$. By using eq. (S37) and the assumption on $\boldsymbol{\Sigma}_0$ we have that

$$\begin{aligned}\boldsymbol{C}_i = \boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal &= \boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal\\ &= \boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\end{aligned}$$

This proves that

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_0 - \boldsymbol{C}_i = \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i(\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{K}}\boldsymbol{S}_i)^{-1}\boldsymbol{S}_i^\intercal\hat{\boldsymbol{K}}\boldsymbol{\Sigma}_0$$

Now by eq. (S45) it holds that $\boldsymbol{C}_i(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{v}_i$. This proves the claim. □

Proposition S4 (Posterior Contraction)
Let $\boldsymbol{S}_i \in \mathbb{R}^{n \times i}$ be the actions chosen by Algorithm 1, then its posterior contracts as

$$\operatorname{tr}\big(\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_0^{-1}\big) = \operatorname{tr}(\boldsymbol{\Sigma}_i \hat{\boldsymbol{K}}) = n - \operatorname{rank}(\boldsymbol{S}_i).$$

Proof. Since $\boldsymbol{\Sigma}_0 = \hat{\boldsymbol{K}}^{-1}$, we have by eq. (S37), that

$$\begin{aligned}
\operatorname{tr}\big(\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_0^{-1}\big) &= \operatorname{tr}((\boldsymbol{\Sigma}_0 - \boldsymbol{C}_i)\hat{\boldsymbol{K}}) \\
&= \operatorname{tr}\big(\boldsymbol{I}_n - \boldsymbol{S}_i(\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^\dagger \boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}}\big) \\
&= \operatorname{tr}(\boldsymbol{I}_n) - \operatorname{tr}(\underbrace{\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i (\boldsymbol{S}_i^\intercal \hat{\boldsymbol{K}} \boldsymbol{S}_i)^\dagger}_{\in \mathbb{R}^{i \times i}}) \\
&= n - \operatorname{rank}(\boldsymbol{S}_i)
\end{aligned}$$

Now, if the actions $\boldsymbol{S}_i$ are chosen linearly independent, then $\operatorname{rank}(\boldsymbol{S}_i) = i$. □

Theorem S7 (Online GP Approximation with Algorithm 1)
Let $n, n' \in \mathbb{N}$ and consider training data sets $\boldsymbol{X} \in \mathbb{R}^{n \times d}, \boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{X}' \in \mathbb{R}^{n' \times d}, \boldsymbol{y}' \in \mathbb{R}^{n'}$. Consider two sequences of actions $(\boldsymbol{s}_i)_{i=1}^n \in \mathbb{R}^n$ and $(\tilde{\boldsymbol{s}}_i)_{i=1}^{n+n'} \in \mathbb{R}^{n+n'}$ such that for all $i \in \{1, \dots, n\}$, it holds that

$$\tilde{\boldsymbol{s}}_i = \begin{pmatrix} \boldsymbol{s}_i \\ \boldsymbol{0} \end{pmatrix} \tag{S50}$$

Then the posterior returned by Algorithm 1 for the dataset $(\boldsymbol{X}, \boldsymbol{y})$ using actions $\boldsymbol{s}_i$ is identical to the posterior returned by Algorithm 1 for the extended dataset using actions $\tilde{\boldsymbol{s}}_i$, i.e. it holds for any $i \in \{1, \dots, n\}$, that

$$\textsc{IterGP}(\mu, k, \boldsymbol{X}, \boldsymbol{y}, (\boldsymbol{s}_i)_i) = (\mu_i, k_i) = (\tilde{\mu}_i, \tilde{k}_i) = \textsc{IterGP}\left(\mu, k, \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}' \end{pmatrix}, \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{pmatrix}, (\tilde{\boldsymbol{s}}_i)_i\right).$$

Proof. Define $\tilde{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}' \end{pmatrix}$ and $\tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{pmatrix}$. We begin by showing that the search directions of both methods satisfy

$$\boldsymbol{d}_i' = \begin{pmatrix} \boldsymbol{d}_i \\ \boldsymbol{0} \end{pmatrix}. \tag{S51}$$

We proceed by induction. For $i = 0$ it holds by definition of Algorithm 1 and eq. (S50) that

$$\tilde{\boldsymbol{d}}_0 = \tilde{\boldsymbol{s}}_0 = \begin{pmatrix} \boldsymbol{s}_0 \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d}_0 \\ \boldsymbol{0} \end{pmatrix}. \tag{S52}$$

Now for the induction step $i \to i+1$, assume that (S51) holds for $j \in \{1, \dots, i\}$. Then, we have

$$\begin{aligned}
\tilde{\boldsymbol{d}}_{i+1} &= \tilde{\boldsymbol{\Sigma}}_{i-1}(k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_{n+n'})\tilde{\boldsymbol{s}}_{i+1} \\
&= (\boldsymbol{I}_{n+n'} - \tilde{\boldsymbol{C}}_i(k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_{n+n'}))\tilde{\boldsymbol{s}}_{i+1} \\
&= \tilde{\boldsymbol{s}}_{i+1} - \sum_{j=1}^{i} \frac{1}{\tilde{\eta}_j} \tilde{\boldsymbol{d}}_j (\tilde{\boldsymbol{d}}_j)^\intercal (k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_{n+n'})\tilde{\boldsymbol{s}}_{i+1} \\
&\overset{\text{IH}}{=} \begin{pmatrix} \boldsymbol{s}_{i+1} \\ \boldsymbol{0} \end{pmatrix} - \sum_{j=1}^{i} \frac{1}{\tilde{\eta}_j} \begin{pmatrix} \boldsymbol{d}_j \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_j^\intercal & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} k(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{I}_n & k(\boldsymbol{X}, \boldsymbol{X}') \\ k(\boldsymbol{X}', \boldsymbol{X}) & k(\boldsymbol{X}', \boldsymbol{X}') + \boldsymbol{I}_{n'} \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_{i+1} \\ \boldsymbol{0} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{s}_{i+1} - \sum_{j=1}^{i} \frac{1}{\eta_j} \boldsymbol{d}_j (\boldsymbol{d}_j)^\intercal \hat{\boldsymbol{K}} \boldsymbol{s}_{i+1} \\ \boldsymbol{0} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{d}_{i+1} \\ \boldsymbol{0} \end{pmatrix}
\end{aligned}$$

where we used that $\tilde{\eta}_j = \tilde{\boldsymbol{s}}_j^\intercal (k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_{n+n'})\tilde{\boldsymbol{d}}_j = \boldsymbol{s}_j^\intercal \hat{\boldsymbol{K}} \boldsymbol{d}_j = \eta_j$. This proves eq. (S51). Now recognize that

$$\begin{aligned}
\tilde{\alpha}_j = \tilde{\boldsymbol{s}}_j^\intercal \tilde{\boldsymbol{r}}_j = \tilde{\boldsymbol{s}}_j^\intercal (\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{K}}\tilde{\boldsymbol{C}}_i(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}))
&= \tilde{\boldsymbol{s}}_j^\intercal (\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}} - (\tilde{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}) \sum_{\ell=1}^{j} \frac{1}{\tilde{\eta}_\ell} \tilde{\boldsymbol{d}}_\ell \tilde{\boldsymbol{d}}_\ell^\intercal (\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}})) \\
&= \boldsymbol{s}_j^\intercal (\boldsymbol{y} - \boldsymbol{\mu}) - \sum_{\ell=1}^{j} \frac{1}{\eta_\ell} \boldsymbol{s}_j^\intercal \hat{\boldsymbol{K}} \boldsymbol{d}_\ell \boldsymbol{d}_\ell^\intercal (\boldsymbol{y} - \boldsymbol{\mu}) \\
&= \boldsymbol{s}_j^\intercal (\boldsymbol{y} - \boldsymbol{\mu} - \hat{\boldsymbol{K}} \boldsymbol{C}_j (\boldsymbol{y} - \boldsymbol{\mu})) \\
&= \boldsymbol{s}_j^\intercal \boldsymbol{r}_j \\
&= \alpha_j
\end{aligned}$$

Therefore, we finally have that

$$\begin{aligned}
\tilde{\mu}_i(\cdot) &= \mu(\cdot) + k(\cdot, \tilde{\boldsymbol{X}})\tilde{\boldsymbol{v}}_i = \mu(\cdot) + k(\cdot, \tilde{\boldsymbol{X}}) \sum_{j=1}^{i} \frac{\tilde{\alpha}_j}{\tilde{\eta}_j} \tilde{\boldsymbol{d}}_j \\
&= \mu(\cdot) + k(\cdot, \boldsymbol{X})\boldsymbol{v}_i
\end{aligned}$$

as well as

$$\begin{aligned}
\tilde{k}_i(\cdot, \cdot) &= k(\cdot, \cdot) - k(\cdot, \tilde{\boldsymbol{X}})\tilde{\boldsymbol{C}}_i k(\tilde{\boldsymbol{X}}, \cdot) = k(\cdot, \cdot) - k(\cdot, \tilde{\boldsymbol{X}}) \sum_{j=1}^{i} \frac{1}{\tilde{\eta}_j} \tilde{\boldsymbol{d}}_j (\tilde{\boldsymbol{d}}_j)^\intercal k(\tilde{\boldsymbol{X}}, \cdot) \\
&= k(\cdot, \cdot) - k(\cdot, \boldsymbol{X}) \sum_{j=1}^{i} \frac{1}{\eta_j} \boldsymbol{d}_j (\boldsymbol{d}_j)^\intercal k(\boldsymbol{X}, \cdot) = k(\cdot, \cdot) - k(\cdot, \boldsymbol{X})\boldsymbol{C}_i k(\boldsymbol{X}, \cdot) = k_i(\cdot, \cdot).
\end{aligned}$$

□

Remark S2 (Streaming Gaussian Processes)
Theorem S7 shows that any variant of IterGP can be used in the online setting where data arrives sequentially *while* the algorithm is running. Now, if we assume data points arrive one at a time, we choose unit vector actions (IterGP-Chol) and perform one iteration of Algorithm 1 after each data point, then Algorithm 1 simply computes the mathematical GP posterior.

S2.2 Approximation of Representer Weights

Proposition 2 (Relative Error Bound for the Representer Weights)
For any choice of actions a relative error bound $\rho(i)$, s.t. $\|\boldsymbol{v}_* - \boldsymbol{v}_i\|_{\hat{\boldsymbol{K}}} \leq \rho(i)\|\boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}}$ is given by

$$\rho(i) = (\bar{\boldsymbol{v}}_*^\intercal \underbrace{(\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}})}_{\textit{projection onto } \operatorname{span}\{\boldsymbol{S}_i\}^{\perp \hat{\boldsymbol{K}}}} \bar{\boldsymbol{v}}_*)^{\frac{1}{2}} \leq \lambda_{\max}(\boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}) \leq 1 \tag{9}$$

where $\bar{\boldsymbol{v}}_* = \boldsymbol{v}_* / \|\boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}}$. If the actions $\{\boldsymbol{s}_i\}_{i=1}^n$ are linearly independent, then $\rho(i) \leq \delta_{n=i}$.

Proof. Define $\boldsymbol{H}_i = \boldsymbol{\Sigma}_i \hat{\boldsymbol{K}} = \boldsymbol{I} - \boldsymbol{C}_i \hat{\boldsymbol{K}}$. We have by Lemma S2, that

$$\|\boldsymbol{v}_* - \boldsymbol{v}_i\|_{\hat{\boldsymbol{K}}}^2 = \|\boldsymbol{H}_i \boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}}^2 = (\boldsymbol{H}_i \boldsymbol{v}_*)^\intercal \hat{\boldsymbol{K}} \boldsymbol{H}_i \boldsymbol{v}_* \overset{\text{(S44)}}{=} \boldsymbol{v}_*{}^\intercal \boldsymbol{H}_i \boldsymbol{v}_* = \bar{\boldsymbol{v}}_*^\intercal \boldsymbol{H}_i \bar{\boldsymbol{v}}_* \|\boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}}^2$$

This proves the first equality of Proposition 2. Further it holds that

$$\begin{aligned}
\|\boldsymbol{H}_i \boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}} = \|\hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{H}_i \boldsymbol{v}_*\|_2 = \|(\boldsymbol{I} - \hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}) \hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{v}_*\|_2 &\leq \|\boldsymbol{I} - \hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}\|_2 \|\boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}} \\
&= \lambda_{\max}(\boldsymbol{I} - \hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}) \|\boldsymbol{v}_*\|_{\hat{\boldsymbol{K}}}.
\end{aligned}$$

Now by Weyl's inequality and the fact that $\hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}$ is positive semi-definite, it holds that

$$\lambda_{\max}(\boldsymbol{H}_i) = \lambda_{\max}(\boldsymbol{I} - \hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}) \leq \lambda_{\max}(\boldsymbol{I}) - \lambda_{\min}(\hat{\boldsymbol{K}}^{\frac{1}{2}} \boldsymbol{C}_i \hat{\boldsymbol{K}}^{\frac{1}{2}}) \leq 1.$$

Now, recall that similar matrices $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$ have the same eigenvalues. Therefore

$$\boldsymbol{I} - \hat{\boldsymbol{K}}^{\frac{1}{2}}\boldsymbol{C}_i\hat{\boldsymbol{K}}^{\frac{1}{2}} = \hat{\boldsymbol{K}}^{\frac{1}{2}}(\boldsymbol{I} - \boldsymbol{C}_i\hat{\boldsymbol{K}})\hat{\boldsymbol{K}}^{-\frac{1}{2}}$$

and $\boldsymbol{I} - \boldsymbol{C}_i\hat{\boldsymbol{K}}$ have the same eigenvalues. Finally, since by eq. (S39) $\boldsymbol{H}_i$ is a projection onto $\operatorname{span}\{\boldsymbol{S}_i\}^{\perp_{\hat{\boldsymbol{K}}}}$, it has full rank at iteration n if the actions are linearly independent and therefore $\lambda_{\max}(\boldsymbol{H}_n) = 1$. This proves the claim. □

S2.3 Convergence Analysis of the Posterior Mean Approximation

Theorem 1 (Convergence in RKHS Norm of the Posterior Mean Approximation)
Let $\mathcal{H}_k$ be the RKHS associated with kernel $k(\cdot,\cdot)$, $\sigma^2 > 0$ and let $\mu_* - \mu \in \mathcal{H}_k$ be the unique solution to the regularized empirical risk minimization problem

$$\arg\min_{f\in\mathcal{H}_k} \tfrac{1}{n}\big(\textstyle\sum_{j=1}^n (f(\boldsymbol{x}_j) - y_j + \mu(\boldsymbol{x}_j))^2 + \sigma^2\|f\|^2_{\mathcal{H}_k}\big) \tag{11}$$

which is equivalent to the mathematical posterior mean up to shift by the prior μ [e.g. 1, Sec. 6.2]. Then for $i \in \{0,\dots,n\}$ the posterior mean $\mu_i(\cdot)$ computed by Algorithm 1 satisfies

$$\boxed{\|\mu_* - \mu_i\|_{\mathcal{H}_k} \le \rho(i)c(\sigma^2)\|\mu_* - \mu_0\|_{\mathcal{H}_k}} \tag{12}$$

where $\mu_0 = \mu$ is the prior mean and the constant $c(\sigma^2) = \sqrt{1 + \frac{\sigma^2}{\lambda_{\min}(\boldsymbol{K})}} \to 1$ as $\sigma^2 \to 0$.

Proof. Let $\rho(i)$ such that $\|\boldsymbol{v}_* - \boldsymbol{v}_i\|_{\hat{\boldsymbol{K}}} \le \rho(i)\|\boldsymbol{v}_* - \boldsymbol{v}_0\|_{\hat{\boldsymbol{K}}}$, where $\boldsymbol{v}_0 = \boldsymbol{0}$. Then, we have for $i \in \{0,\dots,n\}$, that

$$\begin{aligned}
\|\boldsymbol{v}_* - \boldsymbol{v}_i\|^2_{\boldsymbol{K}} \le \|\boldsymbol{v}_* - \boldsymbol{v}_i\|^2_{\hat{\boldsymbol{K}}} &\le \rho(i)^2\|\boldsymbol{v}_* - \boldsymbol{v}_0\|^2_{\hat{\boldsymbol{K}}} \\
&= \rho(i)^2\big(\|\boldsymbol{v}_* - \boldsymbol{v}_0\|^2_{\boldsymbol{K}} + \sigma^2\frac{1}{\lambda_{\min}(\boldsymbol{K})}\underbrace{\lambda_{\min}(\boldsymbol{K})\|\boldsymbol{v}_* - \boldsymbol{v}_0\|_2^2}_{\le\|\boldsymbol{v}_*-\boldsymbol{v}_0\|^2_{\boldsymbol{K}}}\big) \\
&\le \rho(i)^2\left(1 + \frac{\sigma^2}{\lambda_{\min}(\boldsymbol{K})}\right)\|\boldsymbol{v}_* - \boldsymbol{v}_0\|^2_{\boldsymbol{K}}
\end{aligned}$$

Now by assumption $\mu_i(\cdot) = \mu(\cdot) + \sum_{j=1}^n (\boldsymbol{v}_i)_j k(\cdot, \boldsymbol{x}_j) = \mu(\cdot) + k(\cdot, \boldsymbol{X})\boldsymbol{C}_i\boldsymbol{y}$. By the reproducing property we obtain for $\Delta = \boldsymbol{v}_* - \boldsymbol{v}_i$ that

$$\begin{aligned}
\|\boldsymbol{v}_* - \boldsymbol{v}_i\|^2_{\boldsymbol{K}} &= \Delta^\intercal\boldsymbol{K}\Delta \\
&= \sum_{\ell=1}^n\sum_{j=1}^n \Delta_\ell\Delta_j k(\boldsymbol{x}_\ell, \boldsymbol{x}_j) \\
&= \sum_{\ell=1}^n\sum_{j=1}^n \Delta_\ell\Delta_j \langle k(\cdot,\boldsymbol{x}_\ell), k(\cdot,\boldsymbol{x}_j)\rangle_{\mathcal{H}_k} && k \text{ is the reproducing kernel of } \mathcal{H}_k \\
&= \langle\sum_{\ell=1}^n \Delta_\ell k(\cdot,\boldsymbol{x}_\ell), \sum_{j=1}^n \Delta_j k(\cdot,\boldsymbol{x}_j)\rangle_{\mathcal{H}_k} \\
&= \left\|\sum_{\ell=1}^n \Delta_\ell k(\cdot,\boldsymbol{x}_\ell)\right\|^2_{\mathcal{H}_k} \\
&= \left\|\sum_{\ell=1}^n (\boldsymbol{v}_*)_\ell k(\cdot,\boldsymbol{x}_\ell) - \sum_{\ell=1}^n (\boldsymbol{v}_i)_\ell k(\cdot,\boldsymbol{x}_\ell)\right\|^2_{\mathcal{H}_k} \\
&= \|\mu_* - \mu_i\|^2_{\mathcal{H}_k} && \text{See Theorem 3.4 in Kanagawa et al. [36]}
\end{aligned}$$

Combining the above and setting $c(\sigma^2) = 1 + \frac{\sigma^2}{\lambda_{\min}(\boldsymbol{K})}$ we obtain

$$\|\mu_* - \mu_i\|_{\mathcal{H}_k} = \|\boldsymbol{v}_* - \boldsymbol{v}_i\|_{\boldsymbol{K}} \le \rho(i)c(\sigma^2)\|\boldsymbol{v}_* - \boldsymbol{v}_0\|_{\boldsymbol{K}} = \rho(i)c(\sigma^2)\|\mu_* - \mu_0\|_{\mathcal{H}_k}.$$

□

S2.4 Combined Uncertainty as Worst Case Error

Theorem 2 (Combined and Computational Uncertainty as Worst Case Errors)
Let $\sigma^2 \geq 0$ and let $k_i(\cdot,\cdot) = k_*(\cdot,\cdot) + k_i^{\text{comp}}(\cdot,\cdot)$ be the combined uncertainty computed by Algorithm 1. Then, for any $\boldsymbol{x} \in \mathcal{X}$ (assuming $\boldsymbol{x} \notin \boldsymbol{X}$ if $\sigma^2 > 0$) we have

$$\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} \overbrace{\underbrace{g(\boldsymbol{x}) - \mu_*^g(\boldsymbol{x})}_{\text{error of math. post. mean}} + \underbrace{\mu_*^g(\boldsymbol{x}) - \mu_i^g(\boldsymbol{x})}_{\text{computational error}}}^{\text{error of approximate posterior mean}} = \sqrt{k_i(\boldsymbol{x},\boldsymbol{x}) + \sigma^2}, \quad \textit{and} \tag{13}$$

$$\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} \underbrace{\mu_*^g(\boldsymbol{x}) - \mu_i^g(\boldsymbol{x})}_{\text{computational error}} = \sqrt{k_i^{\text{comp}}(\boldsymbol{x},\boldsymbol{x})} \tag{14}$$

where $\mu_*^g(\cdot) = k(\cdot,\boldsymbol{X})\hat{\boldsymbol{K}}^{-1}g(\boldsymbol{X})$ is the mathematical and $\mu_i^g(\cdot) = k(\cdot,\boldsymbol{X})\boldsymbol{C}_i g(\boldsymbol{X})$ IterGP's posterior mean for the latent function $g \in \mathcal{H}_{k^\sigma}$. If $\sigma^2 = 0$, then the above also holds for $\boldsymbol{x} \in \boldsymbol{X}$.

Proof. Let $\boldsymbol{x}_0 = \boldsymbol{x}$, $c_0 = 1$ and $c_j = -(\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{x}))_j$ for $j = 1,\dots n$, where $k^\sigma(\cdot,\cdot) := k(\cdot,\cdot) + \sigma^2\delta(\cdot,\cdot)$. Then by Lemma 3.9 of Kanagawa et al. [36], it holds that

$$\begin{aligned}
\left(\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} (g(\boldsymbol{x}) - \mu_i^g(\boldsymbol{x}))\right)^2 &= \left(\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} \sum_{j=0}^n c_j g(\boldsymbol{x}_j)\right)^2 \\
&= \left\| k^\sigma(\cdot,\boldsymbol{x}_0) - \sum_{j=1}^n k(\boldsymbol{x},\boldsymbol{x}_j)\boldsymbol{C}_i k^\sigma(\cdot,\boldsymbol{x}_j) \right\|_{\mathcal{H}_{k^\sigma}}^2 \\
&= \|k^\sigma(\cdot,\boldsymbol{x}) - k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\cdot)\|_{\mathcal{H}_{k^\sigma}}^2 \\
&= \langle k^\sigma(\cdot,\boldsymbol{x}), k^\sigma(\cdot,\boldsymbol{x})\rangle_{\mathcal{H}_{k^\sigma}} - 2\langle k^\sigma(\cdot,\boldsymbol{x}), k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\cdot)\rangle_{\mathcal{H}_{k^\sigma}} \\
&\quad + \langle k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\cdot), k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\cdot)\rangle_{\mathcal{H}_{k^\sigma}}
\end{aligned}$$

Now by the reproducing property, it follows that

$$= k^\sigma(\boldsymbol{x},\boldsymbol{x}) - 2k^\sigma(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{x}) + k^\sigma(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{X})\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{x})$$

If $\sigma^2 > 0$ and $\boldsymbol{x} \neq \boldsymbol{x}_j$ or if $\sigma^2 = 0$, it holds that $k^\sigma(\boldsymbol{x},\boldsymbol{X}) = k(\boldsymbol{x},\boldsymbol{X})$. Further by definition $k^\sigma(\boldsymbol{X},\boldsymbol{X}) = \hat{\boldsymbol{K}}$ and finally by (S42), it holds that $\boldsymbol{C}_i\hat{\boldsymbol{K}}\boldsymbol{C}_i = \boldsymbol{C}_i$. Therefore we have

$$\begin{aligned}
&= k(\boldsymbol{x},\boldsymbol{x}) + \sigma^2 - 2k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k(\boldsymbol{X},\boldsymbol{x}) + k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i\hat{\boldsymbol{K}}\boldsymbol{C}_i k(\boldsymbol{X},\boldsymbol{x}) \\
&= k(\boldsymbol{x},\boldsymbol{x}) - k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i k(\boldsymbol{X},\boldsymbol{x}) + \sigma^2 \\
&= k_i(\boldsymbol{x},\boldsymbol{x}) + \sigma^2
\end{aligned}$$

We prove eq. (14) by an analogous argument. Choose $c_j := ((\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_i)k^\sigma(\boldsymbol{X},\boldsymbol{x}))_j$. We have

$$\begin{aligned}
&\left(\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} (\mu_*^g(\boldsymbol{x}) - \mu_i^g(\boldsymbol{x}))\right)^2 = \left(\sup_{g \in \mathcal{H}_{k^\sigma} : \|g\|_{\mathcal{H}_{k^\sigma}} \leq 1} \sum_{j=0}^n c_j g(\boldsymbol{x}_j)\right)^2 \\
&= \left\| \sum_{j=1}^n k(\boldsymbol{x},\boldsymbol{x}_j)(\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_i)k^\sigma(\cdot,\boldsymbol{x}_j) \right\|_{\mathcal{H}_{k^\sigma}}^2 \\
&= \left\| k(\boldsymbol{x},\boldsymbol{X})(\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_i)k^\sigma(\boldsymbol{X},\cdot) \right\|_{\mathcal{H}_{k^\sigma}}^2 \\
&= k^\sigma(\boldsymbol{x},\boldsymbol{X})\hat{\boldsymbol{K}}^{-1}\hat{\boldsymbol{K}}\hat{\boldsymbol{K}}^{-1}k^\sigma(\boldsymbol{X},\boldsymbol{x}) - 2k^\sigma(\boldsymbol{x},\boldsymbol{X})\hat{\boldsymbol{K}}^{-1}\hat{\boldsymbol{K}}\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{x}) + k^\sigma(\boldsymbol{x},\boldsymbol{X})\boldsymbol{C}_i\hat{\boldsymbol{K}}\boldsymbol{C}_i k^\sigma(\boldsymbol{X},\boldsymbol{x})
\end{aligned}$$

Again, we use that $k^\sigma(\boldsymbol{x},\boldsymbol{X}) = k(\boldsymbol{x},\boldsymbol{X})$ by assumption and (S42). Therefore

$$\begin{aligned}
&= k(\boldsymbol{x},\boldsymbol{X})(\hat{\boldsymbol{K}}^{-1} - \boldsymbol{C}_i)k(\boldsymbol{X},\boldsymbol{x}) \\
&= k_i^{\text{comp}}(\boldsymbol{x},\boldsymbol{x})
\end{aligned}$$

This concludes the proof. □

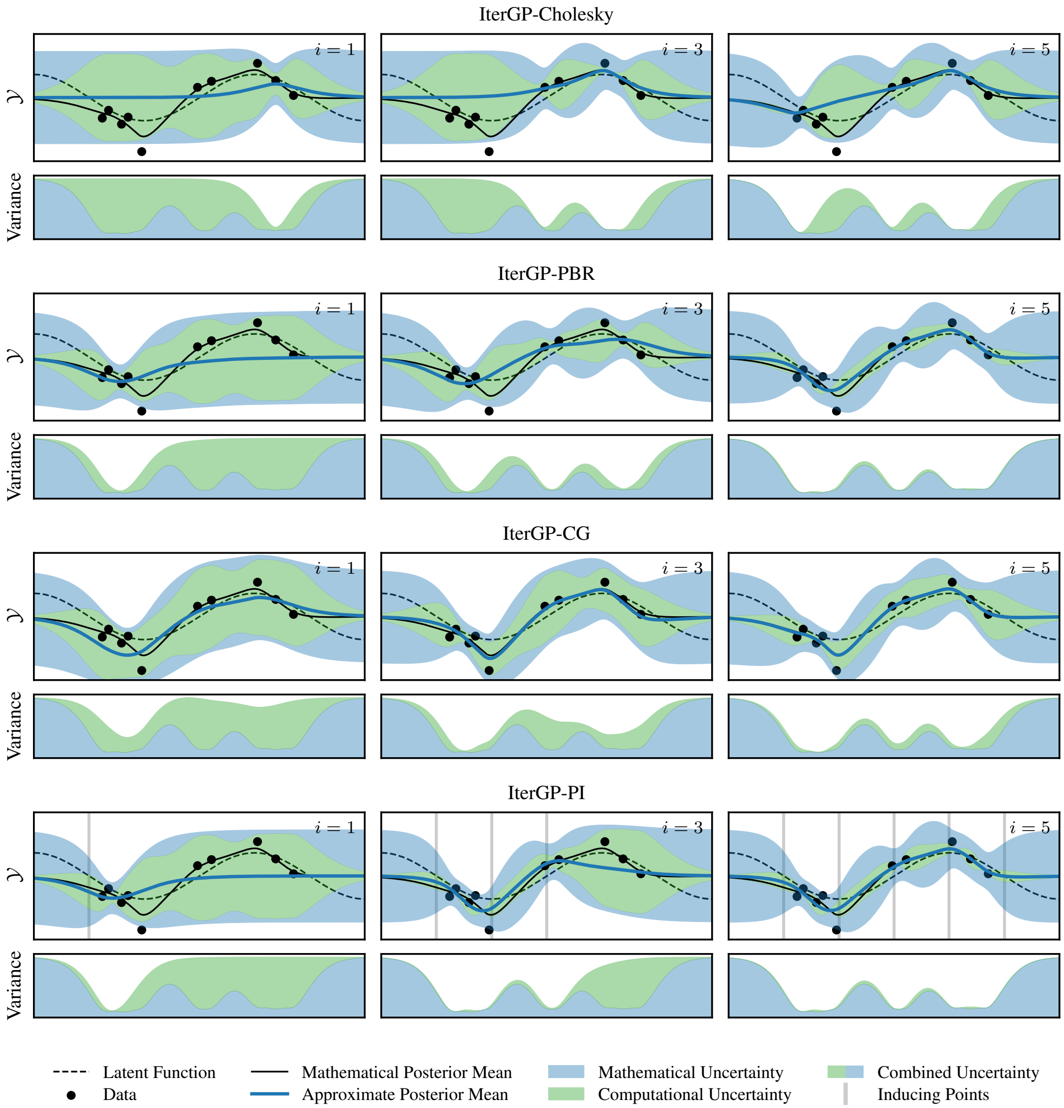


Figure S3: *Illustration of IterGP analogs of commonly used GP approximations.*

S3 Implementation of Algorithm 1

S3.1 Policy Choice

As illustrated in Figure 2, the choice of policy of Algorithm 1 determines where computation in input space is targeted and therefore where the combined posterior contracts first. However, the policy also determines whether the error in the posterior mean or (co-)variance are predominantly reduced first, as Figure S3 shows (cf. IterGP-Chol and IterGP-PBR). Therefore the policy choice is application-dependent. If I am primarily interested in the predictive mean, I may select residual actions (IterGP-CG). If downstream I am making use of the predictive uncertainty, I may want to contract uncertainty globally as quickly as possible at the expense of predictive accuracy (IterGP-PI). Such a choice is not unique to IterGP, but necessary whenever we select a GP approximation. What IterGP adds is computation-aware, meaningful uncertainty quantification in the sense of Corollary 1 no matter the choice of policy.

S3.2 Stopping Criterion

In our implementation of Algorithm 1 we use the following two stopping criteria. Our computational budget can be directly controlled by specifying a *maximum number of iterations*, since each iteration of IterGP needs the same number of matrix-vector multiplies. Alternatively, we terminate if the *absolute or relative norm of the residual* are sufficiently small, i.e. if

$$\|\boldsymbol{r}_i\|_2 < \delta_{\text{abstol}} \qquad \text{or} \qquad \|\boldsymbol{r}_i\|_2 < \delta_{\text{reltol}} \|\boldsymbol{y}\|_2. \tag{S53}$$

Of course other choices are possible. From a probabilistic numerics standpoint one may want to terminate once the combined marginal uncertainty at the training data is sufficiently small relative to the observation noise.

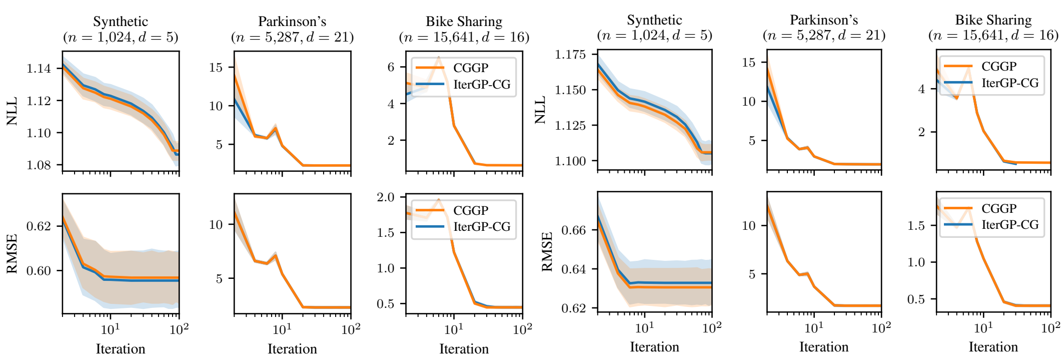
S3.3 Efficient Sampling from the Combined Posterior

Sampling from an exact GP posterior has cubic cost $\mathcal{O}(n_\diamond^3)$ in the number of evaluation points $n_\diamond$, which is prohibitive for many useful downstream applications such as numerical integration over the posterior using Monte-Carlo methods. Wilson et al. [46, 47] recently showed how to make use of *Matheron's rule* [45, 66, 67] to efficiently sample from a GP posterior by sampling from the prior and then performing a pathwise update. We can directly make use of this strategy since Algorithm 1 computes a low-rank approximation to the precision matrix. Assume we are given a draw $f'_{\text{prior}} \in \mathcal{H}_k^\theta$ from the prior[3] such that $\boldsymbol{y}' \sim \mathcal{N}(f'_{\text{prior}}(\boldsymbol{X}), \sigma^2 \boldsymbol{I})$ constitutes a draw from the prior predictive. Then

$$f'(\cdot) = f'_{\text{prior}}(\cdot) + k(\cdot, \boldsymbol{X})\boldsymbol{C}_i(\boldsymbol{y} - \boldsymbol{y}') \tag{S54}$$

is a draw from the combined posterior by Matheron's rule, which we can evaluate in $\mathcal{O}(n_\diamond n i)$ for $n_\diamond$ evaluation points, since $\boldsymbol{C}_i$ has rank i.

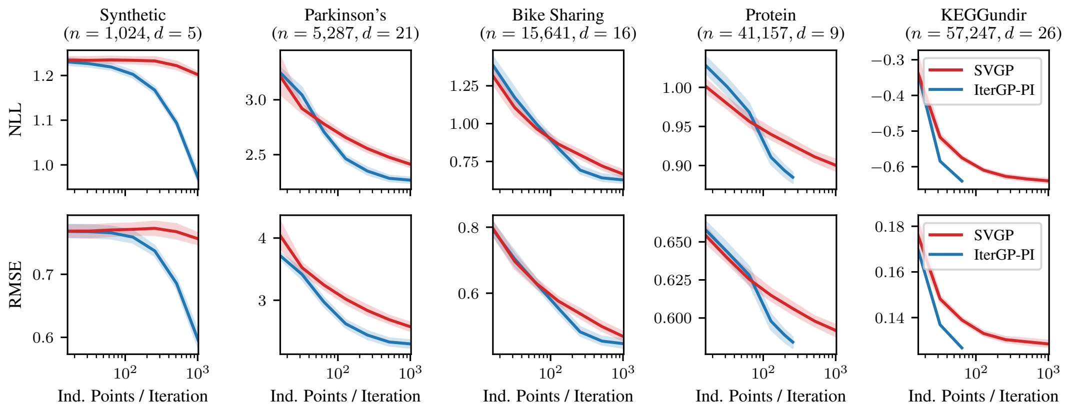
S4 Additional Experimental Results



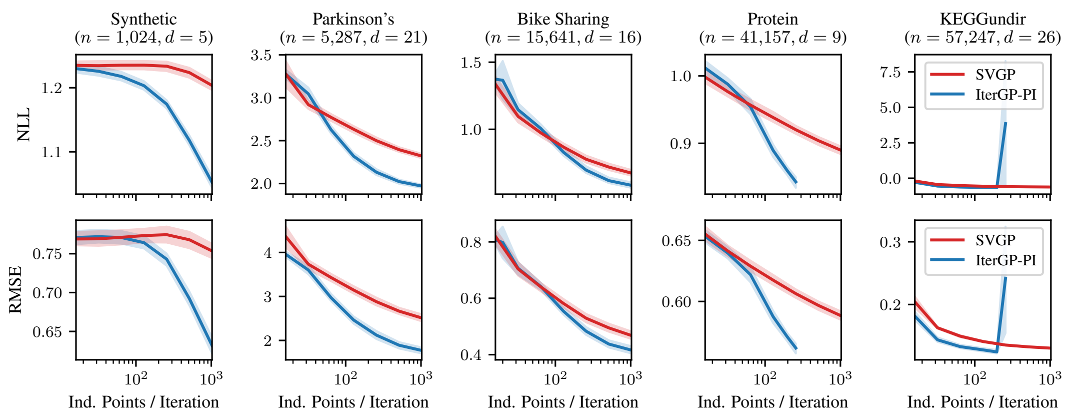
(a) RBF kernel (b) Matérn($\frac{3}{2}$) kernel

Figure S4: *Generalization of CGGP and its closest IterGP analog.* GP regression using an RBF and Matérn($\frac{3}{2}$) kernel on UCI datasets. The plot shows the average generalization error in terms of NLL and RMSE for an increasing number of solver iterations. The posterior mean of IterGP-CG and CGGP is identical, which explains the identical RMSE.

[3] In infinite dimensional reproducing kernel Hilbert spaces samples $f \sim \mathcal{GP}(\mu, k)$ from a Gaussian process almost surely do not lie in the RKHS $\mathcal{H}_k$ [Cor. 4.10, 36]. However, there exists $f' \in \mathcal{H}_k^\theta$ in a larger RKHS $\mathcal{H}_k^\theta \supset \mathcal{H}_k$ such that $f'(\boldsymbol{x}) = f(\boldsymbol{x})$ with probability 1 [Thm. 4.12, 36].



(a) RBF kernel



(b) Matérn($\frac{3}{2}$) kernel

Figure S5: *Generalization of SVGP and its closest IterGP analog.* GP regression using an RBF and Matérn($\frac{3}{2}$) kernel on UCI datasets. The plot shows the average generalization error in terms of NLL and RMSE for an increasing number of identical inducing points. After a small number of inducing points relative to the size of the training data, IterGP has significantly lower generalization error than SVGP. For the "KEGGundir" dataset after ≈ 128 iterations we observe numerical instability in some runs when computing the combined posterior of IterGP using a Matérn($\frac{3}{2}$) kernel.