Pick-to-Learn and Self-Certified Gaussian Process Approximations

Daniel Marks
Imperial College London

Dario Paccagnan
Imperial College London

Abstract

Generalisation bounds are crucial for providing data-driven models with performance and safety guarantees. In this respect, bounds that do not require a held-out test set are particularly valuable as they allow the use of all data for training. While many such bounds do not improve upon the train-test approach, which remains the gold standard, the P2L algorithm (Paccagnan et al., 2023) has shown great potential. However, P2L comes with limitations, including computational overhead, reliance on consistent data, and restriction to non-Bayesian settings. In this work, we overcome these challenges in general settings and employ the corresponding results to show that classical Gaussian process (GP) training procedures can be interpreted as instantiations of P2L, thus inheriting tight, self-certified bounds. Three contributions underpin these conclusions. First, we introduce early stopping in P2L, equipping it with a tight generalisation bound to reduce training costs and address the non-consistent case. Second, we adapt P2L to the Bayesian setting and demonstrate its equivalence to posterior updating in a hierarchical model. Third, we show that greedy subset-of-data GPs are special P2L instantiations. Numerical evidence shows that the resulting P2L bounds we obtain compare favourably with the train-test and PAC-Bayes approaches on various real-world datasets.

1 INTRODUCTION

A crucial requirement for any machine learning pipeline is the ability to assess the performance of a learned model on previously unseen data. This step is especially significant for the deployment of machine learning algorithms to safety-critical settings such as autonomous vehicles (Bojarski et al., 2016), healthcare (Kononenko, 2001), and climate modelling (Rolnick et al., 2023). Different methods for bounding the generalisation error can be broadly divided into two categories: those that require withholding a portion of the data to obtain a bound, and those that do not, each presenting unique advantages and drawbacks.

The most celebrated method in the first category is the train-test approach, which often produces sharp bounds whilst operating with minimal assumptions. While these features make it applicable to a wide range of learning algorithms (Langford, 2005), a crucial disadvantage lies in that a portion of the available data is not used for training, thus often worsening the post-training performance. The second category encompasses numerous frameworks that allow all data to be used for training and instead rely on additional assumptions, such as those related to the structure of the learning problem, the training routine, or the model itself. Notions that characterise the hypothesis space, such as the VC-dimension (Vapnik, 2000) or Rademacher complexity (Bartlett and Mendelson, 2001), have been instrumental in explaining generalisation in classical settings but often result in poor or vacuous bounds when applied to overparameterised models such as deep neural networks (Zhang et al., 2017). In contrast, PAC-Bayesian approaches (Catoni, 2007; Guedj, 2019; Alquier, 2023), which rely on a distribution over predictors, have demonstrated the ability to produce non-vacuous and informative bounds even in the deep learning regime (Dziugaite and Roy, 2017; Perez-Ortiz et al., 2021; Clerico et al., 2022). None of these frameworks, however, has been shown to lead to stronger generalisation bounds than the train-test approach beyond specific settings (Foong et al., 2021).

Recently, a breakthrough in compression theory (Campi and Garatti, 2023) allowed for the development of the Pick-to-Learn (P2L) meta-algorithm that transforms *any* black-box learning algorithm into a *compression-inducing learner* (Paccagnan et al.,

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

2023). Models trained with P2L benefit from a novel generalisation error bound, derived purely from the training procedure, which often yields stronger guarantees than those obtained using the train-test approach. In order to instil the necessary compression properties in any inner learner, P2L iteratively trains on subsets of the available data, progressively incorporating the most poorly predicted samples until all remaining points are deemed sufficiently appropriate based on a predefined criterion. The degree of dataset compression achieved upon termination directly determines the quality of the bound on the generalisation error of the learned model.

Unfortunately, there are three major obstacles preventing the meta-algorithm from being more widely adopted. Firstly, the need for all remaining points to be appropriate enough for termination confines P2L to the consistent case. However, for datasets with high levels of noise or mislabelled examples, this strict requirement can lead to the compressed set being unnecessarily inflated, despite a sufficiently good hypothesis having already been learned, thereby worsening the resulting bound. Secondly, as the level of compression that will be achieved is not known a priori, the amount of computational resources that will be required cannot be bounded beyond the worst case. Crucially, since the quality of the final bound is directly linked to this unpredictable level of compression, one cannot judge whether the potentially high computational cost would even be worth incurring in the first place. Finally, P2L lacks a clear mechanism to account for the stochastic nature of certain models, e.g., Gaussian processes.

In the first part of this paper, we address these limitations in a general setting that is agnostic to the choice of the inner learner. First, we introduce early stopping into P2L, allowing the maximum cardinality of the compressed set to be specified in advance. This extension not only enables the use of P2L on non-consistent data but also mitigates the previously discussed issues of resource allocation and computational overhead. After demonstrating that the original P2L bound of Paccagnan et al. (2023) is violated in this setting, we prove a generalised bound that reduces to the original in the consistent case. We do so by considering a modified compression function that includes a penalty term for any remaining inappropriate points upon termination. Second, we develop a Bayesian formulation of P2L based on posterior updating within a hierarchical model. By formalising the notion of risk for conditional distributions, instead of hypotheses, we obtain an equivalent bound for posterior predictive distributions. We then turn our attention to the specific choice of learner within the meta-algorithm. By employing our results for Gaussian processes, we demonstrate that greedy subset-of-data GP approximations are special instantiations of Bayesian P2L with early stopping, and thus inherit our tight generalisation guarantees. Finally, we show through experiments that the resulting bounds and post-training performance obtained compare favorably against those of the train-test, and PAC-Bayes approaches.

2 BACKGROUND

We begin by introducing essential terminology and results from compression theory before providing an overview of the P2L framework and its theoretical guarantees.

2.1 Compression Theory

We model a dataset $D = \{z_1, \dots, z_N\}$ as a multiset[1] of N training examples, where D is a realisation of $\boldsymbol{D} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_N\}$ and $\boldsymbol{z}_1, \dots, \boldsymbol{z}_N$ are i.i.d. random variables[2] defined over an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and taking values in some measurable space $(\mathcal{Z}, \mathcal{E})$. In this context, a compression function is a permutation invariant map from a multiset to a sub-multiset of examples (Littlestone and Warmuth, 1986; Floyd and Warmuth, 1995; Graepel et al., 2005).

Following Campi and Garatti (2023), we say that a compression function κ satisfies the preference property if, for any $\mathcal{Z}$-valued multisets U and $V \subseteq U$

$$V \neq \kappa(U) \Longrightarrow V \neq \kappa(U \cup \{z\}), \forall z \in \mathcal{Z}.$$

Under the preference assumption, Campi and Garatti (2023) obtained tight upper bounds on a key quantity known as the probability of change of compression

$$\phi_\kappa(\boldsymbol{D}) = \mathbb{P}\left[\kappa(\kappa(\boldsymbol{D}) \cup \{\boldsymbol{z}\}) \neq \kappa(\boldsymbol{D}) \mid \boldsymbol{D}\right],$$

where $\boldsymbol{z}$ is i.i.d. as each element of $\boldsymbol{D}$. Specifically, they showed that for any $\delta \in (0, 1)$ the probability of change of compression is upper-bounded as

$$\mathbb{P}^N\left[\phi_\kappa(\boldsymbol{D}) \leq \bar{\varepsilon}(|\kappa(\boldsymbol{D})|, \delta)\right] \geq 1 - \delta, \tag{1}$$

where $\bar{\varepsilon}(N, \delta) = 1$, while $\bar{\varepsilon}(n, \delta)$ is the unique solution to $\Psi_{n,\delta}(\varepsilon) = 1$ in the interval $[n/N, 1]$ for $n \in 0, \dots, N-1$, with

$$\Psi_{n,\delta}(\varepsilon) = \frac{\delta}{2N} \sum_{m=n}^{N-1} \frac{\binom{m}{n}}{\binom{N}{n}} (1-\varepsilon)^{-(N-m)} + \frac{\delta}{6N} \sum_{m=N+1}^{4N} \frac{\binom{m}{n}}{\binom{N}{n}} (1-\varepsilon)^{m-N}. \tag{2}$$

[1] Multisets are extensions of sets that account for the presence of possibly repeated elements.

[2] Throughout, boldface denotes random quantities.

This result is crucial in the proof of the P2L bound and will also form the basis for our extension of the meta-algorithm to the non-consistent setting.

2.2 The Pick-to-Learn Framework

We begin by presenting the Pick-to-Learn meta-algorithm and its generalisation bound introduced in Paccagnan et al. (2023), before turning our attention to its computational cost. The P2L meta-algorithm (Algorithm 1) takes any learning algorithm and turns it into a compression-inducing learner. Specifically, we consider the supervised learning setting $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with the goal of learning a mapping $h : \mathcal{X} \to \mathcal{Y}$, from a given space of predictors $\mathcal{H}$. Given a multiset $D = \{(x_n, y_n)\}_{n=1}^N$, the algorithm is initialised with a configuration triple consisting of a learner[3] $L : \cup_{n=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$, an initial hypothesis $h_0 : \mathcal{X} \to \mathcal{Y}$ and a hypothesis-dependent total order $\leq_h$ on the augmented multiset $D_S = D \cup \{\texttt{Stop}\}$. `Stop` is an auxiliary symbolic element used as an appropriateness threshold. P2L then iteratively trains on a multiset of points T, where T additionally includes the least appropriate point according to the last learned hypothesis. When all points in $D \setminus T$ are less than `Stop` according to $\leq_h$, the algorithm stops. Upon convergence, P2L returns the last learned hypothesis together with the sub-multiset that produced it.

Observe that, over the sampling of possible datasets, the output tuple of P2L is itself stochastic and we can thus write $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L}(\boldsymbol{D})$, where the configuration triple $(L, h_0, \leq_h)$ is implicit in our notation. The statistical risk $R : \mathcal{H} \to \mathbb{R}_{\geq 0}$ of a learned hypothesis is then defined as

$$R(\boldsymbol{h}) = \mathbb{P}\{\texttt{Stop} \leq_{\boldsymbol{h}} \boldsymbol{z} \mid \boldsymbol{D}\},$$

where $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ is i.i.d. as is each element of $\boldsymbol{D}$.[4] In their original work Paccagnan et al. (2023) make use of two crucial arguments. First, they show that the compression function κ_{P2L} mapping D into the multiset T returned by P2L is preferent, thus licensing the use of (1) to bound $\phi_{\kappa_{P2L}}(\boldsymbol{D})$. Second, they show that the statistical risk is upper bounded by the probability of change of compression, i.e., that $R(\boldsymbol{h}) \leq \phi_{\kappa_{P2L}}(\boldsymbol{D})$ almost surely. As a result, they obtain the following bound on the risk of any hypothesis learned via P2L

$$\mathbb{P}^N \left[R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\boldsymbol{T}|, \delta)\right] \geq 1 - \delta. \tag{3}$$

[3]To ease the presentation, we consider learners that do not account for the order in which elements appear in the multiset. However, all ensuing results continue to hold unchanged in this more general setting.

[4]Note that the statistical risk can be equivalently expressed as the expected value of the zero-one loss, namely $R(\boldsymbol{h}) = \mathbb{E}[\mathbb{I}\{\texttt{Stop} \leq_{\boldsymbol{h}} \boldsymbol{z}\} \mid \boldsymbol{D}]$.

Algorithm 1 $\mathcal{A}_{\text{P2L}}(D)$: The P2L Algorithm

Initialise: $T = \emptyset$, $h = h_0$, $\bar{z} = \max_{h_0}(D_S)$
while $\bar{z} \neq \texttt{Stop}$ **do**
  $T \leftarrow T \cup \{\bar{z}\}$ ▷ Augment T
  $h \leftarrow L(T)$ ▷ Learn hypothesis
  $\bar{z} \leftarrow \max_h(D_S \setminus T)$ ▷ Compute max
end while
return h, T ▷ Hypothesis and multiset

Moving to the computational cost of P2L, the total runtime of the meta-algorithm is given by

$$\mathcal{O}\left(\sum_{m=1}^{|T|} \ell(m) + \alpha(N - m)\right), \tag{4}$$

where we denote with $\ell(m)$ and $\alpha(N - m)$ the cost of the training and determination of appropriateness, respectively, in an arbitrary iteration $m \leq N$. Clearly, the true complexity crucially depends on how $|T|$ scales with N. However, since there is no general way of bounding $|T|$ a priori, we can only perform a worst-case analysis by assuming that $|T| \sim \mathcal{O}(N)$. To illustrate what this scaling entails, if the inner learner uses dot-product self-attention (Vaswani et al., 2017) whose training and inference cost scales quadratically in N, the whole routine will require $\mathcal{O}(N^3)$ steps. This is prohibitive for many real-world applications beyond the small data regime, and motivates the introduction of early stopping in the following section.

3 EARLY STOPPING P2L

As it stands, Pick-to-Learn follows an all-or-nothing philosophy: no generalisation bound is provided unless the algorithm runs to completion, leaving practitioners with no control over the associated runtime or computational cost. In this section, we introduce early stopping, which allows us to reduce the computational complexity of the meta-algorithm. Additionally, we extend the P2L framework to the non-consistent setting arising from early termination and prove a general bound that subsumes the one obtained in the consistent case.

3.1 Introducing Early Stopping

In this section, we introduce early stopping into P2L. The fundamental idea is to allow the end user to specify, a priori, the maximum number of iterations $M \in \mathbb{N}$ that P2L is allowed to perform. While it is clear that such an additional degree of freedom provides control over the computational cost and runtime, the *key challenge* lies in showing that the hypothesis produced by

modifying P2L in this way can still be equipped with a very strong generalisation bound.

The proposed training routine is detailed in Algorithm 2, which we refer to as *P2L with Early Stopping* or *P2L-ES* for short. This modified algorithm requires the initial configuration to additionally contain the maximum number $M \in \mathbb{N}$ of iterations to be performed. Further, it is easy to see that the original P2L algorithm corresponds to a particular initialisation of P2L-ES, where $M = |D|$ (more precisely, where $M \geq |D|$).

Although P2L with Early Stopping can be viewed as an extension of the standard P2L framework, the original risk bound in (3) no longer holds in this setting, as demonstrated by the following proposition.

Proposition 1. *There exists a probability measure $\mathbb{P}$ and initial configuration $(L, h_0, \leq_h)$ such that the corresponding P2L-ES execution with a maximum of M iterations produces a hypothesis $\boldsymbol{h}$ and a compressed multiset $\boldsymbol{T}$, with $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L-ES}(\mathbf{D})$, that violates (3) for any choice of $M < N = |D|$ and $\delta < 1$.*

We provide a constructive proof in Appendix A.1.

3.2 Penalty-Based Compression

The key observation is that, if we want the number of iterations to be specified a priori, we need to additionally account for all points in the training set which are not appropriate enough at termination (Campi and Garatti, 2023). With this in mind, we introduce the following modified compression function

$$\tilde{\kappa}_{P2L-ES}(D) = \kappa_{P2L-ES}(D) \cup \{z \in D \setminus \kappa_{P2L-ES}(D) : \texttt{Stop} \leq_h z\}, \quad (5)$$

which we use to state our first main result.

Theorem 1. *Let $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L-ES}(\boldsymbol{D})$ be the output of P2L-ES. Then for any $\delta \in (0, 1)$ it holds that*

$$\mathbb{P}^N\{R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\boldsymbol{T}| + |\{z \in \boldsymbol{D} \setminus \boldsymbol{T} : \texttt{Stop} \leq_{\boldsymbol{h}} z\}|, \delta)\} \geq 1 - \delta.$$

The proof, which can be found in Appendix A.2, leverages the fact that the penalty-based compression function (5) is preferent and that, unlike κ_{P2L-ES}, the risk of any learned hypothesis is dominated by the probability of change of compression induced by $\tilde{\kappa}_{P2L-ES}$.

With early termination, P2L has the potential to avoid adding unnecessary points to the compressed set, paving the way for improved bounds in the non-consistent setting. This simultaneously gives practitioners control over the meta-algorithm's complexity, replacing the unknown upper limit of summation in (4) with the tunable parameter M. Returning to the dot-product self-attention example, the cost of P2L-ES with this choice of learner is $\mathcal{O}(M^3 + N^2M)$ and by taking $M << N$ one can make the dominant factor reduce to $O(N^2M)$.

Algorithm 2 $\mathcal{A}_{\text{P2L-ES}}(D)$: The P2L-ES Algorithm

Initialise: M, $T_0 = \emptyset$, h_0, $\bar{z}_0 = \max_{h_0}(D_S)$, $m = 1$
while $|T_{m-1}| < M$ **do**
  if $\bar{z}_{m-1} = \texttt{Stop}$ **then**
    return h_{m-1}, T_{m-1} ▷ Exit
  end if
  $T_m \leftarrow T_{m-1} \cup \{\bar{z}_{m-1}\}$ ▷ Augment T
  $h_m \leftarrow L(T_m)$ ▷ Learn hypothesis
  $\bar{z}_m \leftarrow \max_{h_m}(D_S \setminus T_m)$ ▷ Compute max
  $m \leftarrow m + 1$ ▷ Increment m
end while
return h_M, T_M ▷ Hypothesis and multiset

4 BAYESIAN P2L

The general setting of the Pick-to-Learn meta-algorithm allows for great flexibility in the choice of inner black-box learner. However, the initial focus on non-probabilistic models and especially, the frequentist flavour of the guarantees obtained, raises the question of the applicability of P2L to Bayesian models. In what follows, we develop a Bayesian formulation of the Pick-to-Learn framework, which will allow us to nest Gaussian processes inside the meta-algorithm.

4.1 P2L As Bayesian Updating

The iterative training procedure inherent in all Pick-to-Learn variants closely resembles Bayesian updating. To make this connection explicit, we assume a conditional parametric model $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\theta})$, where $\boldsymbol{w}$ and $\boldsymbol{\theta}$ denote the model parameters and hyperparameters, respectively. To ensure an equally expressive formulation, and given that the black-box learning algorithm L can have either preset or learned hyperparameters, we employ a hierarchical Bayesian setup (Gelman et al., 2013), placing a prior distribution $p(\boldsymbol{\theta})$ on the hyperparameters and a conditional prior $p(\boldsymbol{w} \mid \boldsymbol{\theta})$ on the model parameters with the goal of inferring the joint posterior

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid D) \propto p(D \mid \boldsymbol{w}, \boldsymbol{\theta})\, p(\boldsymbol{w} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}).$$

In this light, each step of the meta-algorithm can be seen as having a Bayesian analogue. To begin with, the prior distributions $p(\boldsymbol{\theta})$ and $p(\boldsymbol{w} \mid \boldsymbol{\theta})$ take the role of the initial hypothesis h_0. Moreover, learning a hypothesis at every iteration simply amounts to

Algorithm 3 $\mathcal{A}_{BP2L-ES}$: Bayesian P2L-ES

Initialise: M, $T_0 = \emptyset$, $p(\boldsymbol{\theta})$, $p(\boldsymbol{w} \mid \boldsymbol{\theta})$, $\bar{z}_0 = \max_{p_{T_0}}(D_S)$, $m = 1$
while $|T_{m-1}| < M$ **do**
  if $\bar{z}_{m-1} = \texttt{Stop}$ **then**
    return $p_{T_{m-1}}, T_{m-1}$ ▷ Exit
  end if
  $T_m \leftarrow T_{m-1} \cup \{\bar{z}_{m-1}\}$ ▷ Augment T
  Update posterior via (7)
  Compute predictive posterior via (8)
  $\bar{z}_m \leftarrow \max_{p_{T_m}}(D_S \setminus T_m)$ ▷ Compute max
  $m \leftarrow m + 1$ ▷ Increment m
end while
return p_{T_M}, T_M ▷ Predictive and multiset

updating the joint posterior distribution conditioned on the current compressed multiset. If we denote by $T_m = (\boldsymbol{x}_{T_m}, \boldsymbol{y}_{T_m})$ the multiset constructed by P2L at iteration m, this update can be formalised as

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m})}. \quad (6)$$

Straightforward algebra reveals (6) to be equivalent to a Bayesian update that considers the likelihood of the latest additions to the compressed set, and uses the previous joint posterior as the prior:

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{x}_{T_m \setminus T_{m-1}}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_{m-1}})}{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_m \setminus T_{m-1}})}. \quad (7)$$

A complete derivation can be found in Appendix B. To construct a total order on the augmented training set and select points for the compressed set, it suffices to recognise that the vehicle for prediction is no longer a learned hypothesis h, but rather the predictive posterior distribution:

$$p(\boldsymbol{y}_{D \setminus T_m} \mid \boldsymbol{x}_{D \setminus T_m}, \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \int p(\boldsymbol{y}_{D \setminus T_m} \mid \boldsymbol{x}_{D \setminus T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) d\boldsymbol{w} d\boldsymbol{\theta}. \quad (8)$$

Using this predictive posterior, we construct a distribution-dependent total order on $D_S \setminus T_m$ in order to find the least appropriate remaining point. We will denote the total order constructed using the distribution $p(\cdot \mid \cdot, \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m})$ as $\leq_{p_{T_m}}$. With the above components, we can fully formulate Pick-to-Learn from a Bayesian perspective (Algorithm 3).

A practical approximation to the full hierarchical model can be introduced by disregarding the uncertainty in the hyperparameters and replacing their posterior with a delta distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) \approx \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, where $\boldsymbol{\theta}^*$ can be obtained via type-II MAP or MLE (Kevin P. Murphy, 2022). Since the prior is part of the initial configuration, it remains unchanged for each dataset draw. As such, instead of learning $\boldsymbol{\theta}^*$ in a fully empirical Bayesian manner (Carlin and Louis, 1997), we maximise the marginal likelihood of the latest compressed set at each iteration. Alternatively, we can reserve a portion of the data $S \subseteq D$ to inform the prior, in which case, the resulting bound will be over the sampling of datasets of size $|D| - |S|$.

4.2 Posterior Predictive Risk

Here, we show that an equivalent bound to that of Theorem 1 holds in the Bayesian setting. To this end, we first define the notion of risk for conditional probabilities rather than hypotheses. Assume the sets $\mathcal{X}$ and $\mathcal{Y}$ have associated σ-algebras $\Sigma_{\mathcal{X}}$ and $\Sigma_{\mathcal{Y}}$, respectively. A probability kernel $k : \mathcal{X} \times \Sigma_{\mathcal{Y}} \to [0, 1]$ associates with each element $x \in \mathcal{X}$ a probability measure $\mathbb{P}_{Y|X}(\cdot \mid x) = k(x, \cdot)$ on $(\mathcal{Y}, \Sigma_{\mathcal{Y}})$, and is such that for every measurable $B \in \Sigma_{\mathcal{Y}}$, the map $x \mapsto k(x, B)$ is $\Sigma_{\mathcal{X}}$-measurable (Billingsley, 1995). We define the space of conditional distributions over $(\mathcal{Y}, \Sigma_{\mathcal{Y}})$ for a given $x \in \mathcal{X}$ as

$$\mathcal{M}_x(\mathcal{Y}) = \Big\{ \mathbb{P}_{Y|X=x} : \Sigma_{\mathcal{Y}} \to [0, 1] \mid \exists k \text{ s.t. } \mathbb{P}_{Y|X=x}(B) = k(x, B)\ \forall B \in \Sigma_{\mathcal{Y}} \Big\},$$

and the space of all such conditional distributions as

$$\mathcal{M}_{\mathcal{X}}(\mathcal{Y}) = \bigcup_{x \in \mathcal{X}} \mathcal{M}_x(\mathcal{Y}).$$

For simplicity, we assume that for any $x \in \mathcal{X}$ and $\mathbb{P}_{Y|X=x} \in \mathcal{M}_x(\mathcal{Y})$, $\mathbb{P}_{Y|X=x}$ is absolutely continuous with respect to the Lebesgue measure μ, allowing us to work directly with the probability density function p defined as the Radon-Nikodym derivative

$$p(\cdot) = \frac{d\mathbb{P}_{Y|X=x}}{d\mu}(\cdot).$$

We are now ready to define the statistical risk.

Definition 1. *Let $(\boldsymbol{p_T}, \boldsymbol{T}) = \mathcal{A}_{BP2L-ES}(\boldsymbol{D})$ and let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ be i.i.d. as each element of $\boldsymbol{D}$. The statistical risk $R : \mathcal{M}_{\mathcal{X}}(\mathcal{Y}) \to \mathbb{R}_{\geq 0}$ of the learned conditional distribution is defined as*

$$R(\boldsymbol{p_T}) = \mathbb{P}\{\texttt{Stop} \leq_{\boldsymbol{p_T}} \boldsymbol{z} \mid \boldsymbol{D}\}.$$

Having formalised the statistical risk for predictive posterior distributions obtained via Algorithm 3, one

can verify that the Bayesian variant of P2L does not modify the compression function, meaning that Theorem 1 can be applied mutatis-mutandis to provide an identical risk bound for $(p_{\boldsymbol{T}}, \boldsymbol{T}) = \mathcal{A}_{BP2L-ES}(\boldsymbol{D})$.

5 GP-P2L CORRESPONDENCE

A particular choice of learner we are interested in nesting inside P2L is the widely employed Gaussian process. In what follows, we provide a brief overview of Gaussian processes and leverage the equivalence between their weight-space and function-space formulations to nest them inside Bayesian P2L. Finally, we show that subset-of-data GP approximations are specific instantiations of P2L and therefore naturally inherit its tight generalisation guarantees.

5.1 Gaussian Processes

Gaussian processes are stochastic processes where every finite collection of random variables follows a joint Gaussian distribution; see Rasmussen and Williams (2006) for a detailed exposition. They can be thought of as representing distributions over functions. In particular, a GP prior over real-valued functions is fully specified by its mean function μ and covariance function k

$$f \sim \mathcal{GP}(\mu(\cdot;\theta), k(\cdot,\cdot';\theta))$$

where θ denotes the kernel's hyperparameters (MacKay, David J. C., 1998). Given a dataset $D = \{(x_n, y_n)\}_{n=1}^N$ with $x_n \in \mathcal{X}$ and $y_n \in \mathbb{R}$, we write $X_D \triangleq [x_1, \ldots, x_N]^T$, $y_D \triangleq [y_1, \ldots, y_N]^T$ and $\mathrm{f}_D \triangleq f(X_D)$. The most common choice of likelihood is $y_D|\mathrm{f}_D \sim \mathcal{N}(\mathrm{f}_D, \sigma^2 I)$. However, certain problems require non-Gaussian likelihoods. For example, in binary classification settings, we assume a Bernoulli likelihood $y_D|\mathrm{f}_D \sim \prod_{n=1}^N \mathcal{B}(\Phi(f(x_n)))$ where Φ denotes the normal cumulative density function. Posterior inference can be performed for new latent function values f_* defined on the set $* = \mathcal{X} \setminus D$ using Bayes' rule. The posterior process is denoted as $\mathcal{GP}_{post} \triangleq \mathcal{GP}(\mu_{post}(\cdot;\theta), k_{post}(\cdot,\cdot';\theta))$. In the case of a Gaussian prior, the mean and covariance have the following closed form

$$\begin{aligned}\mu_{post}(\cdot;\theta) &= K^{(\theta)}_{\cdot \mathrm{f}_D}[K^{(\theta)}_{\mathrm{f}_D\mathrm{f}_D} + \sigma^2 I]^{-1} y_D \\ k_{post}(\cdot,\cdot';\theta) &= K^{(\theta)}_{\cdot\cdot'} - K^{(\theta)}_{\cdot \mathrm{f}_D}[K^{(\theta)}_{\mathrm{f}_D\mathrm{f}_D} + \sigma^2 I]^{-1} K^{(\theta)}_{\mathrm{f}_D \cdot'}\end{aligned}$$

where K is shorthand for the corresponding kernel function evaluations. In non-conjugate models, i.e., models where posterior inference cannot be performed in closed form, approximate inference is required. A fully Bayesian treatment of GP inference would require setting a hyperprior $p(\theta)$ and approximating its intractable posterior using variational inference (Lalchand and Rasmussen, 2020) or Markov Chain Monte Carlo (Flaxman et al., 2015). Most commonly, however, their optimal setting is learnt through gradient-based optimisation of the marginal likelihood (Liu and Nocedal, 1989; Zhu et al., 1997).

5.2 Gaussian Processes in Pick-to-Learn

Our Bayesian reformulation of P2L readily allows us to nest GPs inside the meta-algorithm. We formalise this in the remainder of this section. Importantly, the function space view of GPs presented in Sec. 5.1 can be viewed as the limit of a parametric model in weight space (Rasmussen and Williams, 2006). In the context of Bayesian P2L, this means that the parametric prior can be replaced by the non-parametric *prior process* together with the kernel hyperparameters in the initial configuration. This configuration will need to additionally include the marginal likelihood optimiser $\mathfrak{L}$. The total order $\leq_{\mathcal{GP}}$ will be determined by the process's marginal evaluations on the remaining points. Given $(\mathfrak{L}, M, \theta_0, \mathcal{GP}(0, k(\cdot,\cdot';\theta_0)), \leq_{\mathcal{GP}})$ we denote by

$$(\mathcal{GP}(\mu_{post}(\cdot;\theta^\star_{\boldsymbol{T}}), k_{post}(\cdot,\cdot';\theta^\star_{\boldsymbol{T}})), \boldsymbol{T}) = \mathcal{A}_{BP2L-ES}(\boldsymbol{D})$$

the output of P2L-ES with a Gaussian process learner. In the conjugate model, given a Dirac delta hyperprior for $\boldsymbol{\theta}$, step 7 in Algorithm 3 will be replaced by:

$$\begin{aligned}\theta_m &= \operatorname*{argmin}_\theta - \log p(y_{T_m}|X_{T_m}, \theta) \\ \mu_m(\cdot;\theta_m) &= K^{(\theta_m)}_{\cdot \mathrm{f}_{T_m}}\left[K^{(\theta_m)}_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}\right]^{-1} y_{T_m} \\ k_m(\cdot,\cdot';\theta_m) &= K^{(\theta_m)}_{\cdot\cdot'} - \\ &\qquad K^{(\theta_m)}_{\cdot \mathrm{f}_{T_m}}\left[K^{(\theta_m)}_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}\right]^{-1} K^{(\theta_m)}_{\mathrm{f}_{T_m}\cdot'}\end{aligned}$$

In the non-conjugate case, this step will amount to maximising the evidence lower bound for the parameters of a multivariate Gaussian variational distribution. Details of these calculations can be found in Opper and Archambeau (2009). As for the determination of appropriateness, the posterior of $\mathrm{f}_{D\setminus T_m}$ needed for $\leq_{\mathcal{GP}_{T_m}}$ will be a normal distribution governed by the corresponding mean and covariance evaluations. As detailed in Sec. 4.1, instead of optimising the hyperparameters at every iteration, we can use a portion of the data to estimate the prior θ_0 by minimising the NLML and run P2L on the remaining points. This further allows for efficient covariance matrix updates using Schur's complement (Horn and Johnson, 2012). Additional details for this approach are provided in Appendix C.2.

5.3 Greedy SoD GPs as P2L Instantiations

Even with fixed hyperparameters, GP inference scales as $\mathcal{O}(N^3)$, which is prohibitive beyond the small-data regime. Consequently, several approximations have been developed to reduce the computational complexity to $\mathcal{O}(NM^2)$ (Seeger et al., 2003; Snelson and Ghahramani, 2005; Titsias, 2009) or $\mathcal{O}(M^3)$ (Hensman et al., 2013) for $M \ll N$. The simplest of these, the subset-of-data (SoD) approximation, uses a subset of points in the posterior computations (Quiñonero-Candela and Rasmussen, 2005), with the advantage that the approximation to the full model is itself a GP. Existing approaches differ in the subset selection strategy, with greedy criteria being the most common. Algorithm 4 shows a prototypical training routine for such approximations: it starts with an empty set I and a remaining set $Q = \{(x_n, y_n)\}_{n=1}^N$ that initially contains all data points, and iteratively selects the point that maximizes a criterion function $\Delta : \mathcal{X} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, adding it to I and removing it from Q until either a predefined number of points has been selected or a cost function defined on the remaining points $C : \cup_{n=N-M}^N (\mathcal{X} \times \mathbb{R})^n \to \mathbb{R}$ has been sufficiently minimised to satisfy $C(Q) \leq \gamma$ for some $\gamma \in \mathbb{R}$. In the following, we show that such SoD approximations can be viewed as instantiations of P2L-ES and thus inherit tight generalization bounds.

First, observe that the criterion function Δ induces the following P2L total order[5] on Q:

$$z_i \leq_{\mathcal{GP}} z_j \iff \Delta(z_i) \leq \Delta(z_j),\ \forall z_i, z_j \in Q.$$

By additionally selecting the cost function $C(Q) = \max_{z \in Q} \Delta(z)$ we can extend $\leq_{\mathcal{GP}}$ to a total order on $Q \cup \{\texttt{Stop}\}$ that satisfies

$$z \leq_{\mathcal{GP}} \texttt{Stop} \iff \Delta(z) \leq \gamma,\ \forall z \in Q \tag{9}$$

This establishes a one-to-one correspondence to Bayesian P2L-ES with a GP learner. Defining the risk of the learned posterior GP as

$$R(\boldsymbol{\mathcal{GP}}_{post}) = \mathbb{P}^N\{\texttt{Stop} \leq_{\boldsymbol{\mathcal{GP}}_{post}} \Delta(\boldsymbol{z}) \mid \boldsymbol{Q}\}$$

we can state the below corollary as a result.

Corollary 1. *The risk of any greedy SoD GP approximation following Algorithm 4 with* $C(Q) = \max_{z \in Q} \Delta(z)$*, is bounded as*

$$\mathbb{P}^N\{R(\boldsymbol{\mathcal{GP}}_{post}) \leq \bar{\varepsilon}(|\boldsymbol{I}| + |\{z \in \boldsymbol{Q} \setminus \boldsymbol{I} : \Delta(z) > \gamma\}|, \delta)\} \geq 1 - \delta.$$

[5] Technically, Δ induces a total preorder on Q that can be extended to a total order by introducing antisymmetry through a tiebreak such as the lexicographical order.

Algorithm 4 The Greedy SoD GP Approximation

Initialize: M, $I = \emptyset$, $Q = \{z_1, \ldots, z_N\}, \gamma \in \mathbb{R}$
while $|I| \leq M$ and $C(Q) > \gamma$ **do**
  Find the posterior on Q conditioned on I
  Compute $\Delta(z_j)$ for all $j \in \{1, \ldots, |Q|\}$
  $\bar{z} = \text{argmax}_{z_j \in Q} \Delta(z_j)$
  $I \leftarrow I \cup \{\bar{z}\}$, $Q \leftarrow Q \setminus \{\bar{z}\}$
end while
return I

While previous work has established that greedy SoD GP approximations define compression schemes (Herbrich, 2002; Seeger, 2002a), none, to our knowledge, have identified this compression as *preferent*. This distinction, together with the selection of the exit condition, is essential for achieving tight bounds on the probability of misprediction for any chosen criterion function Δ, as we showcase through our numerics in Section 6. If Δ is an interpretable metric, such as the L1 or L2 loss for regression or the cross-entropy loss for classification, this bound becomes informative in predicting the learned model's performance over unseen data. At the same time, approximations whose appropriateness criterion does not directly coincide with the performance metric of interest still carry an additional valuable piece of information. An example is that of informative vector machine (IVM) (Lawrence et al., 2002), which we detail in Appendix C.3.

6 EXPERIMENTS

By nesting GPs within Bayesian P2L and P2L-ES, we aim to demonstrate that our approach can lead to both improved bounds and improved post-training performance when compared to state-of-the-art methods including the train-test and PAC-Bayes approaches. To do so, we consider both regression (Section 6.1) and classification (Section 6.2) tasks.

We perform experiments on multiple datasets. Each d-dimensional training dataset, consisting of N_{train} points, is divided into C equally sized folds, with experiments conducted for each fold, following the approach in Paccagnan et al. (2023). Additionally, we consider nine pretraining fractions of the dataset (ranging from 0.1 to 0.9). For P2L methods, the pretrain portion is used to learn a prior GP, while the remaining points are compressed with and without early stopping to obtain the posterior GP and corresponding bound. For the train-test split, we train a model on the pretrain fraction and use the remaining points to compute the binomial tail inversion (BTI) bound (Langford, 2005). Where applicable, we also compare with PAC-Bayes methods, using the pretrain

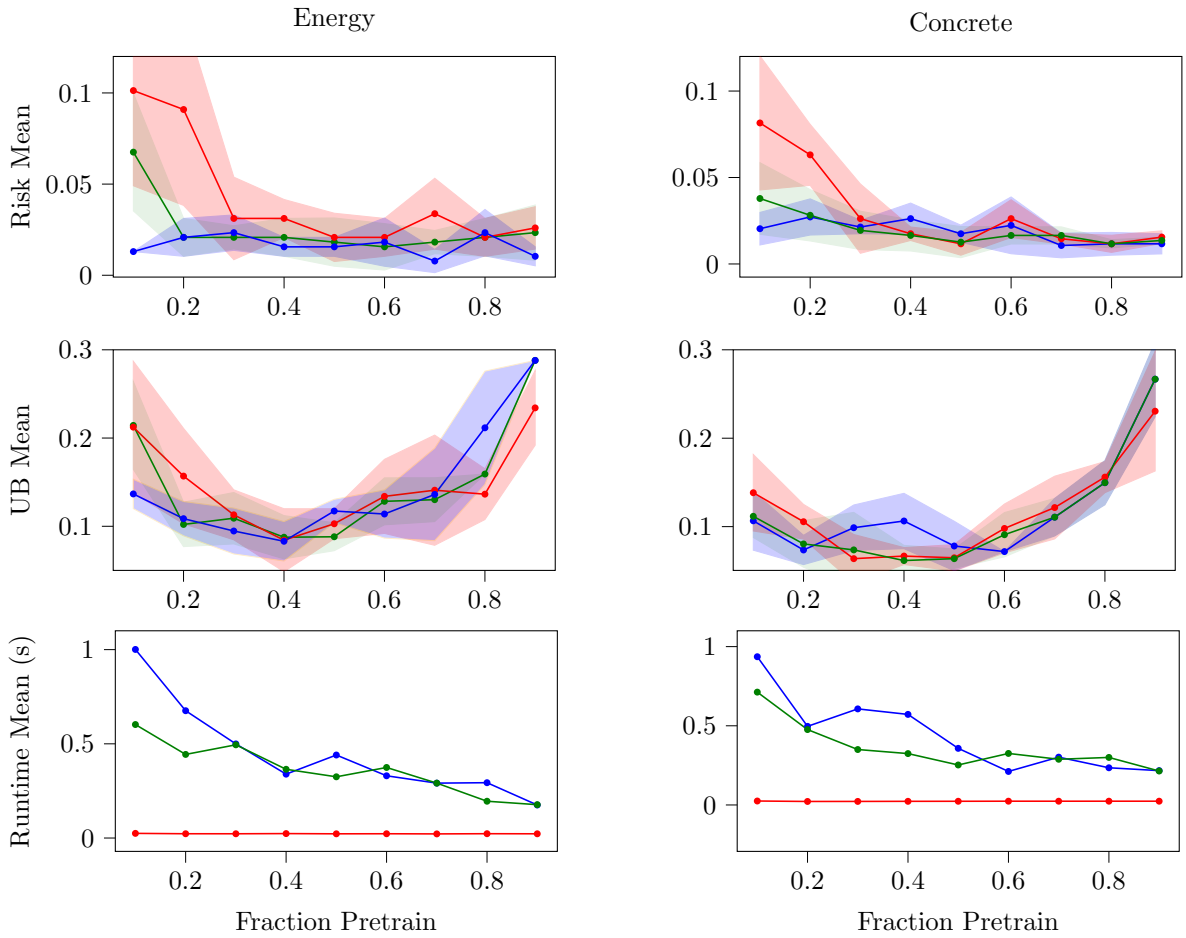

Figure 1: The rows contain from top to bottom: the mean risks on the test set, the mean bounds on the risk, and the average runtimes in seconds for P2L-ES (green), regular P2L (blue) and BTI (red), across different pretrain fractions for the two regression datasets. The shaded regions represent the standard deviations.

fraction to learn a data-dependent prior and the remaining points to derive the posterior and corresponding bounds.[6] We set the confidence level to $\delta = 0.035$ for all PAC bounds and use the mean L1 loss and cross entropy as selection criteria for regression and classification respectively. We evaluate post-training performance on a separate set of N_{test} samples.[7]

6.1 Regression

We start by considering two widely used regression datasets from the UCI repository: *Energy* ($N_{\text{train}} = 691, N_{\text{test}} = 77, d = 8, C = 5$), and *Concrete* ($N_{\text{train}} = 824, N_{\text{test}} = 206, d = 9, C = 5$). We use a square exponential kernel with unit variance for both of them. For the Pick-to-Learn approaches, the appropriateness thresholds are set to $\gamma = 0.53$ and $\gamma = 1.35$, with corresponding cutoffs of $M = 4$ and $M = 5$ points. The results are presented in Figure 1.

For both datasets, P2L and P2L-ES achieve a better risk bound compared to the BTI approach for the majority of the pretrain fractions employed (second row in Figure 1). Additionally, P2L-ES itself returns improved bounds compared to P2L for various pretrain fractions, notably for the Concrete dataset. Interestingly, while providing better bounds, P2L and P2L-ES also achieve better post-training performance when compared to the BTI approach (first row in Figure 1), an effect particularly notable for small pretrain fractions and likely attributable to the active subset selection they perform. In terms of computational cost, early stopping reduces, on average, the runtime relative to regular P2L. However, as expected, both iterative training methods are slower than the train-test approach, although still terminating within seconds.

[6] A discussion of the methods used for comparison, along with the details necessary to reproduce our results can be found in Appendix D.

[7] Our code is available at `https://github.com/MarksDaniel/GPP2L`

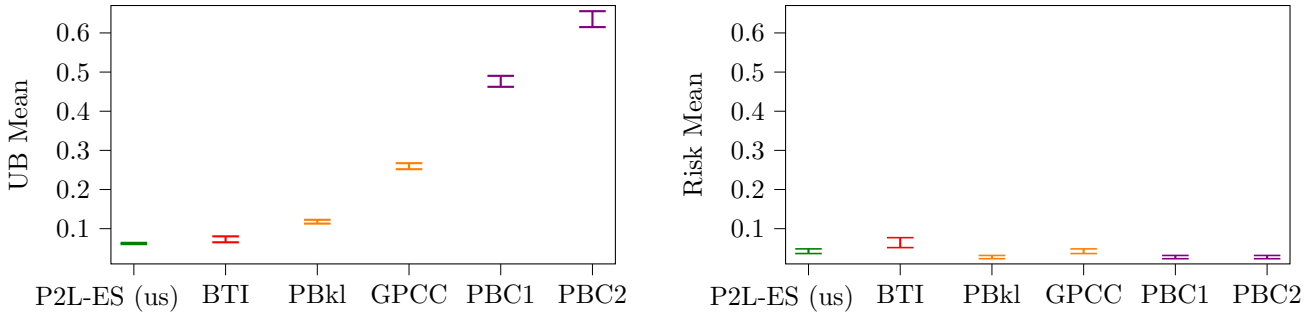

Figure 2: The mean bounds on the risk (left), and the mean risks on the test set (right) for P2L-ES, BTI, and PAC-Bayes. The error bars represent the standard deviations.

6.2 Classification

We now consider the binary classification task of distinguishing handwritten twos against threes from the MNIST dataset ($N_{train} = 12089$, $N_{test} = 2042$, $d = 64$, $C = 10$), similarly to what is done in Seeger (2002a). Specifically, we compare our early-stopping bounds for $M = 100$ and $\gamma = -\log(0.5)$ against several alternatives: the BTI bound, Germain's PAC-Bayes C-bounds, denoted as "PBC1" and "PBC2" (Germain et al., 2015, PAC-Bounds 1 and 2), and Germain's refinement of Seeger's bound for Bayes classifiers, denoted as "PBkl" (Germain et al., 2015, Corollary 20). We also compare against Seeger's adaptation of Herbrich's PAC bound for greedy SoD GP Bayes classifiers, denoted as "GPCC" (Seeger, 2002b, Theorem 3). The results for the best pre-train fraction (0.1) are shown in Figure 2.

Remarkably, the P2L-ES bound is sharper than the BTI bound and all PAC-Bayes bounds. Among the PAC-Bayes bounds, the PBkl bound is the tightest with the C bounds being especially loose. This is commonly the case for majority votes with low disagreement (Lorenzen et al., 2019), with Seeger's bound often being the tightest PAC-Bayesian bound for binary classification (Masegosa et al., 2020). Notably, the improvement of P2L-ES over GPCC, a former state-of-the-art compression bound for greedy SoD GPs, highlights the benefits of inducing *preferent* compressions.

As far as the realised risk is concerned, although we consider six distinct bounds on the risk, some of the training routines required to produce them overlap, resulting in identical learned hypotheses and, consequently, the same post-training error. Specifically, the three PAC-Bayesian methods (PBkl, PBC1, PBC2) yield identical post-training risks, as do the two compression-based methods (P2L-ES, GPCC). While a clear hierarchy emerges among the methods based on their risk bounds (left panel), they all achieve comparable and very low risk on the held-out test set (right panel). When evaluating them jointly on the two risk objectives, P2L-ES outperforms GPCC among the compression-based methods – benefiting from preferential compression –, and also outperforms the test-set approach by achieving a sharper bound with an equivalent or better post-training performance. In a similar manner, PBkl outperforms the two C bounds among the PAC-Bayesian approaches. When comparing P2L-ES with PBkl, P2L-ES achieves a better bound on the risk, while PBkl attains a slightly lower risk on the test set. In this case, the tiebreaker appears to be that P2L-ES provides a much tighter bound, as measured by the gap between its bound and realized risk, compared to the PAC-Bayesian PBkl approach.

7 CONCLUSIONS

In this work, we have shown that the Pick-to-Learn algorithm introduced by Paccagnan et al. (2023) can be refined so as to jointly reduce its runtime, improve its generalisation bounds, and accommodate stochastic learners. Building upon these results, we have then demonstrated how SoD GP approximations are naturally self-certified. While the bounds we obtain compare favourably with the state of the art, we see at least two directions for future work. First, the current version of P2L does not directly allow decoupling the choice of the termination condition from the appropriateness criterion. Doing so will bring additional flexibility to the approach. Second, we see significant opportunities for equipping, through P2L, other, different, GP approximations with strong generalization bounds – a direction which we leave for future work.

Acknowledgements

We would like to thank Mark van der Wilk for the useful discussions and the anonymous reviewers for their valuable feedback. This work was partially supported by EPSRC grant EP/Y001001/1, funded by the International Science Partnerships Fund (ISPF) and UKRI.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G.,

Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: A system for large-scale machine learning.

Alquier, P. (2023). User-friendly introduction to PAC-Bayes bounds.

Bartlett, P. L. and Mendelson, S. (2001). Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. In Goos, G., Hartmanis, J., Van Leeuwen, J., Helmbold, D., and Williamson, B., editors, *Computational Learning Theory*, volume 2111, pages 224–240. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Billingsley, P. (1995). *Probability and measure*. Wiley series in probability and mathematical statistics. Wiley, New York, 3rd ed edition.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to End Learning for Self-Driving Cars.

Campi, M. C. and Garatti, S. (2023). Compression, Generalization and Learning. In *Journal of Machine Learning Research, 24(339):1-74*.

Carlin, B. P. and Louis, T. A. (1997). Bayes and Empirical Bayes Methods for Data Analysis. *Statistics and Computing*, 7(2):153–154.

Catoni, O. (2007). Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. *IMS Lecture Notes Monograph Series*, 56.

Clerico, E., Deligiannidis, G., and Doucet, A. (2022). Conditionally Gaussian PAC-Bayes. In *International Conference on Artificial Intelligence and Statistics*, pages 2311–2329. PMLR.

Dziugaite, G. K. and Roy, D. M. (2017). Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*.

Flaxman, S., Gelman, A., Neill, D., Smola, A., Vehtari, A., and Wilson, A. G. (2015). Fast hierarchical Gaussian processes.

Floyd, S. and Warmuth, M. (1995). Sample Compression, Learnability, and the Vapnik-Chervonenkis Dimension. *Machine Learning*, 21(3):269–304.

Foong, A., Bruinsma, W., Burt, D., and Turner, R. (2021). How Tight Can PAC-Bayes be in the Small Data Regime? In *Advances in Neural Information Processing Systems*, volume 34, pages 4093–4105. Curran Associates, Inc.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Germain, P., Lacasse, A., Laviolette, F., March, M., and Roy, J.-F. (2015). Risk Bounds for the Majority Vote: From a PAC-Bayesian Analysis to a Learning Algorithm. *Journal of Machine Learning Research*, 16(26):787–860.

Graepel, T., Herbrich, R., and Shawe-Taylor, J. (2005). PAC-Bayesian Compression Bounds on the Prediction Error of Learning Algorithms for Classification. *Machine Learning*, 59(1):55–76.

Guedj, B. (2019). A Primer on PAC-Bayesian Learning. In *Congres de la Societe Mathematique de France*, volume 33. Collection SMF.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian Processes for Big Data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290.

Herbrich, R. (2002). *Learning kernel classifiers: theory and algorithms*. Adaptive computation and machine learning. The MIT Press, Cambridge, Mass.

Horn, R. A. and Johnson, C. R. (2012). *Matrix Analysis:*. Cambridge University Press, 2 edition.

Kevin P. Murphy (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109.

Lalchand, V. and Rasmussen, C. E. (2020). Approximate Inference for Fully Bayesian Gaussian Process Regression. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, pages 1–12. PMLR.

Langford, J. (2005). Tutorial on Practical Prediction Theory for Classification. *Journal of Machine Learning Research*, 6:273–306.

Lawrence, N., Seeger, M., and Herbrich, R. (2002). Fast Sparse Gaussian Process Methods: The Informative Vector Machine. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Littlestone, N. and Warmuth, M. (1986). Relating data compression and learnability.

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

Lorenzen, S. S., Igel, C., and Seldin, Y. (2019). On PAC-Bayesian Bounds for Random Forests.

MacKay, David J. C. (1998). Introduction to Gaussian Processes. In Bishop, Christopher M., editor, *Neural Networks and Machine Learning*, Springer-Verlag.

Masegosa, A., Lorenzen, S., Igel, C., and Seldin, Y. (2020). Second Order PAC-Bayesian Bounds for the Weighted Majority Vote. In *Advances in Neural Information Processing Systems*, volume 33, pages 5263–5273. Curran Associates, Inc.

Matthews, A. G. d. G., Wilk, M. v. d., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, Pablo, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian Process Library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Opper, M. and Archambeau, C. (2009). The variational gaussian approximation revisited. *Neural Comput.*, 21(3):786–792.

Paccagnan, D., Campi, M., and Garatti, S. (2023). The Pick-to-Learn Algorithm: Empowering Compression for Tight Generalization Bounds and Improved Post-training Performance. In *Advances in Neural Information Processing Systems*, volume 36, pages 18165–18185.

Perez-Ortiz, M., Rivasplata, O., Shawe-Taylor, J., and Szepesvari, C. (2021). Tighter Risk Certificates for Neural Networks. *Journal of Machine Learning Research*, 22:1–40.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6(65):1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge, Mass.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A. S., Maharaj, T., Sherwin, E. D., Mukkavilli, S. K., Kording, K. P., Gomes, C. P., Ng, A. Y., Hassabis, D., Platt, J. C., Creutzig, F., Chayes, J., and Bengio, Y. (2023). Tackling Climate Change with Machine Learning. *ACM Computing Surveys*, 55(2):1–96.

Seeger, M. (2002a). Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of Machine Learning Research*, 3(2):233–269.

Seeger, M. (2002b). PAC-Bayesian generalization error bounds for Gaussian process clas- sification. Technical Report EDI-INF-RR-0094, University of Edinburgh, Division of Informatics.

Seeger, M. W., Williams, C. K. I., and Lawrence, N. D. (2003). Fast Forward Selection to Speed Up Sparse Gaussian Process Regression. In *International Workshop on Artificial Intelligence and Statistics*, pages 254–261. PMLR.

Snelson, E. and Ghahramani, Z. (2005). Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Titsias, M. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 567–574. PMLR.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Springer New York, New York, NY.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*.

Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable] In the appendix.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable] In the appendix.
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]
   In Section 6 and in the appendix.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable] In Section 6 and in the appendix.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable] In Section 6.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable] In the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable] In the appendix.
   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

Appendix

A ADDITIONAL MATERIAL FOR SECTION 3

This section contains the proofs of the results presented in Section 3.

A.1 Proof of Proposition 1

Here we show that the P2L bound (3) does not hold when applied directly to P2L-ES.[8] To prove this, it is sufficient to find a probability distribution $\mathbb{P}$, a learning algorithm L, a total order $\leq_h$, and an initial hypothesis h_0 for which (3) is violated.

Proposition 1. *There exists a probability measure $\mathbb{P}$ and $(L, h_0, \leq_h)$, such that the corresponding P2L-ES execution with M maximum iterations produces a hypothesis $\boldsymbol{h}$ and a compressed multiset $\boldsymbol{T}$ that violates* (3) *for any choice of $M < N = |D|$ and $\delta < 1$.*

Proof. We construct the following counterexample. Let $\mathcal{Z} = \mathbb{R}^2$, and let $\mathbb{P}$ be a uniform probability distribution over the unit circle $\mathcal{S}^1 \subset \mathbb{R}^2$ used to generate samples $\mathbf{z}$. Let $D = \{z_1, \dots, z_N\}$ be a realization of $\mathbf{D} \sim \mathbb{P}^N$. Consider a learning algorithm L that, given a set of points T constructs their convex hull, denoted with $conv(T)$. Note that, in this setting, a hypothesis h consists of a convex hull of points. Fix h_0 as the convex hull of the set containing only the origin in $\mathbb{R}^2$, i.e., $h_0 = L(\{(0,0)\})$.[9]

Let $d(z, conv(T))$ denote the Euclidean distance between a point $z \in \mathbb{R}^2$ and the closest point inside $conv(T)$. This distance will be 0 if and only if $z \in conv(T)$. For any compressed set T, define the P2L-ES total order as the order ranking points according to their distance from $conv(T)$ and such that `Stop` is the largest element if all points in T belong to $conv(T)$. Formally, define $\leq_h$ as

$$\begin{aligned} z_i \leq_h z_j &\iff d(z_i, conv(T)) \leq d(z_j, conv(T)) \\ z \leq_h \texttt{Stop} &\iff d(z, conv(T)) = 0. \end{aligned}$$

Since ($\mathbb{P}$-almost surely) no point on the unit circle can lie in the convex hull of a finite set of distinct points on the unit circle, the risk of any hypothesis learned by P2L-ES is equal to 1, regardless of the choice of M. However, since P2L-ES stops at iteration M, with $M < N$, we have $\bar{\varepsilon}(|T|, \delta) < 1$ for any choice of $\delta < 1$. The latter fact follows immediately by the definition of $\bar{\varepsilon}(\cdot, \cdot)$ in (2). This suffices to conclude, as it implies that $\mathbb{P}^N\{R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\boldsymbol{T}|, \delta)\} = 0 < 1 - \delta$ for any $\delta < 1$.

□

In addition to the previous example, we present another instructive counterexample arising directly from the supervised learning setting. Consider a binary classification task with feature space $\mathcal{X} = [0, 1]$ and class labels $\mathcal{Y} = \{0, 1\}$. Suppose the distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$ is uniform, and consider any inner learning algorithm that, given a dataset of training points, produces a decision boundary in $\mathcal{X}$ to separate the two classes. Since the distribution is uniform—implying no correlation between the input x and the class label y—any chosen decision boundary inevitably misclassifies exactly 50% of points. Formally, this implies that the true risk (the probability of misclassification) is precisely 0.5, regardless of the hypothesis selected. However, if we run P2L-ES with any positive confidence parameter $\delta > 0$, for instance $\delta = 0.01$, and terminate early, say at $M = 2$, with $|D| = 100$ data points, the resulting P2L bound will incorrectly indicate that the risk is upper-bounded by 0.106.

[8] Naturally, this is meaningful only in the nontrivial case when the maximum number of iterations M is strictly less than the size of D. Otherwise P2L-ES and P2L coincide and therefore (3) holds.

[9] This choice is completely arbitrary. Other choices will not impact the ensuing argument.

A.2 Proof of Theorem 1

Here we prove Theorem 1. The proof is divided into four parts. In the first part, we show that the compression function κ_{P2L-ES} satisfies the property of preference. This property will be used repeatedly in the ensuing parts. In the second part, we show that $\tilde{\kappa}_{P2L-ES}$ also satisfies the property of preference. In the third part, we show that the risk of any hypothesis learned via P2L-ES is almost-surely bounded by the probability of change of compression $\phi_{\tilde{\kappa}_{P2L-ES}}$. Finally, we combine our results from parts two and three to derive the final bound. Throughout our proof, we will make use of the following lemma, which we state without proof.

Lemma 1 (Lemma 3 in Campi and Garatti (2023))**.** *A compression function c satisfies the property of preference if and only if $c(U) = c(V)$ for all multisets U, V such that $c(U) \subseteq V \subseteq U$.*

Before moving to the proof of Theorem 1, we relax the assumption made in the main body of the paper, and allow for the inner learner output to depend on the order of the elements in the compressed set. Indeed, our results hold unchanged in this more general setting. Towards this goal, we denote by $[T]$, the ordered list of examples generated by P2L-ES through its execution. In what follows, we also use the shorthand notation $\kappa(D, z) \triangleq \kappa(D \cup \{z\})$ for any compression function κ, any $\mathcal{Z}$-valued multiset D and any example $z \in \mathcal{Z}$ to enhance readability. We similarly use $\mathcal{A}_{P2L-ES}(D, z) \triangleq \mathcal{A}_{P2L-ES}(D \cup \{z\})$ for P2L-ES executions. We begin with the proof of preference for the compression function induced by P2L-ES.

Lemma 2. *The compression function κ_{P2L-ES} satisfies the property of preference.*

Proof. We need to show that for an arbitrary pair of multisets D and $V \subseteq D$ of elements in $\mathcal{Z}$, if $\kappa_{P2L-ES}(D) \neq V$ then $\kappa_{P2L-ES}(D, z) \neq V$ for any $z \in \mathcal{Z}$. We equivalently show that if $\kappa_{P2L-ES}(D, z) \neq \kappa_{P2L-ES}(D)$ then $\kappa_{P2L-ES}(D, z) \neq V$ for any $V \subseteq D$, which immediately implies that κ_{P2L-ES} is preferent.

Let T_m, h_m and T'_m, h'_m denote the multisets and hypotheses constructed by $\mathcal{A}_{P2L-ES}(D)$ and $\mathcal{A}_{P2L-ES}(D, z)$ respectively, at some iteration $m < M$. We additionally denote the augmented multisets for the two runs of the meta-algorithm by $D_S = D \cup \{\texttt{Stop}\}$ and $D'_S = D_S \cup \{z\}$. We have $T_0 = \emptyset = T'_0$ and $h_0 = h'_0$ by the shared initialisation step. Following Paccagnan et al. (2023), we show that since $\kappa_{P2L-ES}(D, z) \neq \kappa_{P2L-ES}(D)$ there must exist an iteration $\overline{m} < M$ up to which the two executions agree, meaning $T_m = T'_m$ and $h_m = h'_m$ for all $m \leq \overline{m}$, but differ at iteration $\overline{m}$. This means that the maximal elements selected in the $\overline{m}$-th iteration of the two runs differ. Since $T_{\overline{m}} = T'_{\overline{m}}$ and $h_{\overline{m}} = h'_{\overline{m}}$ we have

$$\max_{h_{\overline{m}}}(D_S \setminus T_{\overline{m}}) \neq \max_{h'_{\overline{m}}}(D'_S \setminus T'_{\overline{m}}) = \max_{h_{\overline{m}}}(D'_S \setminus T_{\overline{m}}).$$

It therefore must be that

$$\max_{h_{\overline{m}}}(D'_S \setminus T_{\overline{m}}) = z \notin D_S \setminus T_{\overline{m}},$$

or we would otherwise arrive at a contradiction. Since $z \neq \texttt{Stop}$ by construction, and since $\overline{m} < M$, $\mathcal{A}_{P2L-ES}(D, z)$ runs for at least one more iteration, adding z to the compressed set and constructing

$$T'_{\overline{m}+1} = T'_{\overline{m}} \cup \{z\} = T_{\overline{m}} \cup \{z\}.$$

At this point, $T'_{\overline{m}+1}$ contains the element z as many times as it appears in D, plus one, otherwise, even if z was the least appropriate element, we would have $z \in D_S \setminus T_{\overline{m}}$ and the maximal elements selected in the two runs would be the same. Therefore, we have $T'_{\overline{m}+1} \not\subseteq D$. Since the size of the compressed set increases at every iteration we know that $T'_m \not\subseteq D$ for any $m > \overline{m}$ and so this certainly holds for the terminating iteration, meaning that $\kappa_{P2L-ES}(D, z) \not\subseteq D$ and so $\kappa_{P2L-ES}(D, z) \neq V$, for any $V \subseteq D$ as required. □

We now use this lemma to study the modified compression function $\tilde{\kappa}_{P2L-ES}$.

Lemma 3. *The compression function $\tilde{\kappa}_{P2L-ES}$ also satisfies the property of preference.*

Proof. Take arbitrary multisets U, V such that $\tilde{\kappa}_{P2L-ES}(U) \subseteq V \subseteq U$. We want to show that $\tilde{\kappa}_{P2L-ES}(V) = \tilde{\kappa}_{P2L-ES}(U)$ which guarantees preference by Lemma 1. By definition of $\tilde{\kappa}_{P2L-ES}$, we have $\kappa_{P2L-ES}(U) \subseteq \tilde{\kappa}_{P2L-ES}(U)$ and so the following inclusions also hold: $\kappa_{P2L-ES}(U) \subseteq V \subseteq U$. Since κ_{P2L-ES} is preferent by Lemma 2, Lemma 1 guarantees that

$$\kappa_{P2L-ES}(V) = \kappa_{P2L-ES}(U). \tag{A.1}$$

Denote by $(h_V, \kappa_{P2L-ES}(V)) = \mathcal{A}_{P2L-ES}(V)$ and $(h_U, \kappa_{P2L-ES}(U)) = \mathcal{A}_{P2L-ES}(U)$ the results from running P2L-ES on V and U respectively. Then, (A.1) implies that

$$h_V = L([\kappa_{P2L-ES}(V)]) = L([\kappa_{P2L-ES}(U)]) = h_U.$$

Using the above when unfolding the definition of $\tilde{\kappa}_{P2L-ES}(V)$ we have:

$$\begin{aligned}\tilde{\kappa}_{P2L-ES}(V) &= \kappa_{P2L-ES}(V) \cup \{z \in V \setminus \kappa_{P2L-ES}(V) : \texttt{Stop} \leq_{h_V} z\} \\ &= \kappa_{P2L-ES}(U) \cup \{z \in V \setminus \kappa_{P2L-ES}(U) : \texttt{Stop} \leq_{h_U} z\} \\ &= \kappa_{P2L-ES}(U) \cup (\{z \in V : \texttt{Stop} \leq_{h_U} z\} \setminus \kappa_{P2L-ES}(U)),\end{aligned} \tag{A.2}$$

similar to Campi and Garatti (2023). By definition, $\tilde{\kappa}_{P2L-ES}(U)$ contains all elements of U that are not appropriate enough under h_U and since $\tilde{\kappa}_{P2L-ES}(U) \subseteq V$ it follows that

$$\{z \in V : \texttt{Stop} \leq_{h_U} z\} = \{z \in U : \texttt{Stop} \leq_{h_U} z\}. \tag{A.3}$$

Substituting (A.3) into equation (A.2) we recognise the definition of $\tilde{\kappa}_{P2L-ES}(U)$

$$\begin{aligned}\tilde{\kappa}_{P2L-ES}(V) &= \kappa_{P2L-ES}(U) \cup (\{z \in U : \texttt{Stop} \leq_{h_U} z\} \setminus \kappa_{P2L-ES}(U)) \\ &= \kappa_{P2L-ES}(U) \cup \{z \in U \setminus \kappa_{P2L-ES}(U) : \texttt{Stop} \leq_{h_U} z\} \\ &= \tilde{\kappa}_{P2L-ES}(U).\end{aligned}$$

This implies that the compression function is preferent, as desired. □

We now show that the risk of a hypothesis learned via P2L-ES is almost-surely bounded by the probability of change of compression using the modified compression function $\tilde{\kappa}_{P2L-ES}$.

Lemma 4. *Let $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L-ES}(\boldsymbol{D})$ be the output of P2L-ES. It holds that*

$$\mathbb{P}\{R(\boldsymbol{h}) \leq \phi_{\tilde{\kappa}_{P2L-ES}}(\boldsymbol{D})\} = 1$$

Proof. Assume that for a given multiset D and an arbitrary element $z \in \mathcal{Z}$ we have

$$\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) = \tilde{\kappa}_{P2L-ES}(D) \tag{A.4}$$

and denote by $(h', T') = \mathcal{A}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)$, $(h, T) = \mathcal{A}_{P2L-ES}(D)$ the results of running P2L-ES on $\tilde{\kappa}_{P2L-ES}(D) \cup \{z\}$ and D respectively. Then, the multisets in the assumption represent the following unions of multisets

$$\begin{aligned}\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) &\triangleq T' \cup \{z \in (\tilde{\kappa}_{P2L-ES}(D) \cup \{z\}) \setminus T' : \texttt{Stop} \leq_{h'} z\} \\ \tilde{\kappa}_{P2L-ES}(D) &\triangleq T \cup \{z \in D \setminus T : \texttt{Stop} \leq_{h} z\}.\end{aligned} \tag{A.5}$$

We can apply P2L-ES to both sides of (A.4) and obtain

$$\mathcal{A}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)) = \mathcal{A}_{P2L-ES}\left(\tilde{\kappa}_{P2L-ES}(D)\right). \tag{A.6}$$

By the properties of the respective compression functions we can write the inclusions

$$\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) \subseteq \tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) \subseteq \tilde{\kappa}_{P2L-ES}(D) \cup \{z\}$$

and since κ_{P2L-ES} is preferent it follows by Lemma 1 that

$$\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)) = \kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z).$$

We can therefore simplify the left-hand side of (A.6) to

$$\begin{aligned}&\mathcal{A}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)) = \\ &\quad= \left(L\left([\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z))]\right), \kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z))\right) \\ &\quad= \left(L\left([\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)]\right), \kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)\right) \\ &\quad= \mathcal{A}_{P2L-ES}\left(\tilde{\kappa}_{P2L-ES}(D), z\right).\end{aligned} \tag{A.7}$$

Similarly, we can also write the inclusions $\kappa_{P2L-ES}(D) \subseteq \tilde{\kappa}_{P2L-ES}(D) \subseteq D$ and since κ_{P2L-ES} is preferent we have, once again, $\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D)) = \kappa_{P2L-ES}(D)$ by Lemma 1. Therefore the right-hand side of (A.6) similarly simplifies to

$$\begin{aligned}\mathcal{A}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D)) &= \left(L\left([\kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D))]\right), \kappa_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D))\right) \\ &= \left(L\left([\kappa_{P2L-ES}(D)]\right), \kappa_{P2L-ES}(D)\right) \\ &= \mathcal{A}_{P2L-ES}(D).\end{aligned} \tag{A.8}$$

Using (A.6) and combining (A.7) and (A.8) gives

$$\mathcal{A}_{P2L-ES}\left(\tilde{\kappa}_{P2L-ES}(D), z\right) = \mathcal{A}_{P2L-ES}(D), \tag{A.9}$$

and so $h' = h$. By definition, $\tilde{\kappa}_{P2L-ES}$ is such that if $\texttt{Stop} \leq_{h'} z$, then z must appear in the modified compressed multiset $\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)$ as many times as it appears in the multiset that produced the hypothesis, namely, $\tilde{\kappa}_{P2L-ES}(D) \cup \{z\}$. But since z appears in $\tilde{\kappa}_{P2L-ES}(D) \cup \{z\}$ one more time than it appears in $\tilde{\kappa}_{P2L-ES}(D)$, it also appears there one more time than it appears in $\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z)$ by the assumption. We therefore have $z \leq_{h'} \texttt{Stop}$ and so, by the equality of hypotheses established in (A.9) we have $z \leq_h \texttt{Stop}$.

Since this is a consequence of (A.4) we can write

$$\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) = \tilde{\kappa}_{P2L-ES}(D) \implies z \leq_h \texttt{Stop},$$

or equivalently, its contrapositive

$$\texttt{Stop} \leq_h z \implies \tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(D), z) \neq \tilde{\kappa}_{P2L-ES}(D).$$

In probabilistic terms, taking $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L-ES}(\boldsymbol{D})$, we can say that $\texttt{Stop} \leq_{\boldsymbol{h}} \boldsymbol{z}$ is a sub-event of $\tilde{\kappa}_{P2L-ES}(\tilde{\kappa}_{P2L-ES}(\boldsymbol{D}), \boldsymbol{z}) \neq \tilde{\kappa}_{P2L-ES}(\boldsymbol{D})$, that is, inappropriateness is a sub-event of change of compression under $\tilde{\kappa}_{P2L-ES}$. We recognise the probabilities of these two events as the risk of the learned hypothesis and the probability of change of compression under $\tilde{\kappa}_{P2L-ES}$ respectively and we can thus say, as desired, that almost surely

$$R(\boldsymbol{h}) \leq \phi_{\tilde{\kappa}_{P2L-ES}}(\boldsymbol{D}).$$

□

We finally put everything together to derive the final bound on the risk of any hypothesis learned via P2L-ES.

Theorem 1. *Let $(\boldsymbol{h}, \boldsymbol{T}) = \mathcal{A}_{P2L-ES}(\boldsymbol{D})$ be the output of P2L-ES. Then for any $\delta \in (0, 1)$ it holds that*

$$\mathbb{P}^N\{R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\boldsymbol{T}| + |\{z \in \boldsymbol{D} \setminus \boldsymbol{T} : \texttt{Stop} \leq_{\boldsymbol{h}} z\}|, \delta)\} \geq 1 - \delta.$$

Proof. Since $\tilde{\kappa}_{P2L-ES}$ is preferent by Lemma 3, we know by (1) that

$$\mathbb{P}^N\{\phi_{\tilde{\kappa}_{P2L-ES}}(\boldsymbol{D}) \leq \bar{\varepsilon}(|\tilde{\kappa}_{P2L-ES}(\boldsymbol{D})|, \delta)\} \geq 1 - \delta.$$

Moreover, since the risk is dominated by the probability of change of compression $\phi_{\tilde{\kappa}_{P2L-ES}}(\boldsymbol{D})$ by Lemma 4, it follows that

$$\mathbb{P}^N\{R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\tilde{\kappa}_{P2L-ES}(\boldsymbol{D})|, \delta)\} \geq \mathbb{P}^N\{\phi_{\tilde{\kappa}_{P2L-ES}}(\boldsymbol{D}) \leq \bar{\varepsilon}(|\tilde{\kappa}_{P2L-ES}(\boldsymbol{D})|, \delta)\} \geq 1 - \delta.$$

Using the disjointness of the multisets in (A.5) we can re-write the cardinality of the modified compressed set as

$$|\tilde{\kappa}_{P2L-ES}(\boldsymbol{D})| = |\boldsymbol{T} \cup \{z \in \boldsymbol{D} \setminus \boldsymbol{T} : \texttt{Stop} \leq_{\boldsymbol{h}} z\}| = |\boldsymbol{T}| + |\{z \in \boldsymbol{D} \setminus \boldsymbol{T} : \texttt{Stop} \leq_{\boldsymbol{h}} z\}|,$$

thus obtaining our final claim

$$\mathbb{P}^N\{R(\boldsymbol{h}) \leq \bar{\varepsilon}(|\boldsymbol{T}| + |\{z \in \boldsymbol{D} \setminus \boldsymbol{T} : \texttt{Stop} \leq_{\boldsymbol{h}} z\}|, \delta)\} \geq 1 - \delta.$$

□

B ADDITIONAL MATERIAL FOR SECTION 4

In this section, we show that the iterative training procedure of Pick-to-Learn is analogous to Bayesian updating in a hierarchical model. Recall our setup from Section 4. where we assumed a conditional parametric model $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\theta})$ where $\boldsymbol{w}$ and $\boldsymbol{\theta}$ denote the model parameters and hyperparameters respectively, with joint prior $p(\boldsymbol{w}, \boldsymbol{\theta})$ with the goal of inferring the joint posterior

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid D) \propto p(D \mid \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

If we denote by $T_m = (\boldsymbol{x}_{T_m}, \boldsymbol{y}_{T_m})$ the multiset constructed by P2L at iteration m, updates in this model can be formalised as

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m})}. \tag{B.1}$$

This joint posterior can be broken down into two parts; the original posterior over $\boldsymbol{w}$, and a posterior over $\boldsymbol{\theta}$ using the model evidence as the likelihood

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \underbrace{\frac{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m}, \boldsymbol{\theta})}}_{p(\boldsymbol{w} \mid \boldsymbol{\theta}, \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m})} \underbrace{\frac{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m})}}_{p(\boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m})}.$$

Given the sequential nature of the training procedure, we seek a different re-arrangement of the terms in (B.1), in terms of the joint posterior from the previous iteration $p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_{m-1}})$. We start by separating the likelihood of the existing points in the compressed set $\boldsymbol{y}_{T_{m-1}}$ and the likelihood of the points added at the current iteration, which we express as the multiset $\boldsymbol{y}_{T_m \setminus T_{m-1}}$ to obtain

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_{m-1}} \mid \boldsymbol{y}_{T_m \setminus T_{m-1}}, \boldsymbol{x}_{T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{x}_{T_m}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_m}) p(\boldsymbol{y}_{T_{m-1}} \mid \boldsymbol{x}_{T_m})}.$$

Simplifying the conditional dependencies gives

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_{m-1}} \mid \boldsymbol{x}_{T_{m-1}}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{x}_{T_m \setminus T_{m-1}}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_m \setminus T_{m-1}}) p(\boldsymbol{y}_{T_{m-1}} \mid \boldsymbol{x}_{T_{m-1}})}. \tag{B.2}$$

At this point, we recognise part of (B.2) as the applications of Bayes' theorem to the joint posterior from the previous iteration leading to

$$p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{x}_{T_m \setminus T_{m-1}}, \boldsymbol{w}, \boldsymbol{\theta}) p(\boldsymbol{w}, \boldsymbol{\theta} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_{m-1}})}{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_m \setminus T_{m-1}})}.$$

The above posterior considers the likelihood of the latest additions to the compressed set, and uses the previous joint posterior as the prior. This illustrates that from a Bayesian standpoint, the learning step of any Pick-to-Learn based meta-algorithm is analogous to performing a conditional update in a hierarchical model. If we choose to ignore the uncertainty in the hyperparameters and instead perform type-II maximum likelihood, the above joint posterior update will instead consist of two steps. First, the minimisation of the negative log marginal likelihood

$$\boldsymbol{\theta}^*_{T_m} = \operatorname*{argmin}_{\theta} - \log p\left(\boldsymbol{y}_{T_m} \mid \boldsymbol{x}_{T_m} \boldsymbol{\theta}\right)$$

in order to set $p(\boldsymbol{\theta} \mid \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) \approx \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*_{\boldsymbol{T_m}})$, followed by the conditional update

$$p(\boldsymbol{w} \mid \boldsymbol{\theta}^*_{\boldsymbol{T_m}}, \boldsymbol{y}_{T_m}, \boldsymbol{x}_{T_m}) = \frac{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{x}_{T_m \setminus T_{m-1}}, \boldsymbol{w}, \boldsymbol{\theta}^*_{\boldsymbol{T_m}}) p(\boldsymbol{w} \mid \boldsymbol{\theta}^*_{\boldsymbol{T_m}}, \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_{m-1}})}{p(\boldsymbol{y}_{T_m \setminus T_{m-1}} \mid \boldsymbol{y}_{T_{m-1}}, \boldsymbol{x}_{T_m \setminus T_{m-1}})}.$$

C ADDITIONAL MATERIAL FOR SECTION 5

In this section, we detail a more efficient strategy for performing updates to the covariance matrix of Gaussian process models with conjugate likelihoods nested inside Pick-to-Learn based on Schur's complement, provided that the hyperparameters remain fixed across iterations. We begin by introducing the necessary theory of block matrices. We then employ the corresponding results to bring down the complexity of P2L-ES with a GP learner using a priori fixed hyperparameters. Finally, we demonstrate that a celebrated greedy SoD GP approximation, the Informative Vector Machine (Lawrence et al., 2002), can be viewed as a particular instantiation of P2L-ES.

C.1 Block Matrices

Let $M \in \mathbb{C}^{(p+q)\times(p+q)}$ be a block matrix given by

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

where $A \in \mathbb{C}^{p\times p}$, $B \in \mathbb{C}^{p\times q}$, $C \in \mathbb{C}^{q\times p}$ and $D \in \mathbb{C}^{q\times q}$ for some $p, q \in \mathbb{N}$. If A is invertible[10], the Schur complement of A in M is defined as

$$S \triangleq D - CA^{-1}B.$$

When the inverse of S exists, the inverse of M can be written in terms of it as

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix}. \tag{C.1}$$

The Schur complement S of A in M, when it exists, can further be used to write the determinant of M as

$$\det(M) \triangleq \det(A)\det(S).$$

C.2 Fast Fixed-Parameter Training

In this section, we will show how the Schur complement of the covariance matrix can be used to speed up the incremental training of GPs inside Pick-to-Learn. Recall that the complexity of P2L-ES is given by

$$\mathcal{O}\left(\sum_{m=1}^{M} \ell(m) + \alpha(N-m)\right).$$

In Gaussian process models, the learning step corresponds to the minimisation of the negative log marginal likelihood using the latest compressed set in order to find the optimal setting of the hyperparameters $\boldsymbol{\theta}$, while the determination of appropriateness step corresponds to the computation of the posterior on the remaining points. Assuming that the marginal likelihood optimiser is a variant of BFGS as is common in the literature, and given the matrix operations involved in the posterior equations for conjugate likelihood models, the costs of training and inference are

$$\begin{aligned} \ell(m) &\propto k(m^3 + dim(\boldsymbol{\theta})m^2) \\ \alpha(N-m) &\propto m^3 + m^2(N-m) + m(N-m)^2 + m(N-m). \end{aligned}$$

The corresponding costs across all P2L-ES iterations are then given by

$$\begin{aligned} \mathcal{O}\left(\sum_{m=1}^{M} \ell(m)\right) &= \mathcal{O}\left(\sum_{m=1}^{M} k(m^3 + dim(\boldsymbol{\theta})m^2)\right) = \mathcal{O}\left(kM^4 + kdim(\boldsymbol{\theta})M^3\right) \\ \mathcal{O}\left(\sum_{m=1}^{M} \alpha(N-m)\right) &= \mathcal{O}\left(\sum_{m=1}^{M} m^3 + m^2(N-m) + m(N-m)^2 + m(N-m)\right) = \mathcal{O}\left(M^4 + M^3N + M^2N^2\right). \end{aligned}$$

To reduce the complexity of the meta-algorithm, we start by fixing the hyperparameters to their prior settings at the start of training[11], thus making $\ell(m) = 0$. As we will show, fixing the hyperparameters proves sufficient to alleviate the computational bottleneck of Pick-to-Learn with GP learners, as it also allows for the reduction of the cost of the determination of appropriateness step across multiple iterations. The technique facilitating this – based on Schur's complement – has precedent in the sparse GP literature (Rasmussen and Williams, 2006).

From some iteration m to iteration $m+1$, the $|T_m| \times |T_m|$ matrix $[K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]$ being inverted[12], grows by one row and one column to become the $|T_{m+1}| \times |T_{m+1}|$ matrix $[K_{\mathrm{f}_{T_{m+1}}\mathrm{f}_{T_{m+1}}} + \sigma^2 I_{|T_{m+1}|}]$. Knowing the inverse

[10] A more general pseudo-inverse based formulation of Schur's complement also exists (Horn and Johnson, 2012).

[11] Should one wish to do so, one could learn a suitable prior using a small subset $S \subset D$. P2L-ES would then be executed using the remaining $|D| - |S|$ points.

[12] Note that we have dropped the explicit dependence of the kernel matrix on the hyperparameters used in the main body of the paper, since we now assume they remain fixed across iterations.

of the former from iteration m, we can use Schur's complement to efficiently update the inverse of the latter. Consider re-writing the updated kernel matrix as

$$\begin{aligned} K_{\mathrm{f}_{T_{m+1}}\mathrm{f}_{T_{m+1}}} + \sigma^2 I_{|T_{m+1}|} &= \begin{bmatrix} k(X_{T_m}, X_{T_m}) + \sigma^2 I_{T_m} & k(X_{T_m}, X_{T_{m+1}\setminus T_m}) \\ k(X_{T_{m+1}\setminus T_m}, X_{T_m}) & k(X_{T_{m+1}\setminus T_m}, X_{T_{m+1}\setminus T_m}) + \sigma^2 \end{bmatrix} \\ &= \begin{bmatrix} K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|} & K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} \\ K_{\mathrm{f}_{T_{m+1}\setminus T_m}\mathrm{f}_{T_m}} & K_{\mathrm{f}_{T_{m+1}\setminus T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} + \sigma^2 \end{bmatrix}, \end{aligned} \tag{C.2}$$

where the top-left block is the $|T_m| \times |T_m|$ kernel matrix from the previous iteration, the top-right and bottom left blocks correspond to the $|T_m| \times 1$ cross-covariance vector between the new point and all points previously in the compressed set and its $1 \times |T_m|$ transpose respectively, and the bottom-right block is the scalar covariance of the new point with noise added. Since the upper-left block is invertible, its Schur complement in $K_{\mathrm{f}_{T_{m+1}}\mathrm{f}_{T_{m+1}}} + \sigma^2 I_{|T_{m+1}|}$ exists and is given by the following scalar

$$\begin{aligned} S &= K_{\mathrm{f}_{T_{m+1}\setminus T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} + \sigma^2 - K_{\mathrm{f}_{T_{m+1}\setminus T_m}\mathrm{f}_{T_m}} [K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1} K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} \\ &= K_{\mathrm{f}_{T_{m+1}\setminus T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} + \sigma^2 - K^T_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} [K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1} K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}}, \end{aligned}$$

where in the last equation, we have used the symmetry of covariance functions. Given that $[K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1}$ is known from the previous iteration of Pick-to-Learn, we examine the cost of inverting $[K_{\mathrm{f}_{T_{m+1}}\mathrm{f}_{T_{m+1}}} + \sigma^2 I_{|T_{m+1}|}]$ using (C.1). Without the burden of computing the inverse of the previous covariance matrix, the cost of finding S is $\mathcal{O}(|T_m|^2 + |T_m|)$ where the first term comes from computing $K^T_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} [K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1}$ and the second from computing the product of the above with $K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}}$. Since S is a scalar, its inverse can be found in constant time, giving us the bottom-right block. This further gives us the bottom-left block as all involved products have been computed. The top-right block additionally requires computing $[K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1} K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}}$ which incurs an $\mathcal{O}(|T_m|^2)$ cost. Finally, the top-left block requires computing

$$\left([K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1} K_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}\setminus T_m}} \right) \left(K^T_{\mathrm{f}_{T_m}\mathrm{f}_{T_{m+1}}\setminus T_m} [K_{\mathrm{f}_{T_m}\mathrm{f}_{T_m}} + \sigma^2 I_{|T_m|}]^{-1} \right),$$

and since both matrices have been computed for the rest of the blocks, this also has $\mathcal{O}(|T_m|^2)$ cost. Notice that this would not have been possible if we had allowed the hyperparameters to vary across iterations since the top-left block in (C.2) would not have been fixed from the previous iteration. Overall, this means that $\alpha(N-m) \propto m^2$, which reduces the cost over multiple iterations to a computational complexity comparable to standard sparse GP approximations:

$$\mathcal{O}\left(\sum_{m=1}^{M} \ell(m) + \alpha(N-m) \right) = \mathcal{O}\left(\sum_{m=1}^{M} 0 + m^2 \right) = \mathcal{O}(M^3).$$

C.3 The IVM as a P2L Instantiation

In this section, we show that one of the most popular greedy SoD GP approximations, the Informative Vector Machine (IVM) by Lawrence et al. (2002), can be viewed as an instantiation of P2L-ES with a GP learner. Moreover, we demonstrate that even though the IVM's information theoretic criterion function Δ based on the maximum differential entropy score does not directly correspond to an evaluation or safety metric, it can nevertheless be made to yield a guarantee on an interpretable quantity, namely the marginal variance.

Given a remaining set Q and an active set I the IVM uses the following criterion for any $z_j \in Q \setminus I$

$$\Delta(z_j) = H[p(\mathrm{f}_j | I)] - H[p(\mathrm{f}_j | I, z_j)],$$

where $H[p(\mathrm{f}_j | I, z_j)]$ represents the entropy at the site after the inclusion of z_j. Given that for a Gaussian random variable $\boldsymbol{x}$ with variance v, the differential entropy is given by

$$H(\boldsymbol{x}) = \frac{1}{2} \log(2\pi e v),$$

we can write the criterion function as

$$\begin{aligned}\Delta(z_j) &= H[p(\mathrm{f}_j|I)] - H[p(\mathrm{f}_j|I,z_j)] = \frac{1}{2}\log(2\pi e\mathbb{V}[\mathrm{f}_j|y_I]) - \frac{1}{2}\log(2\pi e\mathbb{V}[\mathrm{f}_j|y_I,y_j]) \\ &= \frac{1}{2}\log\left(\frac{\mathbb{V}[\mathrm{f}_j|y_I]}{\mathbb{V}[\mathrm{f}_j|y_I,y_j]}\right) = \frac{1}{2}\log\left(\frac{\mathbb{V}[\mathrm{f}_j|y_I]}{\frac{1}{\mathbb{V}[\mathrm{f}_j|y_I]}+\frac{1}{\sigma^2}}\right) = \frac{1}{2}\log\left(\frac{\mathbb{V}[\mathrm{f}_j|y_I]}{\frac{\mathbb{V}[\mathrm{f}_j|y_I]\sigma^2}{\mathbb{V}[\mathrm{f}_j|y_I]+\sigma^2}}\right) \\ &= \frac{1}{2}\log\left(1+\frac{\mathbb{V}[\mathrm{f}_j|y_I]}{\sigma^2}\right).\end{aligned}$$

Since $\Delta(z_j)$ is monotonic in $\mathbb{V}[\mathrm{f}_j|y_I]$, this is equivalent to selecting the point with the largest variance at each iteration. Under the cost function $C(Q) = \max_{z_j\in Q}\Delta(z_j)$ the termination condition becomes

$$C(Q)\leq\gamma \iff \max_{z_j\in Q}\frac{1}{2}\log\left(1+\frac{\mathbb{V}[\mathrm{f}_j|y_I]}{\sigma^2}\right) \iff \max_{z_j\in Q}\mathbb{V}\left[\mathrm{f}_j|y_I\right]\leq\frac{e^{2\gamma}-1}{\sigma^2}.$$

We are interested in a bound of the opposite form. By setting $\beta \triangleq \frac{e^{2\gamma}-1}{\sigma^2}$ we have

$$\max_{z_j\in Q}\mathbb{V}\left[\mathrm{f}_j|y_I\right]\leq\beta \iff C(Q)\leq\frac{1}{2}\log\left(1+\beta\sigma^2\right).$$

In Pick-to-Learn terms, at any iteration m with compressed set T_m, this induces a total order on D_S given by

$$\begin{aligned}(x_i,y_i)\leq_{\mathcal{GP}}(x_j,y_j) &\iff \mathbb{V}[\mathrm{f}_i|\boldsymbol{y}_{T_m}]\leq\mathbb{V}[\mathrm{f}_j|\boldsymbol{y}_{T_m}] \\ (x_j,y_j)\leq_{\mathcal{GP}}\texttt{Stop} &\iff \mathbb{V}\left[\mathrm{f}_j|\boldsymbol{y}_{T_m}\right]\leq\beta\end{aligned},$$

resulting in a PAC bound on the probability of the marginal variance of a new sample exceeding β. Note that while we originally stated P2L-ES with a GP learner in full generality, with the entire posterior process being calculated at each iteration, the above criterion does not require the predictive mean and further, only makes use of the diagonal of the predictive covariance matrix, further bringing down the computational cost.

D ADDITIONAL MATERIAL FOR SECTION 6

The implementation of Gaussian processes inside Pick-to-Learn is based on GPflow (Matthews et al., 2017), a TensorFlow (Abadi et al., 2016) based library for GPs. The optimisation of hyperparameters is performed using L-BFGS-B with a maximum of 1.5×10^4 iterations. All experiments were run on an 11-core CPU and 14-core GPU Apple M3 Pro MacBook Pro with 18GB of unified memory.

For our regression experiments, we standardise the data independently along each dimension. For our classification experiment we use the MNIST dataset, consisting of 28×28 bitmaps representing handwritten digits. MNIST consists of 60000 training and 10000 testing examples. We specifically consider the task of discriminating handwritten twos against threes used in Seeger (2002a) for which the training and testing set sizes reduce to 12089 and 2042 respectively. We follow the same pre-processing pipeline as in Seeger (2002a). In particular, we cut away 2-pixel margins, average the intensities of the remaining pixels over 3×3 patches and flatten the resulting 8×8 pixel representations into 64-dimensional vectors.

The PAC-Bayesian C bounds[13] of Germain et al. (2015) as well as the greedy SoD GP classification bound of Seeger (2002b) are straightforwardly applied to our GP classification task. To apply Seeger's bound (Seeger, 2002a), which bounds the risk of Gibbs classifiers, we leverage the symmetry of the posterior to transform it into a bound on the risk of the Bayes classifier by multiplying it by a factor of 2, as is common in the literature (Seeger, 2002a; Lorenzen et al., 2019; Masegosa et al., 2020).

Overall, the early stopping thresholds employed were found to be, on average, less than regular P2L.

[13]Implementations taken directly from the official repository: `http://graal.ift.ulaval.ca/majorityvote`