Scalable and adaptive prediction bands with kernel sum-of-squares

Louis Allain[1,2] **Sébastien Da Veiga**[2] **Brian Staber**[1]
[1]Safran Tech, Digital Sciences & Technologies,78114 Magny-Les-Hameaux, France
[2]Univ Rennes, Ensai, CNRS, CREST - UMR 9194, F-35000 Rennes, France
{louis.allain,brian.staber}@safrangroup.com
sebastien.da-veiga@ensai.fr

Abstract

Conformal Prediction (CP) is a popular framework for constructing prediction bands with valid coverage in finite samples, while being free of any distributional assumption. A well-known limitation of conformal prediction is the lack of adaptivity, although several works introduced practically efficient alternate procedures. In this work, we build upon recent ideas that rely on recasting the CP problem as a statistical learning problem, directly targeting coverage and adaptivity. This statistical learning problem is based on reproducible kernel Hilbert spaces (RKHS) and kernel sum-of-squares (SoS) methods. First, we extend previous results with a general representer theorem and exhibit the dual formulation of the learning problem. Crucially, such dual formulation can be solved efficiently by accelerated gradient methods with several hundreds or thousands of samples, unlike previous strategies based on off-the-shelf semidefinite programming algorithms. Second, we introduce a new hyperparameter tuning strategy tailored specifically to target adaptivity through bounds on test-conditional coverage. This strategy, based on the Hilbert-Schmidt Independence Criterion (HSIC), is introduced here to tune kernel lengthscales in our framework, but has broader applicability since it could be used in any CP algorithm where the score function is learned. Finally, extensive experiments are conducted to show how our method compares to related work. All figures can be reproduced with the accompanying code at gitlab.com/drti/ksos-bands.

1 Introduction

In many applications, machine learning regression models require a trustworthy uncertainty quantification in their predictions. This is especially true for high-stakes applications such as design optimization, non-destructive testing, medical diagnostics, autonomous vehicles, or financial forecasting, where decisions based on model predictions can have significant impacts. Having a reliable uncertainty quantification is thus fundamental. Several machine learning models come with uncertainty quantification in their predictions, such as Gaussian Processes [Rasmussen and Williams, 2005], Random Forests [Breiman, 2001] or Bayesian Neural Networks [Wang and Yeung, 2020], among many others, but these models generally provide inaccurate prediction bands: coverage guarantees typically hold asymptotically or with strong distributional assumptions. In practice, however, especially for high-stakes decisions, we should at least provide marginal coverage guarantees that hold in finite sample and without making any distributional assumptions on the data. In addition, a desirable feature is adaptivity: we would like prediction bands to be wide when either the model lacks confidence or if the variability in the data is high, and narrow when both the model is confident and the variability is low. Having adaptive prediction bands means having an uncertainty quantification that is informative on either the performance of the model or the variability of the data, which is key in crucial applications.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Conformal Prediction (CP) (see, e.g., [Gammerman et al., 1998, Papadopoulos et al., 2002, Shafer and Vovk, 2008] or [Angelopoulos and Bates, 2023] for a modern introduction) has been designed from the ground up to be an uncertainty quantification statistical framework that provides marginal coverage guarantees in finite sample while being distribution-free. In particular, the split conformal procedure proposed by Papadopoulos et al. [2002] is especially easy to implement. CP is becoming widely used in many different applications (see [Balasubramanian et al., 2014, Vazquez and Facelli, 2022] and references therein), but unfortunately, by construction, standard CP does not provide adaptive prediction bands. A lot of research has been done in this direction, which we will review later.

In parallel, for specific statistical learning problems, Marteau-Ferey et al. [2020] introduced a new kernel framework known as kernel sum-of-squares (SoS), tailored specifically to estimate non-negative functions. Their key idea is to characterize such functions of interest by a linear positive semidefinite Hermitian operator, which admits a finite-dimensional representation through a representer theorem. Since then, it has been leveraged for non-convex optimization [Rudi et al., 2025], estimation of optimal transport distances [Vacher et al., 2021], modeling of probability densities [Rudi and Ciliberto, 2021] and PSD-constrained functions [Muzellec et al., 2022]. Very recently, kernel SoS was also identified as a powerful framework for constructing more adaptive prediction bands by Liang [2022] and Fan et al. [2024]. They were the first to propose to build prediction bands as solutions of such a learning problem, where adaptive coverage is targeted with additional constraints. This point of view is the one we adopt and generalize in this paper.

Outline and contributions. We start by presenting CP in Section 2 as well as recent variants developed to improve adaptivity. We also introduce the kernel SoS framework, since our proposed method extensively relies on it. In Section 3 we introduce our approach that learns a CP score function by solving a statistical optimization problem with several ingredients: an objective function controlling both the width and the regularity of the prediction bands, and constraints for coverage. We also discuss a new criterion dedicated to the tuning of kernel lengthscales, with local coverage as an objective. In Section 4, we finally conduct extensive experiments to compare our method to other conformal prediction methods that provide adaptive bands.

Our contributions are as follows:

- We generalize the previously introduced kernel SoS point of view for prediction bands and precisely analyze the contribution and practical effect of each term in the objective function,
- We provide a representer theorem that makes the problem numerically tractable,
- We derive a dual formulation of this problem and propose an accelerated gradient algorithm to enable faster computation on large datasets, unlike previous work that was limited to small datasets,
- We introduce a new criterion to tune kernel hyperparameters based on the Hilbert-Schmidt Independence Criterion (HSIC), which is also applicable to any other CP method. In particular, we provide both theoretical and empirical evidence of the effectiveness of this metric to achieve better adaptivity.

2 Conformal prediction and kernel sum-of-squares

2.1 Conformal prediction

Split conformal prediction. CP was introduced by Gammerman et al. [1998], with the so-called full-variant, but we focus here on the split variant, introduced by Papadopoulos et al. [2002]. Suppose we have a training dataset $\mathcal{D}_N = \{(X_i, Y_i)\}_{i=1}^N$ from a pair $(X, Y) \sim P_{XY}$ where $X \in \mathcal{X} \subset \mathbb{R}^d$ and $Y \in \mathcal{Y} \subset \mathbb{R}$. This dataset is split in two parts: a *pre-training* dataset $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ and a *calibration* one $\mathcal{D}_m = \{(X_i, Y_i)\}_{i=1}^m$ with $N = n + m$.

The pre-training dataset $\mathcal{D}_n$ is used to first fit a predictive model $\widehat{m}_n(\cdot)$, which can be any machine learning algorithm. The second step consists in computing performance scores associated to $\widehat{m}_n$ on the hold-out calibration dataset $\mathcal{D}_m$. The usual score in the literature is defined as the absolute errors $S(X_i, Y_i) := S_i = |Y_i - \widehat{m}_n(X_i)|$ for $i \in \mathcal{D}_m$, which are used to compute the quantile $\widehat{q}_\alpha$ of the set $\{S_i\}_{i \in \mathcal{D}_m}$ with an adjusted level $\lceil (1-\alpha)(m+1) \rceil / m$, where α is the desired error rate. Finally,

for a new observation X_{N+1}, the split CP prediction bands are $\widehat{C}_N(X_{N+1}) = [\widehat{m}_n(X_{N+1}) \pm \widehat{q}_\alpha]$, which satisfy the marginal coverage

$$\mathbb{P}\left(Y_{N+1} \in \widehat{C}_N(X_{N+1})\right) \geq 1 - \alpha \tag{1}$$

for any N as long as $(X_1, Y_1), \ldots, (X_N, Y_N), (X_{N+1}, Y_{N+1})$ are exchangeable. Unfortunately, as underlined by Romano et al. [2019], these prediction bands cannot be adaptive since they have constant width $2\widehat{q}_\alpha$. The research direction in recent years has been to adjust CP procedures to target more adaptive bands.

The quest for adaptivity. Historically the first idea was to change the score function by rescaling the scores with an estimate of the variability $\widehat{\sigma}_n(\cdot) \geq 0$. The new scores are thus defined as $S_i = |Y_i - \widehat{m}_n(X_i)|/\widehat{\sigma}_n(X_i),\ i \in \mathcal{D}_m$ with prediction bands $\widehat{C}_N(X_{N+1}) = [\widehat{m}_n(X_{N+1}) \pm \widehat{q}_\alpha \widehat{\sigma}_n(X_{N+1})]$. Lei and Wasserman [2014] first proposed to use an estimate of the conditional mean absolute deviation for $\widehat{\sigma}_n(\cdot)$. Another sensible choice is to scale the scores by an estimate of the standard deviation, as was suggested for several machine learning models like Gaussian Processes, Random Forests and Bayesian Neural Networks [Johansson et al., 2014, Papadopoulos, 2024, Jaber et al., 2025]. However, such scaling functions are rarely estimated in a goal-oriented way, without quantitative and explicit consideration for adaptive coverage. This often leads in practice to poorly adaptive prediction bands. In a parallel line of work, the popular Conformalized Quantile Regression (CQR) [Romano et al., 2019] proposes to change the score function by leveraging quantile regression. Instead of using an interval built around an estimate $\widehat{m}_n(\cdot)$ of the regression function, they rely on estimates $\widehat{q}_n^{\alpha_{lower}}(\cdot)$ and $\widehat{q}_n^{\alpha_{upper}}(\cdot)$ of the conditional quantiles, and build an interval $\widehat{C}_N(X_{N+1}) = \left[q_n^{\alpha_{lower}}(X_{N+1}) - \widehat{q}_\alpha, q_n^{\alpha_{upper}}(X_{N+1}) + \widehat{q}_\alpha\right]$ where now $\widehat{q}_\alpha$ is the adjusted quantile of the set $\{\max\left(q_n^{\alpha_{lower}}(X_i) - Y_i, Y_i - q_n^{\alpha_{upper}}(X_i)\right),\ i \in \mathcal{D}_m\}$. In other words, the score function is chosen as $S(X, Y) = \max\left(q_n^{\alpha_{lower}}(X) - Y, Y - q_n^{\alpha_{upper}}(X)\right)$. By design, the CQR model is adaptive and generally provides sensible prediction bands. However, it suffers from two practical limitations: (a) decision-making people usually prefer a point estimate with an interval *around* this estimate and (b) quantile regression in a small data regime can be quite challenging. Also note that a regularized version of CQR tailored to target test-conditional coverage was recently proposed by Feldman et al. [2021].

Another line of work consists in modifying the calibration step. For example, Guan [2023] and Hore and Barber [2024] propose to weight the scores when computing the adjusted level quantile, where the weights depend on the test point X_{N+1}: this directly implies that the quantile changes with X_{N+1}, and as a result the prediction bands are adaptive. More precisely, given a kernel $H(\cdot, \cdot)$ that defines a density $H(x, \cdot)$ for all $x \in \mathcal{X}$, sample $\widetilde{X}_{N+1}$ from $H(X_{N+1}, \cdot)$. The quantile $\widehat{q}_\alpha(X_{N+1}, \widetilde{X}_{N+1})$ is computed on the empirical distribution $\sum_{i=1}^m \widetilde{w}_i \delta_{S_i} + \widetilde{w}_{N+1}\delta_{+\infty}$, where the weights are computed as $\widetilde{w}_i = H(X_i, \widetilde{X}_{N+1})/(\sum_{j=1}^m H(X_j, \widetilde{X}_{N+1}) + H(X_{N+1}, \widetilde{X}_{N+1}))$. Although appealing, such modifications to the score function have some practical shortcomings. First, computing different weights for all test points can be computationally demanding during inference. Second, the method suffers in practice from the randomization induced by the sampling of $\widetilde{X}_{N+1}$. To overcome this issue Hore and Barber [2024] propose the m-RLCP methods that averages the predictions bands over m sampling of $\widetilde{X}_{N+1}$. However this leads to a marginal coverage equal to $1 - 2\alpha$ and the computational cost increases significantly. Finally, and perhaps more importantly, the kernel $H(\cdot, \cdot)$ involved in the definition of the weights depends on a bandwidth hyperparameter that must be tuned. This choice has a strong impact on the shape of the prediction bands as shown in Hore and Barber [2024]. We will come back to this point later since our framework shares the same characteristic. Finally, another reweighting method based instead on Jackknife+ was proposed by Deutschmann et al. [2024].

2.2 Kernel sum-of-squares

Since our proposed method is based on kernel SoS for positive functions, we give a brief overview following Marteau-Ferey et al. [2020].

Let $\mathcal{H}$ be a RKHS with associated kernel k and $\phi\colon \mathcal{X} \to \mathcal{H}$ one of its feature map such that $k(x, x') = \phi(X)^\top \phi(X)$. Let $\mathcal{S}(\mathcal{H})$ be the set of bounded Hermitian linear operator from $\mathcal{H}$ to $\mathcal{H}$. For $\mathcal{A} \in \mathcal{S}(\mathcal{H})$, we write $\mathcal{A} \succeq 0$ when $\mathcal{A}$ is a positive semi-definite (PSD) operator, and $\mathcal{S}_+(\mathcal{H})$ the

set of such PSD operators. For all $x \in \mathcal{X}$, we define $f_{\mathcal{A}}(X) = \phi(X)^\top \mathcal{A}\phi(X)$, $\mathcal{A} \in \mathcal{S}_+(\mathcal{H})$ which is non-negative by construction. Kernel SoS refers to a statistical learning problem where the unknown nonparametric function is constrained to be non-negative and obtained as the solution of

$$\inf_{\mathcal{A}\succeq 0} L(f_{\mathcal{A}}(X_1), \dots, f_{\mathcal{A}}(X_n)) + \lambda_1 \|\mathcal{A}\|_\star + \lambda_2 \|\mathcal{A}\|_{\mathrm{F}}^2 \tag{2}$$

where $\|\mathcal{A}\|_\star$ and $\|\mathcal{A}\|_F$ denote the nuclear norm and the Frobenius norm of operator $\mathcal{A}$, respectively. Interestingly, Marteau-Ferey et al. [2020] show a representer theorem for Equation (2), which makes kernel SoS computationally tractable in finite-dimension and is recalled below.

Theorem 1 (Marteau-Ferey et al. [2020]) *Assume $L\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ to be a lower semi-continuous and bounded below loss function. Equation* (2) *admits a solution $\mathcal{A}^\star$ which can be written $\mathcal{A}^\star = \sum_{i,i=1}^n B_{ij}^\star \phi(X_i)\phi(X_j)^\top$ for some matrix $\mathbf{B}^\star \in \mathbb{R}^{n\times n}, \mathbf{B}^\star \succeq 0$. Furthermore $\mathcal{A}^\star$ is unique if L is convex and $\lambda_2 > 0$. The corresponding non-negative function is given by $f_{\mathcal{A}^\star}(X) = \sum_{i,i=1}^n B_{ij}^\star k(X_i, X)k(X_j, X)$.*

Theorem 1 provides a finite-dimensional equivalent problem which involves an unknown PSD matrix $\mathbf{B}$. But Marteau-Ferey et al. [2020] also propose an equivalent formulation: considering $\mathbf{K}$ the kernel matrix with elements $K_{ij} = k(X_i, X_j)$ and $\mathbf{V}$ its upper Cholesky decomposition, we can define $\mathbf{\Phi}(X) = \mathbf{V}^{-\top}\mathbf{k}_X$ with $\mathbf{k}_X = (k(X, X_i))_{i=1,\dots,n}$ and $\tilde{f}_{\mathbf{A}}(X) = \mathbf{\Phi}(X)^\top \mathbf{A}\mathbf{\Phi}(X)$. With these notations, the following proposition shows that we obtain the same solution if we optimize the PSD matrix $\mathbf{A}$ instead of $\mathbf{B}$.

Proposition 1 (Marteau-Ferey et al. [2020]) *Under the assumptions of Theorem 1, the following problem has at least one solution, which is unique if $\lambda_2 > 0$ and L is convex:*

$$\inf_{\mathbf{A}\succeq 0} L(\tilde{f}_{\mathbf{A}}(X_1), \dots, \tilde{f}_{\mathbf{A}}(X_n)) + \lambda_1 \|\mathbf{A}\|_\star + \lambda_2 \|\mathbf{A}\|_F^2. \tag{3}$$

For any given solution $\mathbf{A}^\star \in \mathbb{R}^{n\times n}$ of Equation (3)*, the function $\tilde{f}_{\mathbf{A}^\star}$ is also solution of Equation* (2)*.*

Although such a result may appear of minor impact, the $\mathbf{A}$ formulation actually yields significant computational savings in practice, as we illustrate in our numerical experiments (see Appendix B.4).

Remark 1 *Operator $\mathcal{A}$ is PSD, hence it admits an eigendecomposition $\mathcal{A} = \sum_{l\geq 0} \lambda_l u_l \otimes u_l$ with $\lambda_l \geq 0$ and $u_l \in \mathcal{H}$. By the reproducing property, we have $f_{\mathcal{A}}(X) = \sum_{l\geq 0} \lambda_l u_l(X) u_l(X)^\top$. Hence, $f_{\mathcal{A}}(X)$ is an infinite sum of squared functions in $\mathcal{H}$. Equivalently in finite dimension, for a PSD matrix $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ we have $f_{\mathbf{B}}(X) = \mathbf{k}_X^\top \mathbf{U}\mathbf{D}\mathbf{U}^\top \mathbf{k}_X = \sum_{l=1}^n \lambda_l (\sum_{i=1}^n u_{il} k(X, X_i))^2$. Thus $f_{\mathbf{B}}$ is a function defined as a linear combination of squared functions from $\mathcal{H}$.*

3 Regularized kernel SoS for adaptive prediction bands

We now come back to our initial problem of building adaptive prediction bands. To do so, we focus on the split CP setting with two i.i.d. datasets $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ and $\mathcal{D}_m = \{(X_i, Y_i)\}_{i=1}^m$, and consider estimating a score function in a specific supervised learning problem to achieve better adaptivity.

3.1 Learning the scores through an optimization problem

A general framework for learning score functions was recently introduced by Xie et al. [2024a]: given a task-specific loss (e.g. conditional coverage or minimum interval width), an optimized score function is obtained via a boosting algorithm. In this work, the score function $S(X, Y) = \max\left(\mu_1(X) - Y, Y - \mu_2(X)\right)/\sigma(X)$ is inspired by CQR and is parameterized by three unknown functions (μ_1, μ_2, σ) such that $\mu_1(\cdot) \leq \mu_2(\cdot)$ and $\sigma(\cdot) \geq 0$, which are iteratively optimized during boosting rounds. From a practical viewpoint however, the constraints on these functions are not inherently accounted for in the boosting algorithm.

We advocated before the use of prediction intervals built around a point estimate, which corresponds to the particular case $m = \mu_1 = \mu_2$ and yields a score function $S(X, Y) = |Y - m(X)|/\sigma(X)$, or equivalently $S(X, Y) = (Y - m(X))^2/f(X)$ with $f(\cdot) := \sigma(\cdot)^2$ since only the score quantiles are

involved in the final prediction interval: we thus recover the rescaled conformal score setting where we *learn* the rescaling function $f(\cdot)$, with an additional non-negativity constraint. The kernel SoS framework is thus a natural candidate for this learning task.

Kernel SoS formulation for prediction intervals. We introduce two RKHSs $\mathcal{H}^m$ and $\mathcal{H}^f$ associated to kernels k^m with lengthscales θ^m and k^f with lengthscales θ^f, respectively. The regression function m will be estimated in the RKHS $\mathcal{H}^m$ while the non-negative scaling function f will be estimated using the kernel SoS framework in the RKHS $\mathcal{H}^f$. The first step is to derive our proposed infinite-dimensional learning problem from the properties that we impose on prediction bands:

$$\inf_{m\in\mathcal{H}^m,\ \mathcal{A}\in\mathcal{S}_+(\mathcal{H}^f)} \quad \frac{a}{n}\sum_{i=1}^{n}\left(Y_i - m(X_i)\right)^2 + \frac{b}{n}\sum_{i=1}^{n} f_{\mathcal{A}}(X_i) + \lambda_1\|\mathcal{A}\|_\star + \lambda_2\|\mathcal{A}\|_F^2 \tag{4}$$

$$\text{s.t.} \quad f_{\mathcal{A}}(X_i) \geq \left(Y_i - m(X_i)\right)^2,\ i \in [n]\,, \tag{5}$$

$$\|m\|_{\mathcal{H}^m}^2 \leq s. \tag{6}$$

In this problem, we propose to include several key components:

1. Accurate mean estimation (first term in (4)) with regularity penalty (6), as in standard kernel ridge regression [Schölkopf and Smola, 2002],
2. Prediction intervals with minimum mean width (second term in (4)) and an additional regularity penalty (third and fourth terms in (4)),
3. 100% coverage on pre-training data (5), later calibrated on the calibration dataset.

Minimizing the nuclear norm with coverage constraints was originally proposed by Liang [2022], and minimizing the mean width was later proposed by Fan et al. [2024]. Our proposition essentially differs from their work for a broader and more efficient practical applicability: (a) we place ourselves in the split CP setting, whereas Liang [2022] and Fan et al. [2024] propose different calibration procedures that are harder to implement in practice, with theoretical coverage guarantees that depend on hyperparameters (see Appendix A.3), (b) we rely on a dual formulation which can handle several hundreds of samples, (c) we propose a goal-oriented tuning strategy for θ^f, and (d) our proposal is more general and we give better understanding of why this targets adaptivity[1]. In particular, adaptivity can actually be controlled through the complexity of the scaling function f, in three different ways:

1. The sparsity in the linear combination. This can be controlled by the nuclear norm $\|\mathcal{A}\|_\star$, which acts as a lasso-type penalty (see Recht et al. [2010]),
2. The ℓ_2 norm of the coefficients in the linear combination, which is equal to the Frobenius norm $\|\mathcal{A}\|_F$ and is similar to a ridge penalty,
3. The RKHS $\mathcal{H}^f$ which impacts the functions $u_l(\cdot)$, and notably the lengthscales θ^f.

To further illustrate point 3., if we take $k^f(x, x') = \langle x, x'\rangle$ (RHKS of linear functions), the rescaling function will be a second-order polynomial (thus not complex), while with a Gaussian kernel if $\theta^f \to +\infty$ the prediction intervals will be constant (not adaptive enough) and if θ^f is small they will be very wiggly (too adaptive). We thus see that in between, we can target better adaptivity: we propose in Section 3.2 a new goal-oriented criterion related to local coverage to tune the lengthscales θ^f. Note also that $1-\alpha$ instead of 100% coverage can be considered in the constraints, but this unfortunately leads to a non-convex optimization problem [Braun et al., 2025].

Before discussing in detail the choice of our problem hyperparameters, we first derive a representer theorem which makes the optimization numerically tractable.

Theorem 2 (Representer theorem) *Let $(a, b, s, \lambda_1) \in \mathbb{R}_+^4$ and $\lambda_2 > 0$. Then Equation (4) admits a unique solution $(m^\star, f_{\mathbf{A}^\star})$ of the form $m^\star(X) = \sum_{i=1}^n \gamma_i^\star k^m(X_i, X) = \boldsymbol{\gamma}^{\star\top}\mathbf{k}_X^m$ and $f_{\mathbf{A}^\star}(X) = \boldsymbol{\Phi}(X)^\top\mathbf{A}^\star\boldsymbol{\Phi}(X)$ for some vector $\boldsymbol{\gamma}^\star \in \mathbb{R}^n$ and matrix $\mathbf{A}^\star \in \mathbb{R}^{n\times n}$, $\mathbf{A}^\star \succeq 0$.*

The detailed proof, based on Marteau-Ferey et al. [2020] and Muzellec et al. [2022], can be found in Appendix A.1. This representer theorem leads to the following tractable semi-definite programming

[1]This lack of understanding is for example pointed out by Liang et al. [2024].

(SDP) problem:

$$\begin{aligned} \inf_{\boldsymbol{\gamma}\in\mathbb{R}^n,\, \mathbf{A}\in\mathbb{S}_+^n} \quad & \frac{a}{n}\sum_{i=1}^n \left(Y_i - \boldsymbol{\gamma}^\top \mathbf{k}_{X_i}^m\right)^2 + \frac{b}{n}\sum_{i=1}^n \tilde{f}_{\mathbf{A}}(X_i) + \lambda_1 \|\mathbf{A}\|_\star + \lambda_2 \|\mathbf{A}\|_F^2 \\ \text{s.t.} \quad & \tilde{f}_{\mathbf{A}}(X_i) \geq \left(Y_i - \boldsymbol{\gamma}^\top \mathbf{k}_{X_i}^m\right)^2,\ i \in [n]\,, \\ & \boldsymbol{\gamma}^\top \mathbf{K}^m \boldsymbol{\gamma} \leq s. \end{aligned} \tag{7}$$

In practice, such SDP problem can be solved using off-the-shelf solvers like `SCS` [O'Donoghue et al., 2016], as was advocated by Liang [2022] and Fan et al. [2024]. However, our numerical experiments show that this strategy does not scale past a few hundreds pre-training samples: this is thus a severe practical limitation which, in our opinion, heavily weakens the kernel SoS point of view. To circumvent this major issue, we rely instead on a dual formulation of Equation (7).

Dual formulation. First note that this formulation is possible only when $\lambda_2 > 0$ (thus excluding Liang [2022] framework) and that it consists of an optimization problem over $(n+1)$ variables rather than $(n + n \times n)$ variables.

Proposition 2 (Dual formulation) *Let $(a, b, s, \lambda_1) \in \mathbb{R}_+^4$, $\lambda_2 > 0$ and $\Delta := \mathbb{R}_+^{n+1}$. Equation (7) admits a dual formulation of the form*

$$\sup_{(\boldsymbol{\Gamma},\theta)\in\Delta} \mathbf{r}(\boldsymbol{\Gamma},\theta)^\top \mathrm{Diag}(\boldsymbol{\Gamma}_{\mathbf{a}})\mathbf{r}(\boldsymbol{\Gamma},\theta) + \theta(\boldsymbol{\gamma}(\boldsymbol{\Gamma},\theta)^\top \mathbf{K}^m \boldsymbol{\gamma}(\boldsymbol{\Gamma},\theta) - s) - \Omega^\star(\mathbf{V}\mathrm{Diag}(\boldsymbol{\Gamma}_{-\mathbf{b}})\mathbf{V}^\top) \tag{8}$$

where $\mathbf{r}(\boldsymbol{\Gamma},\theta) = \mathbf{Y} - \mathbf{K}^m\boldsymbol{\gamma}(\boldsymbol{\Gamma},\theta)$, $\boldsymbol{\gamma}(\boldsymbol{\Gamma},\theta) = \mathbf{C}\left(\boldsymbol{\Gamma},\theta\right)^{-1}\mathrm{Diag}(\boldsymbol{\Gamma}_a)\mathbf{Y}$, $\mathbf{C}\left(\boldsymbol{\Gamma},\theta\right) = \mathrm{Diag}\left(\boldsymbol{\Gamma}_a\right)\mathbf{K}^m + \theta\mathbf{I}_n$, $\Omega^\star(\mathbf{B}) = \frac{1}{4\lambda_2}\|[\mathbf{B} - \lambda_1\mathbf{I}_n]_+\|_F^2$ and $\forall x \in \mathbb{R},\ \mathrm{Diag}\left(\boldsymbol{\Gamma}_x\right) := \mathrm{Diag}\left(\boldsymbol{\Gamma}\right) + \frac{x}{n}\mathbf{I}_n$. Moreover, if $(\widehat{\boldsymbol{\Gamma}}, \widehat{\theta})$ is solution of Equation (8), a solution of Equation (4) can be retrieved as

$$\widehat{\boldsymbol{\gamma}} = \left(\mathrm{Diag}(\widehat{\boldsymbol{\Gamma}}_{\mathbf{a}})\mathbf{K}^m + \widehat{\theta}\mathbf{I}_n\right)^{-1}\mathrm{Diag}(\widehat{\boldsymbol{\Gamma}}_{\mathbf{a}})\mathbf{Y} \quad \textit{and} \quad \widehat{\mathbf{A}} = \frac{1}{2\lambda_2}\left[\mathbf{V}\mathrm{Diag}(\widehat{\boldsymbol{\Gamma}}_{-\mathbf{b}})\mathbf{V}^\top - \lambda_1\mathbf{I}_n\right]_+$$

where $[\mathbf{A}]_+$ denotes the positive part of $\mathbf{A}$[2] .

A detailed proof is given in Appendix A.2 as well as the dual analytical gradient. Interestingly, this dual formulation can be efficiently optimized using accelerated gradient-ascent algorithms [Ruder, 2017, Xie et al., 2024b], see Figure 3 for an illustration of the numerical speed-up.

Final prediction bands. Solving the dual formulation yields estimates $\widehat{m}(X) = \widehat{\boldsymbol{\gamma}}^\top \mathbf{k}_X^m$ and $\widehat{f}_{\mathbf{A}}(X) = \boldsymbol{\Phi}(X)^\top \widehat{\mathbf{A}}\boldsymbol{\Phi}(X)$, from which we derive the estimated score function $S(X,Y) = |Y - \widehat{m}(X)|/\sqrt{\widehat{f}_{\mathbf{A}}(X)}$. Following standard split CP, our prediction interval is given by

$$\widehat{C}_N(X_{N+1}) = \left[\widehat{m}(X_{N+1}) - \widehat{q}_\alpha\sqrt{\widehat{f}_{\mathbf{A}}(X_{N+1})}, \widehat{m}(X_{N+1}) + \widehat{q}_\alpha\sqrt{\widehat{f}_{\mathbf{A}}(X_{N+1})}\right] \tag{9}$$

where $\widehat{q}_\alpha$ is the adjusted quantile of the estimated score function on the calibration set. By design (see Appendix A.3), they satisfy the same marginal coverage guarantee as split CP, as formalized below.

Proposition 3 *If the samples $(X_1, Y_1), \ldots, (X_N, Y_N), (X_{N+1}, Y_{N+1})$ are exchangeable, then the prediction interval (9) satisfies the marginal coverage (1).*

Extensions. Similarly to Liang [2022], if a point estimate $\widehat{m}(\cdot)$ is available beforehand from any machine learning model trained on a separate dataset, our dual formulation can be easily modified. More importantly, it is also straightforward to consider non-symmetric intervals by estimating two functions $\widehat{f}_{\mathbf{A}^{\text{low}}}(\cdot)$ and $\widehat{f}_{\mathbf{A}^{\text{up}}}(\cdot)$, at the cost of increasing the dual problem dimension to $(2n+1)$, see Appendix B.4. We also strongly believe that incorporating constraints of the form $\mu_1(\cdot) \leq \mu_2(\cdot)$ as proposed by Xie et al. [2024a] is possible with kernel SoS, by taking inspiration from Aubin-Frankowski and Rudi [2024]. Kernel SoS is also complementary to Hore and Barber [2024], as our learned score can be post-processed with their approach. Finally, since we aim for 100% coverage, our framework may lack robustness with respect to outliers: however, we can take advantage of our preliminary estimate $m_{\text{GP}}(\cdot)$ (see next section) to filter out samples with large residuals.

[2]For a PSD matrix $\mathbf{A}$ with eigendecomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, its positive part is defined as $[\mathbf{A}]_+ = \mathbf{U}\max(0,\mathbf{D})\mathbf{U}^\top$.

3.2 Hyperparameter tuning

First, the hyperparameters θ^m and s related to the regression function $m(\cdot)$ are fixed with a preliminary estimate $m_{\text{GP}}(\cdot)$ obtained from Gaussian Process (GP) regression with kernel k^m: θ^m is optimized by maximum likelihood and $s = \|m_{\text{GP}}\|^2_{\mathcal{H}^m}$. Second, our numerical experiments in Appendix B.1 show that λ_2 has a very small impact on the shape of the intervals, as long as $\lambda_2 > 0$. Not only does it make the initial problem strongly convex and makes the dual formulation possible, but it also facilitates numerical optimization. In the following, we thus fix it at $\lambda_2 = 1$.

Hyperparameters a, b, λ_1 and θ^f necessitate more attention. We observe numerically that a has very little influence on mean squared-errors when the noise is symmetric (see Appendix B.1). This phenomenon was commented in Fan et al. [2024], where they argue that the coverage constraints have the additional effect to reduce mean squared-errors, but here we insist that this is true only for symmetric noise. Our experiments confirm this intuition, we simply set $a = 0$ for the symmetric case. Furthermore, they also show that b and λ_1 have highly different influence. This may seem surprising, given how they interact in the dual formulation through the last term in (8). A potential explanation is that b controls a data-driven term, unlike λ_1, and the latter does not have a strong impact on the optimal solution provided it is positive. As a result, we choose to set $\lambda_1 = 1$. On the contrary, b has the expected behavior, in the sense that higher values yield narrower intervals, at the cost of increasing the nuclear norm, that is the function complexity. But the last hyperparameter θ^f also controls such complexity: we thus expect a compensation between these two hyperparameters through an interaction. Numerical experiments in Section 4 actually confirm this phenomenon and show that as long as b is sufficiently large (*e.g.* 10 or 100, we recommend to test these two values), it is possible to tune only θ^f to reach the same level of adaptivity. θ^f thus requires specific consideration, since it has the largest impact when b is fixed. This is the one we focus on below.

Goal-oriented criterion. Adaptivity is tightly related to local coverage $\mathbb{P}(Y_{N+1} \in \widehat{C}(X_{N+1}) \mid X_{N+1} = x) \geq 1 - \alpha$, which is impossible to achieve exactly in a distribution-free setting [Vovk, 2012, Barber et al., 2021]. We focus ourselves on a weaker version, where we condition on X being in a small neighborhood $\omega_X \in \mathcal{F}_X$ from the event space $\mathcal{F}_X$ such that for all $x \in \mathcal{X}, \mathbb{P}(x \in \omega_X) \geq \delta$: $\mathbb{P}(Y_{N+1} \in \widehat{C}(X_{N+1}) | X_{N+1} \in \omega_X) \geq 1 - \alpha$. Recently, Deutschmann et al. [2024] showed that such coverage with split CP can be controlled with the mutual information (MI) between the inputs and the score function, namely

$$\mathbb{P}(Y \in \widehat{C}_{\mathcal{D}_N}(X) | \mathcal{D}_N, X \in \omega_X) \geq 1 - \alpha - \frac{1}{\delta}\sqrt{1 - \exp(-\text{MI}(X, S_{\mathcal{D}_n}(X, Y)))}. \tag{10}$$

Note that contrary to Deutschmann et al. [2024], we explicitly write the coverage conditionally on the training set $\mathcal{D}_N$. Inherently this bound is uninformative for low-probability sets, but interestingly we can counterbalance this effect by choosing a score function which is as independent as possible of the inputs in order to get $\text{MI}(X, S(X, Y))$ close to 0, see Deutschmann et al. [2024]. However, this implies computing MI for a random vector in dimension d: from a practical perspective, MI suffers from the curse of dimensionality and rapidly becomes numerically unstable. Instead, for a re-scaled score function, we show that a similar bound holds with MI between one-dimensional variables, which is more robust. Furthermore, using recent inequalities between the total variation distance and the maximum mean discrepancy [Wang and Tay, 2023], we also extend our result to replace MI with HSIC [Gretton et al., 2005], for which we observe in practice much improved numerical stability.

Proposition 4 *Let $\widehat{C}_{\mathcal{D}_N}$ be the prediction intervals built from a score function $S(X, Y) = |Y - m(X)|/\sqrt{f(X)}$ through split CP with $\mathcal{D}_N = \mathcal{D}_n \cup \mathcal{D}_m$. Then for any ω_X in $\mathcal{F}_X$ such that $\mathbb{P}(X \in \omega_X) \geq \delta$, denoting $p_{\mathcal{D}_N} = \mathbb{P}(Y_{N+1} \in \widehat{C}_{\mathcal{D}_N}(X_{N+1}) | \mathcal{D}_N, X_{N+1} \in \omega_X)$ we have:*

$$p_{\mathcal{D}_N} \geq 1 - \alpha - \frac{1}{\delta}\sqrt{1 - \alpha_1 \exp(\text{MI}(r_{\mathcal{D}_n}(X_{N+1}, Y_{N+1}), \widehat{f}_{\mathcal{D}_n}(X_{N+1})))} \tag{11}$$

where α_1 does not depend on $f(\cdot)$ and $r_{\mathcal{D}_n}(X, Y) = |Y - \widehat{m}_{\mathcal{D}_n}(X)|$. In addition, we have

$$p_{\mathcal{D}_N} \geq 1 - \alpha - \frac{1}{\delta}\sqrt{1 - \frac{\alpha_1}{1 - \alpha_2 \text{HSIC}(r_{\mathcal{D}_n}(X_{N+1}, Y_{N+1}), \widehat{f}_{\mathcal{D}_n}(X_{N+1}))}} \tag{12}$$

where α_2 only depends on the kernel used for HSIC, which must be characteristic.

To target local coverage, we can then maximize $\text{HSIC}(r(X,Y), f(X))$: in our kernel SoS procedure, this means that θ^f can be tuned efficiently according to this criterion. For HSIC estimation, we need samples X_{N+1} independent of $\mathcal{D}_n$ and thus rely on a cross-validation procedure. Finally, to handle cases where the noise is homoscedastic, we implement as a last step a HSIC test of independence: if the p-value is large, we set θ^f to an arbitrary large value. See Appendix A.4 for the proof and B.2 for implementation details.

Remark 2 *We believe that Proposition 4 is also of interest beyond kernel SoS, in the sense that it provides a principled way to tune hyperparameters in all score functions, such as in Hore and Barber [2024] or Braun et al. [2025] for example. However, its true potential would lie in generalizing Equation* (12) *to other score functions, which is left as future work. It could also be directly used as a loss function in Xie et al. [2024a], instead of estimating local coverage.*

4 Experiments

4.1 Small data experiments

For all experiments, we consider a Matérn $5/2$ kernel for k^m and k^f. Our proposed kernel SoS algorithm is first compared in the small data regime to standard competitors for split CP: CQR (with random forests, following experiments from Romano et al. [2019], Hore and Barber [2024]) and rescaled scores. For the latter, we only consider two variants of GP for fair comparison, since we also place ourselves in the RKHS setting. We focus on a homoscedastic and heteroscedastic [Binois et al., 2018] GP model, and consider here $Y = m(X) + \sigma(X)\epsilon$ with:

	$m(X),\ Z = 10X + 1$	$\sigma(X)$	X	ϵ
Case 1	$\sin(\frac{2\pi Z}{5} + 0.2\frac{4\pi Z}{5})\mathbb{1}_{X\leq 9.6} + (\frac{Z}{10} - 1)\mathbb{1}_{X>9.6}$	$\sqrt{0.1 + 2X^2}$	$\mathcal{U}[-1,1]$	$\mathcal{N}(0,1)$
Case 2	$X/2$	$\lvert\sin(X)\rvert$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$

We begin by illustrating the interaction between b and θ^f in Figure 1 left. For all values of b, we observe a consistent HSIC behaviour: it first increases with θ^f, thus improving adaptivity, until it reaches a peak and then decreases. Interestingly, we also note that the higher b, the higher the optimal θ^f: this clearly shows that both hyperparameters have opposite effects on adaptivity. Furthermore, we see that for $b \geq 10$ we reach a plateau for the optimal HSIC. In practice it is thus sufficient to only optimize θ^f as soon as b is fixed at a large enough value. Figure 1 also shows that a small value for θ^f leads to overly adaptive bands, while the HSIC-optimized θ^f produces smooth and adaptive ones.

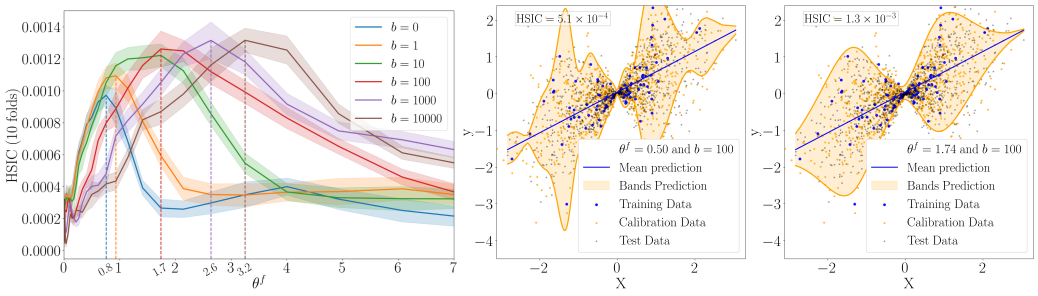


Figure 1: Test case 2 with $n = 100$. Left: HSIC criterion between $r(X,Y)$ and $f(X)$ as a function of b and θ^f (confidence intervals obtained by bootstrap and optimal values of θ^f in dashed lines). Middle / Right: optimal prediction bands with too small and optimized lengthscale, respectively.

In the following we now fix $b = 10$ and compare kernel SoS with HSIC-optimized θ^f to CQR and rescaled GPs on test case 1 with 20 replications. In Figure 2, we investigate several metrics related to adaptivity: mean width, MI and local coverage, see also Appendix B.4 for complementary discussion. We note that CQR and standard GP yield larger intervals and local coverage with great variability around the target. Heteroscedastic GP produces intervals similar to ours, but with higher mean width. In contrast, kernel SoS gives prediction intervals with both small width and satisfying local coverage properties. Additional experiments confirming this behavior are to be found in Appendix B.4.

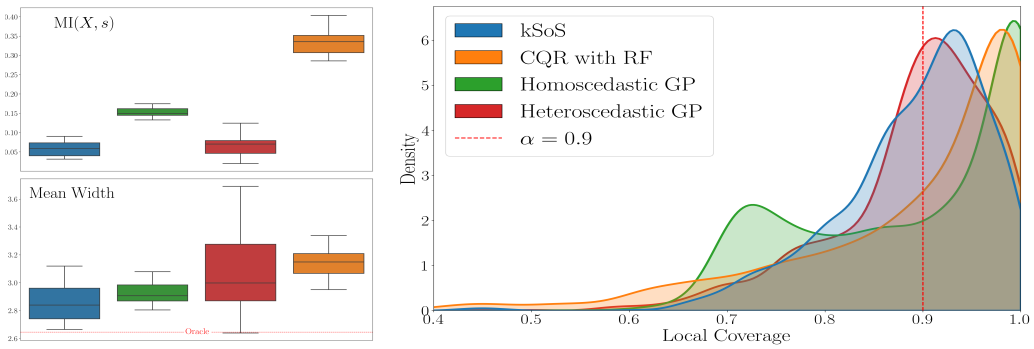


Figure 2: Test case 1 with $n = 100$. Adaptivity metrics and density of local coverage.

4.2 Real-world and large scale experiments

Finally, we demonstrate that our dual formulation for solving the kernel SoS problem can scale to several hundreds or thousands of training points, unlike previous work that relies on SDP solvers. Figure 3 left, shows that a SDP solver can only handle up to $n = 200$ training samples, while our dual solver scales easily to $n = 1000$ (see Appendix B.3). When we optimize θ^f with HSIC, we retrieve the optimal solution in Figure 3 right.

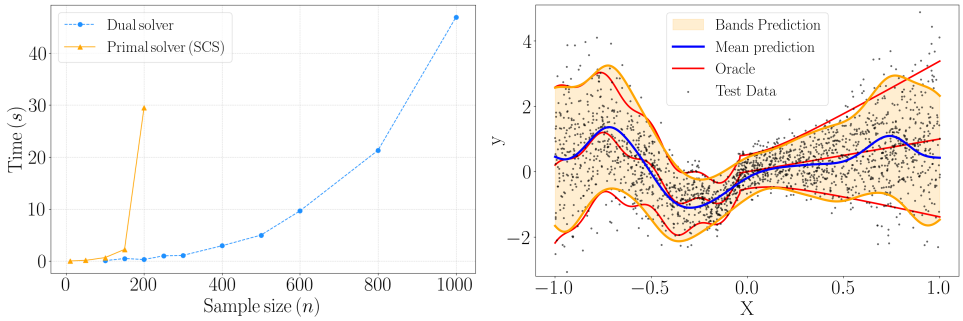


Figure 3: Test case 1. Left: time for SDP and dual formulation as a function of n ($a = b = 0$, $\theta^f = 0.3$, max iter $= 10^4$). Right: optimal solution of dual formulation for $n = 2000$.

At last, we consider six real-world datasets commonly used for regression, which are detailed in Appendix B.4. Table 1 reports the mean width of prediction intervals on a test set for all methods. We consider two variants of kSoS: one where θ^f is chosen to minimize the mean width, and the other where we optimize it with HSIC. Note that for Bio, which exhibits strong asymmetric noise, we use our asymmetric kSoS variant and do not use the HSIC criterion to tune θ^f, since at the moment it only applies to symmetric intervals. Such asymmetric noise heavily favors CQR, which outperforms all methods on this example. For all other cases except for Concrete which we comment below, kernel SoS targeting mean width achieves better adaptivity when measured by mean width. No other competitor, although they generally have strong performance, achieves this robustness. For completeness, we also provide the marginal coverage for all datasets and methods in Appendix B.4.

Dataset	CQR	Het GP	Hom GP	kSoS Best mean width	kSoS Opt. HSIC
Concrete	0.586 ± 0.032	$\mathbf{0.508} \pm 0.052$	0.543 ± 0.044	0.556 ± 0.044	0.568 ± 0.06
Bike	1.114 ± 0.062	1.000 ± 0.079	$\mathbf{0.809} \pm 0.024$	$\mathbf{0.803} \pm 0.031$	$\mathbf{0.803} \pm 0032$
Bio	$\mathbf{1.879} \pm 0.046$	2.21 ± 0.100	2.194 ± 0.119	2.03 ± 0.07	–
Diabetes	188.62 ± 9.33	191.24 ± 11.95	190.58 ± 11.19	$\mathbf{185.83} \pm 14.47$	187.6 ± 16.18
MPG	9.89 ± 0.82	9.70 ± 1.06	9.71 ± 0.73	$\mathbf{9.15} \pm 0.8$	9.36 ± 0.82
Housing	1.816 ± 0.045	1.585 ± 0.099	$\mathbf{1.453} \pm 0.068$	$\mathbf{1.468} \pm 0.094$	1.586 ± 0.104

Table 1: Mean width of prediction intervals for six real-world datasets (mean±sd on 10 repetitions).

At first sight, the kSoS variant based on HSIC produces larger bands and appears to be less adaptive if only mean width is considered. However, a more meaningful criterion to evaluate adaptivity is local coverage. Since it is unfortunately out of reach for real datasets, we rely instead on the worst-set coverage $\min_{l=1,\dots,L} \mathbb{P}(Y_{N+1} \in \widehat{C}_{\mathcal{D}_N}(X_{N+1}) | X_{N+1} \in \mathcal{R}_l)$ where $\{\mathcal{R}_l\}_{l=1,\dots,L}$ is a partition of the input space, see Thurin et al. [2025]. We observe in Figure 4 that kSoS with HSIC stays closer to the target $\alpha = 0.9$ than all other methods. For Concrete, heteroscedastic GP has the smaller mean width, but at the cost of producing intervals with lower worst-set coverage. Similarly, on Housing both homoscedastic GP and kSoS focusing on mean width exhibit the lowest mean width, but are further from the target. This illustrates once again that our HSIC criterion focuses on local adaptivity.

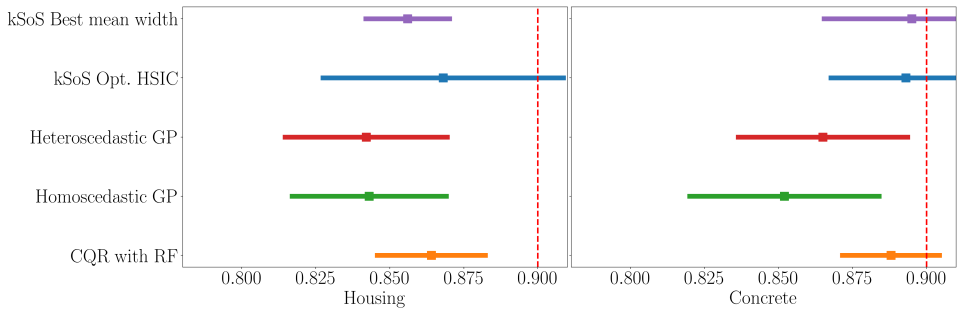


Figure 4: Housing (left, with $b = 100$) and Concrete (right, with $b = 10$) datasets. Mean and standard deviation of worst-set coverage on 10 regions with 100 test points each, 10 repetitions.

5 Limitations

Although our experiments show that kernel SoS is a competitor worth trying to build adaptive bands in a moderate sample size regime, we can still identify limitations that would necessitate extensions:

- Kernel SoS performs well for dimensions up to 10 or 15, a limitation shared by all kernel methods without specific strategies. For high-dimensional objects with structure (time series, probability distributions, molecules, strings, graphs,...), it is however straightforward to plug specifically designed kernels in our framework. On the other hand, for high-dimensional tabular data, a common strategy is to rely on additive kernels, which are left as future work.
- Kernel SoS can scale up to 2000 samples thanks to the dual formulation, but handling much more samples than that would be highly time consuming. This is inherently a limitation that comes from kernels rather than from the dual formulation. Indeed, each objective function and gradient computation involves the usual kernel regression formulas, which scale as $O(n^3)$. A usual workaround is to use Nystrom-type approximations or Random Fourier Features, which could be easily integrated in our problem.
- Due to the 100% coverage constraint, kernel SoS is less robust to outliers than CQR. In such cases, a pre-processing step using another model (e.g. the initial GP model we use in our procedure or an heteroscedastic GP model) is recommended.

6 Conclusion

We introduce a generalized kernel sum-of-squares framework for building scalable and adaptive prediction bands for conformal prediction. Scalability is achieved through a new representer theorem together with a dual formulation which can be solved efficiently with accelerated gradient algorithms. Unlike previous work, this makes the kernel SoS paradigm for CP able to scale up to thousands of training points, as we illustrate in our experiments. On the other hand, adaptivity and local coverage are targeted by optimizing hyperparameters with a new HSIC-based criterion, which is numerically robust and has excellent practical performance. Since such a criterion appears promising, as a perspective, we plan to investigate its extension to more general score functions.

References

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *Association for Computing Machinery*, 53(5), 2020. ISSN 0360-0300. doi: 10.1145/3409383. URL https://doi.org/10.1145/3409383.

A Gammerman, V Vovk, and V Vapnik. Learning by transduction. In *Conference on Uncertainty in Artificial Intelligence*, 1998.

Harris Papadopoulos, Kostas Proedrou, Vladimir Vovk, and Alexander Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, 2002. URL https://api.semanticscholar.org/CorpusID:42084298.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, mar 2008. URL http://jmlr.org/papers/volume9/shafer08a/shafer08a.pdf. Submitted 8/07; Published 3/08.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591, March 2023. ISSN 1935-8237. doi: 10.1561/2200000101. URL https://doi.org/10.1561/2200000101.

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

Janette Vazquez and Julio C Facelli. Conformal prediction in clinical medical sciences. *Journal of Healthcare Informatics Research*, 6(3):241–252, 2022.

Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Non-parametric models for non-negative functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12816–12826. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/968b15768f3d19770471e9436d97913c-Paper.pdf.

Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding global minima via kernel approximations. *Mathematical Programming*, 209(1):703–784, jan 2025. ISSN 1436-4646. doi: 10.1007/s10107-024-02081-4. URL https://doi.org/10.1007/s10107-024-02081-4.

Adrien Vacher, Boris Muzellec, Alessandro Rudi, Francis Bach, and Francois-Xavier Vialard. A dimension-free computational upper-bound for smooth optimal transport estimation. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4143–4173. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/vacher21a.html.

Alessandro Rudi and Carlo Ciliberto. Psd representations for effective probability models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19411–19422. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a1b63b36ba67b15d2f47da55cdb8018d-Paper.pdf.

Boris Muzellec, Francis Bach, and Alessandro Rudi. Learning psd-valued functions using kernel sums-of-squares, 2022. URL https://arxiv.org/abs/2111.11306.

Tengyuan Liang. Universal prediction band via semi-definite programming. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1558–1580, August 2022. ISSN 1467-9868. doi: 10.1111/rssb.12542. URL http://dx.doi.org/10.1111/rssb.12542.

Jianqing Fan, Jiawei Ge, and Debarghya Mukherjee. Utopia: Universally trainable optimal prediction intervals aggregation, 2024. URL https://arxiv.org/abs/2306.16549.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, jan 2014. doi: 10.1111/rssb.12021. URL https://doi.org/10.1111/rssb.12021.

U. Johansson, Henrik Boström, Tuwe Löfström, and Henrik Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97:155 – 176, 2014. URL https://api.semanticscholar.org/CorpusID:14015369.

Harris Papadopoulos. Guaranteed coverage prediction intervals with gaussian process regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9072–9083, December 2024. ISSN 1939-3539. doi: 10.1109/tpami.2024.3418214. URL http://dx.doi.org/10.1109/TPAMI.2024.3418214.

Edgar Jaber, Vincent Blot, Nicolas Brunel, Vincent Chabridon, Emmanuel Remy, Bertrand Iooss, Didier Lucor, Mathilde Mougeot, and Alessandro Leite. Conformal approach to gaussian process surrogate evaluation with coverage guarantees. *Journal of Machine Learning for Modeling and Computing*, 6(3), 2025.

Shai Feldman, Stephen Bates, and Yaniv Romano. Improving conditional coverage via orthogonal quantile regression. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2060–2071. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1006ff12c465532f8c574aeaa4461b16-Paper.pdf.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, mar 2023. doi: 10.1093/biomet/asac040. URL https://doi.org/10.1093/biomet/asac040.

Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables robust guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2024. doi: 10.1093/jrsssb/qkae103. URL https://doi.org/10.1093/jrsssb/qkae103.

Nicolas Deutschmann, Mattia Rigotti, and Maria Rodriguez Martinez. Adaptive conformal regression with split-jackknife+ scores. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=1fbTGC3BUD.

Ran Xie, Rina Foygel Barber, and Emmanuel J. Candès. Boosted conformal prediction intervals. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 71868–71899. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/842714f78c95096e20ac7d2591c5a24b-Paper-Conference.pdf.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Ruiting Liang, Wanrong Zhu, and Rina Foygel Barber. Conformal prediction after efficiency-oriented model selection, 2024. URL https://arxiv.org/abs/2408.07066.

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Sacha Braun, Liviu Aolaritei, Michael I Jordan, and Francis Bach. Minimum volume conformal sets for multivariate regression. *arXiv preprint arXiv:2503.19068*, 2025.

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016. URL http://stanford.edu/~boyd/papers/scs.html.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL https://arxiv.org/abs/1609.04747.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9508–9520, 2024b. doi: 10.1109/TPAMI.2024.3423382.

Pierre-Cyril Aubin-Frankowski and Alessandro Rudi. Approximation of optimization problems with constraints through kernel sum-of-squares. *Optimization*, pages 1–26, 2024.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In Steven C. H. Hoi and Wray Buntine, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pages 475–490, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR. URL https://proceedings.mlr.press/v25/vovk12.html.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(4):455–482, aug 2021. doi: 10.1093/imaiai/iaaa017. URL https://doi.org/10.1093/imaiai/iaaa017.

Chong Xiao Wang and Wee Peng Tay. Semi-nonparametric estimation of distribution divergence in non-euclidean spaces, 2023. URL https://arxiv.org/abs/2204.02031.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

Mickael Binois, Robert B Gramacy, and Mike Ludkovski. Practical heteroscedastic gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27 (4):808–821, 2018.

Gauthier Thurin, Kimia Nadjahi, and Claire Boyer. Optimal transport-based conformal prediction. *PMLR*, 267:59509–59527, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/thurin25a.html.

Kaare B. Petersen and Michael S. Pedersen. The Matrix Cookbook, 2012. URL https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018. doi: 10.1080/01621459.2017.1307116. URL https://doi.org/10.1080/01621459.2017.1307116.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International conference on algorithmic learning theory*, pages 13–31. Springer, 2007.

D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, pages 2263–2291, 2013.

Michael JD Powell et al. The bobyqa algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 26:26–46, 2009.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Brendan O'Donoghue. Operator splitting for a homogeneous embedding of the linear complementarity problem. *SIAM Journal on Optimization*, 31:1999–2023, August 2021.

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. SCS: Splitting conic solver, version 3.2.7. https://github.com/cvxgrp/scs, November 2023.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Mathieu Barré, Adrien Taylor, and Francis Bach. A note on approximate accelerated forward-backward methods with absolute and relative errors, and possibly strongly convex objectives. *Open Journal of Mathematical Optimization*, 3:1, 2022. doi: 10.5802/ojmo.12. URL https://ojmo.centre-mersenne.org/articles/10.5802/ojmo.12/.

Truong T. Truong and Huy T. Nguyen. Backtracking gradient descent method and some applications in large scale optimisation. part 2: Algorithms and experiments. *Applied Mathematics & Optimization*, 84:2557–2586, 2021. doi: 10.1007/s00245-020-09718-8. URL https://doi.org/10.1007/s00245-020-09718-8.

Robert B. Gramacy and Herbert K. H. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, May 2009. ISSN 1537-2723. doi: 10.1198/tech.2009.0015. URL http://dx.doi.org/10.1198/TECH.2009.0015.

Danijel Kivaranovic, Kory D Johnson, and Hannes Leeb. Adaptive, distribution-free prediction intervals for deep networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4346–4356. PMLR, 2020.

I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

Hadi Fanaee-T. Bike Sharing. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Prashant Rana. Physicochemical Properties of Protein Tertiary Structure. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5QW3H.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–451, 2004.

R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33 (3):291–297, 1997.

R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5859H.

A Proofs

A.1 Proof of Theorem 2 - representer theorem

To begin, we first introduce some notation. When considering objects related to functions m and f, we will use the superscripts ^{m} and ^{f} to differentiate them. Associated to each function we consider a RKHS $\mathcal{H}^{(\cdot)}$, a kernel $k^{(\cdot)}$, a kernel matrix $\mathbf{K}^{(\cdot)}$, a feature map $\phi^{(\cdot)}$ and a column vector such that for $x \in \mathcal{X}$

$$\mathbf{k}^{(\cdot)}(X) = \left(k^{(\cdot)}(X_1, X), \ldots, k^{(\cdot)}(X_n, X)\right)^T.$$

For the kernel matrix $[\mathbf{K}^f]_{ij} = k^f(X_i, X_j)$ only, we consider its Cholesky decomposition and empirical feature map

$$\mathbf{K}^f = \mathbf{V}^T\mathbf{V} \quad \text{and} \quad \mathbf{\Phi}(X) = \mathbf{V}^{-T}\mathbf{k}^f(X).$$

Next, for a Hilbert space $\mathcal{H}$ we write $\mathcal{S}(\mathcal{H})$ the set of bounded Hermitian linear operators from $\mathcal{H}$ to $\mathcal{H}$ and $\mathcal{S}_+(\mathcal{H})$ those that are positive-definite. We also write $\mathbb{S}_+^n = \mathbb{S}_+\left(\mathbb{R}^{n\times n}\right)$ the set of real, symmetric and positive-definite square matrices of size n.

To prove our representer theorem, we first rewrite (4) as

$$\begin{aligned} \inf_{m\in\mathcal{H}^m,\, \mathcal{A}\in\mathcal{S}_+(\mathcal{H}^f)} \quad & a\cdot\frac{1}{n}\sum_{i=1}^n \left(Y_i - m(X_i)\right)^2 + b\cdot\frac{1}{n}\sum_{i=1}^n f_{\mathcal{A}}(X_i) + \lambda_1\|\mathcal{A}\|_\star + \lambda_2\|\mathcal{A}\|_F^2 \\ \text{s.t.} \quad & \left(Y_i - m(X_i)\right)^2 - f_{\mathcal{A}}(X_i) \le 0,\ i\in[n]\,, \\ & \|m\|_{\mathcal{H}^m}^2 - s \le 0. \end{aligned} \tag{13}$$

In order to show that there exists a finite-dimensional representation for both $m^\star$ and $\mathcal{A}^\star$, we will first show that for any fixed $m \in \mathcal{H}^m$, the optimal $\mathcal{A}$ has a finite-dimensional representation. Indeed if $m \in \mathcal{H}^m$ is fixed, the problem writes

$$\begin{aligned} \inf_{\mathcal{A}\in\mathcal{S}_+(\mathcal{H}^f)} \quad & \frac{b}{n}\sum_{i=1}^n f_{\mathcal{A}}(X_i) + \lambda_1\|\mathcal{A}\|_\star + \lambda_2\|\mathcal{A}\|_F^2 \\ \text{s.t.} \quad & \left(Y_i - m(X_i)\right)^2 - f_{\mathcal{A}}(X_i) \le 0,\ i\in[n] \end{aligned}$$

or equivalently

$$\inf_{\mathcal{A}\in\mathcal{S}_+(\mathcal{H}^f)} L_m\left(f_{\mathcal{A}}(X_1), \ldots, f_{\mathcal{A}}(X_n)\right) + \Omega\left(\mathcal{A}\right)$$

where

$$L_m\left(f_{\mathcal{A}}(X_1), \ldots, f_{\mathcal{A}}(X_n)\right) = \begin{cases} \frac{b}{n}\sum_{i=1}^n f_{\mathcal{A}}(X_i) & \text{if } \left(Y_i - m(X_i)\right)^2 - f_{\mathcal{A}}(X_i) \le 0,\ i\in[n]\,, \\ +\infty & \text{else,} \end{cases}$$

and

$$\Omega\left(\mathcal{A}\right) = \lambda_1\|\mathcal{A}\|_\star + \lambda_2\|\mathcal{A}\|_F^2.$$

Since $L_m : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is lower semi-continuous and bounded below (notice that it is linear and bounded below by 0), we can apply Theorem 1 and Proposition 3 from Marteau-Ferey et al. [2020] to deduce that the solution is entirely characterized by a PSD matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ given by

$$\begin{aligned} \inf_{\mathbf{A}\in\mathbb{S}_+^n} \quad & \frac{b}{n}\sum_{i=1}^n \tilde{f}_{\mathbf{A}}(X_i) + \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2 \\ \text{s.t.} \quad & \left(Y_i - m(X_i)\right)^2 - \tilde{f}_{\mathbf{A}}(X_i) \le 0,\ i\in[n]\,, \end{aligned}$$

with

$$\tilde{f}_{\mathbf{A}}(X_i) = \langle \mathbf{\Phi}(X_i), \mathbf{A}\mathbf{\Phi}(X_i) \rangle \quad \text{and} \quad \mathbf{\Phi}(X_i) = \mathbf{V}^{-T}\mathbf{k}^f(X_i).$$

Since this representation is valid for any $m \in \mathcal{H}^m$, Problem (13) is thus equivalent to

$$\begin{aligned} \inf_{m \in \mathcal{H}^m,\, \mathbf{A} \in \mathbb{S}_+^n} \quad & \frac{a}{n}\sum_{i=1}^n \left(Y_i - m(X_i)\right)^2 + \frac{b}{n}\sum_{i=1}^n \tilde{f}_{\mathbf{A}}(X_i) + \lambda_1 \|\mathbf{A}\|_\star + \lambda_2 \|\mathbf{A}\|_F^2 \\ \text{s.t.} \quad & \left(Y_i - m(X_i)\right)^2 - \tilde{f}_{\mathbf{A}}(X_i) \leq 0,\ i \in [n]\,, \\ & \|m\|_{\mathcal{H}^m}^2 - s \leq 0. \end{aligned} \tag{14}$$

Now, to show that $m^\star \in \mathcal{H}^m$ has a finite-dimensional representation, we will rely upon the dual problem. To do so, we first need to show that strong duality holds. Since our problem is convex, it is enough to check Slater's constraint qualification, *i.e.* we simply need to exhibit a strictly feasible point $(m_0, \mathbf{A}_0)$. We first take $m_0 = 0 \in \mathcal{H}^m$ which satisfies $\|m_0\|_{\mathcal{H}^m}^2 - s = -s < 0$. If we assume that $\mathbf{K}^f$ is of full rank (this assumption is always satisfied if $k^{(f)}$ is universal and all training points X_i are distinct), it is invertible and we can define $\alpha = (\mathbf{K}^f)^{-1}\mathbf{Y}$ such that $Y_i = \sum_{j=1}^n \alpha_j k^{(f)}(X_i, X_j)$. Denote $\mathbf{B}_0 \succ 0$ the **positive-definite** matrix with elements $[\mathbf{B}_0]_{ii} = \alpha_i^2 + \epsilon \mathbb{1}_{\alpha_i = 0}$ and $[\mathbf{B}_0]_{ij} = 0$ if $i \neq j$ where $\epsilon > 0$ (note that in general all α_i will be non-zeros, but if it it not the case we introduce a small ϵ). Then by Cauchy-Schwartz we have for all $i = 1, \ldots, n$

$$\begin{aligned} Y_i^2 &= \left(\sum_{j=1}^n \alpha_j k^{(f)}(X_i, X_j)\right)^2 \\ &\leq n \sum_{j=1}^n \alpha_j^2 (k^{(f)}(X_i, X_j))^2 \\ &\leq n \sum_{j=1}^n (\alpha_j^2 + \epsilon \mathbb{1}_{\alpha_j = 0})(k^{(f)}(X_i, X_j))^2 \\ &= n \sum_{j=1}^n [\mathbf{B}_0]_{jj} (k^{(f)}(X_i, X_j))^2 = n \tilde{f}_{\mathbf{A}_0}(X_i) \end{aligned}$$

with $\mathbf{A}_0 = \mathbf{V}\mathbf{B}_0\mathbf{V}^\top \succ 0$, which implies $(Y_i - m_0(X_i))^2 = Y_i^2 < \tilde{f}_{\mathbf{A}_0}(X_i)$ as soon as $n \geq 1$. Consequently $(m_0, \mathbf{A}_0)$ is a strictly feasible point and strong duality holds. We thus introduce $\mathbf{\Gamma} \in \mathbb{R}_+^n$ the Lagrangian multipliers associated to the first n constraints and $\theta \in \mathbb{R}_+$ the Lagrangian multiplier associated to the constraint on the norm of m. The Lagrangian function writes:

$$\begin{aligned} \mathcal{L}(m, \mathbf{A}, \mathbf{\Gamma}, \theta) =& \frac{a}{n}\sum_{i=1}^n \left(Y_i - m(X_i)\right)^2 + \frac{b}{n}\sum_{i=1}^n \tilde{f}_{\mathbf{A}}(X_i) + \lambda_1 \|\mathbf{A}\|_\star + \lambda_2 \|\mathbf{A}\|_F^2 \\ &+ \sum_{i=1}^n \Gamma_i \left[\left(Y_i - m(X_i)\right)^2 - \tilde{f}_{\mathbf{A}}(X_i)\right] + \theta\left(\|m\|_{\mathcal{H}^m}^2 - s\right) \\ =& \sum_{i=1}^n \left(\frac{a}{n} + \Gamma_i\right)\left(Y_i - m(X_i)\right)^2 + \theta\left(\|m\|_{\mathcal{H}^m}^2 - s\right) + \lambda_1 \|\mathbf{A}\|_\star + \lambda_2 \|\mathbf{A}\|_F^2 \\ &- \sum_{i=1}^n \left(\Gamma_i - \frac{b}{n}\right) \tilde{f}_{\mathbf{A}}(X_i). \end{aligned}$$

Following Muzellec et al. [2022] Appendix C.5, the optimality conditions of the Lagrangian function w.r.t. m are

$$\begin{aligned}\nabla_m \mathcal{L}(m, \mathbf{A}, \mathbf{\Gamma}, \theta) &= \sum_{i=1}^{n} 2\left(\Gamma_i + \frac{a}{n}\right) k^m(X_i, \cdot)\big(\langle m(\cdot), k^m(X_i, \cdot)\rangle_{\mathcal{H}^m} - Y_i\big) + 2\theta m(\cdot) \\ &= 2\sum_{i=1}^{n}\left(\Gamma_i + \frac{a}{n}\right)\left[k^m(X_i, \cdot) \otimes k^m(X_i, \cdot)\right](m) + 2\theta m(\cdot) \\ &- 2\sum_{i=1}^{n}\left(\Gamma_i + \frac{a}{n}\right) Y_i k^m(X_i, \cdot)\end{aligned}$$

and setting this gradient to 0 leads to

$$\left[\sum_{i=1}^{n}\left(\Gamma_i + \frac{a}{n}\right)\left(k^m(X_i, \cdot) \otimes k^m(X_i, \cdot)\right) + \theta \mathbf{I}_n\right](m) = \sum_{i=1}^{n}\left(\Gamma_i + \frac{a}{n}\right) Y_i k^m(X_i, \cdot).$$

To conclude, we denote

$$\begin{aligned}&\boldsymbol{C}\left(\mathbf{\Gamma}, \theta\right) := \mathrm{Diag}\left(\mathbf{\Gamma_a}\right)\mathbf{K}^m + \theta \mathbf{I}_n, \\ &\boldsymbol{\gamma}\left(\mathbf{\Gamma}, \theta\right) := \boldsymbol{C}\left(\mathbf{\Gamma}, \theta\right)^{-1}\mathrm{Diag}\left(\mathbf{\Gamma_a}\right)\mathbf{Y}, \\ &\mathrm{Diag}\left(\mathbf{\Gamma_a}\right) = \mathrm{Diag}(\mathbf{\Gamma}) + \frac{a}{n}\mathbf{I}_n\end{aligned}$$

such that $m^\star$ has the finite-dimensional representation

$$m^\star(X) = \sum_{i=1}^{n} \gamma_i k^m(X_i, X) = \boldsymbol{\gamma}^T \mathbf{k}^m(X).$$

In the end, plugging this expression in Problem (14) yields the semi-definite problem from Equation (7):

$$\begin{aligned}\inf_{\boldsymbol{\gamma}\in\mathbb{R}^n,\, \mathbf{A}\in\mathbb{S}_+^n} \quad & \frac{a}{n}\sum_{i=1}^{n}\left(Y_i - m(X_i)\right)^2 + \frac{b}{n}\sum_{i=1}^{n}\tilde{f}_{\mathbf{A}}(X_i) + \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2 \\ \text{s.t.} \quad & \left(Y_i - \boldsymbol{\gamma}^T\mathbf{k}^m(X_i)\right)^2 - \tilde{f}_{\mathbf{A}}(X_i) \leq 0,\ i \in [n], \\ & \boldsymbol{\gamma}^T\mathbf{K}^m\boldsymbol{\gamma} - s \leq 0.\end{aligned}$$

A.2 Proof of Proposition 2 - dual formulation

Proof. The dual problem of Equation (7) is defined as

$$d = \sup_{\substack{\mathbf{\Gamma}\in\mathbb{R}_+^n \\ \theta\in\mathbb{R}_+}} \inf_{\substack{m\in\mathcal{H}^m \\ \mathbf{A}\in\mathbb{S}_+^n}} \mathcal{L}(m, \mathbf{A}, \mathbf{\Gamma}, \theta) = \sup_{\substack{\mathbf{\Gamma}\in\mathbb{R}_+^n \\ \theta\in\mathbb{R}_+}} D\left(\mathbf{\Gamma}, \theta\right)$$

where the dual function is $D\left(\mathbf{\Gamma}, \theta\right) := \inf_{m\in\mathcal{H}^m,\, \mathbf{A}\in\mathbb{S}_+^n} \mathcal{L}(m, \mathbf{A}, \mathbf{\Gamma}, \theta)$. Remark first that in the previous section we already introduced the Lagrangian function

$$\begin{aligned}\mathcal{L}(m, \mathbf{A}, \mathbf{\Gamma}, \theta) = &\sum_{i=1}^{n}\left(\frac{a}{n} + \Gamma_i\right)\left(Y_i - m(X_i)\right)^2 + \theta\left(\|m\|_{\mathcal{H}^m}^2 - s\right) + \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2 \\ &- \sum_{i=1}^{n}\left(\Gamma_i - \frac{b}{n}\right)\tilde{f}_{\mathbf{A}}(X_i),\end{aligned}$$

with optimality condition for m given by $m^\star(X) = \boldsymbol{\gamma}^T \mathbf{k}^m(X)$. Now, we need to derive the optimality conditions for $\mathbf{A}$:

$$\begin{aligned}
&\inf_{\mathbf{A}\in\mathbb{S}_+^n} \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2 - \sum_{i=1}^n \left(\Gamma_i - \frac{b}{n}\right) \tilde{f}_{\mathbf{A}}(X_i) \\
&= \inf_{\mathbf{A}\in\mathbb{S}_+^n} \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2 - \langle \mathbf{A}, \mathbf{V}\mathrm{Diag}\left(\boldsymbol{\Gamma}_{-\mathbf{b}}\right)\mathbf{V}^T\rangle \qquad (15)\\
&= - \sup_{\mathbf{A}\in\mathbb{S}_+^n} \langle \mathbf{A}, \mathbf{V}\mathrm{Diag}\left(\boldsymbol{\Gamma}_{-\mathbf{b}}\right)\mathbf{V}^T\rangle - \lambda_1\|\mathbf{A}\|_\star - \lambda_2\|\mathbf{A}\|_F^2 \\
&= - \Omega^\star\left(\mathbf{V}\mathrm{Diag}\left(\boldsymbol{\Gamma}_{-\mathbf{b}}\right)\mathbf{V}^T\right) \qquad (16)
\end{aligned}$$

where $\Omega^\star$ is the Fenchel conjugate of Ω. Equality (15) comes from the fact that $\sum_{i=1}^n \Gamma_i \tilde{f}_{\mathbf{A}}(X_i) = \langle \mathbf{A}, \mathbf{V}\mathrm{Diag}\left(\boldsymbol{\Gamma}\right)\mathbf{V}^T\rangle$. Indeed, recall that $\tilde{f}_{\mathbf{A}}(X_i) = \langle \boldsymbol{\Phi}(X_i), \mathbf{A}\boldsymbol{\Phi}(X_i)\rangle$ and $\boldsymbol{\Phi}(X_i) = \mathbf{V}^{-T}\mathbf{k}^f(X_i)$, which yields

$$\begin{aligned}
\sum_{i=1}^n \Gamma_i \tilde{f}_{\mathbf{A}}(X_i) &= \sum_{i=1}^n \Gamma_i \langle \boldsymbol{\Phi}(X_i), \mathbf{A}\boldsymbol{\Phi}(X_i)\rangle = \sum_{i=1}^n \Gamma_i \,\mathrm{Tr}\left(\boldsymbol{\Phi}(X_i)^T\mathbf{A}\boldsymbol{\Phi}(X_i)\right) \\
&= \sum_{i=1}^n \Gamma_i \,\mathrm{Tr}\left(\mathbf{A}\boldsymbol{\Phi}(X_i)\boldsymbol{\Phi}(X_i)^T\right) = \sum_{i=1}^n \Gamma_i \langle \mathbf{A}, \boldsymbol{\Phi}(X_i)\boldsymbol{\Phi}(X_i)^T\rangle \\
&= \langle \mathbf{A}, \sum_{i=1}^n \Gamma_i \boldsymbol{\Phi}(X_i)\boldsymbol{\Phi}(X_i)^T\rangle
\end{aligned}$$

where the second term inside the brackets can be expressed as

$$\begin{aligned}
\sum_{i=1}^n \Gamma_i \boldsymbol{\Phi}(X_i)\boldsymbol{\Phi}(X_i)^T &= \sum_{i=1}^n \Gamma_i \mathbf{V}^{-T}\mathbf{k}^f(X_i)\mathbf{k}^f(X_i)^T\mathbf{V}^{-1} \\
&= \mathbf{V}^{-T}\left[\sum_{i=1}^n \Gamma_i \mathbf{k}^f(X_i)\mathbf{k}^f(X_i)^T\right]\mathbf{V}^{-1} \\
&= \mathbf{V}^{-T}\mathbf{K}^f\mathrm{Diag}(\boldsymbol{\Gamma})(\mathbf{K}^f)^T\mathbf{V}^{-1} \\
&= \mathbf{V}\mathrm{Diag}(\boldsymbol{\Gamma})\mathbf{V}^T.
\end{aligned}$$

Equation (16) is simply the definition of the Fenchel conjugate of $\Omega\left(\mathbf{A}\right) = \lambda_1\|\mathbf{A}\|_\star + \lambda_2\|\mathbf{A}\|_F^2$. Finally, replacing m by its optimal value and the regularizations involving $\mathbf{A}$ by the above expression, the dual function is

$$\begin{aligned}
D\left(\boldsymbol{\Gamma}, \theta\right) &= \inf_{\substack{m\in\mathcal{H}^m \\ \mathbf{A}\in\mathbb{S}_+^n}} \mathcal{L}(m, \mathbf{A}, \boldsymbol{\Gamma}, \theta) \\
&= \underbrace{\mathbf{r}(\boldsymbol{\Gamma}, \theta)^\top \mathrm{Diag}(\boldsymbol{\Gamma}_{\mathbf{a}})\mathbf{r}(\boldsymbol{\Gamma}, \theta)}_{\text{first term}} + \underbrace{\theta(\boldsymbol{\gamma}(\boldsymbol{\Gamma}, \theta)^\top \mathbf{K}^m \boldsymbol{\gamma}(\boldsymbol{\Gamma}, \theta) - s)}_{\text{second term}} - \underbrace{\Omega^\star(\mathbf{V}\mathrm{Diag}(\boldsymbol{\Gamma}_{-\mathbf{b}})\mathbf{V}^T)}_{\text{third term}}
\end{aligned}$$

where $\mathbf{r}(\boldsymbol{\Gamma}, \theta) := \mathbf{Y} - \mathbf{K}^m\boldsymbol{\gamma}(\boldsymbol{\Gamma}, \theta)$, which corresponds to Proposition 2.

Gradient computation. To solve the dual problem

$$d = \sup_{\substack{\boldsymbol{\Gamma}\in\mathbb{R}_+^n \\ \theta\in\mathbb{R}_+}} D\left(\boldsymbol{\Gamma}, \theta\right)$$

we propose to use an accelerated gradient algorithm, which requires the gradient of the dual function w.r.t. the Lagrange multipliers. We now provide the explicit computations for this gradient. In what follows, $\mathbf{J}_{jj}$ denotes the matrix filled with zeros except for a 1 at row j and column j.

Gradient of first term $\mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{r}$.

It is straightforward to show that

$$\begin{aligned}
\frac{\partial \mathbf{C}}{\partial \Gamma_j} &= \frac{\partial \mathrm{Diag}(\mathbf{\Gamma}_{\boldsymbol{a}})}{\partial \Gamma_j}\mathbf{K}^m = \mathbf{J}_{jj}\mathbf{K}^m, \\
\frac{\partial \mathbf{C}^{-1}}{\partial \Gamma_j} &= -\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \Gamma_j}\mathbf{C}^{-1} = -\mathbf{C}^{-1}\mathbf{J}_{jj}\mathbf{K}^m\mathbf{C}^{-1}, \\
\frac{\partial \boldsymbol{\gamma}}{\partial \Gamma_j} &= \frac{\partial \mathbf{C}^{-1}}{\partial \Gamma_j}\mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{Y} + \mathbf{C}^{-1}\frac{\partial \mathrm{Diag}(\mathbf{\Gamma})}{\partial \Gamma_j}\mathbf{Y} \\
&= \mathbf{C}^{-1}\mathbf{J}_{jj}\left[-\mathbf{K}^m\mathbf{C}^{-1}\mathrm{Diag}(\mathbf{\Gamma_a}) + \mathbf{I}_n\right]\mathbf{Y} \\
&= \mathbf{C}^{-1}\mathbf{J}_{jj}\mathbf{u}, \\
\frac{\partial \mathbf{r}}{\partial \Gamma_j} &= -\mathbf{K}^m\frac{\partial \boldsymbol{\gamma}}{\partial \Gamma_j} = -\mathbf{K}^m\mathbf{C}^{-1}\mathbf{J}_{jj}\mathbf{u}
\end{aligned}$$

where $\mathbf{u} := \left[-\mathbf{K}^m\mathbf{C}^{-1}\mathrm{Diag}(\mathbf{\Gamma_a}) + \mathbf{I}_n\right]\mathbf{Y}$. We then get

$$\begin{aligned}
\frac{\partial \mathbf{r}^T \mathrm{Diag}\left(\mathbf{\Gamma_a}\right)\mathbf{r}}{\partial \Gamma_j} &= r_j^2 + 2\mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\frac{\partial \mathbf{r}}{\partial \Gamma_j} \\
&= r_j^2 - 2\mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{K}^m\mathbf{C}^{-1}\mathbf{J}_{jj}\mathbf{u} \\
&= r_j^2 - 2\mathbf{s}^T\mathbf{J}_{jj}\mathbf{u}, \\
\frac{\partial \mathbf{r}^T \mathrm{Diag}\left(\mathbf{\Gamma_a}\right)\mathbf{r}}{\partial \mathbf{\Gamma}} &= \mathbf{r}^{\odot 2} - 2\mathbf{s} \odot \mathbf{u}
\end{aligned}$$

where $\mathbf{s} := \mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{K}^m\mathbf{C}^{-1}$. Similarly,

$$\begin{aligned}
\frac{\partial \mathbf{C}}{\partial \theta} &= \mathbf{I}_n, \\
\frac{\partial \mathbf{C}^{-1}}{\partial \theta} &= -\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta}\mathbf{C}^{-1} = -\mathbf{C}^{-2}, \\
\frac{\partial \boldsymbol{\gamma}}{\partial \theta} &= \frac{\partial \mathbf{C}^{-1}}{\partial \theta}\mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{Y} = -\mathbf{C}^{-2}\mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{Y}, \\
\frac{\partial \mathbf{r}}{\partial \theta} &= -\mathbf{K}^m\frac{\partial \boldsymbol{\gamma}}{\partial \theta} = \mathbf{K}^m\mathbf{C}^{-2}\mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{Y} := \mathbf{t},
\end{aligned}$$

such that

$$\begin{aligned}
\frac{\partial \mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{r}}{\partial \theta} &= 2\mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\frac{\partial \mathbf{r}}{\partial \theta} \\
&= 2\mathbf{r}^\top \mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{t}.
\end{aligned}$$

Gradient of second term $\theta(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s)$.

$$\begin{aligned}
\frac{\partial \theta(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s)}{\partial \Gamma_j} &= 2\theta\boldsymbol{\gamma}^\top\mathbf{K}^m\frac{\partial \boldsymbol{\gamma}}{\partial \Gamma_j} \\
&= 2\theta\boldsymbol{\gamma}^\top\mathbf{K}^m\mathbf{C}^{-1}\mathbf{J}_{jj}\mathbf{u}, \\
&= 2\theta\mathbf{p}^T\mathbf{J}_{jj}\mathbf{u} \\
\frac{\partial \theta(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s)}{\partial \mathbf{\Gamma}} &= 2\theta\mathbf{p} \odot \mathbf{u}
\end{aligned}$$

where $\mathbf{p} := \boldsymbol{\gamma}^\top\mathbf{K}^m\mathbf{C}^{-1}$. We also have

$$\begin{aligned}
\frac{\partial \theta\left(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s\right)}{\partial \theta} &= \left(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s\right) + 2\theta\boldsymbol{\gamma}^\top\mathbf{K}^m\frac{\partial \boldsymbol{\gamma}}{\partial \theta} \\
&= \left(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s\right) - 2\theta\boldsymbol{\gamma}^\top\mathbf{K}^m\mathbf{C}^{-2}\mathrm{Diag}(\mathbf{\Gamma_a})\mathbf{Y} \\
&= \left(\boldsymbol{\gamma}^\top\mathbf{K}^m\boldsymbol{\gamma} - s\right) - 2\theta\boldsymbol{\gamma}^\top\mathbf{t}.
\end{aligned}$$

Gradient of third term $\Omega^\star(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T)$.

From Marteau-Ferey et al. [2020] Lemma 5, we have

$$\nabla\Omega^\star\left(\mathbf{A}^\star\right) = \frac{1}{2\lambda_2}\left[\mathbf{A}^\star - \lambda_1\mathbf{I}_n\right]_+ .$$

We then use the chain rule following Equation (137) from Petersen and Pedersen [2012]:

$$\begin{aligned}
\frac{\partial\Omega^\star(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T)}{\partial\Gamma_j} &= \mathrm{Tr}\left[\left[\nabla\Omega^\star\left(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T\right)\right]^T\mathbf{V}\mathbf{J}_{jj}\mathbf{V}^T\right] \\
&= \left(\mathbf{V}^T\left[\nabla\Omega^\star\left(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T\right)\right]^T\mathbf{V}\right)_{jj}, \\
\frac{\partial\Omega^\star(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T)}{\partial\mathbf{\Gamma}} &= \mathrm{Diag}\left(\mathbf{V}^T\left[\nabla\Omega^\star\left(\mathbf{V}\mathrm{Diag}(\mathbf{\Gamma}_{-\mathbf{b}})\mathbf{V}^T\right)\right]^T\mathbf{V}\right).
\end{aligned}$$

Recovering the solution from optimal Lagrange multipliers. Once the dual problem is solved (in practice the convergence of our accelerated gradient algorithm is checked with some small relative tolerances on the constraints and the duality gap, *e.g.* 10^{-2}, see Appendix B.3), we need to recover the optimal solutions of the primal problem. Denoting $\widehat{\mathbf{\Gamma}} \in \mathbb{R}_+^n$ and $\widehat{\theta} \in \mathbb{R}_+$ the optimal Lagrange multipliers, the approximated mean function is recovered by

$$\widehat{\gamma} = \left(\mathrm{Diag}(\widehat{\mathbf{\Gamma}}_{\mathbf{a}})\mathbf{K}^m + \widehat{\theta}\mathbf{I}_n\right)^{-1}\mathrm{Diag}(\widehat{\mathbf{\Gamma}}_{\mathbf{a}})\mathbf{Y}.$$

On the other hand, to reconstruct the matrix $\mathbf{A}$, we follow Theorem 8 from Marteau-Ferey et al. [2020]:

$$\begin{aligned}
\widehat{\mathbf{A}} &= \nabla\Omega^\star\left(\mathbf{V}\mathrm{Diag}(\widehat{\mathbf{\Gamma}}_{-\mathbf{b}})\mathbf{V}^T\right) \\
&= \frac{1}{2\lambda_2}\left[\mathbf{V}\mathrm{Diag}(\widehat{\mathbf{\Gamma}}_{-\mathbf{b}})\mathbf{V}^T - \lambda_1\mathbf{I}_n\right]_+ .
\end{aligned}$$

A.3 Proof of Proposition 3 - marginal coverage

Proposition 3 is a trivial extension of standard split CP: the kernel SoS procedure is conducted on pre-training data only. The scores are then computed on an independent calibration data, and thus training conditional coverage is straightforward, yielding marginal coverage (see for example Lei et al. [2018]). Algorithm 1 gives a detailed overview of the full procedure.

Improved calibration from Liang [2022] and Fan et al. [2024]. The split CP calibration procedure and Fan et al. [2024] are really close. Indeed, the calibration procedure in Fan et al. [2024] is also based on a splitting strategy as for split CP, but without adjusting the quantile level. Their calibration method is thus almost equivalent to split CP, and from a theoretical perspective they recover the target coverage level but with different tools (which allows them to derive explicit coverage bounds for their bands before calibration). The main advantage of using split CP, is that we can benefit from all previous and future extensions of CP to handle specific situations (e.g. covariate shift, jackknife+, ...) at almost no cost, unlike Fan et al. [2024]'s strategy. On the other hand, Liang [2022] also provides pre-calibration coverage bounds, while relying on sample splitting. The main variation with respect to split CP or Fan et al. [2024] is that the hyperparameter which must be optimized on the calibration set no longer appears as a multiplicative constant in front of the band width, but as an additive constant inside a square root, thus implying a more complex dyadic search (in contrast to split CP framework or Fan et al. [2024]'s one, where it is simply given as a sample quantile). This simple difference partly explains why they require other proof strategies to get the target coverage. Once again, we believe that the traditional split CP is more straightforward and easier to apprehend, with no obvious disadvantage (to the best of our knowledge) with respect to Fan et al. [2024] or Liang [2022] procedures.

A.4 Proof of Proposition 4 - local coverage

Before giving a detailed proof of our bounds, we first recall the definitions of the maximum mean discrepancy and the Hilbert-Schmidt independence criterion.

MMD and HSIC.

Definition 1 (Maximum Mean Discrepancy [Smola et al., 2007]) *Let X and Y be random vectors defined on a topological space $\mathcal{Z}$, with respective Borel probability measures P_X and P_Y. Let $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ be a kernel function and let $\mathcal{H}(k)$ be the associated reproducing kernel Hilbert space. The maximum mean discrepancy between P_X and P_Y is defined as*

$$\mathrm{MMD}_k(P_X, P_Y) = \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \mathbb{E}_{X \sim P_X}[f(x)] - \mathbb{E}_{Y \sim P_Y}[f(y)] \right|.$$

The squared MMD admits the following closed-form expression:

$$\begin{aligned}\mathrm{MMD}_k(P_X, P_Y)^2 = \mathbb{E}_{X \sim P_X, X' \sim P_X}[k(X, X')] + \mathbb{E}_{Y \sim P_Y, Y' \sim P_Y}[k(Y, Y')] \\ - 2\mathbb{E}_{X \sim P_X, Y \sim P_Y}[k(X, Y)],\end{aligned}$$

which can be estimated thanks to U- or V-statistics.

Now given a pair of random vectors $(U, V) \in \mathcal{X} \times \mathcal{Y}$ with probability distribution P_{UV}, we define the product RKHS $\mathcal{H} = \mathcal{F} \times \mathcal{G}$ with kernel $k_{\mathcal{H}}((u, v), (u', v')) = k_{\mathcal{X}}(u, u')k_{\mathcal{Y}}(v, v')$. A measure of the dependence between U and V can then be defined as the distance between the mean embedding of P_{UV} and $P_U \otimes P_V$, the joint distribution with independent marginals P_U and P_V:

$$\mathrm{MMD}^2(P_{UV}, P_U \otimes P_V) = \|\mu_{P_{UV}} - \mu_{P_U} \otimes \mu_{P_V}\|^2_{\mathcal{H}}.$$

This measure is the so-called *Hilbert-Schmidt independence criterion* (HSIC, see Gretton et al. [2005]) and can be expanded as

$$\begin{aligned}\mathrm{HSIC}(U, V) &= \mathrm{MMD}^2(P_{UV}, P_U \otimes P_V) \\ &= \mathbb{E}_{U,U',V,V'} k_{\mathcal{X}}(U, U')k_{\mathcal{Y}}(V, V') \\ &+ \mathbb{E}_{U,U'} k_{\mathcal{X}}(U, U')\mathbb{E}_{V,V'} k_{\mathcal{Y}}(V, V') \\ &- 2\mathbb{E}_{U,V}\left[\mathbb{E}_{U'} k_{\mathcal{X}}(U, U')\mathbb{E}_{V'} k_{\mathcal{Y}}(V, V')\right]\end{aligned}$$

where (U', V') is an independent copy of (U, V). Once again, the reproducing property implies that HSIC can be expressed as expectations of kernels, which facilitates its estimation when compared to other dependence measures such as the mutual information.

Bounds on local coverage. To lighten notations, we denote $X = X_{N+1}$, $Y = Y_{N+1}$, $R = |Y - \widehat{m}_{\mathcal{D}_N}(X)|$, $V = \widehat{f}_{\mathcal{D}_N}(X)$ and $S = R/\sqrt{V}$. In the frame of Proposition 4, we work conditionally on $\mathcal{D}_N$, which means that in what follows $\widehat{m}_{\mathcal{D}_N}(\cdot)$ and $\widehat{f}_{\mathcal{D}_N}(\cdot)$ are deterministic functions. The chain rule for mutual information gives

$$\begin{aligned}\mathrm{MI}((X, V), R) &= \mathrm{MI}(X, R) + \mathrm{MI}(V, R|X) \\ &= \mathrm{MI}(V, R) + \mathrm{MI}(X, R|V).\end{aligned}$$

Conditionally on X, V is constant and then R and V are independent. This implies $\mathrm{MI}(V, R|X) = 0$ and

$$\mathrm{MI}(X, R) - \mathrm{MI}(V, R) = \mathrm{MI}(X, R|V).$$

We now write

$$\begin{aligned}\mathrm{MI}(X, S) &= \mathrm{MI}(X, \frac{R}{\sqrt{V}}) \\ &\leq \mathrm{MI}(X, (R, V)) \quad (\forall g,\ \mathrm{MI}(g(X), Y) \leq \mathrm{MI}(X, Y)) \\ &\leq \mathrm{MI}((X, V), (R, V)) \quad (\mathrm{MI}((X_1, X_2), Y) \geq \mathrm{MI}(X_1, Y)) \\ &= \mathrm{MI}(X, R|V) + H(V) \quad (\mathrm{MI}(X, Y|Z) = \mathrm{MI}(X, Z), (Y, Z) - H(Z)) \\ &\leq \mathrm{MI}(X, R|V) + H(X) \quad (\forall g,\ H(g(X)) \leq H(X)) \\ &= \mathrm{MI}(X, R) - \mathrm{MI}(V, R) + H(X)\end{aligned}$$

and we can observe that only $\text{MI}(V, R)$ depends on V. We thus deduce that

$$1 - \exp(-\text{MI}(X, S)) \leq 1 - \alpha_1 \exp(\text{MI}(V, R))$$

where $\alpha_1 = \exp(-\text{MI}(X, R) - H(X))$ is independent from V, which proves Equation (11).

For the second part of the proposition, from Equation (15) in Wang and Tay [2023], we have the bound

$$\text{TV}(\mathbb{P}, \mathbb{Q}) \geq \frac{1}{2\sqrt{M_k}} \text{MMD}_k(\mathbb{P}, \mathbb{Q})$$

where $\text{TV}(\mathbb{P}, \mathbb{Q}) = \sup_{A \in \mathcal{F}} |\mathbb{P}(A) - \mathbb{Q}(A)|$ for $\mathbb{P}$, $\mathbb{Q}$ defined on a measurable space $(\Omega, \mathcal{F})$ and $\text{MMD}_k(\mathbb{P}, \mathbb{Q})$ are the total variation and the maximum mean discrepancy between probability distributions $\mathbb{P}$ and $\mathbb{Q}$, respectively. Here, the MMD depends on the choice of a kernel k, which is bounded by $M_k = \sup_{x \in \mathcal{X}} k(x, x)$, and must be characteristic for the inequality to hold. We then apply this inequality to $\mathbb{P} = P_{VR}$ the joint distribution of (V, R) and $\mathbb{Q} = P_V \otimes P_R$ the joint distribution with independent marginals P_V and P_R, to get

$$1 - \exp(-\text{MI}(V, R)) \geq \text{TV}^2(P_{VR}, P_V \otimes P_R) \geq \alpha_2 \text{HSIC}(V, R),$$

where the inequality on the left is the Bretagnolle-Huber inequality, the inequality on the right comes from the HSIC definition $\text{HSIC}(X, Y) = \text{MMD}^2(P_{XY}, P_X \otimes P_Y)$ and we denote $\alpha_2 = 1/(4M_k)$ with k the kernel used in HSIC. We finally have

$$1 - \alpha_1 \exp(\text{MI}(V, R)) \leq 1 - \frac{\alpha_1}{1 - \alpha_2 \text{HSIC}(V, R)}$$

and Equation (12) follows.

Remark 3 *In our initial mutual information bound, we replace $H(V)$ by $H(X)$ in order to obtain in the end a bound which can be expressed only with HSIC. This may seem crude, and of course we could easily incorporate $H(V)$ in our criterion to get a sharper bound: however, our numerical experiments show that even without this term the criterion yields satisfying adaptivity.*

In order to illustrate the reason why we advocate using HSIC over MI, that is numerical stability, we reproduce here our experiment on test case 2. For several values of b, we compute both criteria for a grid of θ^f candidate values, with $n = 100$ and 10-fold cross-validation. Figure 5 shows that HSIC (left) allows to clearly discriminate the lengthscales, while MI (right) suffers from estimation instability and is thus unusable in practice to identify a satisfying lengthscale.

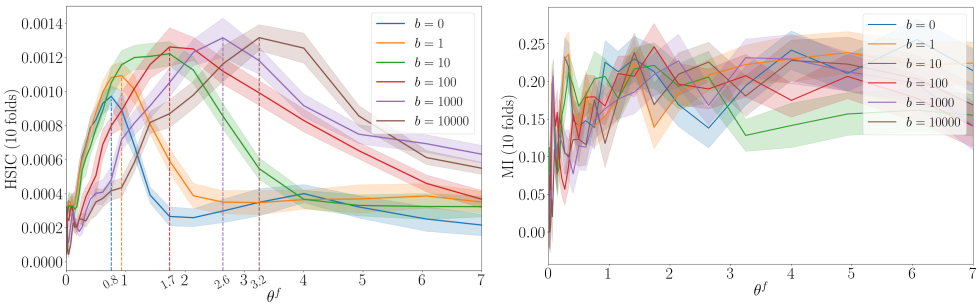


Figure 5: Test case 2 with $n = 100$. HSIC (left) and MI (right) criteria between $r(X, Y)$ and $f(X)$ as a function of b and θ^f (confidence intervals obtained by bootstrap and optimal values of θ^f in dashed lines).

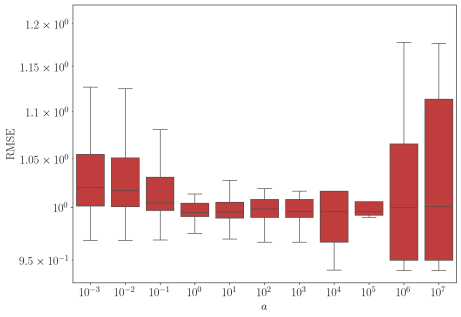
B Additional experiments and details

B.1 Hyperparameter influence

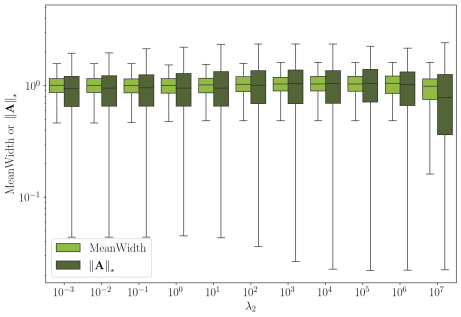
In order to quantitatively evaluate the influence of each hyperparameter in our kernel SoS formulation, we perform the following extensive study:

1. We select test cases 1, 2 and 3 (see Appendix B.4 for details)
2. For each test case:
   (a) We generate a training dataset of size $n = 50, 100$ for 3 different random seeds and 10 different values for θ^f
   (b) For each hyperparameter of interest a, b, λ_1 and λ_2
      i. We fix all hyperparameters to 1 except the one of interest which varies between 10^{-3} and 10^7
      ii. We solve the kernel SoS problem
      iii. We compute the root mean-squared error, the mean width, the nuclear norm and the Frobenius norm of the optimal solution
   (c) We normalize these indicators with their median over all combinations to be able to compare the different test cases on a common ground

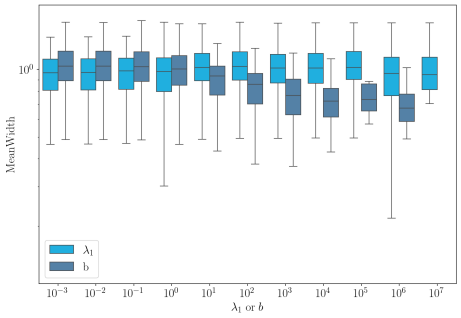
Boxplots of the combined indicators are given in Figure 6. We first observe that on average a does not have a large impact on the root mean-squared errors (top left). On the other hand, λ_2 does not influence at all the optimal Frobenius norm or the interval mean width (top right). However, the bottom row provides interesting insights. Indeed at first glance, since b and λ_1 both appear explicitly in the formula for the PSD matrix A, we may have intuitively thought that they should have a similar effect. Our results actually show that λ_1 has a very small influence, contrary to b which, when it increases, greatly reduces the interval width at a cost of inflating the nuclear norm.



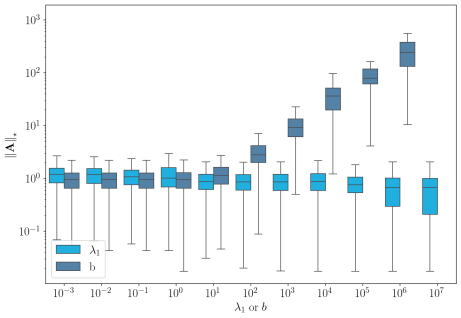
(a) Impact of a on RMSE



(b) Impact of λ_2 on mean interval width and $\|\mathbf{A}\|_\star$



(c) Impact of λ_1 and b on mean interval width



(d) Impact of λ_1 and b on $\|\mathbf{A}\|_\star$

Figure 6: Marginal impact of hyperparameters a, b, λ_1 and λ_2 over several values of θ^f, test cases and random seeds on rmse, mean interval width and regularization norms.

Note however that our conclusions related to a only apply to test cases with symmetric noise (which is the case for test cases 1, 2 and 3). In presence of asymmetry, it may be required to select $a > 0$ to ensure a satisfying estimation of the mean function. We give an illustration in Appendix B.4.

B.2 Cross-validation for kernel hyperparameter estimation and HSIC test of independence

Cross-validation. Let K be the number of folds. For $k \in [K]$, we write $\mathcal{D}_k$ the fold dataset k and $\mathcal{D}_{-k} = \mathcal{D}_n \setminus \mathcal{D}_k$. We denote by $\widehat{m}_{-k}, \widehat{f}_{-k}$ the mean and scaling functions trained on $\mathcal{D}_{-k}$ according to Equation (8). Define two sets,

$$R_K = \bigcup_{k=1}^{K} \{(Y_i - \widehat{m}_{-k}(X_i))^2\}_{i \in \mathcal{D}_k} \quad \text{and} \quad F_K = \bigcup_{k=1}^{K} \{\widehat{f}_{-k}(X_i)\}_{i \in \mathcal{D}_k}.$$

We seek

$$\max_{\theta^f \in \mathbb{R}^d} \quad \widehat{\text{HSIC}}\left(R, F\right), \tag{17}$$

where $\widehat{\text{HSIC}}(R, F)$ is estimated with samples R_K and F_K. In all our experiments, we use the energy distance kernel $k(x, x') = |x| + |x'| - |x - x'|$, which has been shown to be characteristic by Sejdinovic et al. [2013]. In practice, since HSIC is estimated with samples, the objective function of this optimization problem is noisy: consequently we resort to the BOBYQA optimization algorithm [Powell et al., 2009].

Once the lengthscales θ^f are selected, we train the scores on the whole pre-training dataset to learn $\widehat{m}_{\mathcal{D}_n}$ and $\widehat{f}_{\mathcal{D}_n}$. From there, we use the calibration set following the split conformal procedure. Our complete algorithm is summarized in Algorithm 1.

Algorithm 1: Split CP with adaptive kernel SoS

Data: $\mathcal{D}_N = \mathcal{D}_n \cup \mathcal{D}_m$, the pre-training and calibration datasets.
Input: $(a, b, \lambda_1) \in \mathbb{R}_+^3,\ \lambda_2 > 0$, the error-rate $\alpha \in]0, 1[$ and choose kernels k^m and k^f.

1 Fit a GP on $\mathcal{D}_n$ with estimated nugget effect. Retrieve estimated lengthscales θ^m and set $s = \|m_{\text{GP}}\|^2$
2 Solve Equation (17) to estimate θ^f
3 Solve Equation (8) to get $\widehat{m}_{\mathcal{D}_n}$ and $\widehat{f}_{\mathcal{D}_n}$
4 Compute the scores $\{s_i\}_{i \in \mathcal{D}_m} = \{(Y_i - \widehat{m}_{\mathcal{D}_n}(X_i))^2 / \widehat{f}_{\mathcal{D}_n}(X_i)\}_{i \in \mathcal{D}_m}$
5 Compute the adjusted level quantile $\widehat{q}_\alpha(\{s_i\}_{i \in \mathcal{D}_m})$ of these scores

Output: $\widehat{C}_{\mathcal{D}_N}(\cdot) = \left[\widehat{m}_{\mathcal{D}_n}(\cdot) \pm \sqrt{\widehat{q}_\alpha \widehat{f}_{\mathcal{D}_n}(\cdot)}\right]$

HSIC test of independence. Solving Equation (17) will always yield an optimized value for θ^f. However in practice, the maximum attainable value of $\widehat{\text{HSIC}}\left(R, F\right)$ may be small, thus suggesting that the residuals and the width of the most adaptive prediction bands are independent. In such situations, assuming that the noise is homoscedastic is natural, which leads us to design constant prediction bands by choosing an arbitrary large value for θ^f (we use $\theta^f = 20$ in our experiments). The key point is to detect that $\widehat{\text{HSIC}}\left(R, F\right)$ is sufficiently close to 0 in a meaningful statistical way: to do this we use the HSIC test of independence introduced by Gretton et al. [2007]. The null distribution of this test needs to be approximated, we use a permutation procedure. From a practical viewpoint and for numerical stability, we repeat the test 10 times and compute the median of the p-values.

Experiments with naive lengthscales values. We also investigated the performance of naive choices for θ^f. The first one is to choose $\theta^f = \theta^m$, where θ^m is the lengthscales for the mean function. The second naive choice is to take θ^f equal to the median of the feature distances (a usual rule-of-thumb - ROT below - for kernel methods). We report below, in Table 2, the mean width for the two naive choices and for the lengthscales that maximize the HSIC (Opt. HSIC). We can see that HSIC-optimized lengthscales always perform the best in terms of mean width, although with

ties with naive choices in some instances. We also report the local coverage for all cases in Table 2. HSIC-optimized lengthscales are the only one to consistently perform well across all cases.

	Mean-Width		
	θ^m	ROT	Opt. HSIC
Case 1	$\mathbf{2.862} \pm 0.123$	$\mathbf{2.898} \pm 0.155$	$\mathbf{2.877} \pm 0.209$
Case 2	2.260 ± 0.076	$\mathbf{2.032} \pm 0.110$	$\mathbf{2.052} \pm 0.103$
Case 3	1.482 ± 0.061	$\mathbf{1.399} \pm 0.095$	$\mathbf{1.385} \pm 0.080$
Case 4	4.185 ± 0.239	4.227 ± 0.254	$\mathbf{4.063} \pm 0.159$

Table 2: Comparison of mean width for different choices of θ^f.

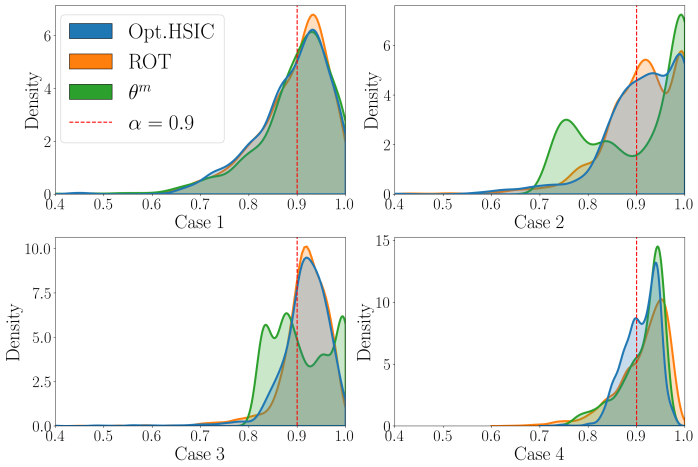


Figure 7: Local coverage for different choices of θ^f.

B.3 Implementation details

This section provides additional details about the SDP and dual solvers used throughout the experiments.

SDP solver. The SDP problem is solved with the SCS algorithm O'Donoghue [2021], O'Donoghue et al. [2023] available in the convex optimization software `CVXPY` Diamond and Boyd [2016], Agrawal et al. [2018]. An example of script that implements the SDP problem defined by Equation (7) is shown below. Here, `vector_variable` and `matrix_variable` denote the unknowns $\boldsymbol{\gamma} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{S}_+^n$. The remaining variables, such as the Gram matrix `Km_train` ($\mathbf{K}^m$) and the feature vector `phi_variance_gram_matrix` ($\boldsymbol{\Phi}(\mathbf{X})$) have to be computed by the user using the training set and the chosen kernel functions.

```
import numpy as np
import cvxpy as cp
import scs

# Variables
matrix_variable = cp.Variable((n, n), symmetric=True)
vector_variable = cp.Variable((n, d))

# Expressions
f_A = cp.reshape(
    cp.diag(phi_variance_gram_matrix.T @ matrix_variable @
        phi_variance_gram_matrix),
    (-1, 1),
    order="C",
)
```

```
mean_estimator = cp.matmul(Km_train, vector_variable)

# Objective
objective = cp.Minimize(
    (a/n) * cp.sum_squares(y_train - mean_estimator)
    + (b/n) * cp.sum(f_A)
    + lambda1 * cp.trace(matrix_variable)
    + lambda2 * cp.square(cp.norm(matrix_variable, "fro"))
)

# Constraints
constraints = [
    matrix_variable >> 0,
    f_A >= (y_train - mean_estimator)**2,
    cp.quad_form(vector_variable, Km_train, assume_PSD=True) <= s,
]

# Problem definition
problem = cp.Problem(objective, constraints)

# Solve
problem.solve(solver=cp.SCS, verbose=False)
```

Listing 1: Solving the SDP problem with CVXPY and SCS.

Dual formulation algorithm. We optimize the dual objective (8) using a projected gradient method with Nesterov acceleration and leveraging forward-backward backtracking line-search on the smoothness parameter, see Barré et al. [2022], Truong and Nguyen [2021], enforcing feasibility by clipping the Lagrange multipliers multipliers and applying convergence checks on constraint satisfaction and duality gap. The gradient is computed analytically and exploits problem structure, including sparsity and Cholesky factorization for efficiency. In Figure 3 left, the time reported are computed with a tolerance of 0.01 for both the convergence and duality gap checks. The simulations were performed on an AMD Ryzen 7 9700X 8-Core Processor (3.80 GHz).

B.4 Additional numerical experiments

Formulation B versus formulation A. As mentioned in Section 2.2, Marteau-Ferey et al. [2020] provide two equivalent formulations of the kernel SoS problems in terms of a PSD matrix $\mathbf{B}$ or $\mathbf{A}$. Though theoretically equivalent, we propose to investigate their respective numerical efficiency. For different test cases, different lengthscales and different random seeds, we record the computational time of the SCS solver to obtain the optimal solution with the $\mathbf{B}$ formulation, the $\mathbf{A}$ formulation and with or without Frobenius regularization for both, see Figure 8. The latter serves as a numerical illustration of its interest from a computational perspective, beyond the theoretical one related to strong convexity.

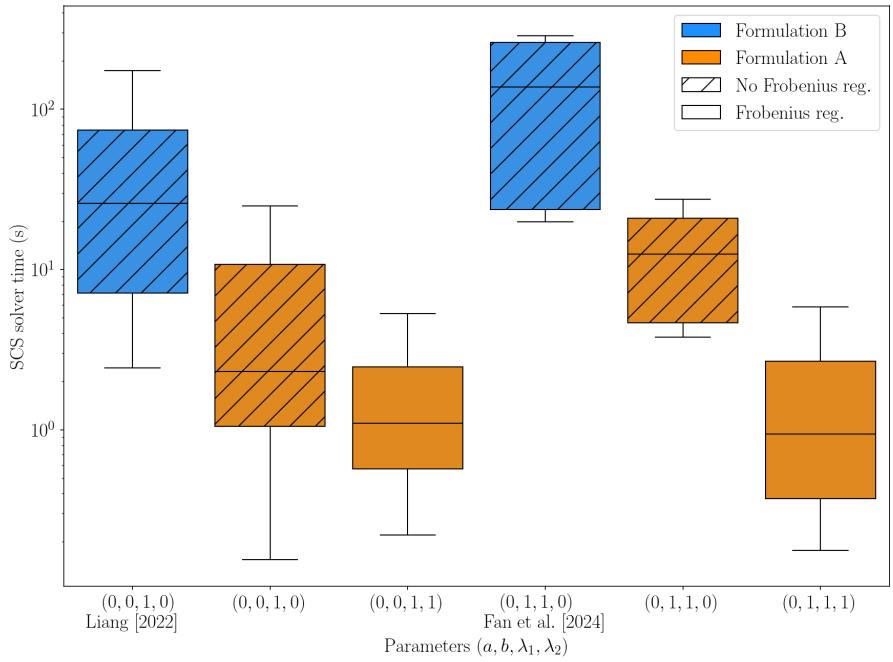


Figure 8: Time comparison between formulation A, B and with or without Frobenius regularization. Each model has been run for test cases 1, 2 and 3 with three random seeds, two sample sizes $n = 50$ and $n = 100$ and ten lengthscales between 0.1 and 1.0. The y-axis is plotted on a logarithmic scale.

In average for different hyperparameter combinations, we observe a significant reduction between both formulations, with formulation $\mathbf{A}$ yielding 90% computational savings. Setting $\lambda_2 > 0$ also improves resolution times, in particular when $b > 0$.

Evaluation metrics. For our experiments on analytical test cases, we compute the following metrics:

- The mean width of the prediction intervals on the test set
- The mutual information between X and $S(X, Y)$ on the test set
- The R^2_{SQI} criterion proposed in Deutschmann et al. [2024]. We first discretize the interval widths of test samples with quantiles, leading to a partition of the test set. For each partition, we compute the $(1 - \alpha)$- quantile of the absolute residuals and the median width: the R^2_{SQI} is the R^2 determination coefficient of the linear regression without intercept between these two
- The local coverage, obtained by approximating $\mathbb{P}(Y_{N+1} \in \widehat{C}(X_{N+1}) \mid X_{N+1} = x)$ by its empirical counterpart with samples from Y_{N+1} (of size n_Y) at different random locations X_{N+1} (of size n_X)

For real-world datasets, we do not compute the mutual information for numerical robustness due to the dimension of X, and cannot estimate the local coverage since we do not have access to multiple samples from Y_{N+1} for a given X_{N+1}. We then consider:

- The mean width of the prediction intervals on the test set as previously
- The worst-set coverage introduced by Thurin et al. [2025]. Starting from a partition $\{\mathcal{R}_l\}_{l=1,\dots,L}$ of the input space, we compute the marginal coverage in each region $\mathbb{P}(Y_{N+1} \in \widehat{C}_{\mathcal{D}_N}(X_{N+1}) | X_{N+1} \in \mathcal{R}_l)$. The worst-set coverage is defined as the minimum of all these coverages: the closer it is to the target α, the more adaptive the intervals. In practice, we follow the ideas of Thurin et al. [2025] to define the regions, with a procedure that may not yield a partition: we randomly select $L = 10$ samples from the test set, and for each of them we identity the 100-th closest neighbors in the feature space to estimate the marginal coverage.

We discuss in further detail the R^2_{SQI} criterion proposed in Deutschmann et al. [2024], using Figure 9 as a visual aid. This illustration excludes the CQR model due to the lack of a predictor for the mean function.

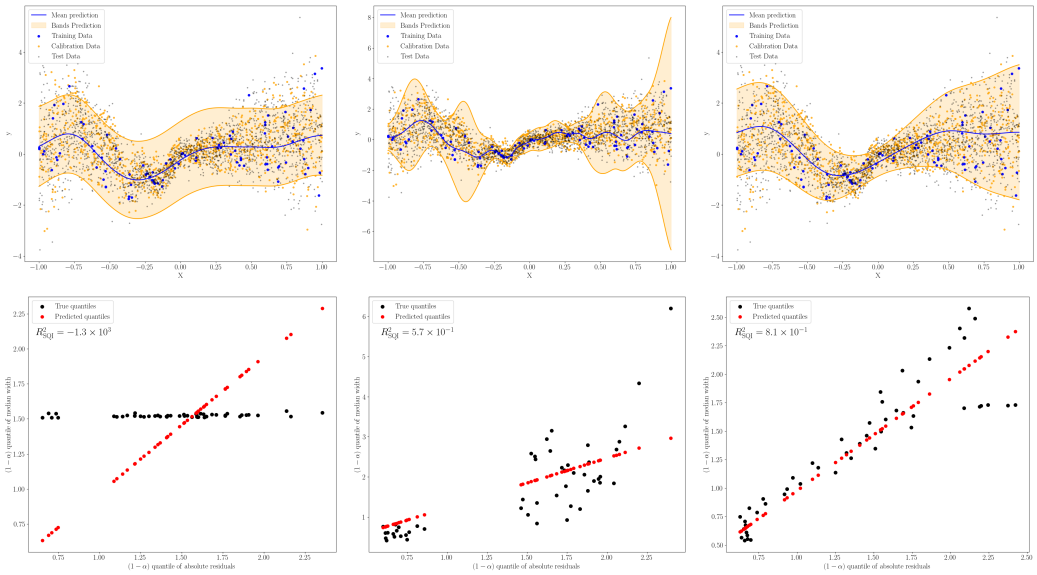


Figure 9: Test case 1 with $n = 100$, $n_X = 100$ and $n_Y = 1000$. We showcase prediction bands (top row) and the absolute residuals quantiles versus the median width quantiles (bottom row) for three models, homoscedastic GP (left column), heteroscedastic GP (middle column) and kernel SoS (right column). We use 50 quantiles to compute R^2_{SQI}.

For the homoscedastic GP model (left) we observe constant width of the prediction bands. Consequently, the median width quantiles form a constant line with respect to the absolute error quantiles. The linear model without intercept thus fits these data very poorly and results in a negative determination coefficient. This behavior is observed throughout cases 1, 2 and 3, where the homoscedastic GP always outputs constant prediction bands.

For the heteroscedastic GP model, we see that the prediction bands are more adaptive than the homoscedastic one, resulting in a positive determination coefficient. However, the linear regression is not perfect: we can spot some bands that overcover (top row, middle column), for $X \in (0.75, 1.0]$ and around $X = 0.5$ and $X = -0.5$. On the corresponding quantile plot, these points correspond to the mean width quantiles that are above the regression line in red. We see that the model mostly overcovers for points with high residuals. In addition, the model also undercovers in some regions, for instance at $X = -0.2$, which corresponds to the black points in the lower left of the quantile regression plot, where the absolute errors are very low, or at $X = -0.7$ and $X = -1$ where the absolute errors are larger. The latter correspond to the black points in the middle of the figure under the red regression line.

Finally, for the kernel SoS model, we observe a better linear regression model. In particular, prediction bands (top row, right column) are indeed narrower in the middle where the mean predictor makes smaller errors and the data are less noisy, and wider at both ends, where the data have more noise which results in a mean predictor with larger errors.

Additional analytical test cases and results. The complete list of test cases we consider is given below. Note that we only provide R^2_{SQI} for kernel SoS and the heteroscedastic GP: homoscedastic GP typically yields constant bands with largely negative determination coefficient, and CQR does not provide a mean function predictor. For all experiments related to adaptivity metrics, we perform 20 replications with different random seeds, and local coverage is estimated with $n_X = 100$ independent random locations X_{N+1} for which we generate $n_Y = 1000$ independent samples from Y_{N+1}. R^2_{SQI}, MI and mean width are estimated with a test set of size $n_{\text{test}} = 1000$. For kSoS, we train first an initial model with $\theta^f = 5$ and extract the values of the mean-width and the norms, which serve as a normalization before setting the hyperparameter values $\lambda_1 = \lambda_2 = 1$ and b (depending on the test case).

Case 1. Inspired from Gramacy and Lee [2009].

$$
\begin{aligned}
&d = 1, \quad X \sim \mathcal{U}[-1, 1], \quad Y = m(X) + \sigma(X)\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1) \\
&m(X) = \begin{cases} \sin(\pi(2X + 1/5)) + 0.2\cos(4\pi(2X + 1/5)) & \text{if } 10X + 1 \leq 9.6 \\ X - 9/10 & \text{otherwise} \end{cases} \\
&\sigma(X) = \sqrt{0.1 + 2X^2}
\end{aligned}
$$

We report below in Figure 10 and 11 the results from the main paper, with the additional R^2_{SQI}.

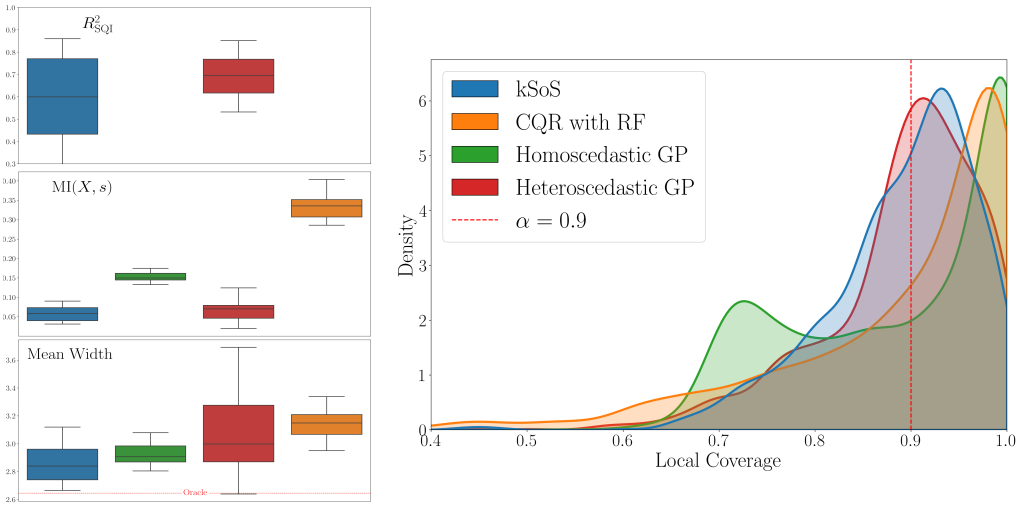


Figure 10: Test case 1 with $d = 1$ and $n = 100$. Adaptivity metrics and density of local coverage for $a = 0$ and $b = 10$.

Case 2. Corresponds to setting 1 in Hore and Barber [2024].

$$
\begin{aligned}
&X \sim \mathcal{N}_d(0, I_d), \quad Y = m(X) + \sigma(X)\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1) \\
&m(X) = 0.5 \sum_{i=1}^{d} X^{(i)} \\
&\sigma(X) = \sum_{i=1}^{d} |\sin(X^{(i)})|
\end{aligned}
$$

We start by setting $d = 1$ and display the adaptivity metrics in Figure 12. For this test case, we observe that all methods tend to produce prediction bands that are too large (hence a local coverage

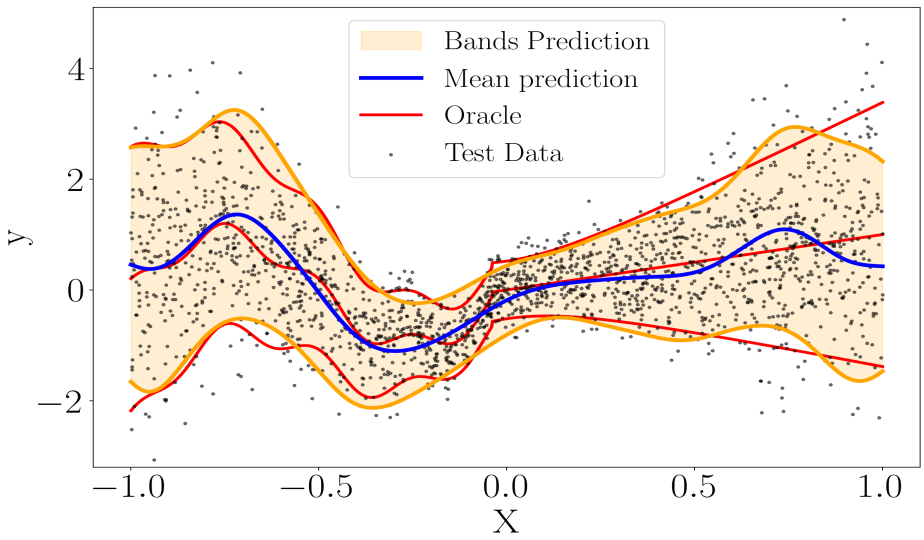


Figure 11: Test case 1 with $d = 1$ and $n = 2000$. Optimal solution of dual formulation for $a = b = 0$, $\lambda_1 = \lambda_2 = 1$ and $\theta^f = 0.3$.

density leaning towards 1), with a notable exception for kernel SoS. Although MI and R^2_{SQI} are similar for the heteroscedastic GP and kernel SoS, the latter has intervals with much smaller mean width.

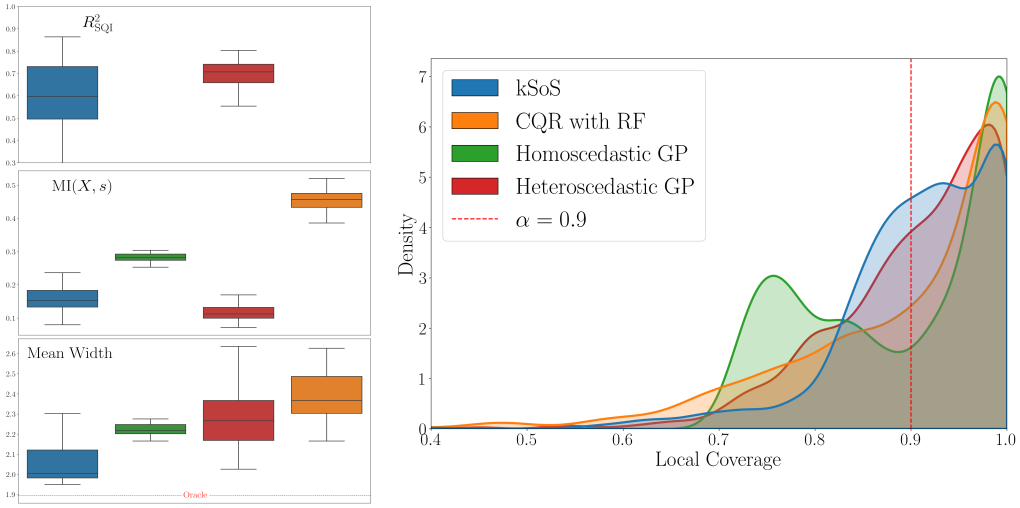


Figure 12: Test case 2 with $d = 1$ and $n = 100$. Adaptivity metrics and density of local coverage for $a = 0$ and $b = 10$.

Figure 13 shows the optimal solution of our dual formulation for $n = 2000$.

Remark 4 *We do not investigate this test case in larger dimension, since by concentration due to the additive structure, the intervals tend to be constant when d increases. The same comment applies to test case 3. For brevity, we thus postpone the investigation of such a setting to test case 4.*

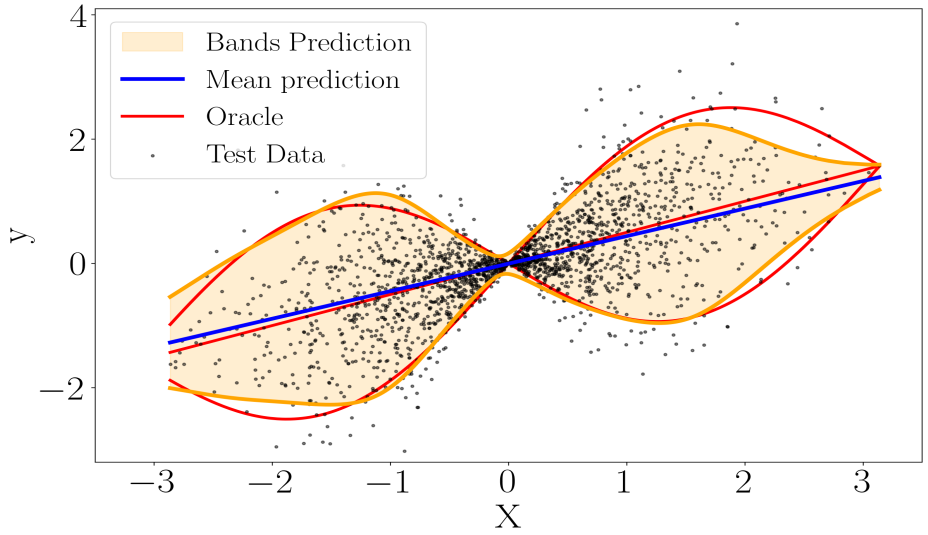


Figure 13: Test case 2 with $d = 1$ and $n = 2000$. Optimal solution of dual formulation for $a = 0$, $b = 1000$, $\lambda_1 = \lambda_2 = 1$ and $\theta^f = 1.2$.

Case 3. Corresponds to setting 2 in Hore and Barber [2024].

$$X \sim \mathcal{N}_d(0, I_d), \quad Y = m(X) + \sigma(X)\epsilon, \quad \epsilon \sim \mathcal{N}(0,1)$$

$$m(X) = 0.5 \sum_{i=1}^{d} X^{(i)}$$

$$\sigma(X) = \sum_{i=1}^{d} \frac{4}{3}\phi(\frac{2X}{3}), \ \phi: \text{ pdf of standard Gaussian}$$

For $d = 1$, the homoscedastic GP has poor local coverage, but with small prediction bands, as can be seen in Figure 14. Kernel SoS and the heteroscedastic GP have highly similar characteristics, with excellent local coverage as well as low MI and intervals with small mean width.

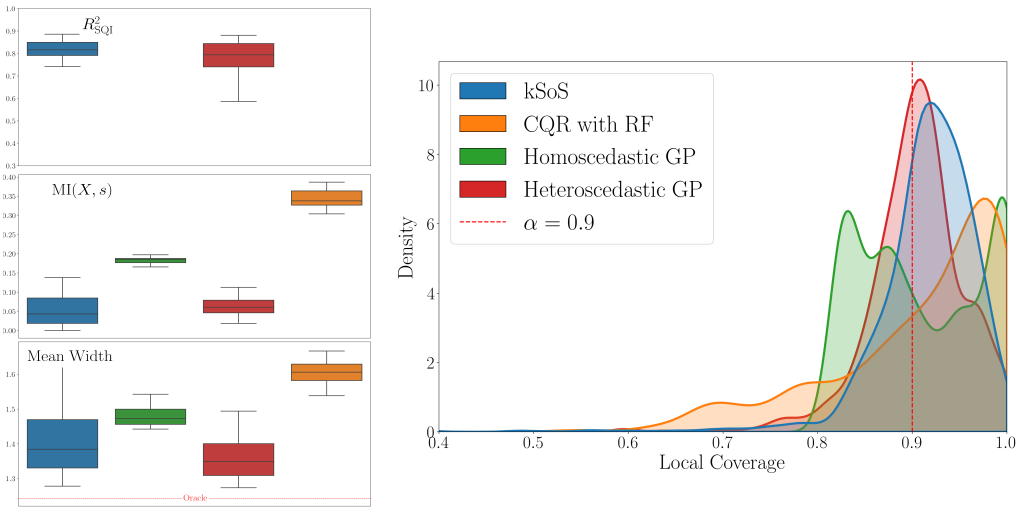


Figure 14: Test case 3 with $d = 1$ and $n = 100$. Adaptivity metrics and density of local coverage for $a = 0$ and $b = 10$.

With $n = 2000$, we obtain in Figure 15 the following optimal solution of the dual formulation.

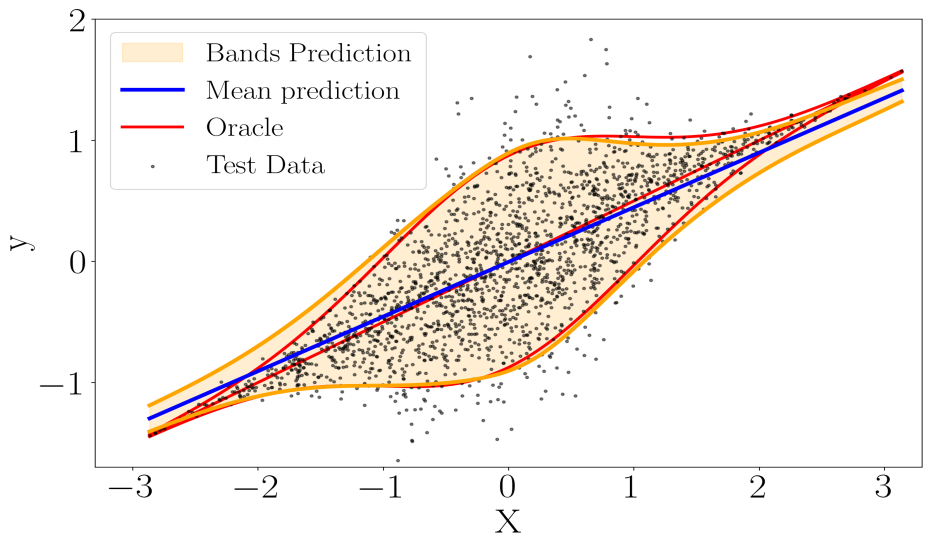


Figure 15: Test case 3 with $d = 1$ and $n = 2000$. Optimal solution of dual formulation for $a = b = 0$, $\lambda_1 = \lambda_2 = 1$ and $\theta^f = 0.9$.

Case 4. Inspired from Kivaranovic et al. [2020].

$$X \sim \mathcal{U}[0,1]^d, \quad Y = m(X) + \sigma(X)\epsilon, \quad \epsilon \sim \mathcal{N}(0,1)$$
$$m(X) = 2\sin(\pi\beta^\top X) + \pi\beta^\top X$$
$$\sigma(X) = \sqrt{1 + (\beta^\top X)^2}$$

For this test case, the oracle prediction bands are actually close to be constant: since we have seen previously that R^2_{SQI} is not a relevant indicator in such situation, we discard it from our analysis. In dimension $d = 1$ we set $\beta = 1$ and obtain the remaining adaptivity metrics given in Figure 16.

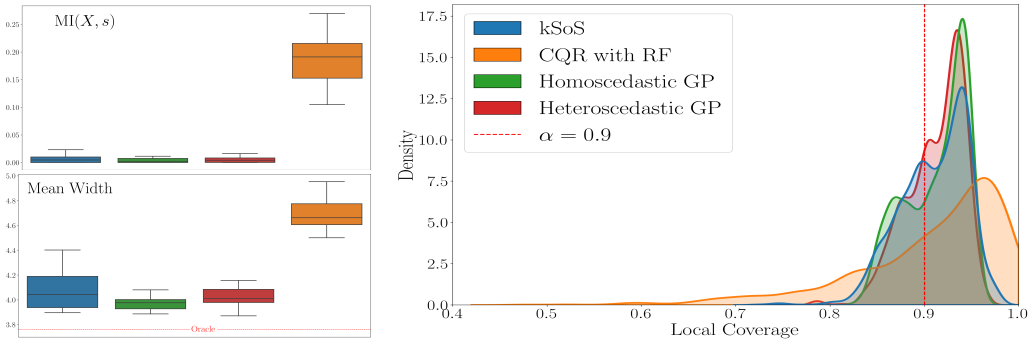


Figure 16: Test case 4 with $d = 1$ and $n = 100$. Adaptivity metrics and density of local coverage for $a = b = 0$.

As expected, homoscedastic GP, which produces almost constant intervals, performs the best in this setting. Except CQR, all methods yield similar MI, as well as similar satisfying local coverage. But this time, kSoS tend to select larger intervals than GPs: the good performance of heteroscedastic GP actually comes from the fact that it often prefers a homoscedastic model during fitting. We give in Figure 17 the optimal solution of the dual formulation obtained with $n = 2000$.

We now turn to the same test case in dimension $d = 5$ with $n = 150$ and $\beta = (1, 0.1, 0.1, 0.1, 0.1)$. Figure 18 shows a comparison with all competitors in terms of adaptivity metrics, but observe that we also do not compute MI since estimation is not robust due to the dimension.

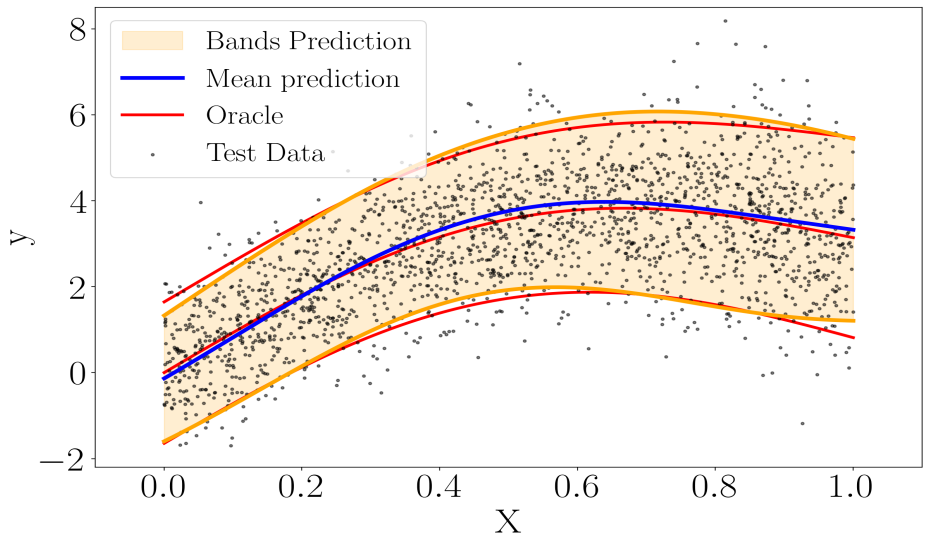


Figure 17: Test case 4 with $d = 1$ and $n = 2000$. Optimal solution of dual formulation for $a = b = 0$, $\lambda_1 = \lambda_2 = 1$ and $\theta^f = 1.6$.

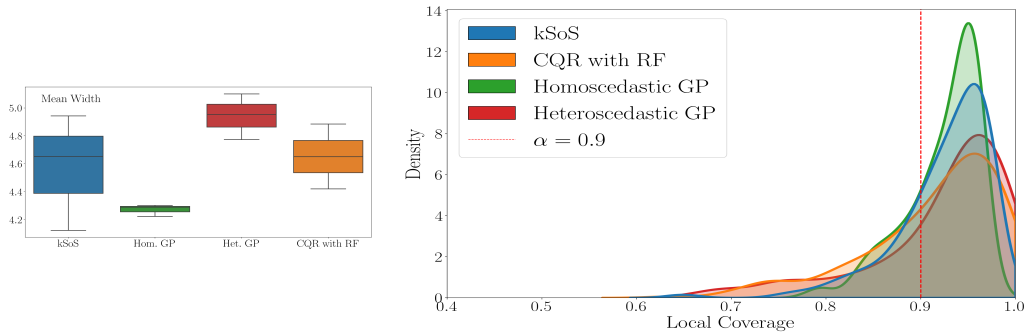


Figure 18: Test case 4 with $d = 5$ and $n = 150$. Adaptivity metrics and density of local coverage.

We can first remark that all methods tend to overcover more than in dimension $d = 1$. But similarly to $d = 1$, here the homoscedastic GP is also the best model in terms of local coverage and mean width. Interestingly, the performance of heteroscedastic GP and CQR degrades, while kSoS still achieves local coverage as good as homoscedastic GP, although with larger intervals on average.

Additional real-world datasets and results. We also investigate six real-world datasets:

1. Three of them (Concrete [Yeh, 1998], Bike [Fanaee-T, 2013] and Bio [Rana, 2013]) are taken from Romano et al. [2019].
2. The last three are standard regression datasets: Diabetes [Efron et al., 2004], Housing [Pace and Barry, 1997] and MPG [Quinlan, 1993].

They vary in terms of number of features and total sample size, see Table 3 for a detailed description. Note that for some of them, we perform a preliminary pre-processing step, by removing outliers (with a heteroscedastic GP model) and removing inactive features (identified with a heteroscedastic GP model and automatic relevance determination [Rasmussen and Williams, 2005]). For Housing specifically, we remove censored data (target ≥ 5) and apply a logarithm transformation of the target.

Dataset	Nb features	Total sample size	n_{train}	n_{cal}	n_{test}	Outliers
Concrete	8	1030	412	412	206	30%
Bike	(18) 13	10886	1000	1000	1000	20%
Bio	(9) 4	45730	1000	1000	1000	NA
Diabetes	10	442	101	170	171	NA
MPG	7	398	100	146	146	NA
Housing	8	20640	1000	1000	1000	NA

Table 3: Description of six real-world datasets: number of features (before and after filtering, if applicable), total sample size, training set sample size, calibration set sample size, test set sample size, percentage of removed outliers in training set.

Each experiment is repeated 10 times, where we randomly sample the training, calibration and test datasets. As for analytical test cases, we compare CQR, homoscedastic GP, heteroscedastic GP, and an isotropic kSoS with two choices for θ^f: the optimal value identified with our HSIC criterion, and the value that yields the smallest mean width. See Table 4 for details on the selection of hyperparameters for both choices: since θ^f optimized with HSIC varies along the experiments, we do not report its value. Note also that for the Bio dataset, which exhibits asymmetric noise, we use the extended variant of kSoS presented in the next paragraph, and thus has not an HISC-optimized version (since our criterion is only valid for symmetric intervals). We also use the kSoS formulation with known mean, given by a homoscedastic GP. To supplement the mean widths given in Table 1, we also provide in Table 5 the estimated marginal coverage on the test set. As expected from theory, all methods achieve the target coverage ($\alpha = 0.9$ here).

Dataset	kSoS Best mean width		kSoS Opt. HSIC
	θ^f	b	b
Concrete	3	10	10
Bike	7	10	10
Bio	2	10000	–
Diabetes	0.3	100	10
MPG	10	10	100
Housing	20	10	100

Table 4: Hyperparameters for six real-world datasets and the choices of lengthscales.

To complement the mean width analysis of each method in terms of adaptivity, we also compute in Figure 19 the worst-set coverage on these datasets except for MPG and Diabetes (very small test set size) and Bio (asymmetric noise). The mean widths are recalled below in Table 6.

Dataset	CQR	Het GP	Hom GP	kSoS Best mean width	kSoS Opt. HSIC
Concrete	0.902 ± 0.020	0.898 ± 0.020	0.898 ± 0.015	0.900 ± 0.034	0.899 ± 0.034
Bike	0.904 ± 0.012	0.903 ± 0.018	0.905 ± 0.012	0.901 ± 0.012	0.900 ± 0.012
Bio	0.902 ± 0.013	0.900 ± 0.013	0.900 ± 0.016	0.905 ± 0.012	–
Diabetes	0.899 ± 0.037	0.896 ± 0.032	0.904 ± 0.025	0.892 ± 0.027	0.886 ± 0.036
MPG	0.889 ± 0.037	0.913 ± 0.026	0.899 ± 0.028	0.895 ± 0.029	0.892 ± 0.03
Housing	0.897 ± 0.011	0.893 ± 0.012	0.900 ± 0.017	0.898 ± 0.016	0.899 ± 0.019

Table 5: Estimated marginal coverage on the test test for six real-world datasets (mean±sd on 10 repetitions).

Dataset	CQR	Het GP	Hom GP	kSoS Best mean width	kSoS Opt. HSIC
Concrete	0.586 ± 0.032	$\mathbf{0.508} \pm 0.052$	0.543 ± 0.044	0.556 ± 0.044	0.568 ± 0.06
Bike	1.114 ± 0.062	1.000 ± 0.079	$\mathbf{0.809} \pm 0.024$	$\mathbf{0.803} \pm 0.031$	$\mathbf{0.803} \pm 0032$
Housing	1.816 ± 0.045	1.585 ± 0.099	$\mathbf{1.453} \pm 0.068$	$\mathbf{1.468} \pm 0.094$	1.586 ± 0.104

Table 6: Mean width of prediction intervals for three real-world datasets (mean±sd on 10 repetitions).

In average, CQR and HSIC-optimized kSoS have better worst-set coverage than both GPs, but CQR systematically yields intervals with much larger mean widths. As for the comparison between kSoS and GPs:

- Housing (Figure 19, left). Homoscedastic GP and kSoS focusing on mean width exhibit the lowest mean width, meaning that they clearly outperform heteroscedastic GP since they all have similar worst-set coverage. But they do not reach the target of $\alpha = 0.9$. On the contrary, kSoS optimized with HSIC stays closer to the target, with a mean width in line with heteroscedastic GP.
- Concrete (Figure 19, middle). In terms of mean width, heteroscedastic GP is the clear winner. However, its worst-set coverage is around 0.87. Both kSoS variants have a larger mean width, but their worst-set coverage reach the target. On this test case, homoscedastic GP has coverage similar to heteroscedastic GP but with a larger mean width.
- Bike (Figure 19, right). All methods have similar worst-set coverage in average, while the lowest mean width is obtained with homoscedastic GP and kSoS.

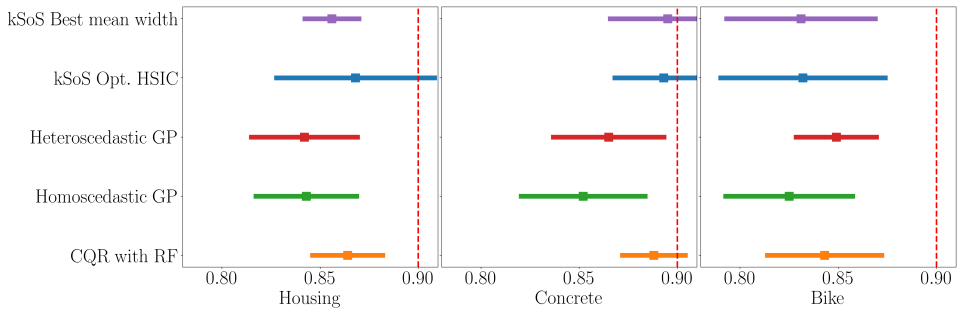


Figure 19: Housing ($b = 100$), Concrete ($b = 10$) and Bike ($b = 10$) dataset. Mean and standard deviation of worst-set coverage, 10 repetitions.

Non-symmetric intervals. We can readily extend our kernel SoS framework to non-symmetric prediction intervals of the form

$$\widehat{C}_N(X) = \left[\widehat{m}(X) - \widehat{f}_{\mathbf{A}^{\text{low}}}(X) - \widehat{q}_\alpha, \widehat{m}(X) + \widehat{f}_{\mathbf{A}^{\text{up}}}(X) + \widehat{q}_\alpha\right]$$

with score function $S(X,Y) = \max\left(m(X) - f_{\mathbf{A}^{\text{low}}}(X) - Y, Y - m(X) - f_{\mathbf{A}^{\text{up}}}(X)\right)$ inspired by CQR. In this setting, the kernel SoS problem writes

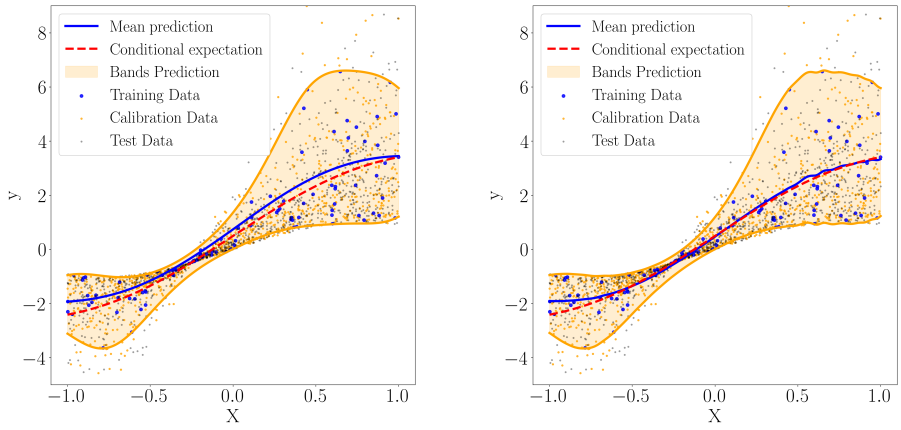
$$\begin{aligned}
\inf_{m\in\mathcal{H}^m,\, \mathcal{A}^{\text{low}},\, \mathcal{A}^{\text{up}}\in\mathcal{S}_+(\mathcal{H}^f)} \quad & \frac{a}{n}\sum_{i=1}^n \left(Y_i - m(X_i)\right)^2 + \frac{b}{n}\sum_{i=1}^n \left(f_{\mathcal{A}^{\text{low}}}(X_i) + f_{\mathcal{A}^{\text{up}}}(X_i)\right) \\
& + \lambda_1\|\mathcal{A}^{\text{low}}\|_\star + \lambda_2\|\mathcal{A}^{\text{low}}\|_F^2 + \lambda_1\|\mathcal{A}^{\text{up}}\|_\star + \lambda_2\|\mathcal{A}^{\text{up}}\|_F^2 \\
\text{s.t.} \quad & f_{\mathcal{A}^{\text{low}}}(X_i) \geq m(X_i) - Y_i,\ i\in[n]\,, \\
& f_{\mathcal{A}^{\text{up}}}(X_i) \geq Y_i - m(X_i),\ i\in[n]\,, \\
& \|m\|_{\mathcal{H}^m}^2 \leq s.
\end{aligned}$$

Contrary to the symmetric noise case, here the constraints no longer ensure a good fit of the regression function: this means that we need to set $a > 0$. Also remark that different RKHSs can be chosen for $\mathcal{A}^{\text{low}}$ and $\mathcal{A}^{\text{up}}$ in order to adapt to different regularities for the left and right tails of the conditional distribution.

As an illustration, we focus on a test case from Braun et al. [2025], which involves an exponentially distributed noise:

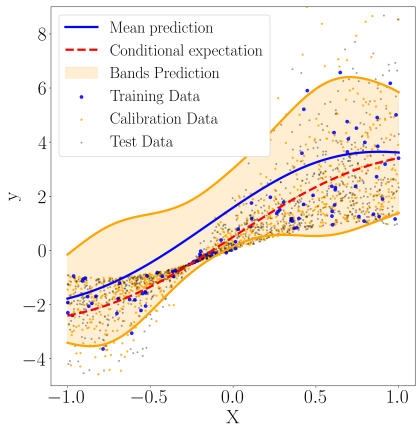
$$\begin{aligned}
& d = 1, \quad X \sim \mathcal{U}[-1,1], \quad Y = m(X) + \sigma(X)\epsilon, \quad \epsilon \sim \mathcal{E}(1) \\
& m(X) = \sin(2X) \\
& \sigma(X) = 0.5 + 2X
\end{aligned}$$

The optimal solution of the kSOS problem is given in Figure 20 for symmetric and non-symmetric intervals with $a = 0$ and $a = 1000$, for $n = 100$. First observe that when $a = 0$ the mean function is biased, unlike for $a = 1000$. In addition, breaking the symmetry clearly improves adaptivity.

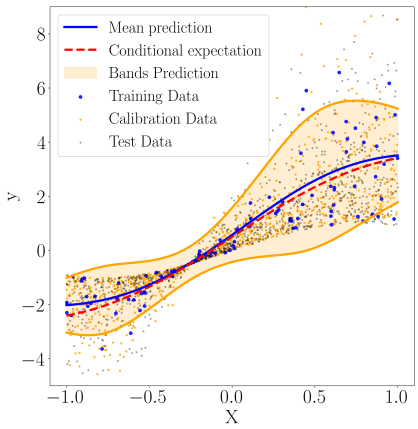


(a) Non-symmetric intervals and $a = 0$

(b) Non-symmetric intervals and $a = 1000$



(c) Symmetric intervals and $a = 0$



(d) Symmetric intervals and $a = 1000$

Figure 20: Test case 5. Optimal mean function and non-symmetric prediction intervals (up) and symmetric intervals (bottom) for $a = 0$ (left), $a = 1000$ (right), with parameters $b = 1$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\theta^f = 0.7$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We rely on numerical experiments in the main paper to support our claims, and also provide detailed proofs and additional experiments in the Appendix.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We comment on the limitations in several remarks and in the final section that guide towards future work. We also provide a discussion on computational efficiency and how it scales with sample size in the main paper and the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Complete proofs and assumptions are available in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We always explicitly mention the hyperparameter values used in all experiments, and also provide reproducible code in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The accompanying code can be used to reproduce all our results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide in the Appendix an entire section dedicated to all the experimental setting.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: When applicable, we always add error bars or boxplots on our experiments. We also describe the factors of variability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We detail in the Appendix the computer resources we used for all our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work has no potential harmful consequence, and as far as data are concerned, we only used synthetic and publicly available datasets.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We only use synthetic datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: We do not rely on existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: NA.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.