Simulator Calibration under Covariate Shift with Kernels

Keiichi Kisamori
NEC and AIST, Japan

Motonobu Kanagawa
EURECOM, France

Keisuke Yamazaki
AIST, Japan

Abstract

We propose a novel calibration method for computer simulators, dealing with the problem of covariate shift. Covariate shift is the situation where input distributions for training and test are different, and ubiquitous in applications of simulations. Our approach is based on Bayesian inference with kernel mean embedding of distributions, and on the use of an importance-weighted reproducing kernel for covariate shift adaptation. We provide a theoretical analysis for the proposed method, including a novel theoretical result for conditional mean embedding, as well as empirical investigations suggesting its effectiveness in practice. The experiments include calibration of a widely used simulator for industrial manufacturing processes, where we also demonstrate how the proposed method may be useful for sensitivity analysis of model parameters.

1 Introduction

Computer simulators are ubiquitous in many areas of science and engineering, examples including climate science, social science, and epidemics, to just name a few (Winsberg, 2010; Weisberg, 2012). Such tools are useful in understanding and predicting complicated time-evolving phenomena of interest. Computer simulators are also widely used in industrial manufacturing process modeling (Mourtzis et al., 2014), and we use one such simulator described in Fig. 1-(A), which models an assembling process of certain products in a factory, as our working example.

In this work we deal with the task of *simulator calibration* (Kennedy and O'Hagan, 2001), which is necessary to make simulation-based predictions reliable. To describe this, we introduce some notation used in the paper. We are interested in a system $R(x)$ that takes x as an input and output $y = R(x) + \varepsilon$ possibly corrupted by a noise ε. This system $R(x)$ is of interest but not known. Instead, we are given data $(X_i, Y_i)_{i=1}^n$ from the system, where input locations $X_1, \ldots, X_n$ are generated from a distribution $q_0(x)$ and outputs $Y_1, \ldots, Y_n$ from the target system $Y_i = R(X_i) + \varepsilon_i$. On the other hand, a simulator is defined as a function $r(x, \theta)$ that takes x as an input and outputs $r(x, \theta)$, where θ is a model parameter. The task of simulator calibration is to tune (or estimate) the parameter θ so that the

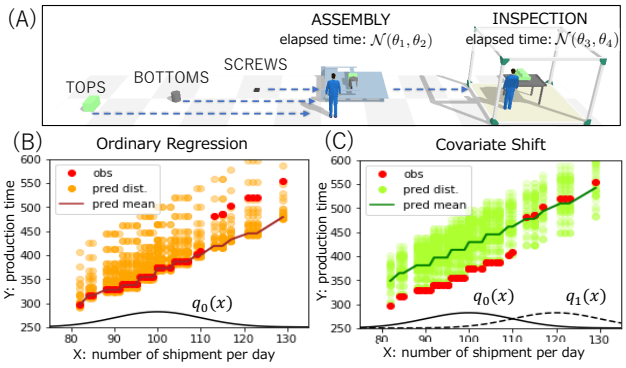

Figure 1: (A) Illustration of a manufacturing process simulator for assembling products. In the factory, one product is made from three items (TOPS, BOTTOMS and SCREWS) by the ASSEMBLY machine, and four such products are checked by the INSPECTION machine at the same time. Parameter θ of the simulation model $r(x, \theta)$ consists of 4 constants: mean θ_1 and variance θ_2 of the distribution of the processing time in the ASSEMBLY machine, and those (described as θ_3 and θ_4) in the INSPECTION machine. (B) Results of our method *without* covariate shift adaptation: training data (red points), generated predictive outputs (orange) and their means (brown curve). (C) Results of our method *with* covariate shift adaptation: training data (red points), generated predictive outputs (light green) and their means (green curve). $q_0(x)$ and $q_1(x)$ are input densities for training and prediction, respectively. More details in Secs. 1 and 5.2.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

$r(x, \theta)$ "approximates well" the unknown target system $R(x)$ by using the data $(X_i, Y_i)_{i=1}^n$. For instance, in Fig. 1, the target system $R(x)$ takes as an input the number x of required products to be manufactured in one day, and outputs the total time $y = R(x) + \varepsilon$ required for producing all the products; the simulator $r(x, \theta)$ models this process (see the "pred mean" curves in Fig. 1-(B)(C)).

There are mainly two challenges in the task of simulator calibration, which distinguish it from standard statistical learning problems. The first one owes to the complexity of the simulation model. Very often, a simulation model $r(x, \theta)$ cannot be written as a simple function of the input x and parameter θ, because the process of producing the output $y = r(x, \theta)$ may involve various numerical algorithms (e.g., solutions for differential equations) and/or IF-ELSE type decision rules of multiple agents. Therefore, one cannot access the gradient of the simulator output $r(x, \theta)$ with respect to the parameter θ, and thus calibration cannot reply on gradient-based methods for optimization (e.g., gradient descent) and sampling (e.g., Hamiltonian Monte Carlo). Moreover, one simulation $y = r(x, \theta)$ for a given input x can be computationally very expensive. Thus only a limited number of simulations can be performed for calibration. To summarise, the first challenge is that calibration should be done by only making use of forward simulations (or evaluations of $r(x, \theta)$), while the number of simulations cannot be large.

The second challenge is that of *covariate shift* (or *sample selection bias*) (Shimodaira, 2000; Sugiyama and Kawanabe, 2012), which is ubiquitous in applications of simulations, but has been rarely discussed in the literature on calibration methods. The situation is that the input distribution $q_1(x)$ for the test (or prediction) phase is *different* from the input distribution $q_0(x)$ generating the training input locations $X_1, \ldots, X_n$. In other words, the parameter θ is to be tuned so that the simulator $r(x, \theta)$ accurately approximates the target system $R(x)$ with respect to the distribution $q_1(x)$ (e.g., the error defined as $\int (R(x) - r(x, \theta))^2 q_1(x) dx$ is to be small), while training data $(X_i, Y_i)_{i=1}^n$ are only given with respect to another distribution $q_0(x)$.

The covariate shift setting is inherently important and ubiquitous in applications of computer simulation, because the purpose of a simulation is often in *extrapolation*. An illustrative example is climate simulations, where the aim is to answer whether global warming will occur in the future. As such, input x is a time point and the target system $R(x)$ is the global temperature. Calibration of the simulator $r(x, \theta)$ is to be done based on data from the past, but prediction is required for the future. This means that training input distribution $q_0(x)$ has a support in the past, but that of test $q_1(x)$ has a support on the future. For our working example in Fig. 1, training input locations $X_1, \ldots, X_n$ from $q_0(x)$ are more densely distributed in the region $x < 110$ than the region $x \geq 110$, since the data are obtained in a trial period. On the other hand, the test phase (i.e., when the factory is deployed) is targeted on mass production, and thus the test input distribution $q_1(x)$ has mass concentrated in the region $x \geq 110$.

Being a parametric model, a simulator only has a finite degree of freedom, and thus cannot capture all the aspects of the target system. Under such a model misspecification, the covariate shift is known to have a huge effect: the optimal model for the test input distribution may be drastically different from that for the training input distribution (Shimodaira, 2000). In climate simulations, care must be taken in how to tune the simulator as the data are only from the past; otherwise, the resulting predictions about the future will not be reliable (Winsberg, 2018). In the example of Fig. 1, the behavior of the target system $R(x)$ changes for the trial and test phases: Figs. 1-(B)(C) describe this situation. As can be seen in training data (red points), the total manufacturing time $R(x)$ becomes significantly larger when the number x of required products is greater than $x = 110$, because of of the overload of workers and machines. However, such structural change of the target $R(x)$ is not modeled in the simulator $r(x, \theta)$ (model misspecification). Thus, if calibration is done without taking the covariate shift into account, the resulting simulator makes predictions that fit well to the data in the region $x < 110$, but do not fit well in the region $x \geq 110$, as described in Fig. 1-(B).

Because of the first challenge of simulator calibration, exiting methods for covariate shift adaptation, which have been developed for standard statistical and machine learning approaches, cannot be directly employed for the simulator calibration problem: see e.g., Shimodaira (2000); Yamazaki et al. (2007); Gretton et al. (2009); Sugiyama and Kawanabe (2012) and references therein. On the other hand, existing approaches to likelihood-free inference, such as Approximate Bayesian Computation (ABC) methods (e.g.Csilléry et al. (2010); Marin et al. (2012); Nakagome et al. (2013)), are applicable to simulator calibration, but they do not address the problem of covariate shift. Our approach combines these two approaches and thus enjoys the best of both worlds, offering a solution to the calibration problem with covariate shift adaptation.

This work proposes a novel approach to simulator calibration, dealing explicitly with the setting of covariate shift. Our approach is Bayesian, deriving a certain posterior distribution over the parameter space given

observed data. The proposed method is based on Kernel ABC (Nakagome et al., 2013; Fukumizu et al., 2013), which is an approach to ABC based on kernel mean embedding of distributions (Muandet et al., 2017), and a certain importance-weighted kernel that works for covariate shift adaptation. We provide a theoretical analysis of this approach, showing that it produces a distribution over the parameter space that approximates the posterior distribution in which the "observed data" is predictions from the model that minimises the importance-weighted empirical risk. In other words, the proposed method approximates the posterior distribution whose support consists of parameters such that the resulting simulator produces a small generalization error for the test input distribution. For instance, Fig. 1-(C) shows predictions obtained with our method, which fit well in the test region $x \geq 110$ as a result of covariate shift adaptation.

This paper is organized as follows. In Sec. 2, we briefly review the setting of covariate shift and the framework of kernel mean embedding. In Sec. 3, we present our method for simulator calibration with covariate shift adaptation, and in Sec. 4 we investigate its theoretical properties. In Sec. 5 we report results of numerical experiments that include calibration of the production simulator in Fig. 1, confirming the effectiveness of the proposed method. Additional experimental results and all the theoretical proofs are presented in Appendix.

2 Background

We here introduce some notation and definitions used in the paper, by reviewing the problem setting of covariate shift, and the framework of kernel mean embeddings.

2.1 Calibration under Covariate Shift

Let $\mathcal{X} \subset \mathbb{R}^{d_\mathcal{X}}$ with $d_\mathcal{X} \in \mathbb{N}$ be a measurable subset that serves as the input space for a target system and a simulator. Denote by $R : \mathcal{X} \to \mathbb{R}$ the regression function of the (unknown) target system, which is deterministic, and define the true data-generating process as

$$y(x) := R(x) + e(x), \tag{1}$$

where $e : \mathcal{X} \to \mathbb{R}$ is a (zero-mean) stochastic process that represent error in observations. Observed data $D_n := \{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathbb{R}$ are assumed to be generated from the process (1) as

$$X_1, \ldots, X_n \sim q_0 \ \text{(i.i.d.)}, \quad Y_i = y(X_i), \quad (i = 1, \ldots, n),$$

where q_0 is a probability density function on $\mathcal{X}$. We use the following notation to write the output values:

$$Y^n := (Y_1, \ldots, Y_n) \in \mathbb{R}^n.$$

Let $\Theta \subset \mathbb{R}^{d_\Theta}$ with $d_\Theta \in \mathbb{N}$ be a measurable subset that serves as a parameter space. Let

$$r : \mathcal{X} \times \Theta \to \mathbb{R}$$

be a (measurable) deterministic simulation model that outputs a real value $r(x, \theta) \in \mathbb{R}$ given an input $x \in \mathcal{X}$ and a parameter $\theta \in \Theta$. Assume that we have a prior distribution $\pi(\theta)$ on the parameter space Θ.

In the setting of *covariate shift*, the input distribution $q_1(x)$ in the test or prediction phase is different from that $q_0(x)$ for training data $X_1, \ldots, X_n$, while the input-output relationship (1) remains the same. Thus, the expected loss (or the generalization error) to be minimized may be defined as

$$\begin{aligned} L(\theta) &:= \int (y(x) - r(x, \theta))^2 \, q_1(x) dx \\ &= \int (y(x) - r(x, \theta))^2 \, \beta(x) q_0(x) dx, \end{aligned}$$

where $\beta : \mathcal{X} \to \mathbb{R}$ is the *importance weight* function, defined as the ratio of the two input densities:

$$\beta(x) := q_1(x)/q_0(x).$$

In this work, we assume for simplicity that importance weights $\beta(X_i)$ at training inputs $X_1, \ldots, X_n$ are known, or estimated in advance. The knowledge of the importance weights is available when $q_0(x)$ and $q_1(x)$ are designed by an experimenter. For estimation of the importance, we refer to Gretton et al. (2009); Sugiyama et al. (2012) and references therein.[1] Using the importance weights, the expected loss can be estimated as

$$L_n(\theta) := \frac{1}{n} \sum_{i=1}^n \beta(X_i) \left(Y_i - r(X_i, \theta)\right)^2. \tag{2}$$

Covariate shift has a strong inference of the generalization performance of an estimated model, when the true regression function $R(x)$ does not belong to the class of functions realizable by the simulation model $\{r(\cdot, \theta) \mid \theta \in \Theta\}$, i.e., when *model misspecification* occurs (Shimodaira, 2000; Yamazaki et al., 2007). Such a misspecification happens in practice, since the simulation model only has a finite degree of freedom, as the parameter space is finite dimensional. To obtain a model with a good prediction performance, one needs to use an importance-weighted loss like (2) for parameter estimation.

[1] Note that kernel mean matching (Gretton et al., 2009) is a method for estimating the importance weights $\beta(X_1), \ldots, \beta(X_n)$, while it is based on kernel mean embeddings as in our method. In this sense, that approach deals with a problem different from ours.

2.2 Kernel Mean Embedding of Distributions

This is a framework for representing probability measures as elements in an Reproducing Kernel Hilbert Space (RKHS). We refer to Muandet et al. (2017) and references therein for details.

Let Ω be a measurable space, $k : \Omega \times \Omega \to \mathbb{R}$ be a measurable positive definite kernel and $\mathcal{H}$ be its RKHS. In this framework, any probability measure P on Ω is represented as a Bochner integral

$$\mu_P := \int k(\cdot, \theta) dP(\theta) \in \mathcal{H},$$

which is called the *kernel mean* of P. Estimation of P can be carried out by that of μ_P, which is usually computationally and statistically easier, thanks to nice properties of the RKHS. Such a strategy is justified if the mapping $P \to \mu_P$ is injective, in which case μ_P maintains all information of P. Kernels satisfying this property are called characteristic, and examples of characteristic kernels on $\Omega = \mathbb{R}^d$ include Gaussian and Matérn kernels (Sriperumbudur et al., 2010).

3 Proposed Calibration Method

We present our approach to simulator calibration with covariate shift adaptation. We take a Bayesian approach, and our target posterior distribution is described in Sec. 3.1. The proposed approach consists of Kernel ABC using a certain importance-weighted kernel (Sec. 3.2) and posterior sampling with the kernel herding algorithm (Sec. 3.3).

3.1 Target Posterior Distribution

We define a vector-valued function $r^n : \Theta \to \mathbb{R}^n$ from the simulator $r(x, \theta)$ as

$$r^n(\theta) := (r(X_1), \ldots, r(X_n))^\top \in \mathbb{R}^n, \quad \theta \in \Theta. \tag{3}$$

Let $\mathrm{supp}(\pi)$ be the support of π. Define $\Theta^* \subset \mathrm{supp}(\pi)$ as the set of parameters that minimize the weighted square error, i.e., for all $\theta \in \Theta^*$ we have

$$\sum_{i=1}^n \beta(X_i)(Y_i - r(X_i, \theta^*))^2 = \min_{\theta \in \mathrm{supp}(\pi)} \sum_{i=1}^n \beta(X_i)(Y_i - r(X_i, \theta))^2. \tag{4}$$

We allow for Θ^* to contain multiple elements, but assume that they all give the same simulation outputs, which we denote by $r^* \in \mathbb{R}^n$:

$$r^* := r^n(\theta^*) = r^n(\tilde{\theta^*}), \quad \forall \theta^*, \tilde{\theta^*} \in \Theta^*. \tag{5}$$

Let $\vartheta \sim \pi$ be a random variable following π. Then $r^n(\vartheta)$ is also a random variable taking values in $\mathbb{R}^n$ and its distribution is the *push-forward measure* of π under the mapping r^n, denoted by $r^n\pi$. We write the distribution of the joint random variable

$$(\vartheta, r^n(\vartheta)) \in \Theta \times \mathbb{R}^n$$

as $P_{\Theta\mathbb{R}^n}$, and their marginal distributions on Θ and $\mathbb{R}^n$ as P_Θ and $P_{\mathbb{R}^n}$, respectively. Then by definition we have $P_\Theta = \pi$ and $P_{\mathbb{R}^n} = r^n\pi$. Let

$$\mathrm{supp}(P_{\mathbb{R}^n}) = \mathrm{supp}(r^n\pi) = \{r^n(\theta) \mid \theta \in \mathrm{supp}(\pi)\}$$

be the support of the push-forward measure, which is the range of the simulation outputs when the parameter is in the support of the prior.

We consider the conditional distribution on Θ induced from the joint distribution $P_{\Theta\mathbb{R}^n}$ by conditioning on $\boldsymbol{y} \in \mathrm{supp}(P_{\mathbb{R}^n})$, which we write

$$P_\pi(\theta | \boldsymbol{y}), \quad \boldsymbol{y} \in \mathrm{supp}(P_{\mathbb{R}^n}) \tag{6}$$

Note that, since the conditional distribution on $\mathbb{R}^n$ given $\theta \in \Theta$ is the Dirac distribution at $r^n(\theta)$, one cannot use Bayes' rule to define the conditional distribution. However, the conditional distribution (6) is well-defined as a *disintegration*, and is uniquely determined up to an almost sure equivalence with respect to $P_{\mathbb{R}^n}$ (Chang and Pollard, 1997, Thm. 1 and Example 9); see also Cockayne et al. (2017, Sec. 2.5).

It will turn out in Sec. 4 that our approach provides an estimator for the kernel mean of the conditional distribution (6) with $\boldsymbol{y} = r^*$:

$$P_\pi(\theta | r^*) \tag{7}$$

where r^* is the outputs of the optimal simulator (5). In other words, (7) is the posterior distribution on the parameters, given that the optimal outputs r^* are observed. Sampling from (7) thus amounts to sampling parameters that provide the optimal simulation outputs.

Finally, we define a predictive distribution of outputs y for any input point $x \in \mathcal{X}$ as the push-forward measure of the posterior (7) under the mapping $r(x, \cdot) : \theta \to r(x, \theta)$, which we denote by

$$P_\pi(y | x, r^*). \tag{8}$$

3.2 Kernel ABC with a Weighted Kernel

Let $k_\Theta : \Theta \times \Theta \to \mathbb{R}$ be a kernel on the parameter space and $\mathcal{H}_\Theta$ be its its RKHS. We define the kernel mean of the posterior (7) as

$$\mu_{\Theta|r^*} := \int k_\Theta(\cdot, \theta) dP_\pi(\theta | r^*) \in \mathcal{H}_\Theta, \tag{9}$$

We propose to use the following weighted kernel on $\mathbb{R}^n$ defined from importance weights. As mentioned, we assume that the importance weight function $\beta(x) = q_1(x)/q_0(x)$ is known or estimated in advance. For $Y^n, \tilde{Y}^n \in \mathbb{R}^n$, the kernel is defined as

$$k_{\mathbb{R}^n}(Y^n, \tilde{Y}^n) = \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n \beta(X_i)(Y_i - \tilde{Y}_i)^2\right), \tag{10}$$

where $\sigma^2 > 0$ is a constant and a parameter of the kernel.

We apply Kernel ABC (Nakagome et al., 2013) with the importance-weighted kernel defined above, to estimate the posterior kernel mean (9). First, we independently generate $m \in \mathbb{N}$ parameters from the prior $\pi(\theta)$

$$\bar{\theta}_1, \ldots, \bar{\theta}_m \sim \pi.$$

Then for each parameter $\bar{\theta}_j, j = 1, \ldots, m$, we run the simulator to generate pseudo observations at $X_1, \ldots, X_n$:

$$\bar{Y}_j^n := r^n(\bar{\theta}_j), \quad j = 1, \ldots, m,$$

where $r^n : \Theta \to \mathbb{R}^n$ is defined in (3). Then an estimator of the kernel mean (9) is given by

$$\hat{\mu}_{\Theta|r^*} := \sum_{j=1}^m w_j k_\Theta(\cdot, \bar{\theta}_j) \ \in \mathcal{H}_\Theta, \tag{11}$$
$$(w_1, ..., w_m)^\top := (G + m\varepsilon I_m)^{-1}\mathbf{k}_{\mathbb{R}^n}(Y^n) \in \mathbb{R}^m,$$

where $I_m \in \mathbb{R}^{m\times m}$ is the identity and $\varepsilon > 0$ is a regularization constant; the vector $\mathbf{k}_{\mathbb{R}^n}(Y^n) \in \mathbb{R}^m$ and the Gram matrix $G \in \mathbb{R}^{m\times m}$ are computed from the kernel $k_{\mathbb{R}^n}$ in (10) with the observed data Y^n as

$$\begin{aligned}\mathbf{k}_{\mathbb{R}^n}(Y^n) &:= (k_{\mathbb{R}^n}(\bar{Y}_1^n, Y^n), \ldots, k_{\mathbb{R}^n}(\bar{Y}_m^n, Y^n))^\top \in \mathbb{R}^m \\ G &:= (k_{\mathbb{R}^n}(\bar{Y}_j^n, \bar{Y}_{j'}^n))_{j,j'=1}^m \in \mathbb{R}^{m\times m}.\end{aligned}$$

3.3 Posterior Sampling with Kernel Herding

We apply Kernel herding (Chen et al., 2010), a deterministic sampling method based on kernel mean embedding, to generate parameters $\check{\theta}_1, ..., \check{\theta}_m \in \Theta$ from the posterior kernel mean $\hat{\mu}_{\Theta|r^*}$ in (11). The procedure is as follows. The initial point $\check{\theta}_1$ is generated as $\check{\theta}_1 := \mathrm{argmax}_{\theta\in\Theta}\hat{\mu}_{\Theta|r^*}(\theta)$. Then the subsequent points $\check{\theta}_t$, $t = 2, \ldots, m$, are generated sequentially as

$$\check{\theta}_t := \operatorname*{argmax}_{\theta\in\Theta} \hat{\mu}_{\Theta|r^*}(\theta) - \frac{1}{t}\sum_{j=1}^{t-1} k_\Theta(\theta, \check{\theta}_j).$$

These points are a sample from the approximate posterior, in the sense that they satisfy $\|\hat{\mu}_{\Theta|r^*} - \frac{1}{t}\sum_{j=1}^t k_\Theta(\cdot, \check{\theta}_j)\|_{\mathcal{H}_\Theta} = O(t^{-1/2})$ under a mild condition (Bach et al., 2012).

Prediction. Let $x \in \mathcal{X}$ be any test input location, and recall that the predictive distribution $P_\pi(y|x, r^*)$ in (8) is defined as the push-forward measure of the posterior $P_\pi(\theta|r^*)$ under the mapping $r(x, \cdot)$. Therefore, predictive outputs can be obtained simply by running simulations with the posterior samples $\check{\theta}_1, \ldots, \check{\theta}_m$:

$$r(x, \check{\theta}_1), \ldots, r(x, \check{\theta}_m),$$

and the predictive distribution is approximated by the empirical distribution

$$\hat{P}_\pi(y|x, r^*) := \frac{1}{m}\sum_{j=1}^m \delta(y - r(x, \check{\theta}_j)),$$

where $\delta(\cdot)$ is the Dirac distribution at 0.

4 Theoretical Analysis

To analyze the proposed method, we first express the estimator (11) in terms of *covariance operators* on the RKHSs, which is how the estimator was originally proposed (Song et al., 2009; Nakagome et al., 2013). To this end, define joint random variables $(\vartheta, \boldsymbol{y}) \in \Theta \times \mathbb{R}^n$ by

$$\vartheta \sim \pi, \quad \boldsymbol{y} := r^n(\vartheta),$$

where $r^n : \Theta \to \mathbb{R}^n$ is defined in (3). Let $\mathcal{H}_\Theta$ and $\mathcal{H}_{\mathbb{R}^n}$ be the RKHSs of k_Θ and $k_{\mathbb{R}^n}$, respectively.

Covariance operators $C_{\vartheta\boldsymbol{y}} : \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_\Theta$ and $C_{\boldsymbol{yy}} : \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_{\mathbb{R}^n}$ are then defined as

$$\begin{aligned}C_{\vartheta\boldsymbol{y}} f &:= \mathbb{E}[k_\Theta(\cdot, \vartheta) f(\boldsymbol{y})] \in \mathcal{H}_\Theta, \quad f \in \mathcal{H}_{\mathbb{R}^n}, \\ C_{\boldsymbol{yy}} f &:= \mathbb{E}[k_{\mathbb{R}^n}(\cdot, \boldsymbol{y}) f(\boldsymbol{y})] \in \mathcal{H}_{\mathbb{R}^n}, \quad f \in \mathcal{H}_{\mathbb{R}^n}.\end{aligned}$$

Note that parameter-data pairs $(\bar{\theta}_j, \bar{Y}_j^n)_{j=1}^m = (\bar{\theta}_j, r^n(\bar{\theta}_j))_{j=1}^m \subset \Theta \times \mathbb{R}^n$ in Kernel ABC (Sec. 3.2) are i.i.d. copies of the random variables $(\vartheta, \boldsymbol{y})$. Thus empirical covariance operators $\hat{C}_{\vartheta\boldsymbol{y}} : \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_\Theta$ and $\hat{C}_{\boldsymbol{yy}} : \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_{\mathbb{R}^n}$ are defined as

$$\hat{C}_{\vartheta\boldsymbol{y}} f := \frac{1}{m}\sum_{j=1}^m k_\Theta(\cdot, \bar{\theta}_j) f(\bar{Y}_j^n), \quad f \in \mathcal{H}_{\mathbb{R}^n},$$

$$\hat{C}_{\boldsymbol{yy}} f := \frac{1}{m}\sum_{j=1}^m k_{\mathbb{R}^n}(\cdot, \bar{Y}_j^n) f(\bar{Y}_j^n), \quad f \in \mathcal{H}_{\mathbb{R}^n}.$$

The estimator (11) is then expressed as

$$\hat{\mu}_{\Theta|r^*} = \hat{C}_{\vartheta\boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}} + \varepsilon I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n). \tag{12}$$

See the above original references as well as Song et al. (2013); Fukumizu et al. (2013); Muandet et al. (2017) for the derivation.

Recall that Y^n is the observed data from the real process. The issue is that, in our setting, Y^n may *not* lie in the support of the distribution $P_{\mathbb{R}^n}$ of $\boldsymbol{y} = r^n(\vartheta)$, since the simulation model $r(\theta, x)$ is misspecified, i.e., there exists no $\theta \in \Theta$ such that $R(x) = r(x, \theta)$ for all $x \in \mathcal{X}$. The misspecified setting where $Y^n \notin \mathrm{supp}(P_{\mathbb{R}^n})$ has not been studied in the literature on kernel mean embeddings, and therefore existing theoretical results on conditional mean embeddings (Grünewälder et al., 2012; Fukumizu, 2015; Singh et al., 2019) are not directly applicable. Our theoretical contribution is to study the estimator (12) in this misspecified setting, which may be of general interest.

4.1 Projection and Best Approximation

Let $\mathcal{H}_{\boldsymbol{y}} \subset \mathcal{H}_{\mathbb{R}^n}$ be the Hilbert subspace of $\mathcal{H}_{\mathbb{R}^n}$ defined as the completion of the linear span of functions $k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n)$ with $\tilde{Y}^n$ from the support of $P_{\mathbb{R}^n}$:

$$\mathcal{H}_{\boldsymbol{y}} := \overline{\mathrm{span}\left\{k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n) \mid \tilde{Y}^n \in \mathrm{supp}(P_{\mathbb{R}^n})\right\}}, \quad (13)$$

where the closure is taken with respect to the norm of $\mathcal{H}_{\mathbb{R}^n}$. In other words, every $h \in \mathcal{H}_{\boldsymbol{y}}$ may be written in the form $h = \sum_{\ell=1}^{\infty} \alpha_\ell k_{\mathbb{R}^n}(\cdot, \tilde{Y}_\ell^n)$ for some $(\alpha_\ell)_{\ell=1}^{\infty} \subset \mathbb{R}$ and $(\tilde{Y}_\ell^n)_{\ell=1}^{\infty} \subset \mathrm{supp}(P_{\mathbb{R}^n})$ such that $\|h\|_{\mathcal{H}_{\mathbb{R}^n}}^2 = \sum_{\ell,j=1}^{\infty} \alpha_\ell \alpha_j k_{\mathbb{R}^n}(\tilde{Y}_\ell^n, \tilde{Y}_j^n) < \infty$.

Since $\mathcal{H}_y$ is a Hilbert subspace, one can consider the orthogonal projection of $k_{\mathbb{R}^n}(\cdot, Y^n)$, the "feature vector" of the observed data Y^n, onto $\mathcal{H}_{\boldsymbol{y}}$, which is uniquely determined and denoted by

$$h^* := \operatorname*{argmin}_{h \in \mathcal{H}_{\boldsymbol{y}}} \|h - k_{\mathbb{R}^n}(\cdot, Y^n)\|_{\mathcal{H}_{\mathbb{R}^n}}. \quad (14)$$

Then $k_{\mathbb{R}^n}(\cdot, Y^n)$ can be written as

$$k_{\mathbb{R}^n}(\cdot, Y^n) = h^* + h_\perp,$$

where $h_\perp \in \mathcal{H}_{\mathbb{R}^n}$ is orthogonal to $\mathcal{H}_{\boldsymbol{y}}$.

Note that the estimator (12) is an approximation to the following population expression:

$$C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{y}\boldsymbol{y}} + \varepsilon I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n). \quad (15)$$

Our first result below shows that (15) can be written in terms of the projection (14).

Lemma 1. *Let k_Θ be a bounded and continuous kernel and assume that $0 < \beta(X_i) < \infty$ holds for all $i = 1, \ldots, n$. Then* (15) *is equal to*

$$C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{y}\boldsymbol{y}} + \varepsilon I)^{-1} h^*$$

We make the following identifiability assumption. It is an assumption on the observed data Y^n (or the data generating process (1)), the simulation model $r(x, \theta)$ and the kernel $k_{\mathbb{R}^n}$ (or the importance weight function $\beta(x) = q_1(x)/q_0(x)$; see the definition of $k_{\mathbb{R}^n}$ in (10)).

Assumption 1. *There exists some $\tilde{Y}^n \in \mathrm{supp}(P_{\mathbb{R}^n})$ such that $k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n) = h^*$, where h^* is the orthogonal projection of $k_{\mathbb{R}^n}(\cdot, Y^n)$ onto the subspace $\mathcal{H}_{\boldsymbol{y}}$ in* (14).

The assumption states that the orthogonal projection of the feature vector $k_{\mathbb{R}^n}(\cdot, Y^n)$ of observed data Y^n onto $\mathcal{H}_{\boldsymbol{y}}$ lies in the set

$$\begin{aligned} &\{k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n) \mid \tilde{Y}^n \in \mathrm{supp}(P_{\mathbb{R}^n})\} \\ &\quad = \{k_{\mathbb{R}^n}(\cdot, r^n(\theta)) \mid \theta \in \mathrm{supp}(\pi)\}. \end{aligned}$$

Thus the assumption implies that the best approximation h^* of the observed data is given by the simulation model with some parameter $\theta^* \in \mathrm{supp}(\pi)$, i.e., $h^* = k_{\mathbb{R}^n}(\cdot, r^n(\theta^*))$. Such θ^* satisfies

$$\begin{aligned} \theta^* &\in \operatorname*{argmin}_{\theta \in \mathrm{supp}(\pi)} \|k_{\mathbb{R}^n}(\cdot, Y^n) - k_{\mathbb{R}^n}(\cdot, r(\cdot, \theta))\|_{\mathcal{H}_{\mathbb{R}^n}}^2 \\ &= \operatorname*{argmax}_{\theta \in \mathrm{supp}(\pi)} k_{\mathbb{R}^n}(Y^n, r(\cdot, \theta)) \\ &= \operatorname*{argmax}_{\theta \in \mathrm{supp}(\pi)} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \beta(X_i)(Y_i - r(X_i, \theta))^2\right) \\ &= \operatorname*{argmin}_{\theta \in \mathrm{supp}(\pi)} \sum_{i=1}^n \beta(X_i)(Y_i - r(X_i, \theta))^2, \end{aligned}$$

where the last identity follows from the exponential function being monotonically increasing. This shows that, under Assumption 1, the parameter θ^* realizing the projection is a least weighted-squares solution, and thus belongs to the set Θ^* defined in (4). Moreover, since h^* is uniquely determined, so is the simulation outputs $r^* := r^n(\theta^*)$, in the sense of (5).

By these arguments, Lemma 1 and Assumption 1 lead to the following result.

Theorem 1. *Suppose that the assumptions in Lemma 1 and Assumption 1 hold. Let $r^* := r^n(\theta^*)$ where θ^* is any element satisfying* (4). *Then* (15) *is equal to*

$$C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{y}\boldsymbol{y}} + \varepsilon I)^{-1} k_{\mathbb{R}^n}(\cdot, r^*).$$

Theorem 1 suggests that the estimator (12) would behave as if the observed data is the optimal simulation outputs r^* obtained as a best approximation for the given data Y^n. The convergence result presented below shows that this is indeed the case.

To state the result, we define a function $G : \mathrm{supp}(P_{\mathbb{R}^n}) \times \mathrm{supp}(P_{\mathbb{R}^n}) \to \mathbb{R}$ as

$$\begin{aligned} G(Y_a^n, Y_b^n) &:= \mathbb{E}[k_\Theta(\vartheta, \vartheta) | \boldsymbol{y} = Y_a^n, \boldsymbol{y}' = Y_b^n], \quad (16) \\ &= \mathbb{E}[k_\Theta(\vartheta, \vartheta) | r^n(\vartheta) = Y_a^n, r^n(\vartheta') = Y_b^n], \end{aligned}$$

where $(\vartheta', \boldsymbol{y}')$ is an independent copy of $(\vartheta, \boldsymbol{y})$.

The following result shows that (12) (or (11)) is a consistent estimator of the kernel mean $\mu_{\Theta|r^*}$ (9) of the posterior $P_\pi(\theta|r^*)$. It is obtained by extending the result of Fukumizu (2015, Theorem 1.3.2) to the misspecified setting where $Y^n \notin \mathrm{supp}(P_{\mathbb{R}^n})$ by using Theorem1. The assumptions made are essentially the same those in Fukumizu (2015, Theorem 1.3.2). Below $\mathrm{Range}(C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}})$ denotes the range of the tensor-product operator $C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}}$ on the tensor-product RKHS $\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}$ (see Appendix for details).

Theorem 2. *Suppose that the assumptions in Lemma 1 and Assumption 1 hold. Assume that the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ of $C_{\boldsymbol{yy}}$ satisfy $\lambda_i \leq \beta i^{-b}$ for all $i \in \mathbb{N}$ for some constants $\beta > 0$ and $b > 1$, and that the function G in (16) satisfies $G \in \mathrm{Range}(C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}})$. Let $C > 0$ be any fixed constant, and set the regularization constant $\varepsilon := \varepsilon_m := Cm^{-\frac{b}{1+4b}}$ of $\hat{\mu}_{\Theta|r^*}$ in (12) (or (11)). Then we have*

$$\left\| \hat{\mu}_{\Theta|r^*} - \mu_{\Theta|r^*} \right\|_{\mathcal{H}_\Theta} = O_p\left(m^{-\frac{b}{1+4b}} \right) \quad (m \to \infty).$$

5 Experiments

We first explain the setting common for all the experiments. In each experiment, we consider both regression problems with and without covariate shift, to see whether the proposed method can deal with covariate shift. In the latter case, which we call "ordinary regression," we set the importance weights to be constant, $\beta(X_i) = 1$ $(i = 1, ..., n)$. The noise process $e(x)$ in (1) is independent Gaussian $\varepsilon \sim N(0, \sigma^2_{\mathrm{noise}})$. We write $N(a, b)$ for the normal distribution with mean a and variance b; the multivariate version is denoted similarly.

For the proposed method, we used a Gaussian kernel $k_\Theta(\theta, \theta') = \exp(-\|\theta - \theta'\|^2/2\sigma^2_\Theta)$ for the parameter space, where $\sigma^2_\Theta > 0$ is a constant. We set the constants $\sigma^2, \sigma^2_\Theta > 0$ in the kernels $k_{\mathbb{R}^n}$ and k_Θ by the median heuristic (e.g. Garreau et al., 2018) using the simulated pairs $(\bar{\theta}_j, \bar{Y}^n_j)_{j=1}^m$.

For comparison, we used Markov Chain Monte Carlo (MCMC) for posterior sampling, more specifically the Metropolis-Hastings (MH) algorithm. For this competitor, we assume that the noise process $e(x)$ in (1) is known, so that the likelihood function is available in MCMC (which is of the form $\exp(-\sum_{i=1}^n \beta(X_i)\left(Y_i - r(X_i, \theta)\right)^2 / 2\sigma^2_{\mathrm{noise}})$ up to constant). In this sense, we give an unfair advantage for MH over the proposed method, as the latter does not assume the knowledge of the noise process, which is usually not available in practice.

For evaluation, we compute Root Mean Square Error (RMSE) in prediction for each method (and for a different number of simulations, m) as follows. Test input locations $\tilde{X}_1, \ldots, \tilde{X}_n$ are generated from $q_0(x)$ in the case of ordinary regression, and from $q_1(x)$ in the covaraite shift setting. After sampling parameters $\check{\theta}_1, \ldots, \check{\theta}_m$ with the method for evaluation, the RMSE is computed as $(\frac{1}{n}\sum_{i=1}^n (R(\tilde{X}_i) - \frac{1}{m}\sum_{j=1}^m r(\tilde{X}_i, \check{\theta}_j))^2)^{1/2}$.

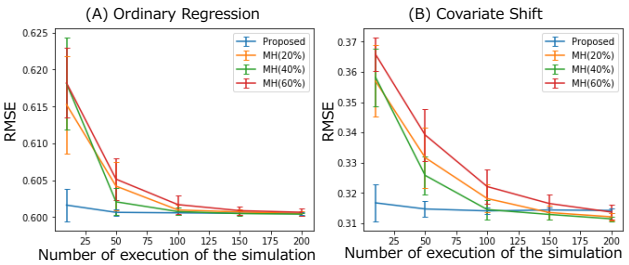

Figure 2: RMSEs for (A) ordinary and (B) covariate shift cases, as a function of the number m of simulations, given by the proposed method (blue) and the MH algorithm with different acceptance ratios about 20% (orange), 40% (red), and 60% (green).

5.1 Synthetic Experiments

We consider the problem setting of the benchmark experiment in Shimodaira (2000).

Setting. The input space is $\mathcal{X} = \mathbb{R}$, and the data generating process (1) is given by $R(x) = -x + x^3$ and $e(x) = \epsilon$ with $\epsilon \sim N(0, 2)$ being an independent noise. The simulation model is defined by $r(x, \theta) = \theta_0 + \theta_1 x$, where $\theta = (\theta_1, \theta_2)^\top \in \Theta = \mathbb{R}^d$. For demonstration, we treat this model as intractable, i.e., we assume that only evaluation of function values $r(x, \theta)$ is possible once x and θ are given. The input densities $q_0(x)$ and $q_1(x)$ for for training and prediction are those of $N(0.5, 0.5)$ and $N(0, 0.3)$, respectively. We define the prior as multivariate Gaussian $\pi = N(\mathbf{0}, 5I_2)$, where $I_2 \in \mathbb{R}^{2\times 2}$ is the identity. We set the size of training data $(X_i, Y_i)_{i=1}^n$ as $n = 100$.

Results. Figure 2 shows RMSEs for (A) ordinary regression and (B) covariate shift as a function of the number m of simulations, with the means and standard deviations calculated from 30 independent trials. For the proposed method, we set the regularization constant to be $\varepsilon = 1.0$. We set the proposal distribution of MH to be $N(\mathbf{0}, \sigma_p^2 I_2)$ with σ_p being 0.08, 0.06, and 0.03, which were tuned so that the acceptance ratios become about 20%, 40%, and 60% respectively. In the horizontal axis, the number of simulations for MH is the number of all MCMC steps (which all require running the simulator) including burn-in and rejected executions. For MH, we used the first 10% MCMC steps

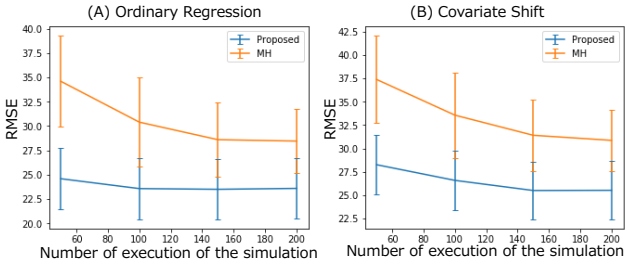

Figure 3: RMSEs in the (A) ordinary and (B) covariate shift settings, as a function of the number m of simulations, for the proposed method (blue) and MH (orange).

for burn-in, and excluded them for predictions. The results show that the proposed method is more efficient than MH, in the sense that it gives better predictions than MH based on a small number of simulations. This is a promising property, since real-world simulators are often computationally expensive, as is the case for the experiment in the next section.

5.2 Experiments on Production Simulator

We performed experiments on the manufacturing process simulator mentioned in Sec. 1 (Fig. 1), and a more sophisticated production simulator with 12 parameters. We only describe the former here, and report the latter in the Appendix due to the space limitation.

Setting. We used a simulator constructed with *WITNESS*, a popular software package for production simulation (`https://www.lanner.com/en-us/`). We refer to Sec. 1 for an explanation of the simulator. This simulator $r(x, \theta)$ has 4 parameters $\theta \in \Theta \subset \mathbb{R}^4$. The input space for regression is $\mathcal{X} = (0, \infty)$.

The data generating process (1) is defined as $R(x) = r(x, \theta^{(0)})$ for $x < 110$ and $R(x) = r(x, \theta^{(1)})$ for $x \geq 110$, where $\theta^{(0)} := (2, 0.5, 5, 1)^\top$ and $\theta^{(1)} := (3.5, 0.5, 7, 1)^\top$; the noise model is an independent noise $e(x) = \epsilon \sim N(0, 30)$. The input densities are defined as $q_0(x) = N(100, 10)$ (training) and $q_1(x) = N(120, 10)$ (prediction). We constructed this model so that the two regions $x < 110$ and $x \geq 110$ correspond to those for training and prediction, respectively, with $\theta^{(0)}$ and $\theta^{(1)}$ being the "true" parameters in the respective regions. We defined the prior $\pi(\theta)$ as the uniform distribution over $\Theta := [0, 5] \times [0, 2] \times [0, 10] \times [0, 2] \subset \mathbb{R}^4$. The size of training data $(X_i, Y_i)_{i=1}^n$ (which are described in Fig. 1 (B)(C) as red points) is $n = 50$.

Results. Figure 3 shows the averages and standard deviations of RMSEs for the proposed method and MH of 10 independent trials, changing the number m of simulations. We set the regularization constant of the proposed method as $\varepsilon = 0.01$, and the proposal distribution of MH as $N(\mathbf{0}, 0.03^2 I_4)$, which was tuned to make the acceptance about 40%.[2] The results show that the proposed method is more accurate than MH with a small number of simulations, even though the latter used the full knowledge of the data generating process (1).

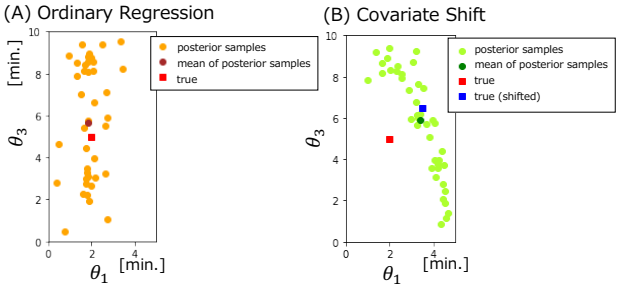

Figure 4: Parameters $\check{\theta}_1, \ldots, \check{\theta}_m$ generated from the proposed method, in the subspace of coordinates of θ_1 and θ_3. (A): Ordinary regression: the generated parameters (orange), the mean of them (brown), and the "true" parameter $\theta^{(0)}$ for the training region $x < 110$ (red). (B) Covariate shift: the generated parameters (light green), the mean of them (green), and the "true" parameter $\theta^{(1)}$ for the prediction region $x \geq 110$ (blue, "true shifted").

Fig. 4 (A) and (B) describe parameters $\check{\theta}_1, \ldots, \check{\theta}_m$ generated in one run of the proposed method in the ordinary and covariate shift settings, respectively; the corresponding predictive outputs are shown in Fig. 1 (B) and (C). In both settings, the estimated posterior mean is located near the "true" parameter of each scenario. Fig. 4 (A) and (B) also demonstrate how our method might be useful for sensitivity analysis. Our method generates parameters $\check{\theta}_1, \ldots, \check{\theta}_m$ so as to approximate the posterior $P_\pi(\theta | r^*)$, where r^* is "optimal" simulation outputs. Therefore, the more variation in the coordinate θ_1 indicates that the value of θ_1 is not very important to obtain optimal simulation outputs. But a comparison between (A) and (B) indicates that, under covariate shift, there should be small correlation between θ_1 and θ_3 to obtain optimal simulation outputs.

Acknowledgements

We would to thank the reviewers and the area chair for their constructive feedback.

[2] In this experiment one simulation is computationally expensive and takes about 2 seconds with the authors' PC, so we decided to only use this acceptance rate, given that the it performed the best in the previous experiment.

References

Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, pages 1359–1366.

Baker, C. R. (1973). Joint Measures and Cross-covariance Operators. *Transactions of the American Mathematical Society*, 186:273–289.

Caponnetto, A. and Vito, E. D. (2007). Optimal rates for regularized least-squares algorithm. *Found. Comput. Math. J.*, 7(4):331–368.

Chang, J. T. and Pollard, D. (1997). Conditioning as disintegration. *Statistica Neerlandica*, 51:287–317.

Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel-herding. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, pages 109–116.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2017). Bayesian probabilistic numerical methods. *ArXiv e-prints*, arXiv:1702.03673v2 [stat.ME].

Csilléry, K., Blum, M. G. B., Gaggiotti, O. E., and François, O. (2010). Approximate Bayesian Computation (ABC) in practice. *Trends in Ecology and Evolution*, 25(7):410–418.

Fukumizu, K. (2015). Nonparametric Bayesian Inference with Kernel Mean Embedding. In *Modern Methodology and Applications in Spatial-Temporal Modeling*, Springer Briefs in Statistics. Springer, Tokyo.

Fukumizu, K., Song, L., and Gretton, A. (2013). Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783.

Garreau, D., Jitkrittum, W., and Kanagawa, M. (2018). Large sample analysis of the median heuristic. *ArXiv*.

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Scholkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

Grünewälder, S., Lever, G., Baldassarre, L., Patterson, S., Gretton, A., and Pontil, M. (2012). Conditional mean embeddings as regressors. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, pages 1823–1830.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B*, 63(3):425–464.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Mourtzis, D., Doukas, M., and Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges. *Procedia CIRP*, 25(C):213–229.

Muandet, K., Fukumizu, K., Sriperumbudur, B. K., and Schölkopf, B. (2017). Kernel mean embedding of distributions : A review and beyonds. *Foundations and Trends in Machine Learning*, 10(1–2):1–141.

Nakagome, S., Fukumizu, K., and Mano, S. (2013). Kernel approximate Bayesian computation in population genetic inferences. *Statistical Applications in Genetics and Molecular Biology*, 12(6):667–678.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244.

Singh, R., Sahani, M., and Gretton, A. (2019). Kernel instrumental variable regression. In *NeurIPS*.

Song, L., Fukumizu, K., and Gretton, A. (2013). Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th International Conference on Machine Learning (ICML2009)*, pages 961–968.

Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. (2011). Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12:2389–2410.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *Jounal of Machine Learning Research*, 11:1517–1561.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Springer.

Sugiyama, M. and Kawanabe, M. (2012). *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density Ratio Estimation in Machine Learning*. Cambridge University Press, Cambridge, UK.

Weisberg, M. (2012). *Simulation and Similarity: Using Models to Understand the World*. Oxford University Press.

Winsberg, E. (2010). *Science in the Age of Computer Simulation*. University of Chicago Press.

Winsberg, E. (2018). *Philosophy and Climate Science*. Cambridge University Press.

Yamazaki, K., Kawanabe, M., Watanabe, S., Sugiyama, M., and Müller, K.-R. (2007). Asymptotic Bayesian generalization error when training and test distributions are different. *Proceedings of the 24th international conference on Machine learning - ICML '07*, pages 1079–1086.

Supplementary Materials

Simulator Calibration under Covariate Shift with Kernels

A Proofs

A.1 Proof of Lemma 1

First we note that from the assumption $0 < \beta(X_i) < \infty$ for all $i = 1, \ldots, n$, the importance-weighted kernel (10) is continuous on $\mathbb{R}^n$. Therefore Steinwart and Christmann (2008, Lemma 4.33) implies that the RKHS $\mathcal{H}_{\mathbb{R}^n}$ of $k_{\mathbb{R}^n}$ is separable.

To prove Lemma 1, we need the following result.

Lemma 2. *Suppose that the assumptions in Lemma 1 hold. Let $(\phi_i)_{i=1}^{\infty} \subset \mathcal{H}_{\mathbb{R}^n}$ be the eigenfunctions of the covariance operator $C_{\boldsymbol{yy}}$ associated with positive eigenvalues, and let $(\tilde{\phi}_j)_{j=1}^{\infty} \subset \mathcal{H}_{\mathbb{R}^n}$ be an ONB of the null space of $C_{\boldsymbol{yy}}$. Then $\tilde{\phi}_j(\tilde{Y}^n) = 0$ holds for $P_{\mathbb{R}^n}$-almost every $\tilde{Y}^n \in \mathbb{R}^n$.*

Proof. By definition of $\tilde{\phi}_j$, its holds that

$$0 = C_{\boldsymbol{yy}}\tilde{\phi}_j = \int k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n)\tilde{\phi}_j(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n) =: \int k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n)d\nu(\tilde{Y}^n),$$

where the measure ν is defined by $d\nu(\tilde{Y}^n) := \tilde{\phi}_j(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n)$. Since the kernel $k_{\mathbb{R}^n}$ is bounded on $\mathbb{R}^n$, $\mathcal{H}_{\mathbb{R}^n}$ consists of bounded functions, and thus $\tilde{\phi}_j \in \mathcal{H}_{\mathbb{R}^n}$ is bounded. Therefore ν a finite measure. But since $k_{\mathbb{R}^n}$ is a Gaussian kernel (see (10)), it is c_0-universal, and so Sriperumbudur et al. (2011, Proposition 2) and the integral being zero imply that ν is the zero measure. Thus for ν to be the zero measure, $\tilde{\phi}_j(\tilde{Y}^n) = 0$ should hold for $P_{\mathbb{R}^n}$-almost every $\tilde{Y}^n$, which concludes the proof.

□

We now prove Lemma 1.

Proof. Let $(\phi_i)_{i=1}^{\infty} \subset \mathcal{H}_{\mathbb{R}^n}$ be the eigenfunctions of the covariance operator $C_{\boldsymbol{yy}}$ associated with positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots > 0$, and let $(\tilde{\phi}_j)_{j=1}^{\infty} \subset \mathcal{H}_{\mathbb{R}^n}$ be an ONB of the null space of $C_{\boldsymbol{yy}}$. To prove the assertion, we first show that (a) $\langle \phi_i, h_\perp \rangle = 0$ for every ϕ_i, and that (b) $C_{\vartheta \boldsymbol{y}}\tilde{\phi}_j = 0$ for every $\tilde{\phi}_j$.

(a) By definition of ϕ_i, it can be written as

$$\phi_i = \lambda_i^{-1} C_{\boldsymbol{yy}}\phi_i = \lambda_i^{-1} \int k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n)\phi_i(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n).$$

Therefore,

$$\begin{aligned}
\langle \phi_i, h_\perp \rangle_{\mathcal{H}_{\mathbb{R}^n}} &= \left\langle \lambda_i^{-1} \int k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n)\phi_i(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n), h_\perp \right\rangle_{\mathcal{H}_{\mathbb{R}^n}} \\
&= \lambda_i^{-1} \int \left\langle k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n), h_\perp \right\rangle_{\mathcal{H}_{\mathbb{R}^n}} \phi_i(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n) = 0,
\end{aligned}$$

where the last identity follows from $\left\langle k_{\mathbb{R}^n}(\cdot, \tilde{Y}^n), h_\perp \right\rangle_{\mathcal{H}_{\mathbb{R}^n}} = 0$ for $\tilde{Y}^n \in \mathrm{supp}(P_{\mathbb{R}^n})$, which follows from the definition of $h_\perp$.

(b) We have

$$\begin{aligned}
C_{\vartheta \boldsymbol{y}}\tilde{\phi}_j &= \int k_\Theta(\cdot, \theta)\tilde{\phi}_j(\tilde{Y}^n)dP_{\Theta\mathbb{R}^n}(\theta, \tilde{Y}^n) \\
&= \int \left(\int k_\Theta(\cdot, \theta)dP_\pi(\theta|\tilde{Y}^n) \right) \tilde{\phi}_j(\tilde{Y}^n)dP_{\mathbb{R}^n}(\tilde{Y}^n) = 0,
\end{aligned}$$

where the last identity follows from Lemma 2.

We now prove the assertion. By using (a) and (b), we obtain

$$
\begin{aligned}
& C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n) \\
= \; & C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon I)^{-1}(h^* + h_\perp) \\
= \; & C_{\vartheta \boldsymbol{y}} \sum_{i=1}^{\infty} (\lambda_i + \varepsilon)^{-1} \langle h^*, \phi_i \rangle_{\mathcal{H}_{\mathbb{R}^n}} \phi_i + C_{\vartheta \boldsymbol{y}} \sum_{j=1}^{\infty} \varepsilon^{-1} \left\langle h^* + h_\perp, \tilde{\phi}_j \right\rangle_{\mathcal{H}_{\mathbb{R}^n}} \tilde{\phi}_j \\
= \; & C_{\vartheta \boldsymbol{y}} \sum_{i=1}^{\infty} (\lambda_i + \varepsilon)^{-1} \langle h^*, \phi_i \rangle_{\mathcal{H}_{\mathbb{R}^n}} \phi_i \\
= \; & C_{\vartheta \boldsymbol{y}} \sum_{i=1}^{\infty} (\lambda_i + \varepsilon)^{-1} \langle h^*, \phi_i \rangle_{\mathcal{H}_{\mathbb{R}^n}} \phi_i + C_{\vartheta \boldsymbol{y}} \sum_{j=1}^{\infty} \varepsilon^{-1} \left\langle h^*, \tilde{\phi}_j \right\rangle_{\mathcal{H}_{\mathbb{R}^n}} \tilde{\phi}_j \\
= \; & C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon I)^{-1} h^*,
\end{aligned}
$$

which completes the proof.

□

A.2 Proof of Theorem 2

Theorem 2 can be easily proven by combining the proof idea of Fukumizu (2015, Theorem 1.3.2) and Theorem 1, but for completeness we present the proof.

Before presenting, we introduce some notation and definitions. Below $\|A\|$ for an operator A denotes the operator norm. $\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}$ denotes the tensor-product RKHS of $\mathcal{H}_{\mathbb{R}^n}$ and $\mathcal{H}_{\mathbb{R}^n}$, which is the RKHS of the product kernel $k_{\mathbb{R}^n \times \mathbb{R}^n} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by $k_{\mathbb{R}^n \times \mathbb{R}^n}((Y_a^n, \tilde{Y}_a^n), (Y_b^n, \tilde{Y}_b^n)) = k_{\mathbb{R}^n}((Y_a^n, Y_b^n)) k_{\mathbb{R}^n}((\tilde{Y}_a^n, \tilde{Y}_b^n))$. $C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}} : \mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}$ is the covariance operator defined by

$$
C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}} F := \mathbb{E}[k_{\mathbb{R}^n \times \mathbb{R}^n}(\cdot, (\boldsymbol{y}, \boldsymbol{y}')) F(\boldsymbol{y}, \boldsymbol{y}')], \quad F \in \mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n},
$$

where $\boldsymbol{y}'$ is an independent copy of the random variable $\boldsymbol{y}$.

Note that the covariance operator $C_{\vartheta \boldsymbol{y}}$ satisfies $\langle C_{\vartheta \boldsymbol{y}} f, g \rangle_{\mathcal{H}_\Theta} = \mathbb{E}[f(\boldsymbol{y}) g(\vartheta)]$ for any $f \in \mathcal{H}_{\mathbb{R}^n}$ and $g \in \mathcal{H}_\Theta$. Similarly, $C_{\boldsymbol{yy}}$ satisfies $\langle C_{\boldsymbol{yy}} f, h \rangle_{\mathcal{H}_{\mathbb{R}^n}} = \mathbb{E}[f(\boldsymbol{y}) h(\boldsymbol{y})]$ for any $f, h \in \mathcal{H}_{\mathbb{R}^n}$, and $C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}}$ satisfies $\langle C_{\boldsymbol{yy}} F_a, F_b \rangle_{\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}} = \mathbb{E}[F_a(\boldsymbol{y}, \boldsymbol{y}') F_b(\boldsymbol{y}, \boldsymbol{y}')]$ for any $F_a, F_b \in \mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}$.

Proof. By the triangle inequality,

$$
\begin{aligned}
& \left\| \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n) - \mu_{\Theta|r^*} \right\|_{\mathcal{H}_\Theta} \\
\leq \; & \left\| \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n) - C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n) \right\|_{\mathcal{H}_\Theta} \\
& + \left\| C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, Y^n) - \mu_{\Theta|r^*} \right\|_{\mathcal{H}_\Theta} \\
\leq \; & \left\| \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} - C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} \right\| \left\| k_{\mathbb{R}^n}(\cdot, Y^n) \right\|_{\mathcal{H}_\Theta} \qquad (17) \\
& + \left\| C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, r^*) - \mu_{\Theta|r^*} \right\|_{\mathcal{H}_\Theta}, \qquad (18)
\end{aligned}
$$

where we used Theorem 1 in the last line. Below we derive convergence rates of the two terms (17)(18) separately, and then determine the decay schedule of ε_m as $m \to \infty$ so that the two terms have the same rate.

The first term (17)**.** We first have

$$\begin{aligned}
& \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1} \\
= & \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-\hat{C}_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1} \\
& +\hat{C}_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1} \\
= & \hat{C}_{\vartheta \boldsymbol{y}}\left[(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right] \\
& +(\hat{C}_{\vartheta \boldsymbol{y}}-C_{\vartheta \boldsymbol{y}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1} \\
= & \hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}(C_{\boldsymbol{yy}}-\hat{C}_{\boldsymbol{yy}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1} \\
& +(\hat{C}_{\vartheta \boldsymbol{y}}-C_{\vartheta \boldsymbol{y}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1},
\end{aligned}$$

where the last equality follows from the formula $A^{-1}-B^{-1}=A^{-1}(B-A)B^{-1}$ that holds for any invertible operators A and B. Note that $\hat{C}_{\vartheta \boldsymbol{y}}=\hat{C}_{\vartheta\vartheta}^{1/2} W_{\vartheta \boldsymbol{y}} \hat{C}_{\boldsymbol{yy}}^{1/2}$ holds for some $W_{\Theta\mathcal{F}}: \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_\Theta$ with $\|W_{\vartheta \boldsymbol{y}}\| \leq 1$ (Baker, 1973, Theorem 1). Using this, we have

$$\begin{aligned}
& \left\|\hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
\leq & \left\|\hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}(C_{\boldsymbol{yy}}-\hat{C}_{\boldsymbol{yy}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
& +\left\|(\hat{C}_{\vartheta \boldsymbol{y}}-C_{\vartheta \boldsymbol{y}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
= & \left\|\hat{C}_{\vartheta\vartheta}^{1/2} W_{\vartheta \boldsymbol{y}} \hat{C}_{\boldsymbol{yy}}^{1/2}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}(C_{\boldsymbol{yy}}-\hat{C}_{\boldsymbol{yy}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
& +\left\|(\hat{C}_{\vartheta \boldsymbol{y}}-C_{\vartheta \boldsymbol{y}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
\leq & \left\|\hat{C}_{\vartheta\vartheta}^{1/2}\right\| \varepsilon_m^{-1/2}\left\|(C_{\boldsymbol{yy}}-\hat{C}_{\boldsymbol{yy}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
& +\left\|(\hat{C}_{\vartheta \boldsymbol{y}}-C_{\vartheta \boldsymbol{y}})(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \\
= & O_p\left(\varepsilon_m^{-3/2} m^{-1/2}+\sqrt{N(\varepsilon_m)} \varepsilon_m^{-1} m^{-1/2}\right) \quad (m \to \infty,\ \varepsilon_m \to 0),
\end{aligned}$$

where the second inequality follows from $\|W_{\vartheta y}\| \leq 1$ and $\|\hat{C}_{\boldsymbol{yy}}^{1/2}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\| \leq \varepsilon_m^{-1/2}$, and the last line from Fukumizu (2015, Lemma 1.5.1); the quantity $N(\varepsilon)$ for any $\varepsilon>0$ is defined by $N(\varepsilon):=\mathrm{Tr}[C_{\boldsymbol{yy}}(C_{\boldsymbol{yy}}+\varepsilon I)^{-1}]$, where $\mathrm{Tr}(A)$ denotes the trace of an operator A. Under our assumption on the eigenvalue decay rate of $C_{\boldsymbol{yy}}$, we have $N(\varepsilon) \leq \frac{\beta b}{b-1}\varepsilon^{-1/b}$ (Caponnetto and Vito, 2007, Proposition 3), which implies that the above rate becomes

$$O_p\left(\varepsilon_m^{-3/2} m^{-1/2}+\varepsilon_m^{-1-1/2b} m^{-1/2}\right) \quad (m \to \infty,\ \varepsilon_m \to 0).$$

From $m\varepsilon_m \to \infty$ and $\varepsilon_m \to 0$ (as we determine the schedule of ε_m below), it is easy to show that the second term is slower and thus dominates the above rate. This concludes that the rate of the first term (17) is

$$\left\|\hat{C}_{\vartheta \boldsymbol{y}}(\hat{C}_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}-C_{\vartheta \boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}\right\| \left\|k_{\mathbb{R}^n}(\cdot, Y^n)\right\|_{\mathcal{H}_\Theta}=O_p\left(\varepsilon_m^{-1-1/2b} m^{-1/2}\right) \quad (m \to \infty,\ \varepsilon_m \to 0).$$

The second term (18)**.** Let $(\vartheta', \boldsymbol{y}')$ be an independent copy of the random variables $(\vartheta, \boldsymbol{y})$. Note that for any $\psi \in \mathcal{H}_{\mathbb{R}^n}$, we have

$$\begin{aligned}
\langle C_{\vartheta \boldsymbol{y}}\psi, C_{\vartheta \boldsymbol{y}}\psi\rangle_{\mathcal{H}_\Theta} &= \mathbb{E}\left[k_\Theta(\vartheta, \vartheta')\psi(\boldsymbol{y})\psi(\boldsymbol{y}')\right] \\
&= \mathbb{E}\left[\mathbb{E}[k_\Theta(\vartheta, \vartheta')|\boldsymbol{y}, \boldsymbol{y}']\psi(\boldsymbol{y})\psi(\boldsymbol{y}')\right] \\
&= \mathbb{E}\left[G(\boldsymbol{y}, \boldsymbol{y}')\psi(\boldsymbol{y})\psi(\boldsymbol{y}')\right] \\
&= \langle (C_{\boldsymbol{yy}} \otimes C_{\boldsymbol{yy}})G, \psi \otimes \psi\rangle_{\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}}.
\end{aligned}$$

Similarly, for any $\psi \in \mathcal{H}_{\mathbb{R}^n}$ and $\tilde{Y}^n \in \mathrm{supp}(P_{\mathbb{R}^n})$, we have

$$\begin{aligned}
\left\langle C_{\vartheta \boldsymbol{y}}\psi, \mathbb{E}[k_\Theta(\cdot,\vartheta)|\boldsymbol{y}=\tilde{Y}^n]\right\rangle_{\mathcal{H}_\Theta} &= \mathbb{E}\left[\psi(\boldsymbol{y}')\mathbb{E}[k_\Theta(\vartheta',\vartheta)|\boldsymbol{y}=\tilde{Y}^n]\right] \\
&= \mathbb{E}\left[\psi(\boldsymbol{y}')\mathbb{E}[k_\Theta(\vartheta',\vartheta)|\boldsymbol{y}=\tilde{Y}^n,\boldsymbol{y}']\right] \\
&= \mathbb{E}\left[\psi(\boldsymbol{y}')G(\tilde{Y}^n,\boldsymbol{y}')\right] \\
&= \left\langle (I\otimes C_{\boldsymbol{yy}})G, k_{\mathbb{R}^n}(\cdot,\tilde{Y}^n)\otimes\psi\right\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}},
\end{aligned}$$

where $I : \mathcal{H}_{\mathbb{R}^n} \to \mathcal{H}_{\mathbb{R}^n}$ is the identity operator and

$$\left((I\otimes C_{\boldsymbol{yy}})G\right)(\cdot,*) := \mathbb{E}[G(\cdot,\boldsymbol{y}')k_{\mathbb{R}^n}(\boldsymbol{y}',*)].$$

Now let $\psi := (C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}k_{\mathbb{R}^n}(\cdot,r^*)$. Recall $\mu_{\Theta|r^*} = \mathbb{E}[k_\Theta(\cdot,\vartheta)|\boldsymbol{y}=r^*]$, which gives $\|\mu_{\Theta|r^*}\|^2_{\mathcal{H}_\Theta} = G(r^*,r^*)$. Then the square of (18) can be written as

$$\begin{aligned}
&\left\|C_{\vartheta\boldsymbol{y}}(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}k_{\mathbb{R}^n}(\cdot,r^*) - \mu_{\Theta|r^*}\right\|^2_{\mathcal{H}_\Theta} \\
=\ & \|C_{\vartheta\boldsymbol{y}}\psi\|^2_{\mathcal{H}_\Theta} - 2\left\langle C_{\vartheta\boldsymbol{y}}\psi, \mu_{\Theta|r^*}\right\rangle_{\mathcal{H}_\Theta} + \|\mu_{\Theta|r^*}\|^2_{\mathcal{H}_\Theta} \\
=\ & \left\langle (C_{\boldsymbol{yy}}\otimes C_{\boldsymbol{yy}})G,\ \ (C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}k_{\mathbb{R}^n}(\cdot,r^*)\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}k_{\mathbb{R}^n}(\cdot,r^*)\right\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
& -2\left\langle (I\otimes C_{\boldsymbol{yy}})G,\ \ k_{\mathbb{R}^n}(\cdot,r^*)\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}k_{\mathbb{R}^n}(\cdot,r^*)\right\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} + G(r^*,r^*) \\
=\ & \left\langle ((C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}})G,\ \ k_{\mathbb{R}^n}(\cdot,r^*)\otimes k_{\mathbb{R}^n}(\cdot,r^*)\right\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
& -2\left\langle (I\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}})G,\ \ k_{\mathbb{R}^n}(\cdot,r^*)\otimes k_{\mathbb{R}^n}(\cdot,r^*)\right\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} + G(r^*,r^*) \\
=\ & \Big\langle \Big\{(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}} - I\otimes(C_{\boldsymbol{yy}}+\varepsilon_m)^{-1}C_{\boldsymbol{yy}} \\
& -(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes I + I\otimes I\Big\}G,\quad k_{\mathbb{R}^n}(\cdot,r^*)\otimes k_{\mathbb{R}^n}(\cdot,r^*)\Big\rangle_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
\le\ & \Big\|\Big\{(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}} - I\otimes(C_{\boldsymbol{yy}}+\varepsilon_m)^{-1}C_{\boldsymbol{yy}} \\
& -(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes I + I\otimes I\Big\}G\Big\|_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \ \Big\|k_{\mathbb{R}^n}(\cdot,r^*)\otimes k_{\mathbb{R}^n}(\cdot,r^*)\Big\|_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}}.
\end{aligned}$$

Let $(\phi_i)_{i=1}^\infty \subset \mathcal{H}_{\mathbb{R}^n}$ be the eigenfunctions of $C_{\boldsymbol{yy}}$ and $(\lambda_i)_{i=1}^\infty$ be the associated eigenvalues such that $\lambda_1 \ge \lambda_2 \ge \cdots \ge 0$. Then the eigenfunctions and eigenvalues of the operator $C_{\boldsymbol{yy}}\otimes C_{\boldsymbol{yy}}$ are given as $(\phi_i\otimes\phi_j)_{i,j=1}^\infty$ and $(\lambda_i\lambda_i)_{i,j=1}^\infty$, respectively. Note that $(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}^2\phi_i = (\frac{\lambda_i^2}{1+\varepsilon_m})\phi_i$. Note also that our assumption $G \in \mathrm{Range}(C_{\boldsymbol{yy}}\otimes C_{\boldsymbol{yy}})$ implies that there exists some $\xi \in \mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}$ such that $G = (C_{\boldsymbol{yy}}\otimes C_{\boldsymbol{yy}})\xi$. Using these identities and Parseval's identity, we have

$$\begin{aligned}
&\Big\|\Big\{(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}} - I\otimes(C_{\boldsymbol{yy}}+\varepsilon_m)^{-1}C_{\boldsymbol{yy}} \\
& -(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}+I\otimes I\Big\}G\Big\|^2_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
=\ & \Big\|\Big\{(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}\otimes(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}} - I\otimes(C_{\boldsymbol{yy}}+\varepsilon_m)^{-1}C_{\boldsymbol{yy}} \\
& -(C_{\boldsymbol{yy}}+\varepsilon_m I)^{-1}C_{\boldsymbol{yy}}+I\otimes I\Big\}(C_{\boldsymbol{yy}}\otimes C_{\boldsymbol{yy}})\xi\Big\|^2_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
=\ & \sum_{i,j}\left\{\frac{\lambda_i^2}{\lambda_i+\varepsilon_m}\frac{\lambda_j^2}{\lambda_j+\varepsilon_m} - \frac{\lambda_i\lambda_j^2}{\lambda_j+\varepsilon_m} - \frac{\lambda_i^2\lambda_j}{\lambda_i+\varepsilon_m} + \lambda_i\lambda_j\right\}^2 \langle\phi_i\otimes\phi_j,\xi\rangle^2_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
=\ & \sum_{i,j}\left\{\frac{\varepsilon_m^2\lambda_i\lambda_j}{(\lambda_i+\varepsilon_m)(\lambda_j+\varepsilon_m)}\right\}^2 \langle\phi_i\otimes\phi_j,\xi\rangle^2_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}} \\
\le\ & \varepsilon_m^4\|\xi\|^2_{\mathcal{H}_{\mathbb{R}^n}\otimes\mathcal{H}_{\mathbb{R}^n}}.
\end{aligned}$$

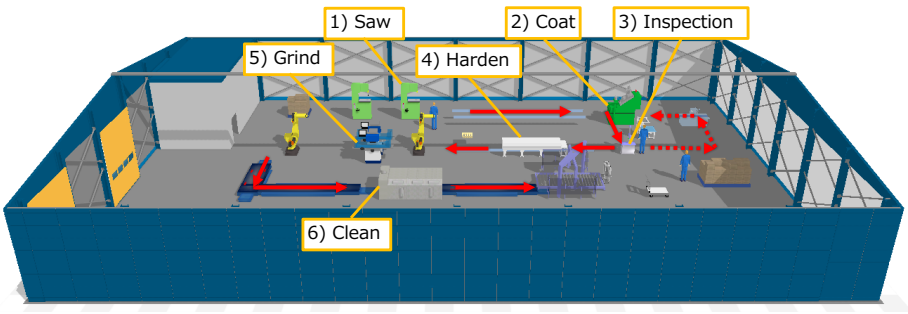

Figure 5: Illustration of the manufacturing process (metal processing factory) for producing valves.

Table 1: Summary of the true and estimated parameters for the experiment on the sophisticated simulation model. T_{BF} represents the mean time between failures, and T_R the mode of repair time for each process. The parameter estimates are the posterior means of the generated parameters, averaged over 10 independent trials, and the corresponding standard deviations are shown in brackets.

Process	Saw		Coat		Inspection		Harden		Grind		Clean	
Parameters	T_{BF} θ_1	T_{R} θ_2	T_{BF} θ_3	T_{R} θ_4	T_{BF} θ_5	T_{R} θ_6	T_{BF} θ_7	T_{R} θ_8	T_{BF} θ_9	T_{R} θ_{10}	T_{BF} θ_{11}	T_{R} θ_{12}
true $\theta^{(0)}$ $(x < 140)$	100	25	200	10	70	20	200	20	75	15	120	20
true $\theta^{(1)}$ $(x > 140)$	100	25	200	10	**50**	20	200	20	75	15	120	20
posterior mean for ordinary reg.	104.6 (4.4)	25.3 (1.2)	181.2 (7.9)	7.1 (0.3)	70.9 (7.6)	18.9 (0.8)	180.1 (8.4)	18.9 (0.3)	72.5 (3.9)	15.2 (0.9)	121.7 (5.1)	20.2 (1.2)
posterior mean for covariate shift	99.4 (6.1)	25.4 (0.9)	181.2 (7.5)	7.9 (0.1)	**54.5** (6.2)	22.1 (2.2)	176.4 (4.4)	17.9 (0.1)	75.6 (3.6)	14.9 (0.5)	120.6 (5.1)	20.4 0.7

From this the second term (18) is upper-bounded as

$$\begin{aligned} & \left\| C_{\vartheta \boldsymbol{y}} (C_{\boldsymbol{yy}} + \varepsilon_m I)^{-1} k_{\mathbb{R}^n}(\cdot, r^*) - \mu_{\Theta|r^*} \right\|_{\mathcal{H}_\Theta} \\ \leq \quad & \varepsilon_m \|\xi\|^{1/2}_{\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}} \left\| k_{\mathbb{R}^n}(\cdot, r^*) \otimes k_{\mathbb{R}^n}(\cdot, r^*) \right\|^{1/2}_{\mathcal{H}_{\mathbb{R}^n} \otimes \mathcal{H}_{\mathbb{R}^n}} = O(\varepsilon_m), \quad (m \to \infty,\ \varepsilon_m \to 0). \end{aligned}$$

The obtained rates for the two terms (17)(18) can be balanced by setting $\varepsilon_m = C m^{-\frac{b}{1+4b}}$ for any fixed constant $C > 0$, and this gives the rate in the assertion. □

B Experiments on Sophisticated Production Simulator

We performed experiments on a sophisticated but more complicated simulator for industrial manufacturing processes than the one in Sec. 5.2. We used a simulation model constructed with the software package WITNESS (`https://www.lanner.com/en-us/`) described in Fig. 5. It models a metal processing factory for producing valves (products) from metal pipes, with six primary processes of 1) "saw", 2) "coat", 3) "inspection", 4) "harden", 5) "grind", and 6) "clean." Each process consists of complicated procedures such preparation, waiting, and machine repair in case of a trouble.

B.1 Setting

As in Sec. 5.2, the input space is $\mathcal{X} = (0, \infty)$ and each input x represents the number of products required to make, and the resulting output $y(x) = R(x) + e(x)$ is the length of time needed to produce that number of products.

The mapping $x \to r(x, \theta)$ consists of the above six processes, and each of them contains two parameters for machine downtime due to failures: the mean time between failures (T_{BF}), and the mode of repair time (T_{R}).

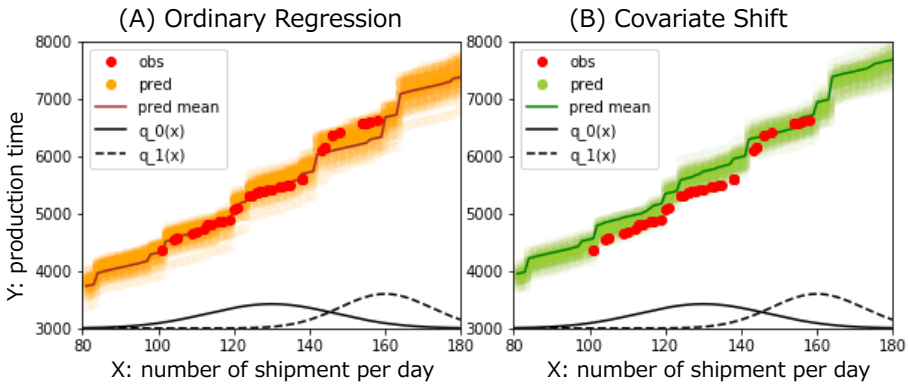

Figure 6: Results of ordinary regression and covariate shift adaptation, for the experiment on the sophisticated model. (A) Results of our method *without* covariate shift adaptation: training data (red points), generated predictive outputs (orange) and their means (brown curve). (B) Results of our method *with* covariate shift adaptation: training data (red points), generated predictive outputs (light green) and their means (green curve). $q_0(x)$ and $q_1(x)$ are input densities for training and prediction, respectively.

Thus, in total, there are 12 parameters, i.e., $\theta = (\theta_1, ..., \theta_{12})^\top \in \Theta \subset \mathbb{R}^{12}$, where $\theta_{2j} = T_{\mathrm{BF}}^{(j)}$ and $\theta_{2j+1} = T_{\mathrm{R}}^{(j)}$ for the j $(= 1, \ldots, 6)$-th process (see Table 1). In each process (say the j-th process), the time between two failures follows the negative exponential distribution with the mean time $\theta_{2j} = T_{\mathrm{BF}}^{(j)}$, and the time required for repair follows the Erlang distribution with the mode of repair time $\theta_{2j+1} = T_{\mathrm{R}}^{(j)}$ and the shape parameter 3. We set the prior distribution $\pi(\theta)$ by defining the uniform distribution over [0, 300] for θ_{2j} and that over [0,30] for θ_{2j+1}, and taking the product of the uniform distributions for all the parameters $(j = 1, \ldots, 6)$.

In a similar manner to the experiment in Section 5.2, we defined the regression function $R(x)$ of the data generating process as $R(x) = r(x, \theta^{(0)})$ for $x < 140$ and $R(x) = r(x, \theta^{(1)})$ for $x \geq 140$, where $\theta^{(0)}$ and $\theta^{(1)}$ are the "true" parameters for training and prediction, and defined in Table 1. We set the input densities $q_0(x)$ and $q_1(x)$ for training and prediction as $N(130, 15)$ and $N(160, 12)$, respectively. The size of training data is $n = 50$, and the number of simulations is $m = 400$. We set the noise process of the data generating process to be independent Gaussian, $e(x) = \epsilon \sim N(0, 300)$. We set the constants $\sigma^2, \sigma_\Theta^2 > 0$ in the kernels $k_{\mathbb{R}^n}$ and k_Θ by the median heuristic using the simulated pairs $(\bar{\theta}_j, \bar{Y}_j^n)_{j=1}^m$, and the regularization constant to be $\varepsilon = 0.1$.

B.1.1 Details of the Simulation Model

We explain below qualitative details of the six processes in the simulation model constructed with the WITNESS software package.

Cutting process: The manufacturing process begins with the arrival of pipes, all of which have the same diameter and length of 30 cm. These pipes arrive at a fixed time interval, depending on the vendor's supply schedule. Subsequently, each pipe is cut into 10-cm sections along the length, resulting in three pieces. A worker is assigned for this process to perform changeover, repair, and disconnection operations. This worker takes a break once every eight hours. Then the small pieces obtained are transferred to the coating process by a conveyor belt.

Coating process: The small pieces are coated for protection by a coating machine. The machine processes six pieces in a batch manner at once. A coating material must have been prepared in the coating machine, before those pieces have arrived; otherwise, the quality of those pieces will be degraded by heat. When the pieces ride on the belt conveyor, a sensor detects them and the coating material is prepared.

Inspection process: After the coating process, each piece is placed in an inspection waiting buffer. An inspector picks up those pieces one by one from the waiting buffer, and inspects the coating quality. If a piece fails the quality inspection, the inspector places that piece in the recoating waiting buffer. The coating machine must process the pieces of the recoating buffer preferentially. When pieces pass the quality inspection, the inspector sends those pieces to the curing step.

Harden process: In the harden (quenching) process, up to 10 pieces are processed simultaneously in a first-come first-out basis, and each piece is quenched for at least one hour.

Grind process: The quenched pieces are polished to satisfy a customer's specifications. Two polishing machines with the same priority are available. Each machine uses special jigs to process four pieces simultaneously, and produces two different types of valves. Further, 10 jigs exist in the system, and when not in use, they are placed in a jig storage buffer.

A loader fixes four pieces with a jig and sends it to the polishing machine. The polishing machine sends the jig and the four pieces to an unloader, once polishing is done. The unloader sends the finished pieces to a valve storage area and the jig to a jig return area. The two types of valves are separated, and placed in a dedicated valve storage buffer. When a jig is required to be used again, it is returned by a jig return conveyor to the jig storage buffer.

Cleaning process: Valves issued from a valve storage area are cleaned before shipment. In the washing machine, five stations are available where valves can be placed one at a time, and the valves are cleaned in these stations. Up to 10 valves of each type can be washed simultaneously. When the valve type is changed, the cleaning head must be replaced.

B.2 Results

The true 12 parameters are estimated as the posterior means of generated parameters, and their averages and standard deviations over 10 independent trials are shown in the bottom rows in Table 1. Most of the true parameters are estimated for both of the ordinary regression and covariate shift settings.

Fig. 6 (A) and (B) describe predictive outputs and their means given by the proposed method, which fit well for both the ordinary and covariate shift settings. The RMSE for predictive outputs by the proposed method with covariate shift adaptation, calculated for test data generated from $q_1(x)$, is 1.48×10^2. On the other hand, the RMSE on the same test data for the proposed method *without* covariate shift adaptation (i.e., setting $\beta(X_i) = 1, i = 1, \ldots, n$ in the importance-weighted kernel) is 1.64×10^3. This confirms that the use of the importance-weighted kernel indeed works for covariate shift adaptation.

In this experiment, approximately 3 [s] was required for one evaluation of the simulation model $r(x, \theta)$ with the authors' computational environment. Thus, the dominant factor in the computational cost was that of simulations.