Machine Learning (2022) 111:1671–1694
https://doi.org/10.1007/s10994-021-06101-8

Nested aggregation of experts using inducing points for approximated Gaussian process regression

Ayano Nakai-Kasai[1] · Toshiyuki Tanaka[1]

Received: 2 May 2021 / Revised: 7 October 2021 / Accepted: 8 October 2021 /
Published online: 9 November 2021

Abstract

Gaussian process regression is a flexible regression scheme but suffers from its high computational complexity regarding the inversion of a matrix with the same size as the training dataset. Aggregation method is one of the approximation techniques for reducing the complexity. In this paper, we propose a novel aggregation method, Nested Aggregation of Experts using Inducing Points (NAE-IP), which is an extension of a conventional method and enables dimensionality reduction by making use of the idea of linear sketching. There are some options for selecting inducing points in the proposed method. The options can introduce test points of interest as inducing points, albeit at the cost of slightly higher computational complexity. The other options exploiting less informative inducing points can yield a significant reduction of the computational complexity. The proposed NAE-IP is theoretically guaranteed to have consistency under certain conditions. Results of our computational experiments using synthetic and real data show that the proposed method achieves lower prediction error and even lower computing time than conventional methods.

Keywords Gaussian process regression · Aggregation methods · Linear sketching · Big data · Scalability

1 Introduction

Gaussian process regression (GPR or full GPR) (Rasmussen and Williams 2006) is a nonparametric regression model that assumes a Gaussian process prior on regression functions. Its application includes geostatistics (Cressie 1993; Stein 1999), data visualization (Lawrence 2005), reinforcement learning (Deisenroth et al. 2015), multi-task learning (Ashton and Sollich 2012), distributed learning (Tavassolipour et al. 2020), to mention a few. Despite its advantage of allowing nonlinear regression, its computational complexity

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Ayano Nakai-Kasai
nakai.ayano@sys.i.kyoto-u.ac.jp

[1] Graduate School of Informatics, Kyoto University, 36-1 Yoshida Honmachi, Sakyo-ku, Kyoto, Kyoto 606-8501, Japan

and required memory can be a serious problem in the cases where the number N of training data is large. Full GPR (which we mean the GPR with no approximation) includes inversion of an $N \times N$ matrix, so that it takes $\mathcal{O}(N^3)$ time complexity[1] (via conventional methods like Gauss-Jordan elimination and LU decomposition) for training. This would restrict the applicability of full GPR to problems with $N \lesssim 10^4$.

In order to circumvent the limitation, various approximation methods have been proposed. These can be divided in two main categories, global and local approximations (Liu et al. 2020). The global approximations replace the global representation of the $N \times N$ matrix with small-sized matrices, typically by using some training points or virtual points, called inducing points or pseudo datapoints (Snelson and Ghahramani 2005; Quiñonero-Candela and Rasmussen 2005; Wilson and Nickisch 2015; Bauer et al. 2016). The idea is categorically termed sparse GP approximation. Sparse GP using m inducing points can reduce time complexity to $\mathcal{O}(Nm^2)$. Locations of the inducing points can furthermore be optimized via stochastic variational inference (Hensman et al. 2013). However, the sparse GP methods are not suitable when the underlying function has quick-varying features because in such cases they require a large number of inducing points to achieve good performance, yielding high complexity (Bui and Turner 2014). On the other hand, the local approximations split training data into a number of sub-datasets, assign an "expert" to each of them, and summarize local predictions made by these experts to arrive at the final prediction. The procedure enables us to capture such quick-varying features. One of the state-of-the-art local approximations is the aggregation method, which includes product-of-experts (PoE) (Hinton 2002), generalized PoE (GPoE) (Cao and Fleet 2014), Bayesian committee machine (BCM) (Tresp 2000), robust BCM (RBCM) (Deisenroth and Ng 2015), generalized RBCM (GRBCM) (Liu et al. 2018), query-aware BCM (QBCM) (He et al. 2019), and nested pointwise aggregation of experts (NPAE) (Rullière et al. 2018). Different aggregation methods summarize the local predictions of the experts by using different schemes. The time complexity of the aggregation methods except for NPAE with sub-dataset size n_0 is reduced to $\mathcal{O}(Nn_0^2) + \mathcal{O}(CNn_0)$, where C is independent of N and varies depending on the method.

An important theoretical property for the aggregation methods is *consistency*, which means that the aggregated prediction converges to the value of the true underlying function when N approaches infinity. The aggregation methods without consistency do not necessarily yield good predictions even in large-sample situations. NPAE and GRBCM are proven to have consistency under appropriate conditions (Bachoc et al. 2017, 2021; Liu et al. 2018). Furthermore, NPAE usually achieves better predictive performance than other methods by using richer information but at the same time requires higher computational complexity.

In this paper, we propose a novel aggregation method inspired by NPAE. We first generalize the prediction of NPAE by using the idea of low-dimensional projection known as sketching (Liberty 2013; Woodruff 2014) of the training samples, and then extend it to more informative versions by introducing inducing points. With its higher flexibility this method is expected to achieve a better trade-off between predictive performance and computational complexity. We name the proposed method Nested Aggregation of Experts using Inducing Points (NAE-IP). The Gaussian process approximation via sketching has also been considered by Calandriello et al. (2019) but their construction is based on matrix sketching and falls

[1] In what follows we consider time complexity under the assumption that arithmetic with matrix elements has complexity $\mathcal{O}(1)$.

within the category of sparse GP methods. On the other hand, the dimensionality reduction in NAE-IP is different from that of sparse GP in that NAE-IP exploits linear sketching of signals, extending NPAE. Furthermore, NAE-IP, similarly to NPAE, allows parallelization of some part of processing by employing a block-diagonal sketching matrix. NAE-IP is expected to be advantageous in two alternative fronts: one is that it prioritizes predictive performance at the cost of an increase of computational complexity, and another is that it uses a less informative set of inducing points while allowing reduction of the computational complexity. Furthermore, we prove that NAE-IP has consistency under certain conditions. Simulation results show that the proposed method achieves lower prediction errors than conventional aggregation methods, while keeping less computing time than the original NPAE.

In the rest of the paper, we use the following notations. Boldface indicates vector or matrix. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-\mathrm{T}}$ denote the transpose and the inverse of the transpose, respectively. $\mathbf{0}$, $\boldsymbol{O}$, and $\boldsymbol{I}$ stand for the zero vector, zero matrix, and identity matrix, respectively. $\|\cdot\|$ represents ℓ_2-norm. $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ is Gaussian distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$. $\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}]$ means the covariance matrix of random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. $\ker \boldsymbol{A} := \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \mathbf{0}\}$ is the kernel (null space) of the matrix $\boldsymbol{A}$. $\det[\cdot]$ and $\mathrm{Tr}[\cdot]$ stand for the determinant and trace operator, respectively. $[\cdot]_{ab}$, $[\cdot]_a$, and $[\cdot]_{[a][b]}$ denote the (a, b)th element of the matrix, the ath row of the matrix or ath element of the vector, and the (a, b)th block of the block matrix, respectively. $\mathrm{diag}[\cdot]$ is the diagonal matrix composed of the elements in the square brackets.

2 Gaussian process regression and aggregation methods

2.1 Full GPR

Consider the full GPR on a region $\mathcal{Q} \subset \mathbb{R}^D$. Given a training dataset with N samples, $\mathcal{D} = \{(\boldsymbol{x}_n, z_n) \in \mathcal{Q} \times \mathbb{R}\}_{n=1,\ldots,N}$, the regression model is

$$z_n = f(\boldsymbol{x}_n) + \epsilon_n, \tag{1}$$

where the regression function f is assumed to follow a Gaussian process (GP), and where the residual error ϵ_n is assumed to be a white Gaussian noise with mean 0 and variance σ^2, that is, one has $[\epsilon_1, \ldots, \epsilon_N]^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$. The mean function of the GP can be assumed to be 0 without loss of generality. The covariance function $k(\cdot,\cdot)$ of the GP represents properties of the regression function. Commonly used covariance functions are the squared exponential (SE) function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{r^2}{2}\right), \tag{2}$$

and the Matérn-$(\nu + 1/2)$ function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\sqrt{2\nu + 1}r\right) \frac{\nu!}{(2\nu)!} \sum_{\nu'=0}^{\nu} \frac{(\nu + \nu')!}{\nu'!(\nu - \nu')!} \left(2\sqrt{2\nu + 1}r\right)^{\nu - \nu'}, \tag{3}$$

where $r = \sqrt{(\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}}\boldsymbol{L}^{-1}(\boldsymbol{x} - \boldsymbol{x}')}$ is the Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ and covariance matrix $\boldsymbol{L} = \mathrm{diag}[\ell_1, \ldots, \ell_D]$, where $\nu \in \mathbb{N}^+$ is a model parameter for the Matérn function, and where $\sigma_f^2 > 0$ and $\ell_d > 0$ $(d = 1, \ldots, D)$ are hyperparameters. The hyperparameters of these models are thus $\boldsymbol{\Theta} = \{\sigma_f^2, \{\ell_d\}_{d=1,\ldots,D}, \sigma^2\}$, the values of which may be determined via maximizing the log-marginal likelihood

$$\log p(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{\Theta}) = -\frac{1}{2}\Big(\boldsymbol{z}^{\mathrm{T}}(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{z} + \log\det\big[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I}\big]\Big), \tag{4}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N\times D}$, $\boldsymbol{z} = [z_1, \cdots, z_N]^{\mathrm{T}} \in \mathbb{R}^N$, and where the covariance matrix $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}')$ is such that $[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}')]_{nn'} = k(([\boldsymbol{X}]_n)^{\mathrm{T}}, ([\boldsymbol{X}']_{n'})^{\mathrm{T}})$.

Assume that we wish to estimate the values of f at N_T test points $\{\boldsymbol{x}_t^*\}_{t=1,\dots,N_T}$. All the test points and the corresponding outputs are summarized as $\boldsymbol{X}^* = [\boldsymbol{x}_1^*, \cdots, \boldsymbol{x}_{N_T}^*]^{\mathrm{T}} \in \mathbb{R}^{N_T\times D}$ and $\boldsymbol{z}^* = [z_1^*, \dots, z_{N_T}^*]^{\mathrm{T}} \in \mathbb{R}^{N_T}$, respectively. The values of the regression function corresponding to $\boldsymbol{X}$ and $\boldsymbol{X}^*$ are summarized as $\boldsymbol{f} = [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_N)]^{\mathrm{T}} \in \mathbb{R}^N$ and $\boldsymbol{f}^* = [f(\boldsymbol{x}_1^*), \cdots, f(\boldsymbol{x}_{N_T}^*)]^{\mathrm{T}} \in \mathbb{R}^{N_T}$, respectively. On the assumption that the prior of f is GP, the joint distribution of $\boldsymbol{z}$ and $\boldsymbol{f}^*$ is given by

$$p\left(\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{f}^*\end{bmatrix}\right) = \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^*)\\ \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X}^*)\end{bmatrix}\right). \tag{5}$$

The predictive distribution of $\boldsymbol{f}^*$ given $\mathcal{D}$ is obtained as $p(\boldsymbol{f}^*|\boldsymbol{X}^*,\mathcal{D}) = \mathcal{N}\big(\boldsymbol{\mu}_{\text{full}}(\boldsymbol{X}^*), \boldsymbol{\Sigma}_{\text{full}}(\boldsymbol{X}^*)\big)$, where

$$\boldsymbol{\mu}_{\text{full}}(\boldsymbol{X}^*) = \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{z}, \tag{6}$$

$$\boldsymbol{\Sigma}_{\text{full}}(\boldsymbol{X}^*) = \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X}^*) - \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^*). \tag{7}$$

The prediction of $\boldsymbol{z}^*$ is similarly obtained as $p(\boldsymbol{z}^*|\boldsymbol{X}^*,\mathcal{D}) = \mathcal{N}\big(\boldsymbol{\mu}_{\text{full}}(\boldsymbol{X}^*), \boldsymbol{\Sigma}_{\text{full}}(\boldsymbol{X}^*) + \sigma^2\boldsymbol{I}\big)$. The matrix inversion in Eqs. (6) and (7) has $\mathcal{O}(N^3)$ time complexity and $\mathcal{O}(N^2)$ memory consumption, so that various approximations have been proposed to circumvent the complexity.

2.2 Aggregation methods

2.2.1 Problem settings and training

In this subsection, we introduce the common settings among the aggregation methods. The whole training dataset is first divided into p subsets, $\mathcal{D}_i = (\boldsymbol{X}_i, \boldsymbol{z}_i)$ $(i = 1, \dots, p)$, where each subset has $n^{(i)}$ data points, namely, $\boldsymbol{X}_i \in \mathbb{R}^{n^{(i)}\times D}$ and $\boldsymbol{z}_i \in \mathbb{R}^{n^{(i)}}$. Sub-models that make predictions using the sub-datasets are referred to as "experts". Each expert $\mathcal{M}_i$ makes own predictions by using its own sub-dataset $\mathcal{D}_i$. The local prediction $p_i(z^*|\boldsymbol{x}^*,\mathcal{D}_i) = \mathcal{N}\big(\mu_i(\boldsymbol{x}^*), \sigma_i^2(\boldsymbol{x}^*)\big)$ at a test point $\boldsymbol{x}_t^* := \boldsymbol{x}^*$ $(t = 1, \dots, N_T)$ is obtained by applying full GPR to the sub-dataset $\mathcal{D}_i$, as

$$\mu_i(\boldsymbol{x}^*) = \boldsymbol{K}_{*i}\big(\boldsymbol{K}_{ii} + \sigma^2\boldsymbol{I}\big)^{-1}\boldsymbol{z}_i, \tag{8}$$

$$\sigma_i^2(\boldsymbol{x}^*) = k(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{K}_{*i}\big(\boldsymbol{K}_{ii} + \sigma^2\boldsymbol{I}\big)^{-1}\boldsymbol{K}_{i*} + \sigma^2, \tag{9}$$

respectively, where $\boldsymbol{K}_{*i} = \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{X}_i)$, $\boldsymbol{K}_{i*} = \boldsymbol{K}_{*i}^{\mathrm{T}}$, and $\boldsymbol{K}_{ij} = \boldsymbol{K}(\boldsymbol{X}_i,\boldsymbol{X}_j)$ for $i,j = 1, \dots, p$. Aggregation methods described in the subsequent sections integrate the experts' predictions and yield the final prediction in different manners.

For learning hyperparameters $\boldsymbol{\Theta}$, it is reasonable under these settings to introduce a factorized training process (Deisenroth and Ng 2015). In the process, the exact marginal

Table 1 Recommended weight choice for aggregation methods

Method	β_{i1}	β_{i2}	Constraint
PoE	1	$1/p$	
GPoE	$1/p\dagger$	$1/p$	$\sum_{i=1}^{p} \beta_{i1} = 1$
BCM	1	β_{i1}	
RBCM	$\frac{\log \sigma_{**}^2 - \log \sigma_i^2}{2}\dagger$	β_{i1}	

The choice marked † means that it lacks theoretical justification

likelihood (Eq. (4)) is approximated by assuming independence of the marginal likelihoods of the experts, i.e.,

$$p(z|\boldsymbol{X}, \boldsymbol{\Theta}) \approx \prod_{i=1}^{p} p_i(z_i|\boldsymbol{X}_i, \boldsymbol{\Theta}), \tag{10}$$

where the experts share the same hyperparameters $\boldsymbol{\Theta}$. The computational complexity for the training process can be reduced compared with full GPR thanks to the independence assumption.

2.2.2 Predictions that ignore some covariance of experts

PoE (Hinton 2002), GPoE (Cao and Fleet 2014), BCM (Tresp 2000), and RBCM (Deisenroth and Ng 2015) are aggregation methods that ignore covariance between experts. The original PoE and GPoE assume independence of experts $\{\mathcal{M}_i\}_{i=1,\dots,p}$. BCM and RBCM assume conditional independence of the experts given the value $f(\boldsymbol{x}^*)$ of the regression function at a test point $\boldsymbol{x}^*$. The aggregated prediction $p\big(z^*|\boldsymbol{x}^*, \{\mu_i(\boldsymbol{x}^*), \sigma_i^2(\boldsymbol{x}^*)\}_{i=1,\dots,p}\big)$ with mean $\mu_{\text{poe/bcm}}(\boldsymbol{x}^*)$ and variance $\sigma^2_{\text{poe/bcm}}(\boldsymbol{x}^*)$ can be collectively formulated as

$$\mu_{\text{poe/bcm}}(\boldsymbol{x}^*) = \sigma^2_{\text{poe/bcm}}(\boldsymbol{x}^*) \sum_{i=1}^{p} \beta_{i1} \sigma_i^{-2}(\boldsymbol{x}^*) \mu_i(\boldsymbol{x}^*), \tag{11}$$

$$\sigma^{-2}_{\text{poe/bcm}}(\boldsymbol{x}^*) = \sum_{i=1}^{p} \beta_{i1} \sigma_i^{-2}(\boldsymbol{x}^*) + \left(1 - \sum_{i=1}^{p} \beta_{i2}\right) \sigma_{**}^{-2}, \tag{12}$$

where $\sigma_{**}^2 = k(\boldsymbol{x}^*, \boldsymbol{x}^*) + \sigma^2$, and where β_{i1} and β_{i2} are the weights assigned to expert $\mathcal{M}_i$. The choices of the weights recommended in the respective papers, as well as constraints, of those aggregation methods are summarized in Table 1.

Two extensions of RBCM, called GRBCM (Liu et al. 2018) and QBCM (He et al. 2019), are recently proposed. These methods assume existence of an informative "global expert" $\mathcal{M}_g := \mathcal{M}_1$, and that every expert can access, in addition to the sub-dataset assigned to it, the sub-dataset $\mathcal{D}_g = \mathcal{D}_1$ assigned to the global expert. Therefore, these methods take account of covariances between the global expert and other experts, but ignore covariance between non-global experts, and assume conditional independence $\mathcal{D}_i \perp \mathcal{D}_j \mid z^*, \mathcal{D}_g$ for $i, j = 2, \dots, p$ and $i \neq j$. Each expert $\mathcal{M}_i$ $(i = 2, \dots, p)$ possesses sub-dataset $\mathcal{D}_{+i} = \mathcal{D}_g \cup \mathcal{D}_i$ and makes own predictions with mean $\mu_{+i}(\boldsymbol{x}^*)$ and variance $\sigma^2_{+i}(\boldsymbol{x}^*)$. The global expert also makes prediction with mean $\mu_g(\boldsymbol{x}^*)$ and variance $\sigma_g^2(\boldsymbol{x}^*)$ by using only the global sub-dataset $\mathcal{D}_g$. The aggregated predictions are given by the following mean $\mu_{\text{grbcm/qbcm}}(\boldsymbol{x}^*)$ and variance $\sigma^2_{\text{grbcm/qbcm}}(\boldsymbol{x}^*)$,

$$\mu_{\text{grbcm/qbcm}}(\boldsymbol{x}^*) = \sigma^2_{\text{grbcm/qbcm}}(\boldsymbol{x}^*) \cdot \left[\sum_{i=2}^{p} \beta_i \sigma_{+i}^{-2}(\boldsymbol{x}^*) \mu_{+i}(\boldsymbol{x}^*) + \left(1 - \sum_{i=2}^{p} \beta_i\right) \sigma_g^{-2}(\boldsymbol{x}^*) \mu_g(\boldsymbol{x}^*) \right], \tag{13}$$

$$\sigma^{-2}_{\text{grbcm/qbcm}}(\boldsymbol{x}^*) = \sum_{i=2}^{p} \beta_i \sigma_{+i}^{-2}(\boldsymbol{x}^*) + \left(1 - \sum_{i=2}^{p} \beta_i\right) \sigma_g^{-2}(\boldsymbol{x}^*), \tag{14}$$

where the experts' weights $\{\beta_i\}_{i=2,\dots,p}$ are chosen in the same manner as RBCM. For GRBCM, the global sub-dataset $\mathcal{D}_g$ is randomly selected from the entire training samples, and for QBCM, $\mathcal{D}_g$ is selected as the sub-dataset with its centroid closest to the test point.

2.2.3 NPAE: prediction that uses covariance between all experts

NPAE (Rullière et al. 2018) for GPR is also one of the aggregation methods but it yields "consistent" prediction by taking account of covariance between experts, at the cost of computational complexity. The consistency is discussed in the next subsection. In NPAE, the aggregated prediction is obtained as follows:

$$\mu_{\text{npae}}(\boldsymbol{x}^*) = \boldsymbol{k}_{\mathcal{A}*}^{\mathrm{T}} \boldsymbol{K}_{\mathcal{A}*}^{-1} \boldsymbol{\mu}_*, \tag{15}$$

$$\sigma^2_{\text{npae}}(\boldsymbol{x}^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}_{\mathcal{A}*}^{\mathrm{T}} \boldsymbol{K}_{\mathcal{A}*}^{-1} \boldsymbol{k}_{\mathcal{A}*} + \sigma^2, \tag{16}$$

where $\boldsymbol{\mu}_* = [\mu_1(\boldsymbol{x}^*), \dots, \mu_p(\boldsymbol{x}^*)]^{\mathrm{T}} \in \mathbb{R}^p$, $\boldsymbol{k}_{\mathcal{A}*} = \mathrm{Cov}[\boldsymbol{\mu}_*, z^*] \in \mathbb{R}^p$, and $\boldsymbol{K}_{\mathcal{A}*} = \mathrm{Cov}[\boldsymbol{\mu}_*, \boldsymbol{\mu}_*] \in \mathbb{R}^{p \times p}$. This formulation means that NPAE uses the covariance between all experts, that is, uses richer information than those aggregation methods described in Sect. 2.2.2.

Note that the original NPAE is restricted to test-point-wise processing and requires $p \times p$ matrix construction and its inversion $\boldsymbol{K}_{\mathcal{A}*}^{-1}$ for each test point, so that its computational complexity is higher than other aggregation methods. Rullière et al. (2018) have also proposed additional complexity reduction of NPAE by considering hierarchical organization of the experts, in which case the subsequent prediction becomes different from Eqs. (15) and (16).

2.2.4 Consistency

Consistency is one of the important properties for the aggregation methods, which means that the aggregated prediction converges to the value of the true underlying function when the number N of training points approaches infinity. It should be noted that the definition of consistency in this paper is such that an aggregation method for a finite number of test points is said to be consistent if the aggregated predictions provided by the method converges to the values of the true underlying function at those test points in probability, as $N \to \infty$. In particular, the definition is different from, and much weaker than, the consistency in functional spaces (van der Vaart and van Zanten 2011): Consistency of a method in the above definition does not necessarily imply that the posterior on the functional space provided by the method converges to the Dirac measure at the true underlying function in the limit $N \to \infty$.

NPAE in Sect. 2.2.3 is proven to be consistent in the noiseless case ($\sigma^2 = 0$) (Bachoc et al. 2017) and the noisy case ($\sigma^2 \neq 0$) (Bachoc et al. 2021). For the latter case, NPAE is consistent when the placement of all input points is not too irregular on $\mathcal{Q}$ or when the training data is divided by typical clustering algorithms, e.g., k-means. Consistency including noisy observations is also discussed in Liu et al. (2018), where they have concluded that GRBCM is consistent as long as the input points in the global sub-dataset are randomly selected on $\mathcal{Q}$. Bachoc et al. (2017, 2021) have also proven that, under some assumptions on the kernel[2], there are cases where consistency of PoE, GPoE, BCM, and RBCM does not hold depending on the distribution of the input points.

3 NAE using inducing points

3.1 Reformulation of NPAE via sketching

In this subsection, we represent the predictions by NPAE (Eqs. (15) and (16)) in an alternative formulation, with the aim of extending it to a generalized method. As mentioned in Sect. 1, the high computational complexity of full GPR arises primarily from the necessity of inverting the Gram matrix $(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I})$ with size equal to the number N of training samples. Consequently, all the existing approximation schemes include some ideas of reducing the size of the matrix to be inverted, and accordingly, when evaluating the conditional mean in these schemes one projects $\boldsymbol{z}$ to a low-dimensional subspace determined by the matrix of reduced size. In this paper, rather than considering a reduced-size matrix to be inverted, we focus on the latter projection procedure. More specifically, we consider a *linear sketch* $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{z} \in \mathbb{R}^{N_u}$ of $\boldsymbol{z}$, where N_u is the dimension of the linear sketch $\boldsymbol{u}$, and where $\boldsymbol{A} \in \mathbb{R}^{N_u \times N}$ is a sketching matrix, and study the problem of estimating the function values at test points not on the basis of $\boldsymbol{z}$ but on the basis of its sketch $\boldsymbol{u}$. As detailed in the following, this approach has advantages in that it provides a novel interpretation of NPAE as well as its extensions, and that it allows us to provide a full characterization of the optimal sketching matrix.

In what follows we assume, without loss of generality, that the rows of the sketching matrix $\boldsymbol{A}$ are linearly independent, as adding linearly dependent rows does not add any useful information of $\boldsymbol{z}$ to its linear sketch $\boldsymbol{u}$. The joint probability of $\{\boldsymbol{u}, \boldsymbol{z}^*\}$ is

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{z}^* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{A}\left(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I}\right)\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{A}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^*) \\ \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X})\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{K}(\boldsymbol{X}^*,\boldsymbol{X}^*) + \sigma^2\boldsymbol{I} \end{bmatrix} \right). \tag{17}$$

The conditional distribution of $\boldsymbol{z}^*$ given $\boldsymbol{u}$ is calculated as

$$\boldsymbol{z}^* | \boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathcal{A}}(\boldsymbol{X}^*), \boldsymbol{\Sigma}_{\mathcal{A}}(\boldsymbol{X}^*) \right), \tag{18}$$

where

[2] Many stationary kernels including the Matérn kernel satisfy the assumption, but the SE kernel does not.

$$\boldsymbol{\mu}_{\mathcal{A}}(\boldsymbol{X}^*) = \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{u}, \tag{19}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathcal{A}}(\boldsymbol{X}^*) = {} & \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) + \sigma^2\boldsymbol{I} \\ & - \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*).\end{aligned} \tag{20}$$

The matrix to be inverted in the above formulae is of size $N_u \times N_u$, implying that the time complexity can be significantly reduced by taking $N_u \ll N$. It should be noted that this reduction is different from that of sparse GP methods and that in Calandriello et al. (2019), where the matrix to be inverted is $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_u)\boldsymbol{K}(\boldsymbol{X}_u, \boldsymbol{X}_u)^{-1}\boldsymbol{K}(\boldsymbol{X}_u, \boldsymbol{X})$ ($\boldsymbol{X}_u \in \mathbb{R}^{N_u \times D}$ is the set of inducing points) with size $N \times N$ and the reduction is granted via Woodbury matrix identity.

For sketching matrix $\boldsymbol{A}$ with a general structure, the following proposition holds.

Proposition 1 *Assume row independence of the sketching matrix* $\boldsymbol{A}$. *The conditional distribution of* $\boldsymbol{z}^*$ *given the linear sketch* $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{z}$ *depends on* $\boldsymbol{A}$ *only through its kernel* $\ker \boldsymbol{A}$.

Proof Under the row independence, the size of the sketching matrix $\boldsymbol{A}$ is $N_u \times N$ with $N_u = N - \dim\ker\boldsymbol{A}$. For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{N_u \times N}$, $\ker\boldsymbol{A} = \ker\boldsymbol{B}$ holds if and only if $\boldsymbol{A}$ and $\boldsymbol{B}$ are row equivalent, that is, there exists an invertible matrix $\boldsymbol{T} \in \mathbb{R}^{N_u \times N_u}$ satisfying $\boldsymbol{B} = \boldsymbol{T}\boldsymbol{A}$. The conditional distribution of $\boldsymbol{z}^*$ given a linear sketch $\boldsymbol{u}' = \boldsymbol{B}\boldsymbol{z}$ with $\boldsymbol{B} = \boldsymbol{T}\boldsymbol{A}$ is the same as that given the linear sketch $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{z}$, as can be confirmed by the fact that replacing $\boldsymbol{A}$ with $\boldsymbol{B} = \boldsymbol{T}\boldsymbol{A}$ in Eqs. (19) and (20) with $\boldsymbol{z}$ fixed keeps $\boldsymbol{\mu}_{\mathcal{A}}(\boldsymbol{X}^*)$ and $\boldsymbol{\Sigma}_{\mathcal{A}}(\boldsymbol{X}^*)$ invariant. □

One expects that the conditional mean with sketching given in Eq. (19) would give a good approximation of the conditional mean in the full GPR. Goodness of this approximation may be measured via the mean squared error $\mathcal{E} = \mathrm{E}[\|\boldsymbol{\mu}_{\mathcal{A}}(\boldsymbol{X}^*) - \boldsymbol{\mu}_{\mathrm{full}}(\boldsymbol{X}^*)\|^2)$ between the conditional means with and without sketching. It is evaluated as

$$\begin{aligned}\mathcal{E} = {} & \mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\right] \\ & - \mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\right].\end{aligned} \tag{21}$$

The next proposition provides a full characterization of the optimal sketching matrix in the sense of minimizing $\mathcal{E}$.

Proposition 2 *For a given dimension* N_u *of the linear sketching* $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{z} \in \mathbb{R}^{N_u}$ *of* $\boldsymbol{z}$, *the optimal sketching matrix* $\boldsymbol{A} \in \mathbb{R}^{N_u \times N}$ *in the sense of minimizing the mean squared error* $\mathcal{E}$ *is such that the* N_u *row vectors of* $\boldsymbol{A}$ *span the subspace spanned by the eigenvectors of* $(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})$ *corresponding to its* N_u *largest eigenvalues.*

Proof Since the first term on the right-hand side of Eq. (21) is independent of $\boldsymbol{A}$, the optimal sketching matrix $\boldsymbol{A}$ minimizing the mean squared error $\mathcal{E}$ is the matrix that maximizes

$$J(\boldsymbol{A}) = \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\right].$$

The matrix $\boldsymbol{C} = \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}$ is symmetric and positive definite, so that it is diagonalized by an orthogonal matrix $\boldsymbol{V}$ as $\boldsymbol{C} = \boldsymbol{V}^{\mathrm{T}} \boldsymbol{\Lambda} \boldsymbol{V}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with the diagonal elements consisting of the eigenvalues of $\boldsymbol{C}$. Letting $\boldsymbol{C}^{1/2} = \boldsymbol{V}^{\mathrm{T}} \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}$ and $\boldsymbol{A}' = \boldsymbol{T}\boldsymbol{A}\boldsymbol{C}^{1/2}$, where $\boldsymbol{T}$ is an invertible matrix corresponding to the Gram-Schmidt orthogonalization applied to the row vectors of $\boldsymbol{A}\boldsymbol{C}^{1/2}$ such that $\boldsymbol{A}'(\boldsymbol{A}')^{\mathrm{T}} = \boldsymbol{I}$ holds, one has

$$\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{C}^{1/2}\left(\boldsymbol{A}\boldsymbol{C}^{1/2}\right)^{\mathrm{T}} = \boldsymbol{T}^{-1}\boldsymbol{A}'(\boldsymbol{A}')^{\mathrm{T}}\boldsymbol{T}^{-\mathrm{T}} = \boldsymbol{T}^{-1}\boldsymbol{T}^{-\mathrm{T}}.$$

The cost function $J(\boldsymbol{A})$ can then be written as

$$\begin{aligned} J(\boldsymbol{A}) &= \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{C}^{-1/2}(\boldsymbol{A}')^{\mathrm{T}}\boldsymbol{A}'\boldsymbol{C}^{-1/2}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\right] \\ &= \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{A}'\boldsymbol{C}^{-1/2}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{C}^{-1/2}(\boldsymbol{A}')^{\mathrm{T}}\right]. \end{aligned}$$

Therefore, the optimal sketching matrix is such that the N_u row vectors of $\boldsymbol{A}' = \boldsymbol{T}\boldsymbol{A}\boldsymbol{C}^{1/2}$ span the subspace spanned by the eigenvectors of $\boldsymbol{C}^{-1/2}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})\boldsymbol{C}^{-1/2}$ corresponding to its N_u largest eigenvalues. This coincides with the statement of the proposition. □

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$ be the eigenvalues of $(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X})$. Then, the mean squared error with the optimal sketching matrix is given by

$$\mathcal{E} = \sum_{i=N_u+1}^{N} \lambda_i.$$

Since $\mathrm{rank}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*) = \mathrm{rank}\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}) \leq \min\{N, N_T\}$, one has $\lambda_i = 0$ for $i > \min\{N, N_T\}$. Therefore, in order to make the mean squared error $\mathcal{E}$ smaller, it would make no sense to take $N_u > N_T$ if there is no restriction in the choice of the sketching matrix $\boldsymbol{A}$, because $N_u > N_T$ allows us to make $\mathcal{E} = 0$ with the optimal choice of $\boldsymbol{A}$.

The approach of optimizing the sketching matrix with a general structure, however, would require inversion of $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I}$ and/or solving a (generalized) eigenvalue problem with a large full-rank matrix, so that its computational complexity should be high.

We next consider block-structured sketching, in which one assumes $\boldsymbol{A}$ to have the following block structure:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2 & \cdots & \boldsymbol{O} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{A}_p \end{bmatrix},$$

where $\boldsymbol{A}_i \in \mathbb{R}^{n_u^{(i)} \times n^{(i)}}$ with $\sum_{i=1}^{p} n_u^{(i)} = N_u$ and $\sum_{i=1}^{p} n^{(i)} = N$. This block-structured sketching allows us to perform a certain fraction of the calculations in a distributed manner, with p computing agents (i.e., experts). The prediction in Eq. (18) exactly coincides with that of NPAE when $n_u^{(i)} = 1$ and $\boldsymbol{A}_i$ is chosen as

$$\boldsymbol{A}_i = \boldsymbol{K}_{*i}\left(\boldsymbol{K}_{ii} + \sigma^2\boldsymbol{I}\right)^{-1}, \tag{22}$$

for all experts. In this case, $\boldsymbol{A}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}^*)$ and $\boldsymbol{A}\left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2\boldsymbol{I}\right)\boldsymbol{A}^{\mathrm{T}}$ in Eq. (17) are replaced as $\boldsymbol{k}_{\mathcal{A}*}$ and $\boldsymbol{K}_{\mathcal{A}*}$, respectively. Thanks to this formulation, we can regard the choice of $\boldsymbol{A}_i$ in NPAE as a dimensionality reduction from the size $n^{(i)}$ of the sub-dataset $\mathcal{D}_i$ to $n_u^{(i)} = 1$.

3.2 NAE-IP

The choice of the matrix $\boldsymbol{A}_i$ in Eqs. (19) and (20) is not limited to that of NPAE (Eq. (22)). Furthermore, $\boldsymbol{A}_i$ does not even have to be dependent on $\boldsymbol{X}^*$. We then propose a novel aggregation method on the basis of Eq. (18) and name it *Nested Aggregation of Experts using Inducing Points (NAE-IP)*, which is not limited to be "pointwise," that is, it allows simultaneous prediction on multiple test points. In the proposed method, we select the following choice for $\boldsymbol{A}_i$:

$$\boldsymbol{A}_i = \boldsymbol{K}_{\eta_i i}\left(\boldsymbol{K}_{ii} + \sigma^2 \boldsymbol{I}\right)^{-1}, \tag{23}$$

where $\boldsymbol{K}_{\eta_i i} = \boldsymbol{K}(\bar{\boldsymbol{X}}_i, \boldsymbol{X}_i) \in \mathbb{R}^{n_u^{(i)} \times n^{(i)}}$ and $\boldsymbol{K}_{i\eta_i} = \boldsymbol{K}_{\eta_i i}^{\mathrm{T}}$ for a collection $\bar{\boldsymbol{X}}_i \in \mathbb{R}^{n_u^{(i)} \times D}$ of $n_u^{(i)}$ inducing points. It should be noticed that Eq. (23) is the same as Eq. (22) except that the test points $\boldsymbol{X}^*$ in the latter is replaced by the collection $\bar{\boldsymbol{X}}_i$ of inducing points. In other words, we consider the projection from the size $n^{(i)}$ of the sub-dataset $\mathcal{D}_i$ to the number $n_u^{(i)}$ of inducing points.

We show the prediction scheme using Eq. (23) in a way that follows NPAE. Assume that each expert $\mathcal{M}_i$ has a set of inducing points $\bar{\boldsymbol{X}}_i$ in addition to its own sub-dataset $\mathcal{D}_i$. There is no constraint on the choice of the inducing points but their total number N_u is assumed to be less than N for achieving dimensionality reduction. First, each expert defines an estimator $\bar{\boldsymbol{\mu}}_i$ on the basis of its observation $\boldsymbol{z}_i$ as

$$\bar{\boldsymbol{\mu}}_i = \boldsymbol{K}_{\eta_i i}\left(\boldsymbol{K}_{ii} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{z}_i = \boldsymbol{A}_i \boldsymbol{z}_i \in \mathbb{R}^{n_u^{(i)}}, \quad i = 1, \dots, p. \tag{24}$$

Second, the estimators are concatenated to form a random vector $\bar{\boldsymbol{\mu}} = [\bar{\boldsymbol{\mu}}_1^{\mathrm{T}}, \dots, \bar{\boldsymbol{\mu}}_p^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{N_u}$. The covariances involving $\bar{\boldsymbol{\mu}}$ and $\boldsymbol{z}^*$ are calculated as

$$\bar{\boldsymbol{k}}_{\mathcal{A}} = \mathrm{Cov}[\bar{\boldsymbol{\mu}}, \boldsymbol{z}^*] = \begin{bmatrix} \boldsymbol{A}_1 \boldsymbol{K}_{1*} \\ \vdots \\ \boldsymbol{A}_p \boldsymbol{K}_{p*} \end{bmatrix} \in \mathbb{R}^{N_u \times N_T}, \tag{25}$$

$$\begin{aligned} \bar{\boldsymbol{K}}_{\mathcal{A}} &= \mathrm{Cov}[\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\mu}}] = \left[\left[\bar{\boldsymbol{K}}_{\mathcal{A}}\right]_{[i][j]}\right]_{i,j=1,\dots,p} \in \mathbb{R}^{N_u \times N_u}, \\ \left[\bar{\boldsymbol{K}}_{\mathcal{A}}\right]_{[i][j]} &= \begin{cases} \boldsymbol{K}_{\eta_i i}\left(\boldsymbol{K}_{ii} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{K}_{i\eta_i} & \text{if } i = j, \\ \boldsymbol{A}_i \boldsymbol{K}_{ij} \boldsymbol{A}_j^{\mathrm{T}} & \text{if } i \neq j. \end{cases} \end{aligned} \tag{26}$$

Finally, the predictive mean $\bar{\boldsymbol{\mu}}_{\mathcal{A}}$ and covariance $\bar{\boldsymbol{\Sigma}}_{\mathcal{A}}$ of NAE-IP are derived as

$$\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*) = \bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}} \bar{\boldsymbol{K}}_{\mathcal{A}}^{-1} \bar{\boldsymbol{\mu}}, \tag{27}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathcal{A}}(\boldsymbol{X}^*) = \boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) - \bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}} \bar{\boldsymbol{K}}_{\mathcal{A}}^{-1} \bar{\boldsymbol{k}}_{\mathcal{A}} + \sigma^2 \boldsymbol{I}. \tag{28}$$

These formulae correspond to $\boldsymbol{\mu}_{\mathcal{A}}(\boldsymbol{X}^*)$ and $\boldsymbol{\Sigma}_{\mathcal{A}}(\boldsymbol{X}^*)$ in Eq. (18), respectively, when the choice of Eq. (23) for $\boldsymbol{A}_i$ is employed.

The following proposition holds and is used for proving the consistency of NAE-IP discussed later.

Proposition 3 $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*)$ *in Eq.* (27) *is the best linear unbiased estimator of* $\boldsymbol{f}^*$ *on the basis of* $\bar{\boldsymbol{\mu}}$, *where the coefficient matrix* $\boldsymbol{\phi} = [\boldsymbol{\phi}_1^{\mathrm{T}} \dots \boldsymbol{\phi}_p^{\mathrm{T}}]^{\mathrm{T}}$ *of* $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*) = \boldsymbol{\phi}^{\mathrm{T}}\bar{\boldsymbol{\mu}} = \sum_{i=1}^{p} \boldsymbol{\phi}_i^{\mathrm{T}}\bar{\boldsymbol{\mu}}_i$ *is given by* $\bar{\boldsymbol{K}}_{\mathcal{A}}^{-1}\bar{\boldsymbol{k}}_{\mathcal{A}}$. *The mean squared error* $v(\boldsymbol{X}^*) = \mathrm{E}\left[\|\boldsymbol{f}^* - \bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*)\|^2\right]$ *of the estimator* $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*)$ *of* $\boldsymbol{f}^*$ *is given by* $\mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) - \bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}}^{-1}\bar{\boldsymbol{k}}_{\mathcal{A}}\right]$.

Proof Using Eqs. (27) and (28), the mean squared error of an estimator $\boldsymbol{\phi}^{\mathrm{T}}\bar{\boldsymbol{\mu}}$ of $\boldsymbol{f}^*$, with $\bar{\boldsymbol{\mu}}$ defined above, is written as

$$\mathrm{E}\left[\|\boldsymbol{f}^* - \boldsymbol{\phi}^{\mathrm{T}}\bar{\boldsymbol{\mu}}\|^2\right] = \mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) - 2\boldsymbol{\phi}^{\mathrm{T}}\bar{\boldsymbol{k}}_{\mathcal{A}} + \boldsymbol{\phi}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}}\boldsymbol{\phi}\right].$$

The value of $\hat{\boldsymbol{\phi}}$ minimizing it is found by differentiation: $-2\bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}} + 2\hat{\boldsymbol{\phi}}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}} = \mathbf{O}$, which leads to $\hat{\boldsymbol{\phi}} = \bar{\boldsymbol{K}}_{\mathcal{A}}^{-1}\bar{\boldsymbol{k}}_{\mathcal{A}}$ and $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*) = \hat{\boldsymbol{\phi}}^{\mathrm{T}}\bar{\boldsymbol{\mu}}$. Then, $v(\boldsymbol{X}^*) = \mathrm{Tr}\left[\boldsymbol{K}(\boldsymbol{X}^*, \boldsymbol{X}^*) - 2\hat{\boldsymbol{\phi}}^{\mathrm{T}}\bar{\boldsymbol{k}}_{\mathcal{A}} + \hat{\boldsymbol{\phi}}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}}\hat{\boldsymbol{\phi}}\right]$ and the statement follows. □

One may perform prediction on the N_T test points in a pointwise manner, repeating predition on a single point N_T times, or all at once, predicting for the N_T test points simultaneously. In view of the computational complexity to be discussed later, we consider a more general framework in which the N_T test points are partitioned into S subsets $\boldsymbol{X}_1^*, \dots, \boldsymbol{X}_S^*$ with $\bigcup_{s=1}^{S} \boldsymbol{X}_s^* = \boldsymbol{X}^*$ and $\boldsymbol{X}_s \cap \boldsymbol{X}_{s'} = \emptyset$ for $s \neq s'$, and the prediction is performed on each of these subsets separately. Assume now that the prediction is to be made on the target subset $\boldsymbol{X}_s^*$ of $n_t^{(s)}$ test points. Then, there are 5 possible options of inducing points $\bar{\boldsymbol{X}}_i$ for expert i in NAE-IP:

1. $\bar{\boldsymbol{X}}_i = \boldsymbol{X}_s^*$: Use the test points themselves as the inducing points. In this case $n_u^{(i)} = n_t^{(s)}$. [Blockwise Test points (**BT**)]
2. $\bar{\boldsymbol{X}}_i = \{\boldsymbol{x} \in \boldsymbol{X}_e^* \mid \boldsymbol{X}_e^* \subset \boldsymbol{X}^*, \boldsymbol{X}_e^* \neq \boldsymbol{X}_s^*\}$: Use a part of test points, $\boldsymbol{X}_e^*$, which is not equal to the target subset $\boldsymbol{X}_s^*$. We can set $n_u^{(i)}$ arbitrarily while satisfying $n_u^{(i)} < N_T$. [Blockwise Test points and Other Test points (**BT+OT**), Arbitrary Test points (**AT**)]
3. $\bar{\boldsymbol{X}}_i = \{\boldsymbol{x} \in (\boldsymbol{X}_o \cup \boldsymbol{X}_s^*) \mid \boldsymbol{X}_o \cap \boldsymbol{X}^* = \emptyset, \boldsymbol{X}_o \neq \emptyset\}$: Use both the target subset of test points, $\boldsymbol{X}_s^*$, and non-test points. We can set $n_u^{(i)}$ arbitrarily while satisfying $n_u^{(i)} > n_t^{(s)}$. [Blockwise Test points and Non-Test points (**BT+NT**)]
4. $\bar{\boldsymbol{X}}_i = \{\boldsymbol{x} \in \boldsymbol{X}_o \mid \boldsymbol{X}_o \cap \boldsymbol{X}^* = \emptyset, \boldsymbol{X}_o \neq \emptyset\}$: Use only non-test points as the inducing points. We can set $n_u^{(i)}$ arbitrarily. [Non-Test points (**NT**)]
5. $\bar{\boldsymbol{X}}_i = \{\boldsymbol{x} \in (\boldsymbol{X}_o \cup \boldsymbol{X}_e^*) \mid \boldsymbol{X}_o \cap \boldsymbol{X}^* = \emptyset, \boldsymbol{X}_e^* \subset \boldsymbol{X}^*, \boldsymbol{X}_e^* \neq \boldsymbol{X}_s^*, \boldsymbol{X}_o \neq \emptyset\}$: Use both a part of test points, $\boldsymbol{X}_e^*$, which is not equal to the target subset $\boldsymbol{X}_s^*$, and non-test points. We can set $n_u^{(i)}$ arbitrarily.

Option 1 is an extension of the original NPAE (Rullière et al. 2018) to multiple dimensions. As option 2, we can consider two natural choices, one that completely includes test points themselves (BT+OT) and another that partially or never includes them (AT). BT+OT and BT+NT use higher-dimensional sketching at each expert by incorporating auxiliary points

Table 2 Definitions

$k(\boldsymbol{x}, \boldsymbol{x}')$	Covariance function (e.g., SE function (Eq. (2)) or Matérn-($\nu + 1/2$) function (Eq. (3))
$\boldsymbol{\theta}$	Set of hyperparameters included in covariance function
σ^2	Noise variance
$\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \sigma^2\}$	Set of hyperparameters
$\boldsymbol{K}_{\theta}(\boldsymbol{X}_i, \boldsymbol{X}_j)$	Covariance matrix whose (m, l)th element is $k(([\boldsymbol{X}_i]_m)^{\mathrm{T}}, ([\boldsymbol{X}_j]_l)^{\mathrm{T}})$ with hyperparameters $\boldsymbol{\theta}$
$\{\boldsymbol{X}_i, \boldsymbol{z}_i\}$	Sub-dataset of ith expert
$\bar{\boldsymbol{X}}_i$	Inducing points of ith expert
$n_u^{(i)}$	Number of inducing points of ith expert
p	Number of experts
$\boldsymbol{X}^*$	Test data points
$\boldsymbol{X}_s^*$	sth subset of test data points
$n_t^{(s)}$	Number of test data points in sth subset
S	Number of subsets of test data points

as its inducing points. Extension of sketching dimensions employed in these options is expected to improve prediction accuracy, at the expense of increased computational complexity. The idea of BT, BT+OT, and BT+NT are known as transduction (Quiñonero-Candela and Rasmussen 2005) that uses the test points of interest for prediction. The transduction could be beneficial because the test points should have some information about the corresponding outputs. A drawback with these options is that the covariance matrix $\bar{\boldsymbol{K}}_{\mathcal{A}}$ depends on all or some test points in the target subset $\boldsymbol{X}_s^*$, so that one has to construct it, as well as to perform matrix inversion, for every target subset. On the other hand, AT and NT require the construction of $\bar{\boldsymbol{K}}_{\mathcal{A}}$ only once for all the target subsets of test points, as long as the inducing points are fixed. It brings about a significant reduction of the complexity. Option 5 might not yield a better prediction than BT+OT or BT+NT. Therefore we focus on BT, BT+OT, AT, BT+NT, and NT in the rest of this paper and expect an improvement of the predictive performance by using the extended dimensions $n_u^{(i)} \geq n_t^{(s)}$.

3.3 Summary of proposed algorithm

In this subsection, we summarize the procedure of the proposed NAE-IP. Definitions of symbols used for NAE-IP are summarized in Table 2. We write the covariance matrix as $\boldsymbol{K}_{\theta}(\cdot, \cdot)$ in order to make explicit its dependence on the hyperparameters $\boldsymbol{\theta}$ of the covariance function.

The whole training dataset is divided into p subsets by using some clustering algorithms or at random. Each sub-dataset $(\boldsymbol{X}_i, \boldsymbol{z}_i)$ is assigned to an expert. To learn hyperparameters, we adopt the factorized training process (Deisenroth and Ng 2015). We first specify an option from BT, BT+OT, BT+NT, AT, or NT and construct inducing points $\{\bar{\boldsymbol{X}}_i\}_{i=1}^{p}$ by Algorithm 1. We then perform NAE-IP as shown in Algorithm 2.

Algorithm 1 Construction of inducing points

```
Input: Option (BT, BT+OT, BT+NT, AT, or NT), $\boldsymbol{X}^*$, $\boldsymbol{X}_s^*$, $n_t^{(s)}$, $p$, $n_u^{(i)}(\geq n_t^{(s)}$, $i = 1, \ldots, p)$
Output: $\{\bar{\boldsymbol{X}}_i\}_{i=1}^p$
switch (Option)
    case BT:
        $\bar{\boldsymbol{X}}_i = \boldsymbol{X}_s^* \quad (i = 1, \ldots, p)$
        $n_u^{(i)} = n_t^{(s)} \quad (i = 1, \ldots, p)$
        break
    case BT+OT:
        Choose $n_u^{(i)} - n_t^{(s)}$ test points $\boldsymbol{X}_e^*$ where $\boldsymbol{X}_e^* \subset \boldsymbol{X}^*, \boldsymbol{X}_e^* \cap \boldsymbol{X}_s^* = \emptyset \quad (i = 1, \ldots, p)$
        $\bar{\boldsymbol{X}}_i = (\boldsymbol{X}_e^* \cup \boldsymbol{X}_s^*) \quad (i = 1, \ldots, p)$
        break
    case BT+NT:
        Choose $n_u^{(i)} - n_t^{(s)}$ non-test points $\boldsymbol{X}_o$ where $\boldsymbol{X}_o \cap \boldsymbol{X}^* = \emptyset \quad (i = 1, \ldots, p)$
        $\bar{\boldsymbol{X}}_i = (\boldsymbol{X}_o \cup \boldsymbol{X}_s^*) \quad (i = 1, \ldots, p)$
        break
    case AT:
        Choose $n_u^{(i)}$ test points $\boldsymbol{X}_e^*$ where $\boldsymbol{X}_e^* \subset \boldsymbol{X}^*, \boldsymbol{X}_e^* \neq \boldsymbol{X}_s^* \quad (i = 1, \ldots, p)$
        $\bar{\boldsymbol{X}}_i = \boldsymbol{X}_e^* \quad (i = 1, \ldots, p)$
        break
    case NT:
        Choose $n_u^{(i)}$ non-test points $\boldsymbol{X}_o$ where $\boldsymbol{X}_o \cap \boldsymbol{X}^* = \emptyset \quad (i = 1, \ldots, p)$
        $\bar{\boldsymbol{X}}_i = \boldsymbol{X}_o \quad (i = 1, \ldots, p)$
        break
end switch
```

Algorithm 2 NAE-IP

```
Input: Option, $\boldsymbol{X}^*$, $p$, $S$, $n_t^{(s)}$ $(s = 1, \ldots, S)$, $\{\boldsymbol{X}_i, \boldsymbol{z}_i\}$, $n_u^{(i)}(\geq n_t^{(s)})$ $(i = 1, \ldots, p)$
Output: $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}^*)$, $\bar{\boldsymbol{\Sigma}}_{\mathcal{A}}(\boldsymbol{X}^*)$
Obtain hyperparameters $\boldsymbol{\Theta}$ via factorized training process
    $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \sigma^2\} = \arg\max_{\tilde{\boldsymbol{\theta}}, \tilde{\sigma}^2} \left\{-\sum_{i=1}^{p} \left(\boldsymbol{z}_i^{\mathrm{T}}(\boldsymbol{K}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{X}_i, \boldsymbol{X}_i) + \tilde{\sigma}^2\boldsymbol{I})^{-1}\boldsymbol{z}_i + \log\det\left[\boldsymbol{K}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{X}_i, \boldsymbol{X}_i) + \tilde{\sigma}^2\boldsymbol{I}\right]\right)\right\}$
Partition test points $\boldsymbol{X}^*$ into $S$ subsets $\boldsymbol{X}_s^*$, where the number of points is $n_t^{(s)}$ $(s = 1, \ldots, S)$
Construct inducing points by Algorithm 1
for each partition $\boldsymbol{X}_s^*$ do
    for each expert $i = 1, \ldots, p$ do
        $\boldsymbol{A}_i = \boldsymbol{K}_{\boldsymbol{\theta}}(\bar{\boldsymbol{X}}_i, \boldsymbol{X}_i)\left(\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_i, \boldsymbol{X}_i) + \sigma^2\boldsymbol{I}\right)^{-1}$
        $\bar{\boldsymbol{\mu}}_i = \boldsymbol{A}_i\boldsymbol{z}_i$
    end for
    $\bar{\boldsymbol{\mu}} = [\bar{\boldsymbol{\mu}}_1^{\mathrm{T}}, \ldots, \bar{\boldsymbol{\mu}}_p^{\mathrm{T}}]^{\mathrm{T}}$
    $\bar{\boldsymbol{k}}_{\mathcal{A}} = [\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_s^*, \boldsymbol{X}_1)\boldsymbol{A}_1^{\mathrm{T}}, \ldots, \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_s^*, \boldsymbol{X}_p)\boldsymbol{A}_p^{\mathrm{T}}]^{\mathrm{T}}$
    $\left[\bar{\boldsymbol{K}}_{\mathcal{A}}\right]_{[i][j]} = \begin{cases} \boldsymbol{K}_{\boldsymbol{\theta}}(\bar{\boldsymbol{X}}_i, \boldsymbol{X}_i)\left(\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_i, \boldsymbol{X}_i) + \sigma^2\boldsymbol{I}\right)^{-1} \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_i, \bar{\boldsymbol{X}}_i) & \text{if } i = j \\ \boldsymbol{A}_i\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}_i, \boldsymbol{X}_j)\boldsymbol{A}_j^{\mathrm{T}} & \text{if } i \neq j \end{cases} \quad (i, j = 1, \ldots, p)$
    $\bar{\boldsymbol{\mu}}_{\mathcal{A}}(\boldsymbol{X}_s^*) = \bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}}^{-1}\bar{\boldsymbol{\mu}}$
    $\bar{\boldsymbol{\Sigma}}_{\mathcal{A}}(\boldsymbol{X}_s^*) = \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}^*, \boldsymbol{X}^*) - \bar{\boldsymbol{k}}_{\mathcal{A}}^{\mathrm{T}}\bar{\boldsymbol{K}}_{\mathcal{A}}^{-1}\bar{\boldsymbol{k}}_{\mathcal{A}} + \sigma^2\boldsymbol{I}$
end for
```

3.4 Consistency of NAE-IP

We study consistency of NAE-IP in the noisy case by extending the proof of consistency of NPAE in Bachoc et al. (2021). The following assumption is necessary only for NAE-IP.

Assumption 4 For a test point $\boldsymbol{x}^* \in \mathcal{Q}$, estimation of $f(\boldsymbol{x}^*)$ is done by including the test point $\boldsymbol{x}^*$ as an inducing point of all experts.

For $N \in \mathbb{N}$, let p_N be the number of experts, which may depend on N, and let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_{p_N}$ be the sub-datasets, where $\boldsymbol{X}_i$ $(i = 1, \dots, p_N)$, being a subset of $\boldsymbol{X}$, is the sub-dataset assigned to expert i. We also require the following assumption on the sub-datasets.

Assumption 5 There exists a sequence $\{j_N(\boldsymbol{x}^*)\}_{N \in \mathbb{N}}$ of indices $j_N(\boldsymbol{x}^*) \in \{1, \dots, p_N\}$ depending on a given test point $\boldsymbol{x}^*$ such that, for any $\rho > 0$, the number of the input points in $\boldsymbol{X}_{j_N(\boldsymbol{x}^*)}$ lying within the ρ-ball $B_\rho(\boldsymbol{x}^*) = \{\boldsymbol{x} \in \mathcal{Q} : \|\boldsymbol{x} - \boldsymbol{x}^*\| < \rho\}$ centered at $\boldsymbol{x}^*$ goes to infinity as $N \to \infty$.

Under these assumptions, Proposition 6 below establishes consistency of NAE-IP at a fixed test point $\boldsymbol{x}^* \in \mathcal{Q}$.

Proposition 6 *Let $\mathcal{Q}$ be a compact nonempty subset of $\mathbb{R}^D$. Let f be a Gaussian process on $\mathcal{Q}$ with mean zero and continuous covariance function k. Let $\{\boldsymbol{x}_{Nn}\}_{1 \le n \le N, N \in \mathbb{N}}$ be a triangular array of input points, all of which lie in $\mathcal{Q}$. For $N \in \mathbb{N}$, let $\boldsymbol{X} = [\boldsymbol{x}_{N1}, \dots, \boldsymbol{x}_{NN}]^{\mathrm{T}}$, and let $\bar{\boldsymbol{\mu}}_1, \dots, \bar{\boldsymbol{\mu}}_{p_N}$ be the collection of p_N experts' estimates defined in Eq. (24) on the basis of respective sub-datasets $(\boldsymbol{X}_1, \boldsymbol{z}_1), \dots, (\boldsymbol{X}_{p_N}, \boldsymbol{z}_{p_N})$ of training points. Assume that each row of $\boldsymbol{X}$ is a row of at least one $\boldsymbol{X}_i$. For a test point $\boldsymbol{x}^* \in \mathcal{Q}$, assume further that $\boldsymbol{X}_1, \dots, \boldsymbol{X}_{p_N}$ satisfy Assumption 5. For such a test point $\boldsymbol{x}^*$, under Assumption 4 we have*

$$\lim_{N \to \infty} \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \bar{\mu}_{\mathcal{A}}(\boldsymbol{x}^*)\right)^2\right] = 0. \tag{29}$$

where $\bar{\mu}_{\mathcal{A}}(\boldsymbol{x}^*)$ is as in Eq. (27).

Proof By Assumption 4, expert $j_N(\boldsymbol{x}^*)$ has the test point $\boldsymbol{x}^*$ as its inducing point, that is, $\boldsymbol{x}^*$ is a component of $\bar{\boldsymbol{X}}_{j_N(\boldsymbol{x}^*)}$. Let $a_j(\boldsymbol{x}^*)$ be the index of the test point $\boldsymbol{x}^*$ in $\bar{\boldsymbol{X}}_{j_N(\boldsymbol{x}^*)}$.

With these notations, since $\bar{\mu}_{\mathcal{A}}(\boldsymbol{x}^*)$ is a linear combination of the elements of $\bar{\boldsymbol{\mu}}$ with minimal square prediction errors from Proposition 3, its square prediction error is not larger than that of any single element of $\bar{\boldsymbol{\mu}}$. We hence have

$$\mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \bar{\mu}_{\mathcal{A}}(\boldsymbol{x}^*)\right)^2\right] \le \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - [\bar{\boldsymbol{\mu}}_{j_N(\boldsymbol{x}^*)}(\boldsymbol{x}^*)]_{a_j(\boldsymbol{x}^*)}\right)^2\right] \tag{30}$$

From Assumption 5, for any fixed $\rho > 0$, the number $\imath$ of input points lying within $B_\rho(\boldsymbol{x}^*)$ goes to infinity as $N \to \infty$. Let $\boldsymbol{x}_{j_N^{(1)}(\boldsymbol{x}^*)}, \dots, \boldsymbol{x}_{j_N^{(\imath)}(\boldsymbol{x}^*)}$ and $z_{j_N^{(1)}(\boldsymbol{x}^*)}, \dots, z_{j_N^{(\imath)}(\boldsymbol{x}^*)}$ be such input points and the corresponding observations, respectively. Since $[\bar{\boldsymbol{\mu}}_{j_N(\boldsymbol{x}^*)}(\boldsymbol{x}^*)]_{a_j(\boldsymbol{x}^*)} = \boldsymbol{K}_{*j_N(\boldsymbol{x}^*)}\left(\boldsymbol{K}_{j_N(\boldsymbol{x}^*)j_N(\boldsymbol{x}^*)} + \sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{z}_{j_N(\boldsymbol{x}^*)}$ is also a linear combination of the elements of $\boldsymbol{z}_{j_N(\boldsymbol{x}^*)}$ with minimal square prediction errors, we have, similarly as above,

$$\mathrm{E}\left[\left(f(\boldsymbol{x}^*) - [\bar{\boldsymbol{\mu}}_{j_N(\boldsymbol{x}^*)}(\boldsymbol{x}^*)]_{a_j(\boldsymbol{x}^*)}\right)^2\right] \leq \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \frac{1}{\iota}\sum_{a=1}^{\iota} z_{j_N^{(a)}(\boldsymbol{x}^*)}\right)^2\right]. \tag{31}$$

From the independence of the noise process and Cauchy-Schwarz inequality, the right-hand side can further be bounded as

$$\begin{aligned}\mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \frac{1}{\iota}\sum_{a=1}^{\iota} z_{j_N^{(a)}(\boldsymbol{x}^*)}\right)^2\right] &= \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \frac{1}{\iota}\sum_{a=1}^{\iota}\left(f\left(\boldsymbol{x}_{j_N^{(a)}(\boldsymbol{x}^*)}\right) + \epsilon_{j_N^{(a)}(\boldsymbol{x}^*)}\right)\right)^2\right] \\ &= \mathrm{E}\left[\left(\frac{1}{\iota}\sum_{a=1}^{\iota}\left(f(\boldsymbol{x}^*) - f\left(\boldsymbol{x}_{j_N^{(a)}(\boldsymbol{x}^*)}\right)\right)\right)^2\right] + \mathrm{E}\left[\left(\frac{1}{\iota}\sum_{a=1}^{\iota}\epsilon_{j_N^{(a)}(\boldsymbol{x}^*)}\right)^2\right] \\ &\leq \left(\max_{a=1,\dots,\iota} \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - f\left(\boldsymbol{x}_{j_N^{(a)}(\boldsymbol{x}^*)}\right)\right)^2\right]\right) + \frac{\sigma^2}{\iota}\end{aligned} \tag{32}$$

The second term on the rightmost side of Eq. (32) converges to zero as $\iota \to \infty$ because σ^2 is finite. From the continuity of k as in Bachoc et al. (2021, Appendix E), one has

$$\begin{aligned}\limsup_{N\to\infty} \mathrm{E}\left[\left(f(\boldsymbol{x}^*) - \frac{1}{\iota}\sum_{a=1}^{\iota} z_{j_N^{(a)}(\boldsymbol{x}^*)}\right)^2\right] &\leq \sup_{\xi\in B_\rho(\boldsymbol{x}^*)} \mathrm{E}\left[(f(\boldsymbol{x}^*) - f(\xi))^2\right] \\ &= \sup_{\xi\in B_\rho(\boldsymbol{x}^*)} \left[k(\boldsymbol{x}^*, \boldsymbol{x}^*) + k(\xi,\xi) - 2k(\boldsymbol{x}^*, \xi)\right] \\ &\overset{\rho\to 0}{\to} 0.\end{aligned} \tag{33}$$

The fact that the limit supremum of a nonnegative sequence converges to 0 implies that the limit also converges. We therefore obtain the statement of the proposition. □

Note that Proposition 6 proves convergence of $\bar{\mu}_{\mathcal{A}}(\boldsymbol{x}^*)$ to $f(\boldsymbol{x}^*)$ in the mean square sense, which in turn implies convergence in probability, hence establishing the desired consistency.

Options BT, BT+OT, and BT+NT satisfy Assumption 4 from their definitions. Options AT and NT can also include the test point as an inducing point of all experts under the additional assumptions.

Corollary 7 *Under the conditions of Proposition 6, NAE-IP-BT, BT+OT, and BT+NT have consistency.*

Corollary 8 *Under the conditions of Proposition 6 and the assumption that $n_u^{(i)} \to N_T$ $(i = 1, \dots, p_N)$ as $N \to \infty$, NAE-IP-AT has consistency.*

Corollary 9 *Under the conditions of Proposition 6 and the assumption that $n_u^{(i)} \to \infty$ $(i = 1, \dots, p_N)$ as $N \to \infty$, NAE-IP-NT has consistency.*

Proof In NAE-IP-NT, let $\{\boldsymbol{x}_{\eta_i u^{(i)}}\}_{1\leq u^{(i)}\leq n_u^{(i)}, 1\leq i\leq p}$ be an array of inducing points. For each $\boldsymbol{x}^* \in \mathcal{Q}$ and for $i = 1, \dots, p$, there exists at least one inducing point such that

Table 3 Time complexity of aggregation methods under sufficiently large N, where α, β, and γ are constants larger than 1

Method	Time complexity
(G)PoE/(R)BCM	$\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(\frac{N_T N^2}{p}\right)$
GRBCM	$\mathcal{O}\left(\frac{\alpha N^3}{p^2}\right) + \mathcal{O}\left(\frac{\beta N_T N^2}{p}\right)$
QBCM	$\mathcal{O}\left(\frac{\gamma\alpha N^3}{p^2}\right) + \mathcal{O}\left(\frac{\beta N_T N^2}{p}\right)$
(original) NPAE	$\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(N_T N^2\right)$
NAE-IP-BT	$\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(N_T N^2\right)$
NAE-IP-BT+OT/BT+NT	$\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(\frac{N_T n_u N^2}{n_t}\right)$
NAE-IP-AT/NT	$\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(n_u N^2\right)$

$\lim_{n_u^{(i)} \to \infty} \min_{u^{(i)}} \|\boldsymbol{x}_{\eta_i u^{(i)}} - \boldsymbol{x}^*\| = 0$ and then the estimation of $f(\boldsymbol{x}^*)$ satisfies Assumption 4. □

Assumption 5 holds in typical division of the training data into sub-datasets (Bachoc et al. 2021). When we adopt clustering algorithms such as k-means for the division, the condition holds under the assumption $\min_{i=1,\ldots,p_N} n^{(i)} \to \infty$ as $N \to \infty$. In the case where the input points are distributed with strictly positive density on $\mathcal{Q}$, it also holds under the assumption that the number p_N of experts is $o(N)$ as $N \to \infty$.

3.5 Time complexity

The complexity in time is one of the main interests of approximated Gaussian process regression. We show the complexity of the conventional aggregation methods and our methods proposed in this paper in Table 3 under a sufficiently large N, where, for simplicity, we consider the case of equal dimensions $n_u^{(i)} = n_u$ and equally divided sub-datasets $n^{(i)} = N/p$ among the experts, and of equally divided partitions $n_t^{(s)} = N_T/S = n_t$. GRBCM and QBCM require slightly higher complexity than PoE, GPoE, BCM, and RBCM because each expert uses the modified sub-dataset $\mathcal{D}_{+i}$. As mentioned in Sect. 2.2.3, the complexity of NPAE is higher than these methods but keeps lower than that of full GPR when $N_T < N$. Rullière et al. (2018) reported the complexity as $\mathcal{O}\left(\frac{N^3}{p^2}\right) + \mathcal{O}\left(N_T N^2\right)$ which is the same as NAE-IP-BT, but the frequency for memory access can be reduced by a factor of n_t in the proposed methods. NAE-IP-BT+OT and NAE-IP-BT+NT require higher complexity than the original NPAE by considering n_u dimensions. On the other hand, the complexity of NAE-IP-AT and NAE-IP-NT can be lower than that of not only the original NPAE but also the other methods, depending on the choice of n_u.

In the following, we briefly describe a proof sketch of the complexity of NAE-IP. The main factors that affect the complexity are calculation of the inverse $\left(\boldsymbol{K}_{ii} + \sigma^2 \boldsymbol{I}\right)^{-1}$ in Eq. (24), which is to be performed by every expert, and the construction of $\bar{\boldsymbol{K}}_{\mathcal{A}}$ in Eq. (26). The former takes $\mathcal{O}((n^{(i)})^3)$ at p experts, thus resulting in $\mathcal{O}(p(n^{(i)})^3) = \mathcal{O}(N^3/p^2)$. Next, each block of $\bar{\boldsymbol{K}}_{\mathcal{A}}$ includes the product of $n_u \times n^{(i)}$ matrix and $n^{(i)} \times n^{(i)}$ matrix, and there are p^2 blocks in $\bar{\boldsymbol{K}}_{\mathcal{A}}$, so that these amount to $\mathcal{O}(p^2 n_u (n^{(i)})^2) = \mathcal{O}(n_u N^2)$ computation. The construction of $\bar{\boldsymbol{K}}_{\mathcal{A}}$ is repeated $S = N_T/n_t$ times for BT, BT+OT, and BT+NT, resulting in $\mathcal{O}(N_T N^2)$ for BT and $\mathcal{O}(N_T n_u N^2/n_t)$ for the others, whereas it is performed only once for AT and NT, resulting in $\mathcal{O}(n_u N^2)$.

4 Numerical experiments

4.1 Datasets and settings

We evaluated the predictive performance and the computing time of NAE-IP in comparison with conventional methods. All the results were obtained by using GPML MATLAB Code[3] (Rasmussen and Williams 2006). We measured total CPU time on a linux computer with two CPUs (Intel Xeon Gold 5222, 4 cores, 3.8 GHz base clock) and 768 GB RAM. The datasets used in the numerical experiments are summarized below.

- 1-D *Synthetic data*: Synthetic data generated by $z_n = \mathrm{sinc}(x_n) + \epsilon_n,\ n = 1, \dots, N$, where the training points lie in the interval $[-4, 4]$ uniformly, where N_T test points are uniformly chosen in $[-5, 5]$, and where $[\epsilon_1, \dots, \epsilon_N]^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, 0.04\boldsymbol{I})$.
- 8-D *KIN8NM dataset*[4] (Vanschoren et al. 2013): The data related to the forward dynamics of an 8-link robot arm. There are 8,192 samples in total. We randomly split them into 7,373 samples for training and 819 samples for testing.
- 21-D *SARCOS dataset*[5] (Rasmussen and Williams 2006): The data related to the inverse dynamics problem of robot arms. There are 44,484 training samples and 4,449 test samples.
- 26-D *POL dataset*[6]: Pole telecom dataset. There are 10,000 training samples and 5,000 test samples.

For each experimental condition, we performed 10 trials or more, each consisting of training and prediction procedures. We have used two performance measures, mean squared error (MSE):

$$\mathrm{MSE} = \frac{1}{N_T} \sum_{i=1}^{N_T} (\hat{\mu}(x_{t,i}) - z(x_{t,i}))^2, \tag{34}$$

and mean standardized log loss (MSLL):

$$\mathrm{MSLL} = \frac{1}{N_T} \sum_{i=1}^{N_T} \left(\frac{1}{2} \log (2\pi \hat{\sigma}^2(x_{t,i})) + \frac{(\hat{\mu}(x_{t,i}) - z(x_{t,i}))^2}{2\hat{\sigma}^2(x_{t,i})} \right), \tag{35}$$

where $\hat{\mu}(x_{t,i})$, $\hat{\sigma}^2(x_{t,i})$, $z(x_{t,i})$ are the predictive mean, variance, and true value at the test point $x_{t,i}$, respectively. MSLL is the mean of pointwise negative log losses of the Gaussian models with mean $\hat{\mu}(x_{t,i})$ and variance $\hat{\sigma}^2(x_{t,i})$ given data $\{z(x_{t,i})\}$, and takes into account uncertainty of the predictions via the posterior variances. The lower MSE and MSLL imply the better prediction.

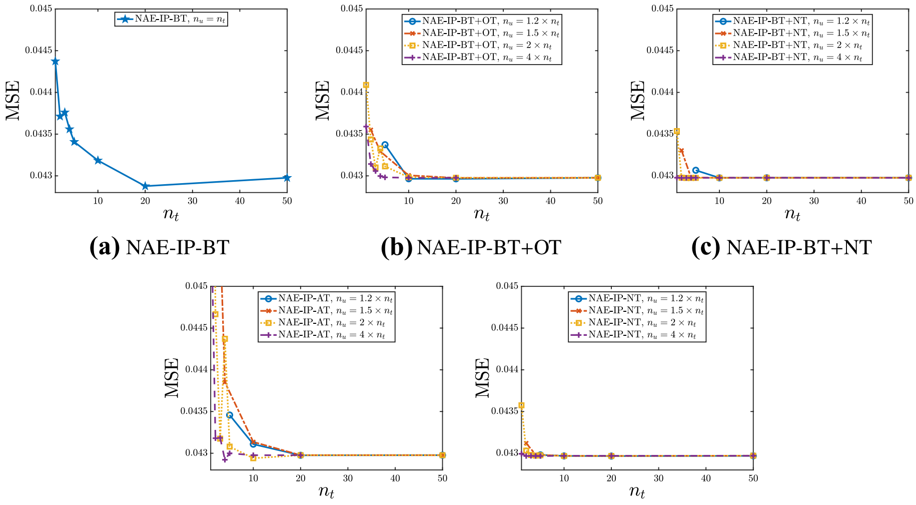
For the proposed methods, we have assumed equal dimensions $n_u^{(i)} = n_u$ among all experts and almost equally divided partitions $n_t^{(s)} = n_t$ of the test points. Other test points for NAE-IP-OT and arbitrary test points for NAE-IP-AT have been chosen randomly from the remaining test points and from the entire test points, respectively.

[3] http://www.gaussianprocess.org/gpml/code/matlab/doc/.

[4] https://www.openml.org/d/189.

[5] http://www.gaussianprocess.org/gpml/data/.

[6] https://cims.nyu.edu/~andrewgw/pattern/.



(a) NAE-IP-BT **(b)** NAE-IP-BT+OT **(c)** NAE-IP-BT+NT

(d) NAE-IP-AT **(e)** NAE-IP-NT

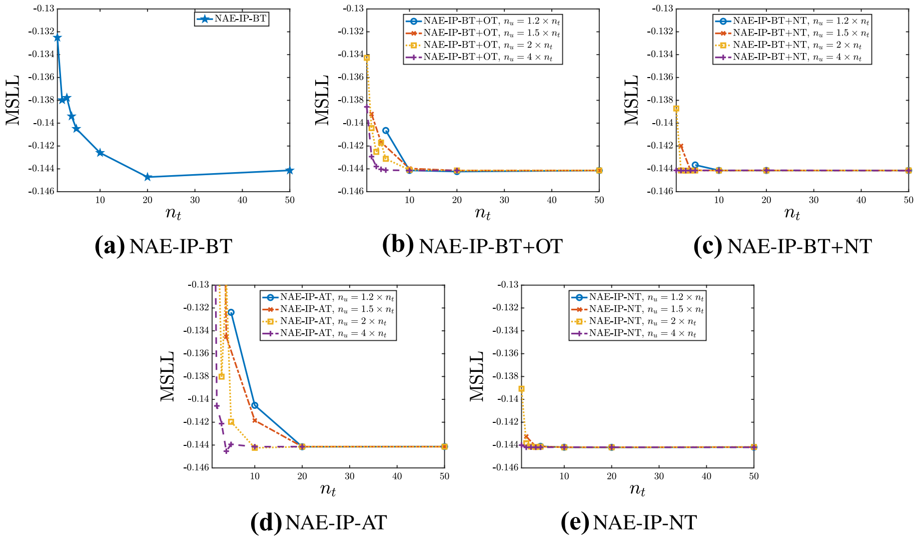
Fig. 1 MSE versus the number n_t of test points

4.2 Synthetic data

For the synthetic data, the training data were divided into sub-datasets by k-means. The SE function (Eq. (2)) was employed as the covariance function of GP. The non-test points of each expert in NAE-IP-BT+NT and NAE-IP-NT were generated from the multivariate Gaussian distribution with the same mean and covariance as those of the sub-dataset assigned to that expert.

First, we investigate the influence on NAE-IP's performance of the dimension n_u and the number of test points n_t processed at once. Figures 1, 2 and 3 show the performance measures and computing time versus n_t when $N = 10^4, N_T = 100$, and $p = 20$. We set the dimension n_u to be $1.2 \times n_t$, $1.5 \times n_t$, $2 \times n_t$, and $4 \times n_t$. When $n_t \leq 20$, the larger n_t and n_u showed the better predictive performance, and the performance became stable in most cases. The higher dimensions required the more computing time, and the computing time of BT, BT+OT, and BT+NT decreased as n_t increased. On the other hand, the computing time of AT and NT was kept small regardless of n_t. Note that, depending on the value of p or N_T, the larger n_t ($\leq n_u$ in this paper) does not always yield the shorter computing time because the complexity required for evaluating the inversion $\boldsymbol{K}_{\mathcal{A}*}^{-1}$ is $\mathcal{O}(n_u^3 p^3)$ and it could be higher than that of other factors.

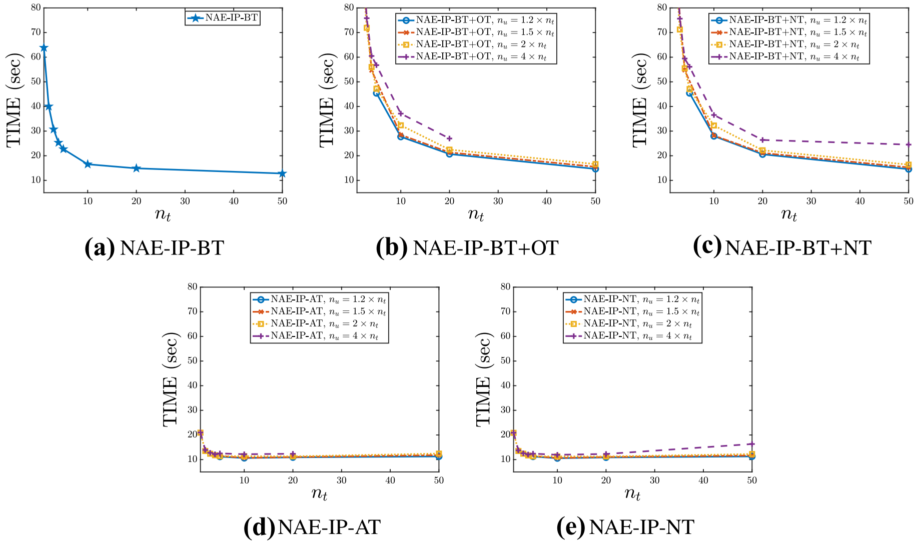
Second, we compare the predictive performance and computing time of NAE-IP with those of conventional aggregation methods, PoE, GPoE, BCM, RBCM, GRBCM, QBCM, and the original NPAE in 30 trials. We evaluated the cases $N = 10^4, 5 \times 10^4$, and 10^5 with $N_T = N \times 10^{-2}$ and $p = N/500$, and chose $(n_t, n_u) = (50, 75)$ for NAE-IP except for NAE-IP-BT. Fig. 4a–c shows the performance measures versus N. Figure 4d summarizes the results of statistical significance testing for a difference between the best-performing method and each of the other 11 methods. We first checked the normality of data via the one-sample two-sided Kolmogorov-Smirnov test ($P < 0.05$), and then employed paired t-test with Bonferroni multiple testing correction ($P < 0.05/11$) for the statistical significance testing. The results for the cases of $N = 5 \times 10^4$ and 10^5 reveal that the MSE of the



(a) NAE-IP-BT **(b)** NAE-IP-BT+OT **(c)** NAE-IP-BT+NT

(d) NAE-IP-AT **(e)** NAE-IP-NT

Fig. 2 MSLL versus the number n_t of test points

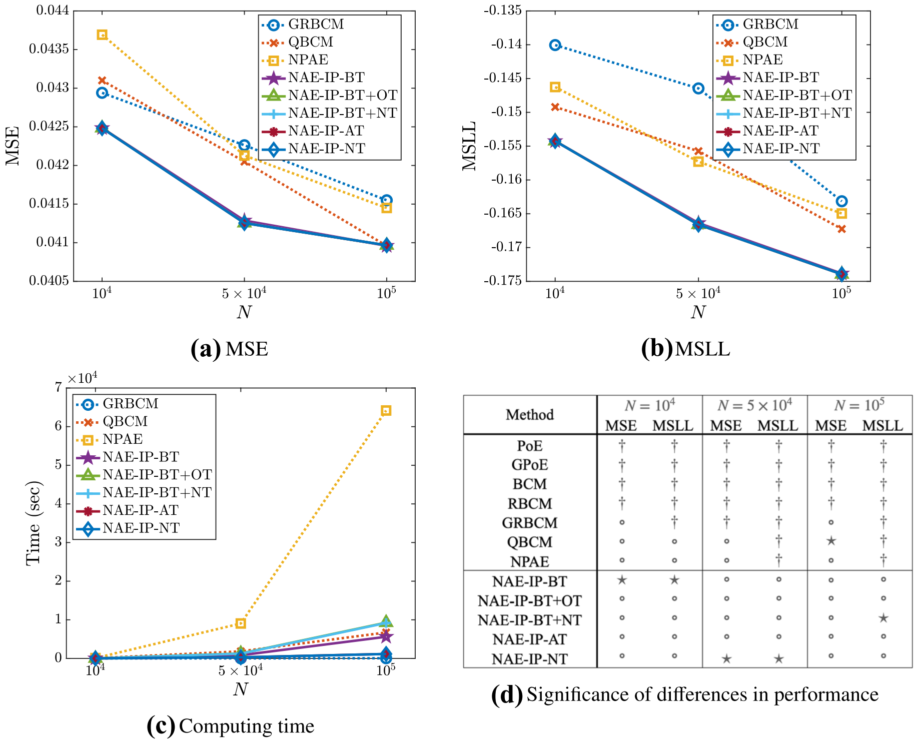


(a) NAE-IP-BT **(b)** NAE-IP-BT+OT **(c)** NAE-IP-BT+NT

(d) NAE-IP-AT **(e)** NAE-IP-NT

Fig. 3 Computing time versus the number n_t of test points

proposed NAE-IP was equal to or better than the other methods and the MSLL was the lowest among the methods. This indicates that NAE-IP can obtain not only better predictive means but also smaller predictive variances. Moreover, NAE-IP took less time than the original NPAE. This might be ascribed to difference in memory access patterns. Especially for NAE-IP-BT, NAE-IP-AT, and NAE-IP-NT, the computing time was shorter than QBCM. Figure 5 shows examples with the predictive means and 95% confidence intervals in the case of $N = 10^4$, $n_t = 20$, and $n_u = 30$ except for NAE-IP-BT. Note that we omitted



(a) MSE

(b) MSLL

(c) Computing time

Method	$N = 10^4$		$N = 5 \times 10^4$		$N = 10^5$	
	MSE	MSLL	MSE	MSLL	MSE	MSLL
PoE	†	†	†	†	†	†
GPoE	†	†	†	†	†	†
BCM	†	†	†	†	†	†
RBCM	†	†	†	†	†	†
GRBCM	∘	†	†	†	∘	†
QBCM	∘	∘	∘	†	⋆	†
NPAE	∘	∘	∘	†	∘	†
NAE-IP-BT	⋆	⋆	∘	∘	∘	∘
NAE-IP-BT+OT	∘	∘	∘	∘	∘	∘
NAE-IP-BT+NT	∘	∘	∘	∘	∘	⋆
NAE-IP-AT	∘	∘	∘	∘	∘	∘
NAE-IP-NT	∘	∘	⋆	⋆	∘	∘

(d) Significance of differences in performance

Fig. 4 **a**–**c** Performance measures versus the number N of training data. **d** Significance of differences in performance. The methods with $\star$ mean that performance of the methods is the best. The results with † mean that differences between each of those and the method with $\star$ are statistically significant (paired t-test with Bonferroni multiple testing correction, $P < 0.05/11$). The results with $\circ$ mean that the null hypothesis is not rejected

the results of PoE, GPoE, BCM, and RBCM from Figs. 4 and 5 because the difference between those and the other methods were significant and those showed the comparable computing time with GRBCM.

4.3 Real data

For SARCOS and POL datasets, the training data were divided respectively by constrained k-means (Bradley et al. 2000), which can avoid generating weak sub-models by setting the minimum size of clusters. We set the minimum size to 300 for SARCOS dataset and 200 for POL dataset. For KIN8NM dataset, the training data were divided by k-means. The Matérn-5/2 function (Eq. (3)) was employed as the covariance function of GP. For NAE-IP-BT+NT and NAE-IP-NT, we employed the optimization of each expert's non-test points by Hensman et al. (2013) under fully independent training conditional assumption

Fig. 5 Examples of predictive distribution with the means and 95% confidence intervals in case of ▶ $N = 10^4, n_t = 20$. The dimension of NAE-IP except for NAE-IP-BT is $n_u = 1.5 \times n_t = 30$. The outside of vertical lines shows extrapolation

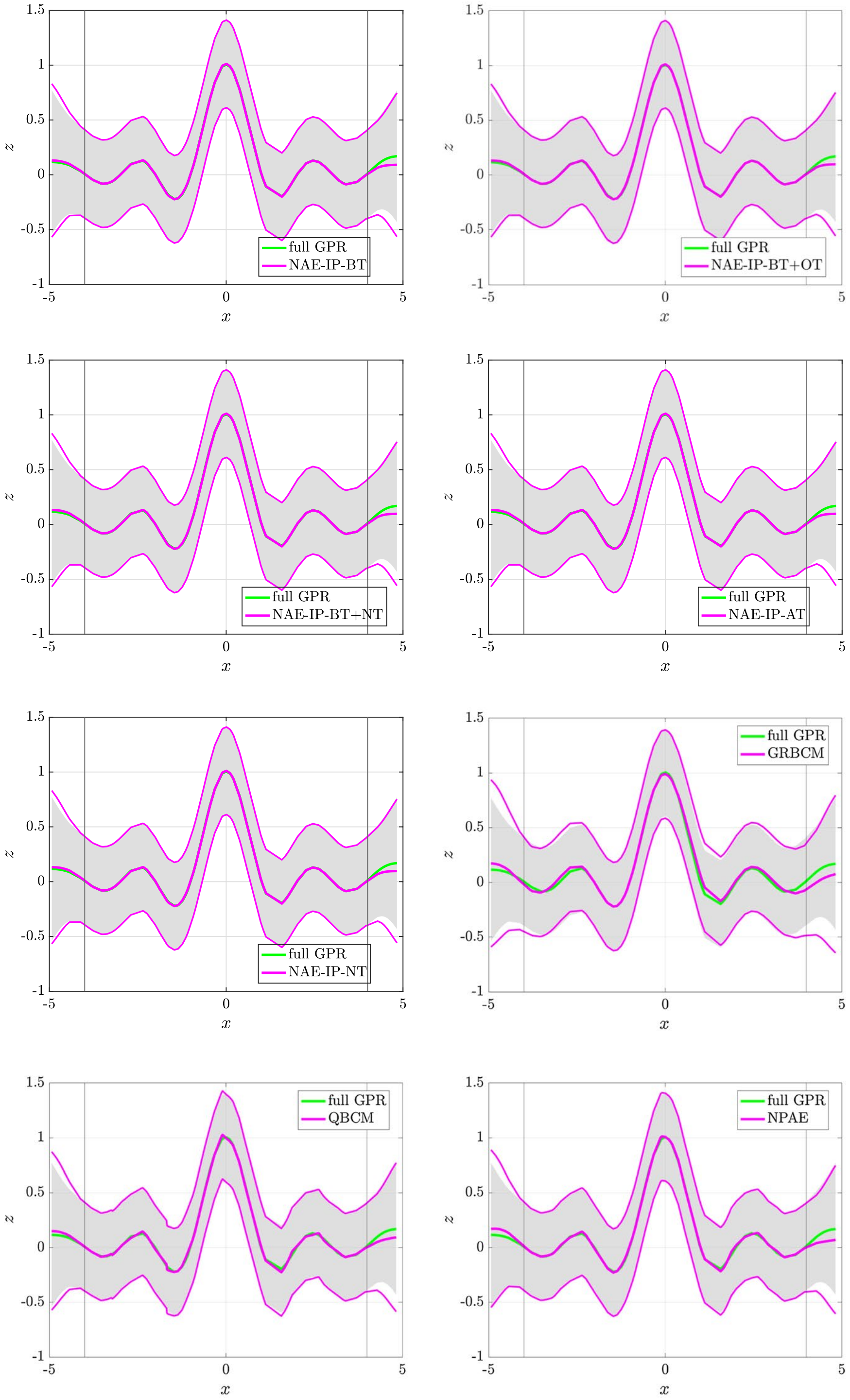

1.5
1
0.5
0
-0.5
-1
-5
0
5
x
z
full GPR
NAE-IP-BT
NAE-IP-BT+OT
NAE-IP-BT+NT
NAE-IP-AT
NAE-IP-NT
GRBCM
QBCM
NPAE

Table 4 Results of the aggregation methods on KIN8NM, SARCOS, and POL datasets

	KIN8NM		SARCOS		POL	
Method	MSE	MSLL	MSE	MSLL	MSE	MSLL
PoE	0.00799	−0.0635	27.1	10.4	116.	11.2
GPoE	0.00799	−0.933	27.1	3.18	116.	3.64
BCM	0.00731	−0.393	3.82	2.73	52.5	3.55
RBCM	0.00625	−0.301	2.24	2.86	22.1	3.19
GRBCM	0.00595	−1.15	1.51	**1.62**⋆	19.5	2.67
QBCM	0.00574	−1.16	1.73	1.73	15.2	2.58
NPAE	0.00540	−1.19	**1.42**⋆	1.68	13.0	2.57
NAE-IP-BT	**0.00532**∘	**−1.20**∘	1.44	1.68	12.3	2.56
NAE-IP-BT+OT	**0.00530**⋆	**−1.20**⋆	1.44	1.68	**12.2**∘	**2.55**∘
NAE-IP-BT+NT	**0.00531**∘	−1.20	1.44	1.68	**12.1**⋆	**2.55**⋆
NAE-IP-AT	0.0178	−0.650	34.4	2.90	54.0	3.27
NAE-IP-NT	0.0141	−0.756	30.2	2.71	26.7	2.85

The methods with ⋆ mean that performance of the methods is the best. The unmarked results mean that differences between each of those and the method with ⋆ are statistically significant (Wilcoxon signed rank test with Bonferroni multiple testing correction, $P < 0.05/11$). The results with ∘ mean that the null hypothesis is not rejected

(Snelson and Ghahramani 2005; Quiñonero-Candela and Rasmussen 2005). We used the assumption only in the optimization of inducing points, and not in the predictions. The mini-batch size and the number of epochs were set to 100 and 10, respectively. It should be noted that the computational complexity of the optimization is $\mathcal{O}((n_u^{(i)})^3)$, so that we can ignore the complexity as long as we set $n_u^{(i)}$ to be smaller than $n^{(i)}$.

We compare the predictive performance of NAE-IP with that of the conventional methods in 10 trials by using the real datasets. We set $(p, n_t) = (8, 20)$ for KIN8NM dataset, (72, 20) for SARCOS dataset, and (25, 50) for POL dataset. The dimension n_u for NAE-IP except for NAE-IP-BT was set to $n_u = 1.5 \times n_t$. Table 4 summarizes the performance measures of the aggregation methods and the results of statistical significance testing for a difference between the best-performing method and each of the other 11 methods (Wilcoxon signed rank test with Bonferroni multiple testing correction, $P < 0.05/11$). For KIN8NM and POL, the extension of sketching dimensions in NAE-IP-BT+NT or NAE-IP-BT+OT improved the performance compared with that of NAE-IP-BT, and those methods achieved better performance than the other methods. On the other hand, for SARCOS, the performance of the conventional methods was the best. The fact that the performance of NAE-IP-AT and NAE-IP-NT was worse might reflect the lack of consistency of these methods under the setting of the dimension n_u.

5 Conclusion

We have introduced the idea of linear sketching into approximate Gaussian process regression and have proposed NAE-IP (Nested Aggregation of Experts using Inducing Points) with five options for the choice of the inducing points. The proposed method inherits consistency under the conditions on the number of inducing points depending on the option. We conducted numerical experiments with synthetic and real datasets. The experimental results show that NAE-IP with the options that include test points as the inducing points achieves a lowest prediction error than the conventional methods. Moreover, the computing time of NAE-IP has been shorter than that of the approximation methods: QBCM and the original NPAE. Future work includes the optimization of a block-structured sketching matrix that projects observations to a low-dimensional subspace.

Author Contributions Conceptualization: Ayano Nakai-Kasai and Toshiyuki Tanaka; Methodology: Ayano Nakai-Kasai and Toshiyuki Tanaka; Formal analysis and investigation: Ayano Nakai-Kasai and Toshiyuki Tanaka; Writing - original draft preparation: Ayano Nakai-Kasai and Toshiyuki Tanaka; Writing - review and editing: Ayano Nakai-Kasai and Toshiyuki Tanaka; Funding acquisition: None; Resources: Ayano Nakai-Kasai and Toshiyuki Tanaka; Supervision: Ayano Nakai-Kasai and Toshiyuki Tanaka.

Funding No funding was received for conducting this study.

Availability of data and material All the real data used in this work are available on the web at https://www.openml.org/d/189, http://www.gaussianprocess.org/gpml/data/, and https://cims.nyu.edu/~andrewgw/pattern/.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code availability The codes for NAE-IP are available at GitHub repository https://github.com/a-nakai-k/NAEIP. Factorized training and predictions of conventional methods are implemented relating to Liu et al. (2018) at https://github.com/LiuHaiTao01/GRBCM.

References

Ashton, S.R.F., & Sollich, P. (2012). Learning curves for multi-task Gaussian process regression. In Advances in Neural Information Processing Systems 25, pp 1393–1428.

Bachoc, F., Durrande, N., Rullière, D., & Chevalier, C. (2017). Some properties of nested Kriging predictors. arXiv preprint arXiv:1707.05708v1.

Bachoc, F., Durrande, N., Rullière, D., & Chevalier, C. (2021). Properties and comparison of some Kriging sub-model aggregation. arXiv preprint arXiv:1707.05708v2.

Bauer, M., van der Wilk, M., & Rasmussen, C.E. (2016). Understanding probabilistic sparse Gaussian process approximations. In Advances in Neural Information Processing Systems 29, pp. 1533–1541.

Bradley, P.S., Bennett, K.P., & Demiriz, A. (2000). Constrained k-means clustering. Tech. rep., MSR-TR-2000-65, Microsoft Research, Redmond, WA.

Bui, T.D., & Turner, R.E. (2014). Tree-structured Gaussian process approximations. In Advances in Neural Information Processing Systems 27, pp. 2213–2221.

Calandriello, D., Carratino, L., Lazaric, A., Valko, M., & Rosasco, L. (2019). Gaussian process optimization with adaptive sketching: Scalable and no regret. *Proceedings of the Thirty-Second Conference on Learning Theory, PMLR, 99*, 533–557.

Cao, Y., & Fleet, D.J. (2014). Generalized product of experts for automatic and principled fusion of Gaussian process predictions. arXiv preprint arXiv:1410.7827.

Cressie, NAC. (1993). Statistics for Spatial Data, Revised Edition. Wiley, New York, NY, https://doi.org/10.1002/9781119115151.

Deisenroth, M.P., & Ng, J.W. (2015) Distributed Gaussian processes. In Proceedings of the 32th International Conference on Machine Learning, PMLR, pp. 1481–1490.

Deisenroth, M. P., Fox, D., & Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(2), 408–423. https://doi.org/10.1109/TPAMI.2013.218

He, J., Qi, J., & Ramamohanarao, K. (2019). Query-aware Bayesian committee machine for scalable Gaussian process regression. In Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 208–216.

Hensman, J., Fusi, N., & Lawrence, N.D. (2013). Gaussian processes for big data. In Proceedings of the 29th Conference on Uncertainly in Artificial Intelligence, pp. 282–290.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*(8), 1771–1800. https://doi.org/10.1162/089976602760128018

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research, 6*, 1783–1816.

Liberty, E. (2013). Simple and deterministic matrix sketching. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 581–588.

Liu, H., Cai, J., Wang, Y., & Ong, Y.S. (2018). Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In Proceedings of the 35th International Conference on Machine Learning, PMLR, pp. 3131–3140.

Liu, H., Ong, Y. S., Shen, X., & Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems, 31*(11), 4405–4423. https://doi.org/10.1109/TNNLS.2019.2957109

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research, 6*, 1939–1959.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Process for Machine Learning*. Cambridge: MIT Press.

Rullière, D., Durrande, N., Bachoc, F., & Chevalier, C. (2018). Nested Kriging predictions for datasets with a large number of observations. *Statistics and Computing, 28*, 849–867.

Snelson, E., & Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. In Advances in Neural Information Processing Systems 18, pp. 1257–1264.

Stein, M. L. (1999). *Interpolation of spatial data: Some theory for kriging*. New York: Springer,. https://doi.org/10.1007/978-1-4612-1494-6.

Tavassolipour, M., Motahari, S. A., & Shalmani, M. T. M. (2020). Learning of Gaussian processes in distributed and communication limited systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(8), 1928–1941. https://doi.org/10.1109/TPAMI.2019.2906207

Tresp, V. (2000). A Bayesian committee machine. *Neural Computation, 12*(11), 2719–2741. https://doi.org/10.1162/089976600300014908

van der Vaart, A., & van Zanten, H. (2011). Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research, 12*, 2095–2119.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). OpenML: Networked science in machine learning. *SIGKDD Explorations, 15*(2), 49–60. https://doi.org/10.1145/2641190.2641198

Wilson, A., & Nickisch, H. (2015). Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In Proceedings of the 32nd International Conference on Machine Learning, PMLR, pp. 1775–1784.

Woodruff, D. P. (2014). Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science, 10*(12), 1–157. https://doi.org/10.1561/0400000060