Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Digital Discovery*, 2024, **3**, 2211

Learning material synthesis–process–structure–property relationship by data fusion: Bayesian co-regionalization N-dimensional piecewise function learning†

A. Gilad Kusne, *[ab] Austin McDannald [a] and Brian DeCost [a]

Autonomous materials research labs require the ability to combine and learn from diverse data streams. This is especially true for learning material synthesis–process–structure–property relationships, key to accelerating materials optimization and discovery as well as accelerating mechanistic understanding. We present the Synthesis–process–structure–property relAtionship coreGionalized lEarner (SAGE) algorithm. A fully Bayesian algorithm that uses multimodal coregionalization and probability to merge knowledge across data sources into a unified model of synthesis–process–structure–property relationships. SAGE outputs a probabilistic posterior including the most likely relationship given the data along with proper uncertainty quantification. Beyond autonomous systems, SAGE will allow materials researchers to unify knowledge across their lab toward making better experiment design decisions.

Received 20th February 2024
Accepted 16th September 2024

DOI: 10.1039/d4dd00048j

rsc.li/digitaldiscovery

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

1. Introduction

Lack of advanced materials stymies many next-generation technologies such as quantum computing, carbon capture, and low-cost medical imaging. However, fundamental challenges stand in the way of discovering novel and optimized materials including (1) the challenge of a high-dimensional, complex materials search space and (2) the challenge of integrating knowledge across instruments and labs, *i.e.*, data fusion. The first challenge arises from the need to explore ever–more complex materials as simpler material systems are exhausted. Here material system refers to the materials resulting from a set of material synthesis and processing conditions. With each new material synthesis or processing condition, the number of potential experiments grows exponentially – rapidly escape the feasibility of Edisonian-type studies, forming a high-dimensional search space. As a result, any data is typically sparse relative to the search space. The search space is also highly complex due to the underlying complex relationship between material synthesis and process conditions and the resulting material structure and functional properties, *i.e.*, the material synthesis-process-structure–property relationship (SPSPR).

Knowledge of this SPSPR plays a fundamental role across materials research, whether the research is performed by hand or through an automated or autonomous system. Researchers use knowledge of the SPSPR as a blueprint to navigate the high-dimensional complex search space toward novel and optimized materials and to explore the underlying mechanistic origins of material properties. As a result, an algorithm that properly unifies diverse materials data into SPSPR models may accelerate all these activities, impacting much of materials research. For example, such an algorithm can exploit the SPSPR to dramatically improve prediction accuracy of a target functional property, despite sparsity of data. This improved prediction would then better guide subsequent research, which would in turn boost SPSPR knowledge.

Building the SPSPR blueprint involves combining knowledge of material synthesis and process conditions, lattice structure (and potentially microstructure), as well as the diverse set of functional properties required to meet the technological requirements. This requires integrating data across different instruments and measurement modalities, each dependent on differing physical principles. Additionally, measurements can vary based on instrument calibration, measurement parameter settings, environmental conditions such as temperature and humidity, and each instrument user's measurement process. Even instruments of the same make and model differ based on unique biases, uncertainties, and data artifacts.

As a very common example, researchers often start their search for improved materials with the phase map of the target material system. A phase map (or 'phase diagram' for equilibrium materials) visualizes the synthesis–structure relationship.

[a]*Materials Measurement Science, Division of the National Institute of Standards and Technology, Gaithersburg, MD 20899, USA. E-mail: aaron.kusne@nist.gov; austin.mcdannald@nist.gov; brian.decost@nist.gov*

[b]*Materials Science and Engineering, Department of the University of Maryland, College Park, MD 20742, USA*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4dd00048j

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

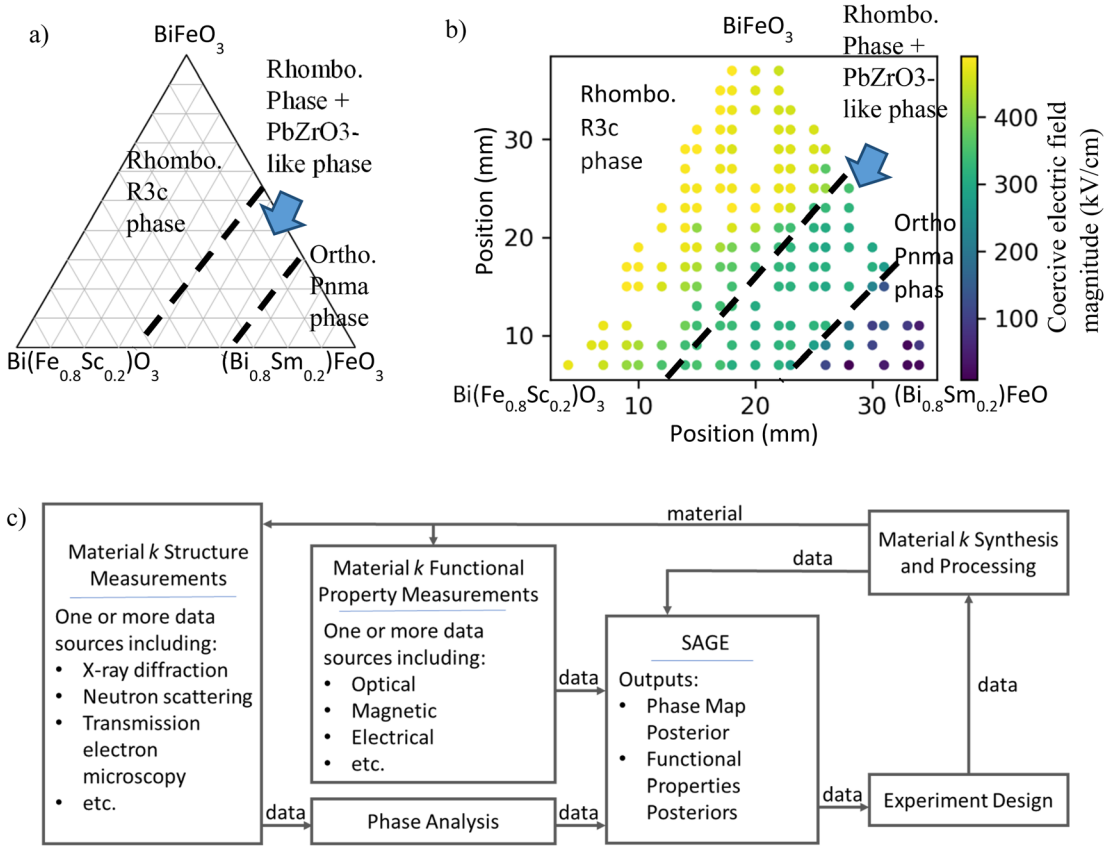

Fig. 1 (a) The (Bi,Sm)(Sc,Fe)O_3 material system experimentally identified phase diagram. Phase boundaries indicated by black dashed lines. (b) (Bi,Sm)(Sc,Fe)O_3 coercive electric field magnitude overlayed with phase diagram. Circles indicate experimentally characterized materials and color indicates coercive electric field magnitude between 0 kV cm^{-1} and 491 kV cm^{-1}. (c) SAGE schematic. A collection of materials spanning a target material system are characterized for multimodal structure data which is then processed through a preliminary phase analysis tool. The materials are also characterized for a range of functional properties. The collected data is passed to SAGE which outputs posterior probabilities for both the material system phase map and the functional properties. These posteriors can then be used in an experiment design (active learning) algorithm to determine the next material to investigate.

An example phase map is shown in Fig. 1a for the (Bi,Sm)(Sc,Fe)O_3 material system.[1] Here the phase map relates material composition (the target synthesis conditions) to resulting lattice structure, described in terms of phases, *i.e.,* composition-structure prototypes. The phase map is divided into phase regions – contiguous regions of synthesis-process space (experiments of varying synthesis and process conditions) that result in materials of the same set of phases. The regions are separated by phase boundaries (dashed black lines). Material phase information is predictive of many functional properties. Materials with property extrema tend to occur either within specific phase regions (*e.g.*, magnetism and superconductivity) or along phase boundaries (*e.g.*, caloric-cooling materials). Thus, a materials researcher can use phase maps to guide their studies toward synthesis and process conditions that are expected to produce materials with more promising properties.

Fig. 1b visualizes a (Bi,Sm)(Sc,Fe)O_3 SPSPR by combining the phase map with the functional property of coercive electric field magnitude (CEFM).[2,3] Circles indicate experimentally characterized materials and circle color indicates measured CEFM. The CEFM is highly dependent on both synthesis conditions and phase, with the highest values occurring with 'open' hysteresis loops in the rhombohedral *R3c* phase region. Additionally, the composition dependence of CEFM significantly differs between phase regions, with greater variation occurring in the *R3c* and *Pnma* phase regions than the intermediary region. In general, discontinuities in functional property values may also occur at phase boundaries. Thus, functional properties can be represented as piecewise functions of the synthesis parameters (in this case composition), with each 'piece' of the piecewise function associated with a phase region. This allows for significant changes in function behavior from region to region and/or discontinuities to occur at phase boundaries.

For this example, data for materials synthesis and structure are used to build a phase map, and that phase map is then used to guide understanding of target property data. Knowledge is one directional, from structure to functional property. With a proper SPSPR learning algorithm, these diverse data could be combined in a unified model where knowledge of the phase map would improve analysis and prediction of functional properties and *vice versa*. For example, significant changes in functional properties may indicate a phase boundary and thus

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

improve analysis and prediction of materials structure. Such an algorithm would boost overall materials research prediction accuracy and subsequent research, but such an algorithm has been lacking.

To overcome the dual challenges of a complex, high-dimensional search space and data fusion in SPSPR learning, we present the synthesis–process–structure–property relAtionship coreGionalized lEarner (SAGE). The SAGE algorithm is available as part of the Hermes library, **https://github.com/usnistgov/hermes** and as a standalone library **https://github.com/KusneNIST/SAGE**. SAGE is a Bayesian machine learning (ML) algorithm that combines three features: (1) ML-based segmentation of the synthesis-processing space using material synthesis, process, and phase data. Segments are phase regions, and the collection of phase regions forms the synthesis–process–phase map. The synthesis–process–phase map is then used to extrapolate the synthesis–process–structure relationship to new materials, (2) piecewise regression to fit and extrapolate synthesis–process–property relationships, and (3) coregionalization. Coregionalization allows multimodal, disparate knowledge of structure[1] and property,[2] both gathered across the shared domain of synthesis and processing conditions, to be combined to exploit shared trends. Here the term multi-modal refers to learning from disparate data sources (similar to its use in the common machine learning challenge of learning from text, audio, and image data). SAGE combines these three features to learn the most likely SPSPR model given material synthesis, process, structure, and property data. Multi-modal learning arises from exploiting both structure and functional property data to improve phase mapping (rather than using just the structure data) and both structure and functional property data to improve functional property regression (rather than just using the functional property data). Here the language of probability is used to unify knowledge across multi-modal data with assumptions represented as priors and data combined through likelihoods. Additionally, SAGE's Bayesian framework allows for full uncertainty quantification and propagation.

Much of machine learning focuses on algorithms that provide "point estimate" outputs, *i.e.*, they provide analysis or prediction without uncertainty. Proper uncertainty quantification and propagation requires explicitly expressing uncertainties in all variables and data, and then propagating these uncertainties through all computations to provide the uncertainty in the algorithm's outputs. A set of statistical learning algorithms such as Gaussian process regression were analytically developed to explicitly and properly manage uncertainties.[4] Due to the complexity (and wide-spread use) of many algorithms, computational methods are often employed to approximate output uncertainties without significantly changing the main algorithm.[5] Alternatively statistical methods such Bayesian inference can be used to build probabilistic models. With Bayesian inference, uncertainties are explicitly expressed and combined with Bayes rule to output the posterior probability (a probabilistic representation of uncertainty) – the probability distribution of the model given the data.[6] When analytically intractable, sampling methods such as Markov Chain Monty Carlo (MCMC) can be used to estimate these posterior probabilities.[6] Implementing these techniques is made easier through probabilistic programming languages such as Pyro and Turing.[7,8] SAGE employes Bayesian inference and MCMC for uncertainty quantification and propagation.

A schematic of SAGE is provided in Fig. 1c. Here SAGE takes in data streams from the material synthesis and processing systems, structure characterization instruments, as well as functional property characterization instruments. Each structure data stream is first processed using a phase analysis algorithm as described below. SAGE then learns the SPSPR from the combined phase analysis data streams and the functional property data streams. SAGE's output SPSPR posterior can be broken down into posteriors over the synthesis–process–structure phase map and the functional properties. These posteriors can then be integrated into either an experiment recommendation engine or a closed-loop autonomous materials laboratory,[9] which can guide subsequent experiments and measurements in structure and functional property. For example, an autonomous system could target maximizing knowledge of the SPSPR or optimizing a material for a set of target functional properties.

Each of SAGE's features has a diverse history. The first feature of ML-based phase mapping has a seen the development of an array of algorithms over the last few decades.[3,10–14] These algorithms combine two tasks, (1) data analysis: analyzing structure data to identify phase abundances or phase regions and (2) extrapolation: extrapolating phase knowledge from measured materials to unmeasured materials. Data analysis techniques (*i.e.*, phase or phase region identification) include matrix factorization, peak detection, graphical model segmentation, constraint programming, mixed integer programming, and deep learning, among others.[3,15–23] For an example of such an algorithm applied to the provided datasets, including a thorough description of these datasets, we refer the reader to ref. 3. Extrapolation algorithms have focused primarily on the use of graph-based models or Gaussian processes (GP).[16,17,24,25] (For a brief overview of Gaussian processes, see Section 4.2.6 Gaussian processes) For the present work, we assume the task of structure data analysis is addressed with one of the many available algorithms. We indicate the use of one of these algorithms with the function $m(D_s)$ applied to structure dataset D_s, as described below. SAGE therefore begins with knowledge of phase and focuses on the task of extrapolating phase map knowledge through Bayesian coregionalized synthesis and process space segmentation.

Piecewise function regression algorithms have a much longer history. This includes the common challenge of detecting data discontinuities – also known as jumps or changepoints, which can be generalized to higher dimensions as edges,[26] change-boundaries, and change-surfaces. Changepoint detection algorithms are quite diverse, using function derivatives, filter convolution, Bayesian inference, and more recently, adaptive design.[27] Common methods for piecewise regression include linear piecewise algorithms and splines. We point the reader to review articles in these fields.[28,29] Specifically for GPs,

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

multiple piecewise modeling methods exist[30] including the use of the changepoint kernel (below called GP-CP).[31]

The field of coregionalization developed from geospatial science to learn functions with shared trends over the same physical domain.[32,33] Data for each target function is not required to be collected for the same set of points in the input domain.[33] For example, if one seeks to learn $f_1{:}x \rightarrow y$ and $f_2{:}x \rightarrow s$, data $D_1 = \{(x_k, y_k)\}_{k=1}^{N}$ and $D_2 = \{(x_l, s_l)\}_{l=1}^{M}$, the set of input locations $\{x_k\}_{k=1}^{N}$ and $\{x_l\}_{l=1}^{M}$ are not required to correspond to the same locations. Alternative methods for jointly learning related functions include multi-task learning, co-kriging, including multi-task Gaussian processes[5,34] as well as constraint programming methods and Bayesian methods.[33,35,36] These algorithms focus on exploiting similarities between functions over the full underlying shared domain, assume the set of output functions are similar (*e.g.*, all continuous), and assume that each experiment is characterized similarly. Recent work tackles learning heterogenous sets of functions such as a mix of continuous, categorical, and binary outputs.[37] These algorithms assume a correlation between a set of latent functions that contribute to the observed output functions.

Our challenge is unique. While we seek to jointly learn the synthesis–process–structure relationship and synthesis–process–property relationships, the correlation of interest between these relationships is purely that of discontinuities, rather than correlations over the full synthesis–process domain. We assume that phase boundaries indicate potential change surfaces in functional properties, and *vice versa*. We wish to jointly learn these phase boundaries and utilize them to define piecewise functions for the functional properties, allowing for different property behavior in different phase regions. Prior algorithms fail for this challenge as the synthesis–process–structure relationship and those of synthesis–process–properties are not correlated over the full synthesis–process domain (this is also true for latent function representations). Additionally, SAGE utilizes coregionalization to allow different measurements to be performed at different locations in the shared synthesis and process domain. This is commonly the case when materials synthesis and processing experiments take equal or less time than the measurements or when combining data collected at different times or by different labs.

To the authors' knowledge, the only algorithm that addresses the same challenge is the closed-loop autonomous materials exploration and optimization (CAMEO) algorithm.[16] CAMEO first learns phase boundaries from synthesis, process, and structure data and then utilizes this knowledge to define the change boundaries in the piecewise function used to fit and model functional property data. This two-step approach was employed in driving an X-ray diffraction-based autonomous (robot) materials research system in the study of phase-change memory material. The study resulted in the discovery of the current best-in-class phase-change memory material – the first autonomous discovery of a best-in-class solid state material.[16] SAGE improves on CAMEO by allowing full Bayesian uncertainty quantification and propagation, thus providing simultaneous information sharing between the structure and property measurements. SAGE jointly solves for the SPSPR to better exploit shared trends across structure and property data and improve SPSPR knowledge. SAGE is offered as a module of CAMEO, *i.e.*, CAMEO-SAGE.

The present data science challenge is generalizable beyond learning SPSPR. One can use SAGE to address the more common issue of having successful and failed experiments across a shared experiment parameter domain. SAGE would then learn and exploit knowledge of the success–failure boundary to improve prediction of properties of either type of experiments. Additionally, SAGE addresses data fusion across instruments, measurement modalities and labs. The common approach to this data fusion challenge is to map data from different sources into the same data space, allowing comparison. For example, data fusion for X-ray diffraction (XRD) measurements from two different XRD instruments requires removing source-based data artifacts including instrument effects that are convolved into the data. To do this, that data must then be mapped from the instrument specific, source-based independent variable space (2θ) to an instrument-free independent variable space (q), while also accounting for differences in finite resolution in 2θ space, absolute intensities and counting times, beam wavelength dispersion, and background signals, amongst other considerations. In general, data mapping to an instrument (also lab, weather, *etc.*) invariant space requires a significant amount of meta data that is often not available.

An alternative is to independently analyze the data from each source and then combine the derived knowledge across sources. SAGE allows such limited-metadata data fusion. The idea behind coregionalization, as implemented in SAGE, is that the boundaries identified by one measurement method are also boundaries in the other measurement methods – regardless of if those measurement methods are all nominally the same technique (*e.g.*, several different XRD instruments) or different techniques (*e.g.*, an XRD instrument and electrical coercivity measurements). For example, for structure data, one performs phase mapping analysis for each data source and then SAGE coregionalization combines knowledge across sources. A similar benefit exists for functional property data by treating data from each source as a different target property, *e.g.*, coercivity_data_source_1 and coercivity_data_source_2. Additionally, SAGE may be applied to cases where only structure data or only functional property data is obtained.

The contributions of this work are:

- Extending Bayesian coregionalization algorithms to 1-dimensional and N-dimensional joint segmentation and piecewise regression.
- Associated constraint programming algorithms for coregionalized joint segmentation and piecewise regression.
- Demonstration of Bayesian algorithms for learning SPSPR in a unified model.

SAGE is a physics-informed (also known as inductive-bias informed) machine learning algorithm.[38] A wide array of methods exists for integrating prior physical knowledge into machine learning methods, including engineering descriptors,[39,40] latent mappings,[12] constrained solution spaces,[41] kernels,[42] among many others. For example, a physics-informed

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

algorithm was designed for autonomous, closed-loop control over neutron scattering to accelerate characterization of temperature-dependent magnetic structure.[43] The authors represent the temperature-dependent structure as a stochastic process with neutron scattering-defined measurement uncertainties as well as a mean function prior defined by magnetics physics. The algorithm resulted in a fivefold acceleration in measurement efficiency. However, no previous algorithms provide the contributions listed above. Such physics-informed methods provide greater performance and lend greater interpretability to the machine learning model – providing more physically meaningful solutions.

While the provided implementation of SAGE is a surrogate model, its framework allows easy modification to embed greater prior knowledge and to increase interpretability. Target functional properties are currently defined through samples of multivariate normal distributions, similar to a Gaussian process. To increase interpretability, users can replace these samples with samples of potentially descriptive parametric models (as well as a parameter that selects between the models). SAGE will then identify the most likely model and posteriors over its parameter values. In this way a user can exploit SAGE's built-in coregionalization of functional property with phase mapping (*i.e.*, enforced SPSPR) to boost data analysis. Additionally, one can modify parameter priors. For example, setting segmentation length scales to a Gamma distribution to increase bias for small or larger phase regions.

2. Results

We demonstrate SAGE for 1D and 2D example challenges. For both 1D and 2D, we first investigate performance for 2 edge cases, each with artificial phase maps of 2 phase regions and one artificial target functional property. In the first edge case, structure data is more informative of the change boundary and in the second edge case the functional property data is more informative of the change boundary. These edge cases demonstrate SAGE's ability to exploit knowledge across both structure and functional property data to improve prediction of both. We then provide an example of SAGE's multi-data source coregionalization capabilities with a challenge of 2 structure data sources and 2 functional property data sources. This is followed by a real-world application to the (Bi,Sm)(Sc,Fe)O_3 and $FeGaPd^3$ material systems.

As described above, SAGE performs the two tasks jointly of (1) segmenting the synthesis–process domain X into phase regions *i.e.*, a phase map and (2) regression of the functional properties. SAGE tackles both tasks by exploiting shared knowledge (*i.e.*, multimodal knowledge) across the structure and functional property data. We compare SAGE to a set of common algorithms and modified SAGE algorithms as described below (see section Additional models). SAGE's phase mapping capabilities are compared to that of GPC, the modified SAGE algorithm SAGE-1D-PM, and SAGE-ND-PM using only synthesis–process and structure data to segment the synthesis–process space. SAGE's functional property regression capability is compared to that of GPR, GP-CP, and the modified SAGE algorithms SAGE-1D-FP and SAGE-ND-FP using only synthesis–process and functional property data. Here the GP-CP algorithm seeks to perform functional property regression while also identifying change points (*i.e.*, phase boundaries) without access to structure data. SAGE's capabilities for jointly performing phase mapping and functional property regression from synthesis, process, structure, and property data is

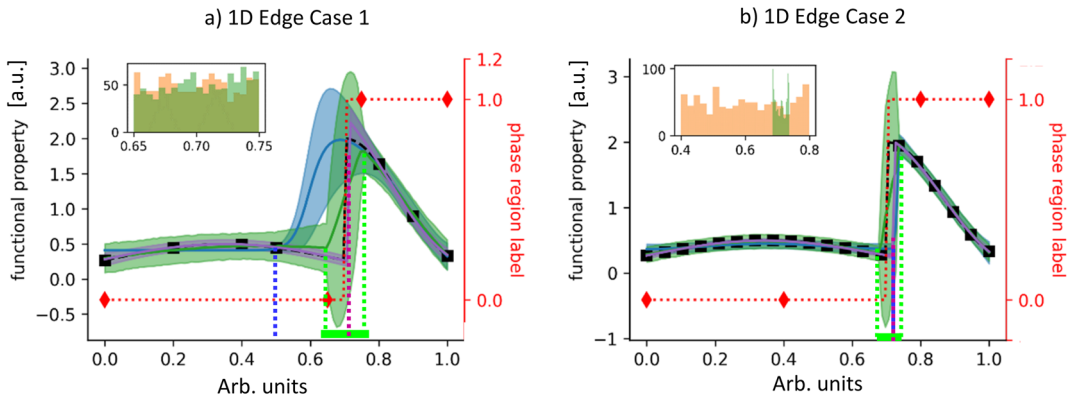

Fig. 2 SAGE-1D performance for two edge cases: (a) edge case 1 where structure data (red diamonds) is more informative of the phase boundary and (b) edge case 2 where functional property data (black squares) is more informative of the phase boundary. In both parts, the main plots show comparison of: the SAGE-1D posterior mean and 95% confidence interval (green line and shaded region), and SAGE-1D maximum likelihood mean estimate (MLE, magenta line and shaded region), and GP-CP – a GP (blue line and shaded region) with a changepoint kernel and two radial basis function kernels on either side of the changepoint. The ground truth phase map is indicated by a red dotted line and the ground truth functional property function is indicated as a black dashed line. The GP identified changepoint is indicated with a blue dotted line and the range of potential changepoints identified by SAGE-1D is indicated by green dotted lines. The SAGE-1D changepoint posterior is shown as the inset (green histogram) and compared to the SAGE-1D-PM changepoint detection algorithm (orange histogram). For the first edge case (a), the GP-CP is unable to find the correct phase boundary, due to only having access to functional property data and thus identifies significant variations in the functional property to be due to data noise. SAGE-1D MLE properly identifies both the phase boundary and the functional property behavior on either side. For the second edge case (b), both GP-CP and SAGE perform well at identifying the phase boundary and the functional property behavior as the functional property data is informative of both.

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

compared to the CAMEO's regression algorithm which first uses synthesis–process and structure data to identify a phase map and then employs the phase map in piecewise regression for functional property data.

For each experiment, an algorithm is given a subset of materials data – for each challenge, all algorithms are given the same set of data (as visualized by red and black markers for 1D challenges and red markers for 2D challenges in the figures below) – and each algorithm is then used to predict one of or both of (based on its capabilities): (1) the phase map over the synthesis–process domain X, (2) functional properties over X. For 1D challenges this is over a 100-point grid and for 2D challenges a 40×40-point grid (see Fig. 2 and 3 for ranges). For instance, GPC takes a subset of data in X and the associated structure data and then predicts the phase map over the associated X grid; GPR takes a subset of data for X and associated functional properties data to provide functional property regression over the X grid.

Phase mapping performance is measured by comparing predicted phase map labels to ground truth (over the grid) using the micro F1 accuracy score. Ground truth for the 1D and 2D cases can be seen in the red and black dashed curves of Fig. 2 and the color-coded values of Fig. 3. Functional property regression performance is quantified by comparing predicted regression models with the ground truth using the typical coefficient of determination R^2. A description of these measures can be found in the Methods section. Furthermore, the performance of MCMC-computed algorithms is based on their posterior mean and the performance of variational inference-based algorithms (*e.g.*, GPs) is based on their maximum likelihood estimate (MLE) mean as given in Table 1.

2.1. 1D examples

The 1D challenges are shown in Fig. 2 with the target functional property shown as a black dashed curve and the phase map shown as a dotted red curve that switches between a value of 0 and 1 at $x = 0.7$. For the first edge case, structure data (red diamonds) is more informative of the phase boundary, compared to functional property data (black squares). The reverse is true for the second edge case. To compare functional property prediction performance, in Fig. 2a and b we plot: (1) SAGE-1D's functional property posterior mean (solid green line) and 95% confidence interval (shaded green area), (2) an off-the-shelf GP with the changepoint kernel[31] (GP-CP, blue line and shaded area) which uses maximum likelihood estimate (MLE), and (3) a plot of SAGE-1D's maximum likelihood sample (MLS,

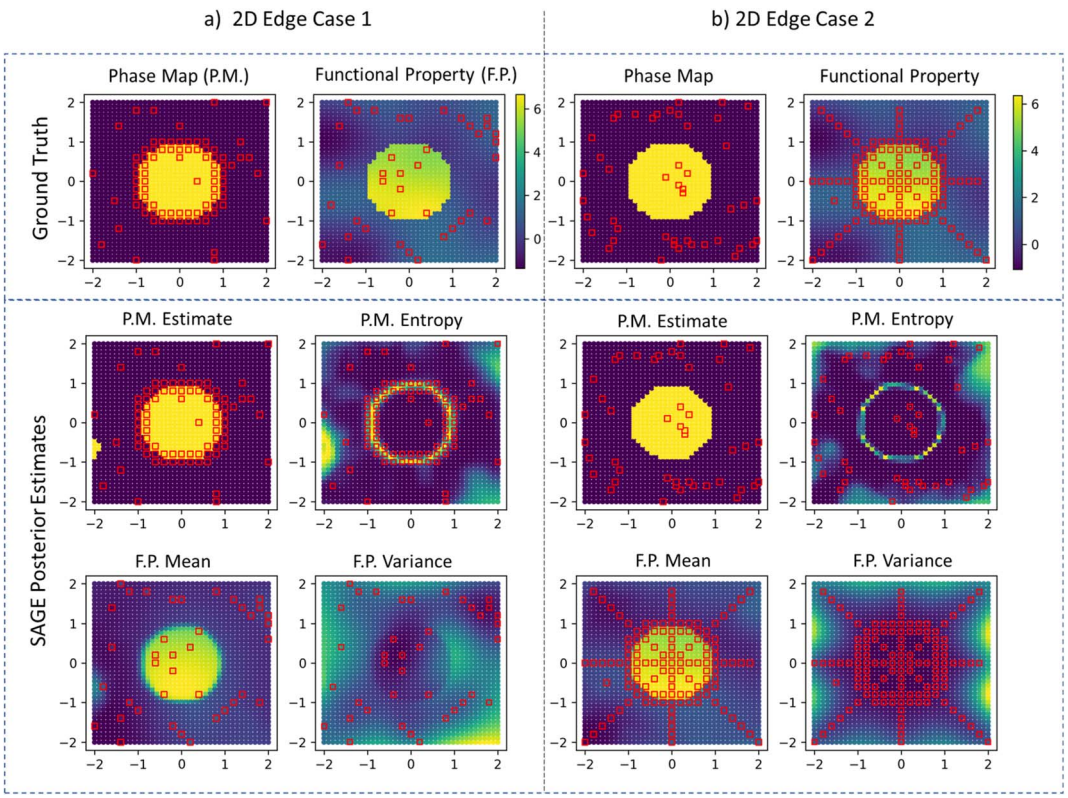

Fig. 3 SAGE-ND demonstration on 2D example for the 2 edge cases: (a) edge case 1 where structure data (red squares in phase map plots) is more informative of the phase boundary and (b) edge case 2 where functional property data (red squares in functional property plots) is more informative of the phase boundary. Here the yellow and blue phase map color coding indicates the two phase regions either as ground truth or predicted (based on figure label). The algorithm shows good agreement with the ground truth for both cases. For the first edge case (a), SAGE-ND properly identifies both the phase boundary and the functional property behavior, using more informative structure data to improve prediction of both. For the second edge case (b) SAGE-ND properly identifies both the phase boundary and the functional property behavior, using the more informative functional property data to improve prediction of both.

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Table 1 Performance scores comparing SAGE with alternative algorithms for the 1D and 2D edge cases and the real-world (Bi,Sm)(Sc,Fe)O_3 and FeGaPd challenges. Here both 1 and N-dimensional Edge Case 1 has structure data that is more informative of the phase boundary and edge case 2 has functional property more informative of the phase boundary

Phase map performance, micro F1 accuracy score [arb. Units]

1D challenges	SAGE-1D (post. mean)	SAGE-1D-PM (post. mean	SAGE-1D-FP (post. mean)	GP-CP (max likelihood)	GP classification (max likelihood)	CAMEO prediction
1D edge case 1	**1.00**	**1.00**	0.89	0.82	**1.00**	**1.00**
1D edge case 2	**0.99**	0.89	**0.99**	0.90	0.90	0.86

Phase map performance, micro F1 accuracy score [arb. Units]

2D challenges	SAGE-ND (post. mean)	SAGE-ND-PM (post. mean)	SAGE-ND-FP (post. mean)	—	GP classification (max likelihood)	CAMEO
2D edge case 1	**0.98**	0.97	0.85	—	**0.98**	0.94
2D edge case 2	**0.98**	0.92	0.97	—	0.93	0.53
(Bi,Sm)(Sc,Fe)O_3	0.97	0.94	0.61	—	0.89	**0.99**
FeGaPd	0.95	0.93	0.13		**0.99**	0.96

Functional property performance R^2 [arb. units]

1D challenges	SAGE-1D (post. mean)	—	SAGE-1D-FP (post. mean)	GP-CP (max likelihood)	—	CAMEO prediction
1D edge case 1	0.99	—	0.98	0.96	—	**1.0**
1D edge case 2	**1.00**	—	**1.00**	0.98	—	0.92

Functional property performance R^2 [arb. units]

2D challenges	SAGE-ND (post. mean)	—	SAGE-ND-FP (post. mean)	—	GP regression (max likelihood)	CAMEO
2D edge case 1	**0.88**	—	0.53	—	0.67	0.86
2D edge case 2	**0.89**	—	0.87	—	0.62	0.67
(Bi,Sm)(Sc,Fe)O_3	**0.91**	—	0.27	—	0.84	0.87
FeGaPd	**0.91**		0.87		0.90	**0.91**

magenta line and shaded area) – the MCMC sample with the maximum computed likelihood. This sample contains an explicit changepoint value and associated piecewise GP regression. For phase boundary prediction comparison, we plot: (1) SAGE-1D's phase boundary posterior distribution (green inset histogram) and (2) SAGE-1D-PM's posterior distribution (orange inset histogram).

For the first edge case, SAGE-1D MLS combines structure and functional property knowledge to outperform GP-CP in predicting both functional property and phase boundary. SAGE-1D's slanted transition at the phase boundary (Fig. 2a) indicates a range of potential phase boundary locations between the two structure data points (range is also indicated by the dotted green lines). SAGE-1D and SAGE-1D-PM have similar performance in identifying the phase boundary location, providing similar posteriors (inset). SAGE-1D employs phase boundary uncertainty to better quantify its regression uncertainty as indicated by the wider confidence intervals.

For the second edge case, SAGE-1D MLS and GP-CP have similar regression performance due to the highly informative functional property data. However, SAGE-1D outperforms SAGE-1D-PM in locating the phase boundary, as it exploits functional property data to greatly narrow in on potential locations. A further comparison between SAGE-1D, GP-CP, SAGE-1D-PM, SAGE-1D-FP, and GP classification are presented in Table 1. Knowledge of the changepoint location is limited to the two nearest data points, either functional property or structure data. As a result, functional property prediction performance is measured outside the range of the two nearest data points.

2.2. 2D examples

We observe similar behavior in the 2D demonstration of SAGE-ND as shown in Fig. 3. The location of structure data (red squares on phase map plots) and functional property data (red squares on functional property plots) are indicated. For phase map prediction, SAGE-ND is compared to SAGE-ND-PM, SAGE-ND-FP, and GP classification. For functional property prediction, SAGE-ND is compared to SAGE-ND-FP and off-the-shelf GP regression. Performance scores are reported in Table 1. SAGE-ND outperforms the other methods in both phase mapping and functional property prediction for both edge cases. In edge case 2, despite highly informative functional property data, SAGE-ND outperforms off-the-shelf GP regression due to its

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

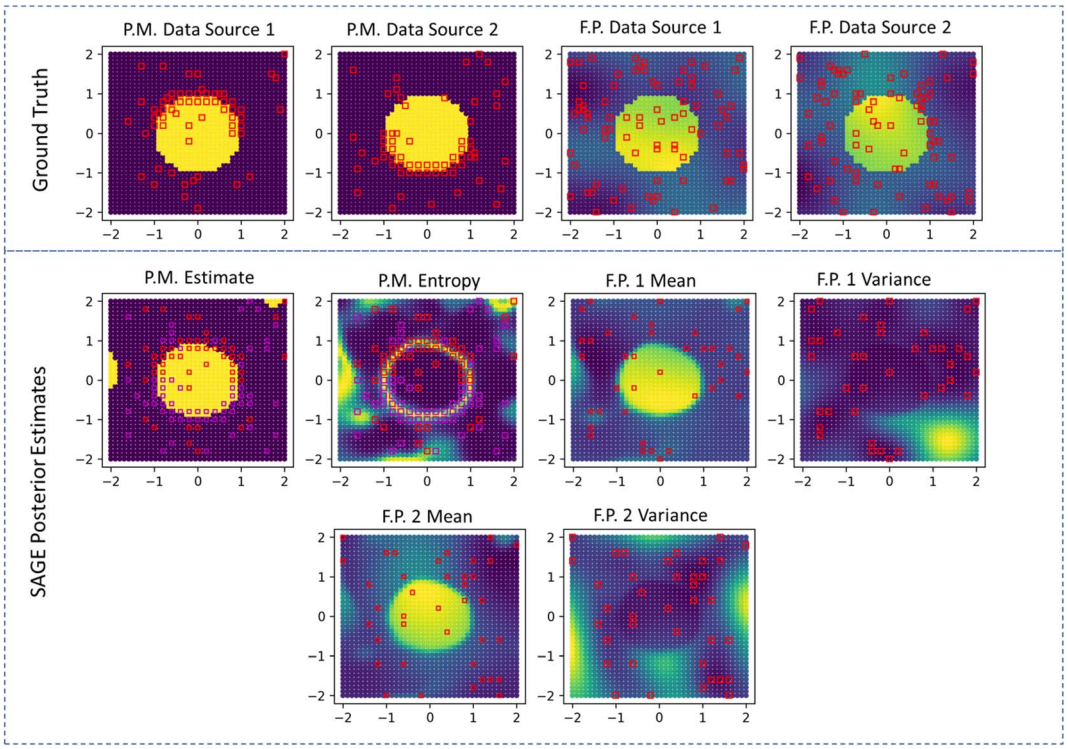

Fig. 4 Demonstration of SAGE-ND algorithm for 2 structure data sources and 2 functional property data sources. The two structure data sources provide phase boundary information in different regions of the phase map. SAGE-ND combines knowledge from these two structure data sources as well as the two functional property data sources to properly identify the phase boundary as well as the behavior of both functional properties.

ability to properly deal with the change in property and change in hyperparameters across the phase boundary.

In Fig. 4 we demonstrate the ND algorithm for the 2D case with 2 structure data sources and 2 functional property sources. Here the first structure data source provides more information for the upper part of the phase boundary and the second source provides more information for the lower part of the boundary. SAGE-ND unifies knowledge across all four data sources to obtain good prediction of both phase map and the two functional properties.

2.3. Materials example

For the first materials challenge demonstration, SAGE-ND is applied to learn a SPSPR for a (Bi,Sm)(Sc,Fe)O_3 composition spread dataset of Raman spectra structure measurements and CEFM as shown in Fig. 5. As structure data is collected primarily to learn the phase map, we present the case where structure data is more informative of the phase boundaries than the functional property data. Phase mapping and CEFM predictions estimates are shown in Fig. 5a1 and 5b1 and uncertainties in Fig. 5a2 and 5b2 respectively.

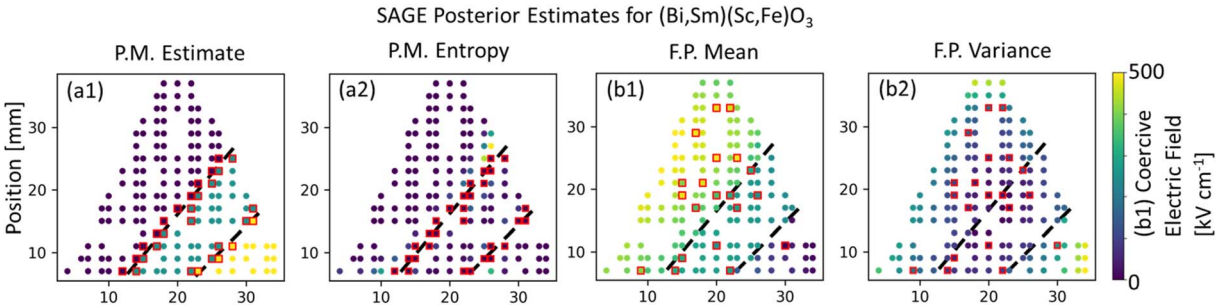

Fig. 5 SAGE-ND applied to (Bi,Sm)(Sc,Fe)O_3 dataset, where structure data is more informative of the phase boundaries. Fig. 1 shows the ground truth. (a1) Phase map estimate indicated by color coding with structure data indicated with red squares and phase boundaries indicated by dashed black lines. (a2) Entropy-measured uncertainty in the phase map of (a1), (b1) CEFM estimate with functional property data indicated with red squares. (b2) Variance-measured uncertainty for the CEFM estimate. SAGE-ND utilizes the more informative structure data to identify the phase regions and utilizes this information to better identify the varying CEFM behavior in each phase region.

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

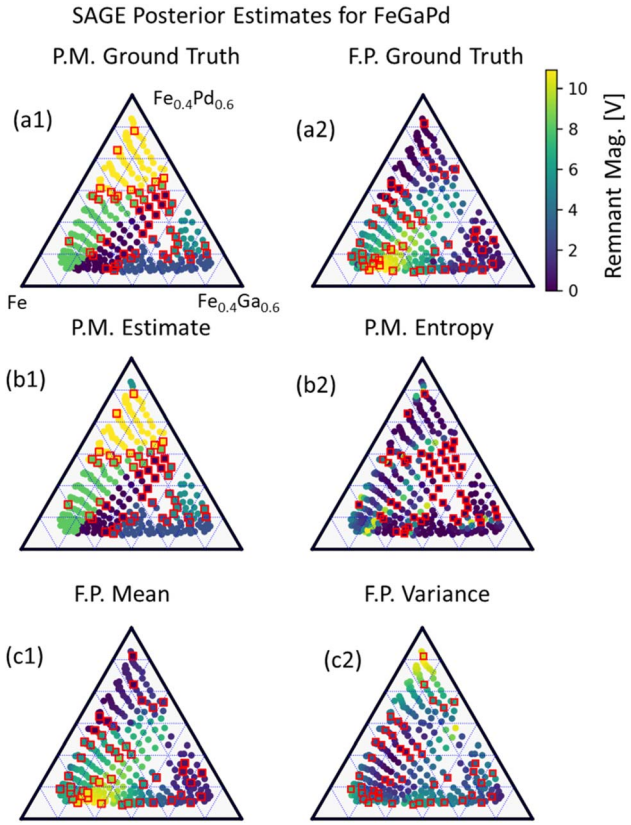

Fig. 6 SAGE-ND applied to FeGaPd dataset, where structure data is more informative of the phase boundaries. (a1) Phase map ground truth with structure data points indicated with red squares, (a2) functional property ground truth with functional property data points indicated with red squares, (b1) phase map estimate indicated by color coding, (b2) entropy-measured uncertainty in the phase map of (b1), (c1) remnant magnetization estimate with measured data indicated with red squares. (c2) Variance-measured uncertainty for the remnant magnetization estimate. From (a2) one can see that functional property behavior (*e.g.*, length scale) is dependent on phase region. Within the yellow, light blue and dark green indicated regions (a1) there are lower remnant magnetization values and more slowly varying values as a function of composition, compared to the behavior in the light green and dark blue indicated phase regions. SAGE-ND utilizes structure data to better identify the phase boundaries and use this to better identify the varying behavior of the functional property across the phase map.

For the second materials challenge demonstration, SAGE-ND is applied to learn a SPSPR for a $FeGaPd^3$ composition spread dataset of X-ray diffraction structure measurements and remanent magnetization as shown in Fig. 6. Ground truth phase mapping and remanent magnetization are shown in Fig. 6a1 and 6a2, respectively. SAGE prediction estimates are shown in Fig. 6b1 and 6c1 and uncertainties in Fig. 6b2 and 6c2, respectively.

For both material systems, a comparison of SAGE-ND with SAGE-ND-PM, SAGE-ND-FP, GP classification, GP regression, and CAMEO are shown in Table 1. For the $(Bi,Sm)(Sc,Fe)O_3$ dataset, SAGE-ND provides 97% (or 0.97 out of 1.00) or greater phase mapping accuracy, though not the top accuracy among algorithms. For $(Bi,Sm)(Sc,Fe)O_3$, CAMEO outperforms SAGE-ND by 2% and for FeGaPd, GPC outperforms SAGE-ND by 4%. For functional property predictions, SAGE-ND provides the best (or tied for best) performance.

Better performance for predicting functional properties over phase mapping is to be expected as there is greater information of the phase boundaries from the structure data than the functional property data. Thus, SAGE exploits knowledge from the structure data (knowledge of the synthesis–process–structure relationship) to boost functional property prediction.

While SAGE-ND does not always provide the best results, it does provide proper uncertainty quantification compared to CAMEO. Additionally, if the SAGE assumption that the target functional property behavior is dependent on phase is true for the material system of interest, SAGE exploits this relationship to improve uncertainty quantification using both structure and functional property data compared to methods (*e.g.*, GP methods) that can only utilize either structure or functional property data. SAGE is also the only algorithm which provides a single model for quantifying both prediction and uncertainty for the synthesis–process–structure–property relationship.

3. Conclusion

SAGE allows one to combine knowledge of material structure and material property from multiple data sources into one unified SPSPR model, exploiting shared data trends to maximize knowledge of the phase diagram and functional properties. The Bayesian inference methodology allows for appropriate quantification of uncertainty. By providing probabilistic descriptions of data of varying quality or fidelity (whether theory-derived or experimental), these uncertainties can then be propagated through the model by sampling the data distributions along with the model parameters and/or by replacing the piecewise GPs with heteroskedastic GPs. Additionally, correlations between functional properties can also be exploited by replacing the functional property-representing independent piecewise GPs with a coregionalized multi-output GP. These points will be the focus of future work.

Model output estimates and uncertainties can be employed in active learning-driven recommendation engines or closed-loop autonomous systems, to ensure optimum selection of subsequent experiments. For example, the phase map estimate and uncertainty can guide subsequent structure measurements toward improved phase map knowledge while the paired functional property estimates and uncertainties guide materials optimization. With each experiment increasing knowledge of separate portions of the SSPR, SAGE can play a part in unifying knowledge across a research lab toward the discovery of advanced materials.

4. Methods

We present coregionalization algorithms for combining multiple data sources for materials synthesis, process, structure, and property to learn the SPSPR over the shared synthesis and processing domain $\boldsymbol{x} \in \boldsymbol{X}$. Structure data from data source i is represented by $D_{s,i} = \{\boldsymbol{x}_k, \boldsymbol{z}_{k,i}\}_{k=1}^{N_i}$ for material $\boldsymbol{x}_k$ (data pair indexed with k) and its associated structure descriptor $\boldsymbol{z}_{k,i}$, with N_i data

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

pairs collected from data source i. The full set of structure data is labeled D_s, where the subscript s indicated structure-associated data. Similarly, property data from data source j is represented by $D_{p,j} = \{\boldsymbol{x}_l, y_{l,j}\}_{l=1}^{N_j}$ for material $\boldsymbol{x}_l$ and its associated material property measurement $y_{l,j}$, and where subscript p indicates functional-property-associated data. $D_{s,i,k}$ and $D_{p,j,k}$ are the i-sourced structure data for $\boldsymbol{x}_k$ and the j-sourced functional property data for $\boldsymbol{x}_l$. For this work we assume each data source provides data for one property. The full set of functional property data is labeled D_p. This representation allows for duplicate measurements of the same material from different data sources. The function $m(D_s)$ maps dataset D_s to a set of phase map labels. It is one of the many such algorithms described above, and as such is not part of SAGE.

4.1. Constraint programming

The constraint programming algorithm (eqn (1)) is defined by finding the set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_s, \boldsymbol{\theta}_p\}$ that minimize the objective function Obj. The phase map is described by the function $f_s(\boldsymbol{x}, \boldsymbol{\theta}_s)$ which maps each point $\boldsymbol{x}$ in the target synthesis-process space $\mathbf{X}$ to a set of phase labels $\mathbf{s}$, f_s:$\boldsymbol{x} \rightarrow \mathbf{s}$, where $\boldsymbol{\theta}_s$ is the associated set of parameters. The functional property is described by the piecewise function $f_p(\boldsymbol{x}, f_s, \boldsymbol{\theta}_p)$ which maps each point $\boldsymbol{x}$ to a set of functional properties $\boldsymbol{y}$, *i.e.*, f_p:$\boldsymbol{x} \rightarrow \boldsymbol{y}$. This function is dependent on the set of parameters $\boldsymbol{\theta}_p$ and its piecewise nature is dependent on f_s. The functions d_s and d_p compute the relationship – typically the loss, between the function f_s and data D_s or between f_p and data D_p, respectively. For example, d_p can combine a measure of goodness of fit of f_p and model complexity, *e.g.*, the Bayesian information criteria.[44] To quantify loss for structure data, the data D_s must also be mapped to a set of phase map labels, here performed by the function $m(D_s)$ (As discussed above, this function is one of the many found in the literature). Minimizing the objective involves: (1) identifying potential values for parameters $\boldsymbol{\theta}_s$, (2) solving for $m(D_s)$ and f_s, (3) identifying potential values for parameters $\boldsymbol{\theta}_p$, (4) solving for f_p, and (5) computing the overall loss for the objective function. This iterative approach allows a target property estimate to inform the subsequent optimization of $f_s(\boldsymbol{x})$.

$$\text{Obj} = \min_{\{\theta_s,\, \theta_p\}} \left[d_s\left(f_s(\boldsymbol{x}, \boldsymbol{\theta}_s), m(D_s)\right) + d_p\left(f_p\left(\boldsymbol{x}, f_s, \boldsymbol{\theta}_p\right), D_p\right)\right] \quad (1)$$

If the loss functions are additive across datasets, we have:

$$\text{Obj} = \min_{\{\theta_s,\, \theta_p\}} \left[\sum_i d_s\left(f_s(\boldsymbol{x}, \boldsymbol{\theta}_s), m(D_{s,i})\right) + \sum_j d_p\left(f_p\left(\boldsymbol{x}, f_s, \boldsymbol{\theta}_p\right), D_{p,j}\right)\right], \quad (2)$$

One implementation has f_s map each point to an integer label associated with a given phase region. The function m_s is then required to map the structure data to potential phase region labels similar to those of.[3,10–13,15] Alternatively, one may want the overall algorithm to identify phase abundances for each material $\boldsymbol{x}$. For this case, m_s identifies phase abundances and maps $\boldsymbol{x}$ to phase region labels. Abundance regression can then be performed by including abundances in the list of target properties $\boldsymbol{y}_p$.

The Bayesian model presented below can be solved using such an objective function. Here, d_s and d_p are the negative log likelihood functions:

$$d_s = -\ln[p(f_s(\boldsymbol{x}, \boldsymbol{\theta}_s)|m(D_s)], \quad (3)$$

$$d_p = -\ln[p(f_p(\boldsymbol{x}, f_s, \boldsymbol{\theta}_p)|D_{p,j})], \quad (4)$$

i.e., the negative log likelihood of data $\{D_s, D_p\}$ being observed for functions $\{f_s, f_p\}$. This gives: Obj $= \min_{\{\theta_s,\, \theta_p\}} [-(L_s + L_p)]$where L_s and L_p are the sum log likelihoods over all structure or all functional property observations, respectively. The notation $p()$ represents a pdf, $p(a|b)$ describes the pdf of a given b, and for the equations below, $a \sim p(b)$ indicates drawing independent and identically distributed samples from $p(b)$. Solving for $\{\theta_s, \theta_p\}$ may be done under the variational inference approximation. The results presented her focus on Markov Chain Monte Carlo (MCMC) computed posteriors. The variational inference approximation can be used to initialize MCMC and speed up calculations.

4.2. Bayesian models

We provide two Bayesian models, one for challenges where $\mathbf{X}$ is one dimensional (*i.e.*, only one synthesis or process parameter is investigated) and one where $\mathbf{X}$ is of arbitrary dimension. Rather than minimizing loss, the aim of these models is to maximize the sum log likelihood L over the set of parameters and observed data (*e.g.*, minimize sum of negative log likelihood as above). Here, MCMC is used to compute a posterior for each model parameter. Additionally, one can incorporate prior physical knowledge by modifying parameter prior probability density functions (pdf). For example, if one believes there to be many small phase regions, the uniform prior for l_s can be replace with a Gamma distribution. Large expected fluctuations in a functional property can be included through modifying the $s_{r,j}$ prior. Both the 1-dimensional and N-dimensional models output an estimate for the posterior of the SPSPR and each parameter (given the model and data) – providing both an estimate and uncertainty, compared to the constraint programming algorithm which outputs a point estimate (estimates of the uncertainty can also be obtained). The posteriors can be used in further Bayesian analysis as demonstrated below. The MCMC Bayesian inference method for evaluating the models consists of: (1) sampling function parameters, (2) using the samples to define f_s and f_p, and then (3) compute the log likelihood L.

Model 1 provides the general model. One samples the function parameter priors for $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_p = \{\boldsymbol{\theta}_{p,j,r}\}$ for each j of J functional properties (or function property data source) and each r of R phase regions. $f_{p,j}$ is a piecewise random process with different behavior $f_{p,j,r}$ for each functional property in each phase region, *i.e.*, different kernel hyperparameters for each phase region. f_s is used to compute the categorical distribution $p(r(\boldsymbol{x}))$ of phase regions labels for each point $\boldsymbol{x}$. $p(r(\boldsymbol{x}))$ is used to compute the sum log likelihood L_s of structure data

 © 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

observations and identify phase region label probabilities for each functional property observation data point $\boldsymbol{x}_\mathrm{p}$. The sum log likelihood of the observed functional properties L_p is computed using these probabilities and the piecewise $f_\mathrm{p,j}$. The total likelihood L is then returned, guiding Bayesian inference sampling. The implementations and associated code can be used with an arbitrary number of data sources. Sampling from GPs uses the Cholesky decomposition method to improve MCMC stability.[4]

Model 1 General SAGE Model

$$\boldsymbol{\theta}_s \sim prior_{\theta_s}$$
$$p_r(r) = p_r(r(\boldsymbol{x}) = l) = \mathrm{Categorical}\left(f_s(\boldsymbol{\theta}_s)\right)$$
$$L_s = \sum_i \sum_k \ln\left[p\left(m\left(D_{s,i,k}\right)|p_r(r)\right)\right]$$
$$\boldsymbol{\theta}_p = \left\{\boldsymbol{\theta}_{p,j,r}\right\}_{j=1,r=1}^{j=J,r=R} \sim prior_{\theta_{p,j,r}}$$
$$f_{p,j} = \sum_r f_{p,j,r}\left(\boldsymbol{\theta}_{p,j,r}\right) p_r(r)$$
$$L_p = \sum_j \sum_k \ln\left[p\left(D_{p,j,k}|N\left(f_{p,j}, \boldsymbol{\theta}_p\right)\right)\right]$$
$$L = L_s + L_p$$

After Bayesian inference is run, *i.e.*, each b sample of B total MCMC samples are collected, the Bayesian posteriors for the phase map and functional-properties-describing functions are approximated. Here the categorical distribution describing the phase map is computed by taking $\mathrm{mean_b}[p_\mathrm{b}(r)]$, the posterior mean over the sampled categorical distributions. The phase map estimate $\hat{p}$ and uncertainty $e_{\hat{p}}$ are then computed with $\hat{p} = \mathrm{argmax_r}[p_{\boldsymbol{M}}]$ and $e_{\hat{p}} = \mathrm{entropy_r}[p_{\boldsymbol{M}}]$. Each functional property is described by the posterior multivariate normal distribution $N(\mathrm{mean_b}[f_\mathrm{p,j,b}], \mathrm{std_b}[f_\mathrm{p,j,b}])$ with additional measurement noise $\mathrm{mean_b}[n_\mathrm{p,j,b}]$.

Algorithm 1 Post MCMC Bayesian Analysis

For the Bayesian inference sample index b:

$$p_{\boldsymbol{M}} = \mathrm{mean_b}\left[p_b(r)\right]$$
$$\hat{p} = \mathrm{argmax}_r\left[p_{\boldsymbol{M}}\right]$$
$$e_{\hat{p}} = \mathrm{entropy}_r\left[p_{\boldsymbol{M}}\right]$$
$$\hat{\mu}_{p,j} = \mathrm{mean_b}\left[f_{p,j,b}\right]$$
$$\hat{\sigma}_{p,j} = \mathrm{standard\_deviation_b}\left[f_{p,j,b}\right]$$
$$\hat{n}_{p,j} = \mathrm{mean_b}\left[n_{p,j,b}\right]$$
$$\hat{f}_{p,j} = N\left(\hat{\mu}_{p,j}, \hat{\sigma}_{p,j}\right)$$
$$\hat{y}_{p,j} = N\left(\hat{\mu}_{p,j}, \hat{\sigma}_{p,j} + \hat{n}_{p,j}\right)$$

The SAGE algorithms make use of latent functions. One set of latent functions are used to identify the probabilities of each point x belonging to a specific phase region. These probabilities are then multiplied by an additional set of latent functions describing target functional properties, in effect weighting these second set of functions to bound them to target phase regions. Through this combination of latent functions, one can identify regions in X that may contain significant changes in phase and/or functional properties and may be of interest for further experiments. Statistical analysis of multiple samples of latent functions provides a posterior distribution for phase map and piecewise functional properties.

4.2.1. One dimensional challenges. When **X** is one dimensional, phase boundaries may be represented as change points. The set of structure model parameters $\boldsymbol{\theta}_\mathrm{s}$ are simply a set of changepoints in **X**. The changepoints $\boldsymbol{\theta}_\mathrm{s}$ are sampled and then converted to categorical distribution $p(r) = f_\mathrm{s}(\boldsymbol{x})$. Each continuous region of **X**, bounded by either a change point or the edge of the search space, defines a phase region r. For example, for a 2-phase region challenge over $\mathbf{X} = [0, 1]$ with one changepoint at arbitrary value 0.5, phase region 0 would have a probability of 1 at $x =< 0.5$ and a probability of 0 for $x > 0.5$ and *vice versa* for phase region 1. The categorical distribution is then used to compute the likelihood of the observations given the samples.

The presented implementation is developed from that of ref. 25. The functional property in each phase region is represented by an independent radial basis function kernel Gaussian process, with $\boldsymbol{\theta}_\mathrm{p}$ including: $l_\mathrm{r,j}$ kernel length scale, $s_\mathrm{r,j}$ kernel standard deviation (also known as 'scale'), and s_j measured noise standard deviation. For this work, we assume that s_j is the same for property j across all phase regions. For each property, the region-specific functions $f_\mathrm{p,j,r}$ are sampled from $\mathrm{GP}(\boldsymbol{\theta}_\mathrm{p,j,r})$ and then combined using the probabilistic weights $p(r)$ to give the piecewise functions $f_\mathrm{p,j}$. $f_\mathrm{p,j}$ describes the sample mean and n_j the sample noise of the multivariate distribution $N(f_\mathrm{p,j}, n_\mathrm{j})$ used to describe a potential generating random process. Data likelihood is then given by $p(D_\mathrm{p,j}|N(f_\mathrm{p,j}, n_\mathrm{j}))$.

Example implementation:

Model 2: 1-Dimensional SAGE

$$\boldsymbol{\theta}_s \sim \mathrm{Uniform}(\boldsymbol{X})$$
$$\boldsymbol{M}_s = \mathrm{membership}(\boldsymbol{\theta}_s, \boldsymbol{x}_s)$$
$$p_r(r) = p_r(r(\boldsymbol{x}) = l) = \mathrm{Categorical}(\boldsymbol{\theta}_s)$$
$$L_s = \sum_i \sum_k \ln\left[p\left(m\left(D_{s,i}\right)|p_r(r)\right)\right]$$
$$l_{r,j} \sim \mathrm{Uniform}(\mathrm{min\_length\_scale_{r,j}}, \mathrm{max\_length\_scale_{r,j}})$$
$$s_{r,j} \sim \mathrm{Uniform}(\mathrm{min\_standard\_deviation_{r,j}}, \mathrm{max\_standard\_deviation_{r,j}})$$
$$n_j \sim \mathrm{Uniform}(\mathrm{min\_noise\_scale_j}, \mathrm{max\_noise\_scale_j})$$
$$\boldsymbol{\theta}_{\boldsymbol{p}} = \left\{l_{r,j}, s_{r,j}, n_j\right\}$$
$$f_{p,j,r} \sim GP_{j,r}\left(\boldsymbol{\theta}_{\boldsymbol{p}}\right)$$
$$f_{p,j} = \sum_r f_{p,j,r}\, p(r)$$
$$L_p = \sum_j \sum_r \ln\left[p\left(D_{p,j}\left|N\left(f_{p,j}, n_j\right)\right.\right)\right]$$
$$L = L_s + L_p$$

Model 3: N-Dimensional SAGE

$$\boldsymbol{l}_s \sim \mathrm{ND\_Uniform}(\mathrm{min\_length\_scale_s}, \mathrm{max\_length\_scale_s})$$
$$\boldsymbol{s}_s \sim \mathrm{ND\_Uniform}(\mathrm{min\_standard\_deviation_s}, \mathrm{max\_standard\_deviation_s})$$
$$\boldsymbol{\theta}_s = \{\boldsymbol{l}_s, \boldsymbol{s}_s\}$$
$$\boldsymbol{W}_k(\boldsymbol{x}) = \{w_h(\boldsymbol{x})\}_{h=1}^{R} \sim N\left(0, K_{Matern\ 5/2}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{\theta}_s)\right)$$
$$p(r) = p(r(\boldsymbol{x}) = l) = \exp w_{r=l}(\boldsymbol{x}) \,/\, \sum_r \exp w_r(\boldsymbol{x})$$
$$L_s = \sum_i \sum_k \ln\left[p\left(m\left(D_{s,i}\right)|p(r)\right)\right]$$
$$\boldsymbol{l}_{r,j} \sim \mathrm{ND\_Uniform}(\mathrm{min\_length\_scale_{r,j}}, \mathrm{max\_length\_scale_{r,j}})$$
$$\boldsymbol{s}_{r,j} \sim \mathrm{ND\_Uniform}(\mathrm{min\_standard\_deviation_{r,j}}, \mathrm{max\_standard\_deviation_{r,j}})$$
$$n_j \sim \mathrm{Uniform}(\mathrm{min\_noise\_scale_j}, \mathrm{max\_noise\_scale_j})$$
$$b_{j,r} \sim \mathrm{Uniform}(\mathrm{min\_bias_{r,j}}, \mathrm{max\_bias_{r,j}})$$
$$\boldsymbol{\theta}_{\boldsymbol{p}} = \left\{\boldsymbol{l}_{r,j}, \boldsymbol{s}_{r,j}, n_j\right\}$$
$$f_{p,j,r} \sim N\left(b_{j,r}, K_{RBF}\left(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{\theta}_{\boldsymbol{p}}\right)\right)$$
$$f_{p,j} = \sum_r f_{p,j,r}\, p(r)$$
$$L_p = \sum_j \sum_r \ln\left[p\left(D_{p,j}\left|N\left(f_{p,j}, n_j\right)\right.\right)\right]$$
$$L = L_s + L_p$$

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

4.2.2. N-dimensional challenges. Change boundaries and surfaces are not easy to define in higher dimensions, so we instead sample latent functions $w(\boldsymbol{x})$ and then transform these latent functions with categorical distributions for phase region labels over **X**. We define an N-dimensional multivariate normal distribution for the latent functions with associated parameters $\boldsymbol{\theta}_s$. For an SPSPR with R phase regions, we take R latent function samples, again, using the Cholesky decomposition method. The samples $w(\boldsymbol{x})$ are then used to define the columns of matrix $\boldsymbol{M}_s$, *i.e.*, $\boldsymbol{M}_s[:, r] = w_r$. Each entry of $\boldsymbol{M}_s[k, r]$ is taken as the unnormalized event log probability (and converted to logits by the Categorical distribution function) for point $\boldsymbol{x}_k$ belonging to phase region label r. Here, each functional property is described by a N-dimensional GP.

4.2.3. Additional Models. In this work, we compare SAGE to a set of algorithms including off-the-shelf GP regression and classification, modified versions of SAGE, and CAMEO. As the space of machine learning algorithms is vast, we down selected comparisons for the following reasons. There are no other algorithms that perform the same joint task as SAGE, *i.e.*, coregionalized joint segmentation and piecewise regression from disparate classification and regression datasets. However, CAMEO works toward the same multitask goal through non-joint learning, and as a result, is one benchmark algorithm. SAGE is also compared to off-the-shelf GP regression and classification algorithms as it shares the assumptions of each of these algorithms, though SAGE also contains the assumption of coregionalization across data sources. By benchmarking against these algorithms, we demonstrate SAGE's improvements over algorithms with shared set of (reduced) assumptions. Similarly, we benchmark SAGE's benefits of coregionalization against limited versions of SAGE, *e.g.*, where SAGE is provided data from only one of the data sources.

We compare SAGE to off the shelf GP algorithms and modified versions of SAGE. We compare SAGE's phase mapping (PM) performance with a version of SAGE which only takes structure data input. For 1D challenges this is Model 4 'SAGE-1D-PM' and for 2D challenges this is Model 6 'SAGE-ND-PM'. We compare SAGE's functional property (FP) prediction performance with versions that only take in functional property data, *i.e.*, piecewise Gaussian process regression. For 1D challenges this is Model 5 'SAGE-1D-FP' and for 2D challenges Model 7 'SAGE-ND-FP'. For these algorithms that rely on just one input data type, it is expected that for exhaustive data, performance will be high, while for partial data, the joint SAGE model will outperform these models. These additional algorithms are available as part of the SAGE library.

• Model 4, SAGE-1D-PM: this algorithm mirrors SAGE-1D but excludes functional property regression. The algorithm is described the same as Model 2 lines 1–4 and returns L_s.

• Model 5, SAGE-1D-FP: this algorithm mirrors SAGE-1D but excludes the phase mapping loss term. It is thus a 1-dimensional piecewise GP. The algorithm is described the same as Model 2 lines 1–3 and 5–11 and returns L_p.

• Model 6, SAGE-ND-PM: this algorithm mirrors SAGE-ND but excludes the task of functional property regression. The algorithm is described the same as Model 3 lines 1–6 and returns L_s.

• Model 7, SAGE-ND-FP: this algorithm mirrors SAGE-ND but excludes the phase mapping loss term. It is thus an N-dimensional piecewise GP. The mode is described the same as Model 3 lines 1–5 and 7–14 and returns L_p.

• GP-CP; GPR; GPC: the implementations use the radial basis function kernel for regression and the Matern 5/2 kernel and MultiClass likelihood for classification. All use the truncated Newton method for optimization.

• CAMEO – only piecewise regression task. This model follows that of []. A Gaussian random field (GRF) is defined for the material system including both characterized and potentially characterized materials. The GRF is applied to the structure data to segment the material system and that segmentation is then combined with off-the-shelf Gaussian process regression, using different hyperparameters for each phase region.

4.2.4. Performance measures

4.2.4.1. Regression: coefficient of determination, R^2.

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where y_i is a ground truth value of target $y(x_i)$ located at point x_i indexed with i, f_i is the associated algorithm-based predicted value, and $\bar{y}$ is the mean over all y values. Implemented with scikit-learn's r^2_score function.

4.2.4.2. Phase mapping: micro F1-Score.

$$\mathrm{mF_1} = \frac{\sum_r \mathrm{TP}_r}{\sum_r (\mathrm{TP}_r + \mathrm{FP}_r + \mathrm{FN}_r)},$$

where TP_r, FP_r, and FN_r are the count of true positive, false positive, and false negative classification values for each region r. Here positive defines points in X predicted to have label r and negative defines points predicted to not have label r, *i.e.*, one-vs-all classification.

4.2.5. Implementation. The provided SAGE implementations are designed for parallel computation across systems with multiple CPUs, allowing for easy scalability for large datasets. Implementations include: SAGE-1D; SAGE-ND for one structure data stream input and multiple functional property data stream inputs, where the functional property data streams are measured over the same materials (though potentially different materials than the structure data stream); and SAGE-ND-MULTI for multiple structure and functional property data streams where materials investigated can be different for all data streams.

SAGE was run on a laptop (6 core 2.7 GHz, 32 GB memory, NVIDIA‡ Quadro P620) and runs within a few minutes, *e.g.*, less than 2 minutes for the $(Bi,Sm)(Sc,Fe)O_3$ material system

‡ NIST disclaimer: certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

example. All implementations are built to boost performance through parallelization across multiple CPUs by changing the "number of available cores" and "number of chains". This dramatically accelerates computation. For example, using parallelization across a 100 CPU node allows MCMC samples for each CPU to be reduced by an order of 100. The choices of MCMC sample number indicated below were found to provide convergence in the posterior predictions.

Here we provide initial values or uniform prior ranges for the implementation. If a parameter is not mentioned, it is the default initial value or range of the library used.

• GPs

○ All GP implementations are written in gpflow.[45]

○ 1D CP-GP: initial length scale = 0.2; initial change point steepness = 100; noise variance = 0.01; max iterations = 10 000;

○ 2D GPR: initial lengthscales = 1; noise variance = 0.005 and range [0.001, 0.01]; max iterations = 1000;

○ 1D and 2D GPC: max iterations = 1000;

• All MCMC algorithms

○ Number warmup samples = 100; number samples = 1000; target acceptance probability = 0.8; max tree depth = 5; jitter = 1×10^{-6}

○ SAGE-1D, SAGE-1D-PM, SAGE-1D-FP are written in numpyro[46] with parameters: $s_{r,j}$ = [0.01, 2.]; $l_{r,j}$ [0.2,1.]; n_j = [0.001,0.01]; Change point bounds, *i.e.*, θ_s = [0.5, 1.]

○ SAGE-ND, SAGE-ND-PM, SAGE-ND-FP are written in numpyro Jax with parameters.

■ SVI initialization of phase map: number of samples = 100 000; Adam step size = 0.05;

■ MCMC algorithm: s_s = [5.,10.]; l_s = [0.1,2.]; $s_{r,j}$ = [0.1, 2.]; $l_{r,j}$ = [1,2.]; n_j = [0.001,0.1]; $b_{r,j}$ = [−2., 2.]

○ SAGE-ND Multiple data sources:

■ SVI initialization of phase map: number of samples: 10 000; Adam step size = 0.01; s_s = [5.,10.]; l_s = [1.,2.];

■ MCMC algorithm: number of warmup steps: 100; number of samples: 2000; number of chains = 100; s_s = [5.,10.]; l_s = [0.1,2.]; $s_{r,j}$ = [0.1, 2.]; $l_{r,j}$ = [0.1,5.]; n_j = [0.001,0.1]; $b_{r,j}$ = [−2., 2.]

• CAMEO: uses the same parameters as in ref. 16.

4.2.6. Gaussian processes. The Gaussian process assumes that the task of regression or classification can be tied to a function f over domain $\boldsymbol{X}$. For common regression tasks $f(x)$ is the model for the target variable $y(x)$, and for classification f is a function (or a set of functions) used to identify classification boundaries. f is described by a prior probability distribution $f \sim p(\mu, k)$, where $\mu : X \rightarrow \mathbb{R}$ is the prior mean function and $k : X \times X \rightarrow \mathbb{R}$ is a kernel function describing the relationship between $f(x_i)$ and $f(x_j)$. The prior is combined with a likelihood probability distribution $p(y|f)$ used to describe the expected relationship between y and f, typically used to describe the expected noise in y given f. The prior and likelihood are combined using Bayes rule to identify a posterior given the data (x, y). When the prior and likelihood are both multivariate normal distributions, the posterior is also a multivariate normal distribution and an analytical solution for the posterior exists.

Numerous kernel functions exist including the radial basis function which is commonly used for regression and the Matern kernels which are commonly used in classification tasks. These kernel functions have parameters (often called hyperparameters) and the GP algorithm typically uses maximum likelihood to identify the most likely value of these parameters given the data. Additionally, numerous likelihood functions exist for various data challenges based on the expected noise in the data. For the regression case of expected normally distributed noise, the likelihood may be a multivariate normal distribution. Specific likelihood functions exist for the challenges of binary and multi-class classification. For more information on GPs, their theory, implementation, and use, please see the excellent resource.[4]

Data availability

The code, analysis scripts, and datasets supporting this article have been uploaded as part of the ESI.†

Author contributions

The authors contributed to the provided work in the following order, with greatest contribution listed first. Initial concept: AGK. Theory: AGK, BD, AM. Implementation: AGK, BD, AM. Testing: AGK, AM, BD. Writing the paper: AGK, AM, BD.

Conflicts of interest

There are no conflicts to declare.

References

1 D. Kan, C. J. Long, C. Steinmetz, S. E. Lofland and I. Takeuchi, Combinatorial search of structural transitions: Systematic investigation of morphotropic phase boundaries in chemically substituted BiFeO3, *J. Mater. Res.*, 2012, **27**(21), 2691–2704.

2 D. Kan, L. Pálová, V. Anbusathaiah, C. J. Cheng, S. Fujino, V. Nagarajan, *et al.*, Universal Behavior and Electric-Field-Induced Structural Transition in Rare-Earth-Substituted BiFeO3, *Adv. Funct. Mater.*, 2010, **20**(7), 1108–1115.

3 A. G. Kusne, D. Keller, A. Anderson, A. Zaban and I. Takeuchi, High-throughput determination of structural phase diagram and constituent phases using GRENDEL, *Nanotechnology*, 2015, **26**(44), 444002.

4 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Cambridge, Mass, The MIT Press, 2005, p. 272.

5 T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman and R. Tibshirani, *The Elements of Statistical Learning*, Springer, 2009, vol. 2, **https://link.springer.com/content/pdf/10.1007/978-0-387-84858-7.pdf**.

6 A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, 1995, **https://www.taylorfrancis.com/books/mono/10.1201/9780429258411/bayesian-data-analysis-andrew-gelman-donald-rubin-john-carlin-hal-stern**.

7 E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, *et al.*, Pyro: Deep universal

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

probabilistic programming, *J. Mach. Learn. Res.*, 2019, **20**(1), 973–978.

8 H. Ge, K. Xu and Z. Ghahramani, Turing: A Language for Flexible Probabilistic Inference, In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 1682–1690, https://proceedings.mlr.press/v84/ge18b.html.

9 E. Stach, B. DeCost, A. G. Kusne, J. Hattrick-Simpers, K. A. Brown, K. G. Reyes, *et al.*, Autonomous experimentation systems for materials development: A community perspective, *Matter*, 2021, **4**(9), 2702–2726.

10 J. K. Bunn, S. Han, Y. Zhang, Y. Tong, J. Hu and J. R. Hattrick-Simpers, Generalized machine learning technique for automatic phase attribution in time variant high-throughput experimental studies, *J. Mater. Res.*, 2015, **30**(07), 879–889.

11 J. R. Hattrick-Simpers, J. M. Gregoire and A. G. Kusne, Perspective: Composition–structure–property mapping in high-throughput experiments: Turning data into knowledge, *APL Mater.*, 2016, **4**(5), 053211.

12 D. Chen, Y. Bai, W. Zhao, S. Ament, J. Gregoire and C. Gomes, Deep Reasoning Networks for Unsupervised Pattern De-mixing with Constraint Reasoning, in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, 2020, pp. 1500–1509, https://proceedings.mlr.press/v119/chen20a.html.

13 C. P. Gomes, J. Bai, Y. Xue, J. Björck, B. Rappazzo, S. Ament, *et al.*, CRYSTAL: a multi-agent AI system for automated mapping of materials' crystal structures, *MRS Commun.*, 2019, **9**(2), 600–608.

14 S. V. Kalinin, M. P. Oxley, M. Valleti, J. Zhang, R. P. Hermann, H. Zheng, *et al.*, Deep Bayesian local crystallography, *npj Comput. Mater.*, 2021, **7**(1), 181.

15 A. G. Kusne, T. Gao, A. Mehta, L. Ke, M. C. Nguyen, K. M. Ho, *et al.*, On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets, *Sci. Rep.*, 2014, 6367, https://www.nature.com/articles/srep06367.

16 A. G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, *et al.*, On-the-fly closed-loop materials discovery via Bayesian active learning, *Nat. Commun.*, 2020, **11**(1), 5966.

17 R. LeBras, T. Damoulas, J. M. Gregoire, A. Sabharwal, C. P. Gomes and R. B. van Dover, Constraint reasoning and Kernel clustering for pattern decomposition with scaling, in *Principles and Practice of Constraint Programming–CP 2011*, Springer, 2011, pp. 508–522.

18 C. P. Gomes, J. Bai, Y. Xue, J. Björck, B. Rappazzo, S. Ament, *et al.*, CRYSTAL: a multi-agent AI system for automated mapping of materials' crystal structures, *MRS Commun.*, 2019, **9**(2), 600–608.

19 Y. Iwasaki, A. G. Kusne and I. Takeuchi, Comparison of dissimilarity measures for cluster analysis of X-ray diffraction data from combinatorial libraries, *npj Comput. Mater.*, 2017, **3**(1), 1–9.

20 S. K. Suram, Y. Xue, J. Bai, R. Le Bras, B. Rappazzo, R. Bernstein, *et al.*, Automated phase mapping with AgileFD and its application to light absorber discovery in the V–Mn–Nb oxide system, *ACS Comb. Sci.*, 2017, **19**(1), 37–46.

21 V. Stanev, V. V. Vesselinov, A. G. Kusne, G. Antoszewski, I. Takeuchi and B. S. Alexandrov, Unsupervised phase mapping of X-ray diffraction data by nonnegative matrix factorization integrated with custom clustering, *npj Comput. Mater.*, 2018, **4**(1), 1–10.

22 P. M. Maffettone, L. Banko, P. Cui, Y. Lysogorskiy, M. A. Little, D. Olds, *et al.*, Crystallography companion agent for high-throughput materials discovery, *Nat. Comput. Sci.*, 2021, **1**(4), 290–297.

23 *Autonomous Experimental Design and Execution | Handbook on Big Data and Machine Learning in the Physical Sciences*, https://www.worldscientific.com/doi/abs/10.1142/9789811204579_0013.

24 S. Ament, M. Amsler, D. R. Sutherland, M. C. Chang, D. Guevarra, A. B. Connolly, *et al.*, Autonomous materials synthesis via hierarchical active learning of nonequilibrium phase diagrams, *Sci. Adv.*, 2021, eabg4930, https://www.science.org/doi/abs/10.1126/sciadv.abg4930.

25 A. G. Kusne and A. McDannald, Scalable multi-agent lab framework for lab optimization, *Matter*, 2023, **6**(6), 1880–1893.

26 J. Jing, S. Liu, G. Wang, W. Zhang and C. Sun, Recent advances on image edge detection: A comprehensive review, *Neurocomputing*, 2022, **503**, 259–271.

27 C. Park, P. Qiu, J. Carpena-Núñez, R. Rao, M. Susner and B. Maruyama, Sequential adaptive design for jump regression estimation, *IISE Trans.*, 2023, **55**(2), 111–128.

28 S. Aminikhanghahi and D. J. Cook, A survey of methods for time series change point detection, *Knowl. Inf. Syst.*, 2017, **51**(2), 339–367.

29 C. Truong, L. Oudre and N. Vayatis, Selective review of offline change point detection methods, *Signal Process.*, 2020, **167**, 107299.

30 Y. Saatçi, R. D. Turner and C. E. Rasmussen, Gaussian process change point models, In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 927–934.

31 J. Lloyd, D. Duvenaud, R. Grosse, J. Tenenbaum and Z. Ghahramani, Automatic construction and natural-language description of nonparametric regression models, In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

32 A. G. Journel and C. J. Huijbregts, *Mining geostatistics*, Academic press, London, 1978, vol. 600.

33 M. A. Alvarez, L. Rosasco and N. D. Lawrence, *Kernels for vector-valued functions: A review*, Foundations and Trends® in Machine Learning, 2012, vol. 4(3), pp. 195–266.

34 E. V. Bonilla, K. Chai and C. Williams, Multi-task Gaussian process prediction, *Advances in neural information processing systems*, ed. J. Platt, D. Koller, Y. Singer and S. Roweis, Curran Associates, Inc., 2007, vol. 20.

35 C. A. Micchelli and M. Pontil, On learning vector-valued functions, *Neural computation*, 2005, **17**(1), 177–204.

© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 27 September 2024. Downloaded on 4/2/2026 5:28:24 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

36 C. Carmeli, E. De Vito and A. Toigo, Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem, *Anal. Appl.*, 2006, **4**(04), 377–408.
37 J. J. Giraldo and M. A. Álvarez, A Fully Natural Gradient Scheme for Improving Inference of the Heterogeneous Multioutput Gaussian Process Model, *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, **33**(11), 6429–6442.
38 N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, *et al. Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence*, USDOE Office of Science (SC), Washington, D.C. (United States), 2019, **https://www.osti.gov/biblio/1478744/**.
39 O. Isayev, D. Fourches, E. N. Muratov, C. Oses, K. Rasch, A. Tropsha, *et al.*, Materials Cartography: Representing and Mining Materials Space Using Structural and Electronic Fingerprints, *Chem. Mater.*, 2015, **27**(3), 735–743.
40 B. F. Grosso, N. A. Spaldin and A. M. Tehrani, Physics-Guided Descriptors for Prediction of Structural Polymorphs, *J. Phys. Chem. Lett.*, 2022, **13**(31), 7342–7349.
41 M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *science*, 2009, **324**(5923), 81–85.
42 Z. Wang, W. Xing, R. Kirby and S. Zhe, Physics informed deep kernel learning, In: *International Conference on Artificial Intelligence and Statistics*, PMLR, 2022, pp. 1206–1218, **https://proceedings.mlr.press/v151/wang22a.html**.
43 A. McDannald, M. Frontzek, A. T. Savici, M. Doucet, E. E. Rodriguez, K. Meuse, *et al.*, On-the-fly autonomous control of neutron diffraction via physics-informed Bayesian active learning, *Applied Physics Reviews*, 2022, **9**(2), 021408.
44 E. Wit, E. van den Heuvel and J. W. Romeijn, 'All models are wrong…': an introduction to model uncertainty, *Statistica Neerlandica*, 2012, **66**(3), 217–236.
45 A. G. de G. Matthews, M. Van Der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, *et al.*, GPflow: A Gaussian Process Library using TensorFlow, *J. Mach. Learn. Res.*, 2017, **18**(40), 1–6.
46 *NumPyro documentation — NumPyro documentation*, **https://num.pyro.ai/en/latest/index.html#**.

© 2024 The Author(s). Published by the Royal Society of Chemistry