# Shared-GP: Learning Interpretable Shared Hidden Structure Across Data Spaces for Design Space Analysis and Exploration

**Wei Xing**
Scientific Computing and Imaging Institute, University of Utah
Email: wxing@sci.utah.edu

**Shireen Y. Elhabian**
Scientific Computing and Imaging Institute, University of Utah
Email: shireen@sci.utah.edu

**Vahid Keshavarzzadeh**
Scientific Computing and Imaging Institute, University of Utah
Email: vkeshava@sci.utah.edu

**Robert M. Kirby**[*]
Scientific Computing and Imaging Institute, University of Utah
Email: kirby@cs.utah.edu

## ABSTRACT

An industrial design process is often highly iterative. With unclear relationships between QoI trade-offs and the design solution, the definition of the cost function usually undergoes several modifications that mandate a continued interaction between the designer and the client to encode all design and mission requirements into an optimization-friendly mathematical formulation. Such an iterative process is time-consuming and computationally expensive. An efficient way to accelerate this process is to derive data-driven mappings between the design/mission and QoI spaces to provide visual insights into the interactions among different QoIs as related to their corresponding simulation parameters. In this paper, we propose Shared-GP, a generative model for the design process that is based on a Gaussian process latent variable model. Shared-GP learns correlations within and across multiple, but implicitly correlated, data spaces considered in the design process (i.e., the simulation parameter space, the design space, and the QoI spaces) to provide data-driven mappings across these data spaces via efficient inference. Shared-GP also provides a structured low-dimensional representation shared among the data spaces (some of which are of very high dimension) that the designer can use to efficiently explore the design space without the need for costly simulations.

## 1 Introduction

Industrial design is often a highly iterative process by which designers and clients continuously interact to translate design and mission requirements into a mathematical language for numerical optimization and simulation. An interesting design optimization process entails multiple, usually conflicting, performance metrics (i.e., *quantities of interest* – QoIs) [1,2]. Hence, an optimal and feasible design typically results from the designer and the client coming to a sufficient understanding of the possible designs from which to choose (i.e., *design space*) to balance the inherent trade-offs among different QoIs. Such an understanding requires defining the criteria that determine the best design in an optimization process (i.e., *cost function*), the mission-specific design requirements (i.e., *constraints*), and the description or parameterization of different designs (i.e., *simulation parameters*).

[*] Corresponding author

### 1.1 Background

A design optimization process can be time-consuming and computationally expensive because (1) the computation of the QoIs is typically performed via computationally expensive numerical solvers used for simulations. (2) The cost function definition undergoes several modifications throughout the design process due to the unclear relationships between QoIs trade-offs and the feasible design solutions. These modifications can be as simple as adjusting the weights used when expressing different additive trade-offs, but they can also involve a change in the choice of metrics of evaluation, the addition or removal of different trade-off terms, etc. (3) An optimal solution could be scientifically infeasible, which mandates revisiting the problem definition and possibly adding or modifying the design requirements/constraints. (4) The design space grows exponentially with the increase of parameters used to describe a design. (5) The response surface of the cost function could be highly nonlinear and nonconvex, which introduces additional optimization challenges (e.g., sensitivity to initialization). To accelerate the design process and reach a better design, a designer needs analysis and visualization tools that correlate simulation parameters with designs and QoIs to explore more broadly across multiple but related *data spaces*.

### 1.2 Related work

Representational learning has been an effective tool for design space exploration by embedding high-dimensional designs into a semantic compact subspace [3, 4]. With low-dimensional representations, visual parameter space analysis [5] can provide an interactive means to navigate multiple data spaces via visual analysis tools such as Tuner [6]. The subspace is normally found using data-driven methods such as multidimensional scaling [7], kernel principal component analysis [8], and deep learning based models [4]. To form an informative subspace that captures the inherent complexity of designs and provides a structural compact semantic representation, *design manifold* [3] has recently been proposed. However, most of the existing techniques for design representation focus on a single design space. The multiple design/QoI space problem remains a challenge.

Another core component for design analysis and visualization is a data-driven *surrogate model* [9–11]. This model provides real-time predictions and inference across different data spaces, i.e., predicting the QoIs and simulation parameters given some designs or predicting the designs given some new simulation parameters. These surrogate models are essentially statistical regression models that fit the input-output paired data generated by the simulation process [12, 13]. As a powerful universal approximator, deep learning has had many successful applications in the context of design optimization [14–16]. In particular, [15] shows that the design constraints can be directly incorporated into a deep net to enable *theory-driven* emulations. Although deep learning models are powerful, they normally lack the ability to quantify the uncertainty associated with the simulators [17]; they are prone to overfitting when dealing with small datasets, which is indeed the case in this paper because a simulation is often expensive to execute.

For surrogate modeling, GPs have gained more popularity than deep learning for the following reasons: (1) GPs can directly capture model uncertainty and make predictions/decisions within a Bayesian paradigm to avoid overfitting [13, 18]. (2) GPs enable applying prior knowledge of the simulator by choosing a proper kernel function [19]. (3) GPs, as nonparametric models, have only a few free parameters and thus do not need a large number of training samples [20], which are normally not available in the context of surrogate models. These advantages in general, and the probabilistic rigor in particular, make GPs preferable surrogate models for design problems, especially when the uncertainty plays a crucial role [21, 22]. The direct implementation of GPs to provide inference across multiple data spaces raises the challenge that $H(H+1)-$pairwise/bidirectional standard GPs are needed to capture all admissible pairwise mappings between any two data spaces (where $H$ is the number of output spaces). The number of models grows quadratically with the number of simulator output spaces and thereby presents a computational overhead even with a few output spaces.

In computer vision, the multiple data space problem is also known as the multiview problem, where multiple spaces usually contain different images (e.g., images taken from different angles) for an object [23–25]. Most existing studies aim to convert the multiple views to a common low-dimensional representation, which is then used as an input feature to accomplish downstream tasks, e.g., classifications [23], object recognitions [24], and pose estimations [25].

### 1.3 Contributions of this work

Similar to the multiview problem, it is possible to consider that multiple design spaces also share some underlying common representations. We thus propose *Shared-GP*, a *generative model* for the design problem, where the simulator inputs, designs, and QoIs are jointly embedded into a structured latent space. This latent space provides not only a joint representation for all our designs/QoIs but also an efficient way to conduct quick inference across data spaces. Our work is consistent with the shared GP latent variable model (shared GPLVM) [26], which solves a multiview facial expression recognition problem using shared latent representations and GPs. Shared-GP, like most existing multiview models, solves a multiple space design problem through a shared latent space. However, our model differs in two ways: (1) Unlike most existing methods that assume no priors or a naive Gaussian prior for the latent space, we equip the latent space with a Dirichlet process prior such that the inherent multimodal nature of the design data can be fully captured. This prior not only improves model accuracy but also leads to a structured latent space, which can help the designers to understand the design, e.g., by identifying different types of designs and when the type changes, as shown in Fig. 15. (2) Rather than using the representations to serve the downstream applications, we focus on accurate inference across data spaces to fulfill the need of understanding the interaction of designs and QoIs in a design process.

The contributions of this work are fourfold. *First*, Shared-GP is the first attempt to introduce the classic multiview techniques to the context of multiple design space exploration. Similar to using representations to understand a single design space [3, 27], the shared representation allows the designer to explore and analyze the relationships among different QoIs, design choices, and simulation parameters by exploring the latent space. *Second*, we further place a Dirichlet process prior in the latent space to harness the inherent structure of the data to produce a structured latent space (e.g., Fig. 6c) and to further improve model accuracy in terms of inference across data spaces. *Third*, we show that a conditional independent GP is sufficient for the emulations of high-dimensional simulation data. *Fourth*, we develop a variational expectation-maximization (EM) method to allow accurate and quick inferences across different data spaces, i.e., predicting the design/QoIs given known values in other data spaces or predicting the simulation inputs given specific QoIs or designs. Note that when a reverse mapping does not exist (i.e., the underlying function is non-bijective), the reverse inference might be misleading because the solution does not exist or is not unique.

To assess the performance of Shared-GP, we first compared it with naive GP implementation, i.e., Pairwise-GPs on a classic beam topology optimization problem. The results indicate that both methods succeed in the prediction task, i.e., predicting the designs given unseen simulation parameters. However, in the inference task, where the models are used to predict the simulation parameters given an unseen design, Shared-GP outperforms Pairwise-GPs in terms of accuracy and consistency. Based on the same dataset, we then investigated how Shared-GP can be used to reduce the dimension of a design problem. Particularly, for the beam topology optimization dataset, we effectively reduced the latent dimension to two because one of three simulation parameters has a weak inference on the design. We also showed how the derived structured latent space can help designers explore the design space. In the last experiment, we evaluated Shared-GP on a more challenging topology optimization problem where the simulation inputs are very high-dimensional, i.e., close to one million, and compared it with the state-of-the-art deep learning models, e.g., deep fully connected network, convolution neural network, and other GP models combined with representational techniques. Results of this experiment indicate that Shared-GP outperforms other models by a large margin in most cases and DPP indeed improves the model accuracy significantly.

## 2 Problem definition

Consider a design optimization process with a *simulator* providing $H-$outputs that include a design solution (in the design space) and associated performance metrics (in the QoI spaces), which can be computed directly from the design or other QoI(s), corresponding to a vector of simulation parameters $\mathbf{x} \in \mathbb{R}^D$ (in the simulation parameter space). It is assumed that the design problem is well posed, i.e., the simulator always finds a unique design solution for the range of values of $\mathbf{x}$ considered. For notation simplicity, we refer to the simulator $H-$outputs as $\mathbf{y}^{(h)} \in \mathbb{R}^{O_h}$, $h = 1, \ldots, H$, where each $\mathbf{y}^{(h)}$ could be a field, a vector, or a scalar value that encodes a simulator output corresponding to the simulation parameters $\mathbf{x}$. For instance, $\mathbf{y}^{(1)}$ could be the density distribution field of a topological structure, $\mathbf{y}^{(2)}$ could be the computed stress field for $\mathbf{y}^{(1)}$, $\mathbf{y}^{(3)}$ could be the maximum load of $\mathbf{y}^{(1)}$, and $\mathbf{y}^{(4)}$ could be the computed compliance based on $\mathbf{y}^{(2)}$. With $N$ conducted simulation experiments, we have a dataset containing $N-$input/output tuples, $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1^{(1)}, \ldots, \mathbf{y}_1^{(H)}), \ldots, (\mathbf{x}_N, \mathbf{y}_N^{(1)}, \ldots, \mathbf{y}_N^{(H)})\}$, where we use the subscript to denote the index of an experiment. In this paper, we are interested in three main *tasks*:

(1) **Shared Parameterization**: Find a low-dimensional representation $\mathbf{z}_n \in \mathbb{R}^L$, $n = 1, \ldots, N$ of all simulator outputs (and possibly the simulation parameters when they are correlated.)

(2) **Prediction**: Predict all simulator outputs $\mathbf{y}_*^{(h)}$, $h = 1, \ldots, H$ given new (unseen) simulation parameters $\mathbf{x}_*$.

(3) **Inference**: Given a subset of the simulator output (e.g., $\mathbf{y}_*^{(1)}$), infer the remaining outputs (e.g., $\mathbf{y}_*^{(h)}, h = 2, \ldots, H$) and the simulation parameters $\mathbf{x}_*$.

These tasks establish the building blocks for exploring multiple correlated data spaces considered in the design process. In particular, accomplishing the first task provides a uniform representation for multiple correlated data spaces such that the classic design space exploration method can be readily implemented. Fulfilling the second and third tasks allows the designer to quickly explore the interactions among QoIs, design, and simulator parameters.

## 3 Model formulation

We start by presenting the basic Gaussian process (GP) model for scalar-valued simulator outputs. We then discuss the high-dimensional output problem, existing multivariate GP models, and why conditional independent GP is sufficient for our problem. Lastly, we introduce the GP latent variable model and propose Shared-GP.

### 3.1 GP surrogate for scalar-valued outputs

Let the $h-$th simulator output $y^{(h)}$ be a scalar that is a functional result of some known (e.g., simulation parameters) or unknown (i.e., latent/hidden variables) $\mathbf{z}$, which can be discrete ($\mathbf{z} \in \mathbb{N}^L$) or continuous ($\mathbf{z} \in \mathbb{R}^L$). The mapping from $\mathbf{z}$ to $y^{(h)}$ can be assumed to be an injective function $\phi(\mathbf{z})$ [28, 29]. A Gaussian process places a Gaussian prior over $\phi(\mathbf{z})$

such that any number of observations have a joint Gaussian distribution. Hence, a GP is fully specified by a mean function $\mathbb{E}[\phi(\mathbf{z})] = m(\mathbf{z})$ and a covariance function $\mathbb{E}[(\phi(\mathbf{z}) - m(\mathbf{z}))^T(\phi(\mathbf{z}') - m(\mathbf{z}'))] = k(\mathbf{z}, \mathbf{z}')$. The mean functions $m(\cdot)$ are usually set to be linear (in the input parameters) or constant values. Constant mean functions have been found to be adequate in most applications [28]. In particular, a zero constant value is frequently assumed after centering the data [29]. When evaluated, the covariance function $k(\cdot,\cdot)$ must generate a symmetric, positive semidefinite covariance matrix, and should be designed to faithfully represent the true correlations, a challenging problem that warrants thorough research on its own [29–32]. In this work, all GP models assume the most commonly used automatic relevance determination (ARD) kernel plus a Gaussian noise term for demonstration and fair comparison purposes. The ARD kernel reads as $k^{(h)}(\mathbf{z}, \mathbf{z}') = \theta^{(h)}_{L+1} \exp(-(\mathbf{z} - \mathbf{z}')^T diag(\theta^{(h)}_1, \dots, \theta^{(h)}_L)(\mathbf{z} - \mathbf{z}'))$, which expresses smoothness of the GP as a function of the inputs $\mathbf{z}$. The *hyperparameters* $\boldsymbol{\theta}^{(h)} = [\theta^{(h)}_1, \dots, \theta^{(h)}_{L+1}]^T$ introduce different degrees of decay in each component of the input and the amplitude. Since any subset of observations forms a joint Gaussian distribution, given $N-$observations and a new input $\mathbf{z}_*$, the posterior mean and variance can be derived analytically as

$$\mathbb{E}[y^{(h)}_*] = \mathbf{k}^{(h)^T} \boldsymbol{\Sigma}^{(h)^{-1}} \mathbf{Y}^{(h)}, \quad \mathbb{V}\mathrm{ar}[y^{(h)}_*] = k^{(h)}(\mathbf{z}_*, \mathbf{z}_*) - \mathbf{k}^{(h)^T} \boldsymbol{\Sigma}^{(h)^{-1}} \mathbf{k}^{(h)}, \tag{1}$$

where $\mathbf{k}^{(h)} = [k^{(h)}(\mathbf{z}_1, \mathbf{z}_*), \dots k^{(h)}(\mathbf{z}_N, \mathbf{z}_*)]^T$ is the covariance between $\mathbf{z}_*$ and all observations, $\boldsymbol{\Sigma}^{(h)}_{ij} = k^{(1)}(\mathbf{z}_i, \mathbf{z}_j) + \theta_{L+2}\varepsilon(\mathbf{z}_i, \mathbf{z}_j)$ is the covariance matrix plus a Gaussian noise term with variance of $\theta_{L+2}$, and $\mathbf{Y}^{(h)} = [y^{(h)}_1; \cdots; y^{(h)}_N] \in \mathbb{R}^{N \times 1}$ is the collection of all observed simulator outputs. Although many design applications generate noise-free data, we still include the noise term in our formulation as it also serves as a regularization parameter for model numerical stability when inverting the covariance matrix. In a Bayesian inference approach, predictions at a new input $\mathbf{z}_*$ are made by marginalizing (i.e., integrating) over the unknown hyperparameters $\boldsymbol{\theta}^{(h)}$ in the joint distribution of $\boldsymbol{\theta}^{(h)}$. The integral is analytically intractable but can be approximated using Monte Carlo integration, e.g., importance sampling or Markov Chain Monte Carlo [33] to sample from the posterior of hyperparameters $p(\boldsymbol{\theta}^{(h)}|\mathbf{Y}^{(h)})$. To derive a practical model, we adopt the commonly used maximum likelihood estimate (MLE) approach [29] that maximizes the model likelihood, i.e., $argmax_{\boldsymbol{\theta}^{(h)}} \mathcal{L}^{(h)}(\boldsymbol{\theta}^{(h)})$, where $\mathcal{L}^{(h)}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(y^{(h)}_n | \boldsymbol{\theta}^{(h)})$ is the log-likelihood of all observations w.r.t $\boldsymbol{\theta}^{(h)}$. This likelihood function has an analytical form:

$$\mathcal{L}^{(h)}(\boldsymbol{\theta}^{(h)}) = -\frac{1}{2} \ln |\boldsymbol{\Sigma}^{(h)}| - \frac{1}{2} \mathbf{Y}^{(h)^T} \boldsymbol{\Sigma}^{(h)^{-1}} \mathbf{Y}^{(h)} - \frac{N}{2} \ln(2\pi). \tag{2}$$

The computational complexity of the likelihood is $O(N^3)$ and $O(N^2)$ for storage; thus, it is much faster than the sampling-based method.

### 3.2 GP surrogate for high-dimensional output

In practice, many simulator outputs are high dimensional. For instance, for structural topology optimization (an emerging component of the engineering design process), we not only are interested in the structure performance concluding metric (e.g., compliance, maximum load, and mass usage) but also need to know the density distribution field and possibly the stress field of the structure. A naive approach is to assign a label to each output dimension and use such a label as an extra model input. However, this method becomes impractical with high dimensions [28] — a typical situation in simulation data. Let $\mathbf{y}^{(h)}$ denote a vector/field-valued simulator output. $\mathbf{y}^{(h)}$ are functional results of some low-dimensional input parameters, which suggests that we could parameterize the high-dimensional outputs using low-dimensional latent variables. For example, Higdon [34] considered the outputs to be a linear combination of principal component analysis (PCA) bases with latent variables treated as realizations sampled from independent univariate GPs. However, this method is applicable only to problems whose data lie in the vicinity of a linear subspace in the ambient space due to its linear assumption. An ad hoc dimensionality reduction was also employed by Bayarri et al. [35], who used a wavelet decomposition and a thresholding procedure to restrict the dimensionality of the latent space.

To relax the linearity assumption, nonlinear dimension reduction methods, e.g., kernel PCA [36], Isomap [37], diffusion map [38], and local tangent space alignment (LTSA) [39], have been applied to improve efficiency. Such methods have shown improved accuracy and efficiency compared to the linear methods, but they lack the tractability for statistical inference. Under the GP framework, a classical solution framework is the linear model of coregionalization (LMC) [40, 41]. Many recent improvements [42–45] are essentially variations of the LMC. The LMC normally assumes a multivariate separable correlation structured GP, which allows the implementation of the outer-product trick [30, 42] to improve computational efficiency. However, this method is limited by the separable correlation assumptions, and the model is also difficult to train as the number of model parameters grows exponentially with the dimensionality of the output. Recent research in machine learning [46] shows a parameterization trick that reduces the model complexity. However, when the observations are noise free or the noise level is low, which is likely to be our case for high-fidelity simulation data, learning the output correlations contributes very little to the model accuracy. This phenomenon is known as autokrigeability [47], which cancels out the learned output correlations in the model predictions, regardless of how complicated they are. We provide a mathematical proof in Appendix A and an empirical proof in Appendix B. Hence, in this paper, we assume a conditional independent assumption as is used in [48] to model the data. Basically, all output dimensions are treated independently and are correlated only through the shared kernel parameters.

Let $\mathbf{Y}_j^{(h)} = [y_{1,j}^{(h)}; \dots ; y_{N,j}^{(h)}] \in \mathbb{R}^{N\times 1}$ denote the data collection of the $j-$th data dimension of the $h-$th simulator output, with $y_{n,j}^{(h)}$ indicating the $n-$th observation at the $j-$th dimension. Using the same formulation of Eq. (2), the log-likelihood of the $j-$th dimension can be written as

$$\mathcal{L}_j^{(h)}(\boldsymbol{\theta}^{(h)}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}^{(h)}| - \frac{1}{2}{\mathbf{Y}_j^{(h)}}^T {\boldsymbol{\Sigma}^{(h)}}^{-1}\mathbf{Y}_j^{(h)} - \frac{N}{2}\ln(2\pi). \tag{3}$$

The joint log-likelihood is the sum over $O_h$ independent log-likelihood with the same covariance matrix:

$$\mathcal{L}^{(h)}(\boldsymbol{\theta}) = -\frac{O_h}{2}\left(\ln|\boldsymbol{\Sigma}^{(h)}| + \mathrm{tr}({\mathbf{Y}^{(h)}}^T {\boldsymbol{\Sigma}^{(h)}}^{-1}\mathbf{Y}^{(h)}) + N\ln(2\pi)\right), \tag{4}$$

where $\mathrm{tr}(\cdot)$ denotes the trace. The computational complexity of this model remains the same as the scalar-value output GP in Eq. (2)

### 3.3 Shared-GP model for multiple simulator outputs

The definition of the likelihood functions of Eqs. (2) and (4) represents model fitness measures for scalar- and vector/field-valued simulator outputs. Inspired by recent works [3, 27, 36–38], we want to discover a low-dimensional latent space that simultaneously parameterizes multiple outputs (i.e., data spaces). Hence, we assume an unknown, but shared, latent variable $\mathbf{z}$ as the input of each GP. We then optimize the joint log-likelihood w.r.t. $\mathbf{z}$ and each independent set of hyperparameters $\boldsymbol{\theta}^{(h)}$. In this joint framework, the simulation parameters $\mathbf{x}$ are also considered as a particular kind of QoI. In particular, we define the $0-$th QoI $\mathbf{y}^{(0)} \equiv \mathbf{x}$ to avoid clutter in the notation. The joint latent GP model of $H+1$ outputs (now including the simulation parameters) is

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{h=0}^{H} \mathcal{L}^{(h)}(\boldsymbol{\theta}^{(h)}), \quad \mathcal{L}^{(h)}(\boldsymbol{\theta}^{(h)}) = -\frac{O_h}{2}\left(\ln|\boldsymbol{\Sigma}^{(h)}| + \mathrm{tr}({\mathbf{Y}^{(h)}}^T {\boldsymbol{\Sigma}^{(h)}}^{-1}\mathbf{Y}^{(h)}) + N\ln(2\pi)\right), \tag{5}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is the joint log-likelihood, $\boldsymbol{\Theta}^{(h)} = diag(\theta_1^{(h)}, \dots, \theta_L^{(h)})$ is a diagonal matrix, $\boldsymbol{\theta} = [\boldsymbol{\theta}^{(0)}; \cdots ; \boldsymbol{\theta}^{(H)}]$ indicates all the hyperparameter of $H+1$ GP models, and $\mathbf{z}_n$ is the latent variables of the $n-$th simulation experiment. By maximizing the joint log likelihood w.r.t. all hyperparameters $\boldsymbol{\theta}$ and $\mathbf{z}_n, n = 1, \dots, N$, we have $H+1$ independent GPs that are connected through a shared low-dimensional variable $\mathbf{z}$. The dimension of $\mathbf{z}$ should reflect the intrinsic dimensionality to characterize the surrogate model; it is normally larger than the dimension of the simulation parameters. In cases where there is a lack of sensitivity to certain simulation parameters (e.g., some parameters that have negligible influence on the QoIs and designs), this latent dimension can be smaller. A low-dimensional $\mathbf{z}$ helps us build a visualization tool for design space exploration. It should be noted that we demonstrate a shared-GP with a Gaussian likelihood function for all output spaces. For simulator outputs with mixed data types, e.g., binary or categorical data, we need to use an appropriate likelihood function for each output space. The model estimation relies on $H+1$ independent GPs, and thus the model complexity is $O((H+1)N^3)$ in time and $O((H+1)N^2)$ in space, compared to $O(H(H+1)N^3)$ and $O(H(H+1)N^2)$ for traditional pairwise GPs.

### 3.4 Dirichlet process for a structured latent space

For real applications, it would be helpful to place a prior knowledge on these variables to lend some structure to the latent space, while accounting for uncertainties associated with the estimated low-dimensional representation. For instance, to encourage sparcity, we can place a Laplacian prior on $\mathbf{z}$, i.e., $p(\mathbf{z}_n) \sim \mathrm{Laplace}(\lambda) \propto \exp(-\lambda||\mathbf{z}_n||_1)$. In practice, the design data is usually multimodal, which requires a more sophisticated prior. Note that we also have no knowledge of the number of modes. To address these issues, we assign a Dirichlet process prior (DPP) over the latent variables similar to [49]. An infinite collection of random variables $\mathbf{v} = \{v_1, v_2, \cdots\}$ and an infinite set of cluster centers $\boldsymbol{\eta} = \{\eta_1, \eta_2, \cdots\}$ are first sampled via

$$p(\mathbf{v}|\alpha) = \prod_{m=1}^{\infty} \mathrm{Beta}(v_m|1,\alpha), \quad p(\boldsymbol{\eta}) = \prod_{m=1}^{\infty} \mathcal{N}(\boldsymbol{\eta}_m|\mathbf{0},\mathbf{I}), \tag{6}$$

where $\alpha > 0$ decides the concentration of the Dirichlet process. The latent variable $\mathbf{z}$ is generated as follows:

$$p(\mathbf{z}_n, w_n|\mathbf{v},\boldsymbol{\eta}) = p(\mathbf{z}_n|w_n,\boldsymbol{\eta})p(w_n|\mathbf{v}) = \prod_{m=1}^{\infty} \mathcal{N}(\mathbf{z}_n|\boldsymbol{\eta}_{w_n}, \lambda\mathbf{I})\pi_m(\mathbf{v})^{\delta(w_n=m)}, \tag{7}$$

where $\pi_m(\mathbf{v})^{\delta(w_n=m)} = v_m \prod_{m'=1}^{m-1}(1-v_{m'})$, $\lambda$ is the variance tolerance of each cluster center, $\mathbf{w} = [w_1, w_2, \cdots]^T$ denotes the assignment variable for each latent variable, and $\delta(\cdot)$ is the indicator function. The latent variables of the multiple spaces are now automatically clustered to reveal the underlying clusters, which can help designers to understand the types of designs and when the type changes. Furthermore, incorporating the inherent structure of the design data improves the model accuracy when conducting prediction and inference tasks.

## 4 Variational model learning

Incorporating the Dirichlet process prior (DPP) into our model, the marginal likelihood function can be written as

$$p(\mathbf{v},\mathbf{w},\boldsymbol{\eta},\mathbf{Z},\mathbf{Y}) = p(\mathbf{Y}|\mathbf{Z})p(\mathbf{v}|\alpha)p(\boldsymbol{\eta}) \times \textstyle\prod_{n=1}^{N} p(\mathbf{z}_n|w_n,\boldsymbol{\eta})p(w_n|\mathbf{v}), \tag{8}$$

where $p(\mathbf{Y}|\mathbf{Z})$ is the model likelihood function in Eq. (5). With DPP, the marginal likelihood does not admit to a closed-form solution. Here, we propose a variational Bayesian expectation maximization (VB-EM) algorithm for model estimation. In the E-step, we approximate the posterior using a fully factorized distribution, i.e., $p(\mathbf{v},\mathbf{w},\boldsymbol{\eta}|\mathbf{Y},\mathbf{Z}) \approx q(\mathbf{v})q(\mathbf{w})q(\boldsymbol{\eta})$. Variational inference minimizes the Kullback-Leibler (KL) divergence between the approximate and exact posterior.

$$\min_{q} \mathrm{KL}\left[q(\mathbf{v})q(\mathbf{w})q(\boldsymbol{\eta})||p(\mathbf{v},\mathbf{w},\boldsymbol{\eta}|\mathbf{Y},\mathbf{Z})\right]. \tag{9}$$

The approximation is achieved using conditional minimization, i.e., optimizing one approximate distribution at a time while keeping the others fixed. The process is iterated until convergence. The calculation of the parameters for these variational posteriors is similar to that in [50], and thus we leave the details to Appendix C. Based on the variational approximation of the posterior in the E-step, we then maximize the expected log-likelihood over $\mathbf{Z}$ and kernel hyperparameters $\boldsymbol{\theta}$ in the M-step,

$$\arg\max_{\mathbf{Z},\boldsymbol{\theta}} \mathbb{E}_q\left[\log p(\mathbf{v},\mathbf{w},\boldsymbol{\eta},\mathbf{Z},\mathbf{Y})\right] = \arg\max_{\mathbf{Z},\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta}) - \sum_{n=1}^{N} \frac{\lambda}{2} \mathbb{E}_q[||\mathbf{z}_n - \sum_{m=1}^{T} \boldsymbol{\eta}_m \delta(z_n = m)||^2] \right\}. \tag{10}$$

## 5 Inference across data spaces

The proposed model provides a data-driven mapping across data spaces where all simulator outputs and the simulation parameters can be inferred given a latent variable $\mathbf{z}_*$. The model also enables predicting the corresponding latent variable $\mathbf{z}_*$ given some simulator output $\mathbf{y}_*^{(k)}$ and consequently predicting the rest of the simulator outputs and simulation parameters. A fully Bayesian approach requires integrating out the latent variable $\mathbf{z}_*$ when making such predictions. However, the integral is intractable and computationally expensive. To improve efficiency, we can use an inverse GP approach similar to [51]. Specifically, for each inference, we first predict the corresponding latent variable $\mathbf{z}_*$ corresponding to the simulator output $\mathbf{y}_*^{(k)}$ using a GP trained on $\mathbf{Z}$ and $\mathbf{Y}^{(k)}$. We then predict other outputs based on existing $H-$GPs and the new latent variable $\mathbf{z}_*$. This method has been shown to be accurate and efficient with a few simulator outputs [51]. Nonetheless, it does not provide a cluster label for the new latent variable that would help in discovering the structure of the latent space. Here, Shared-GP with its DPP estimation can be used to perform the inference for $\mathbf{z}_*$ using the same VB-EM approach but fixing the other posterior. Specifically, in the E-step, we minimize Eq. (11) using the same process as in the model estimation,

$$\mathrm{KL}\left[q(w_*)q(v_*)q(\eta_*)||p\left(w_*,v_*,\eta_*,|\mathbf{v},\mathbf{w},\boldsymbol{\eta},\mathbf{Z},\mathbf{z}_*^{(0)},\mathbf{Y},\mathbf{y}_*^{(h)}\right)\right]. \tag{11}$$

In the M-step, we optimize for $\mathbf{z}_*$,

$$\arg\max_{\mathbf{z}_*} \mathbb{E}_{q_*}\left[\log p\left(w_*,v_*,\eta_*,\mathbf{z}_*,\mathbf{y}_*^{(h)}|\mathbf{v},\mathbf{w},\boldsymbol{\eta},\mathbf{Z},\mathbf{Y}\right)\right]. \tag{12}$$

We cannot always reasonably assume that the underlying functional mappings from simulation parameters to simulator outputs (QoIs in particular) are bijective and hence the inverse exists [28, 38]. In these cases, directly performing predictions of the simulation parameters from a QoI, e.g., compliance, can be misleading. Constructing predictive models with computable inverse maps is beyond the scope of this paper and is left for a future work.

## 6 Experiments

Here, we assess the performance of Shared-GP, compared to Pairwise-GPs, using structural topology optimization (STO) as a use case. STO optimizes the material distribution within a given simulation parameter grid subject to problem-specific parameters and constraints. STO typically entails a large number of simulation parameters and a significant computational cost. Our approach can efficiently traverse the simulation parameter space and provide an interactive tool to identify the optimal simulation parameters for arbitrary, never-tried, simulation parameters. With its image-based representation, we consider STO as an effective showcase example for Shared-GP.

### 6.1 Experiment design for surrogate models

All data-driven surrogate models require good coverage of the simulation parameter space via the $N$ experiments. It is thus important to conduct a well-designed experiment to provide insight into the problem domain. Without prior knowledge, space-filling methods, e.g., Sobol sequence [52], Latin hypercube sampling [53], low-discrepancy sequence [54], and good lattice points [55], are typically used to provide as much information about the response surface as possible. In our experiments, we used the commonly used Latin hypercube sampling to provide the data collections.

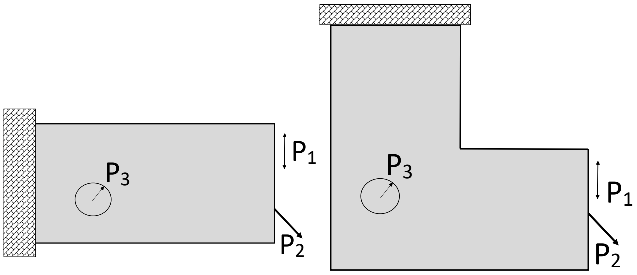


Fig. 1: Geometry, boundary conditions, and simulation parameters for (left) cantilever beam and (right) L-Bracket.

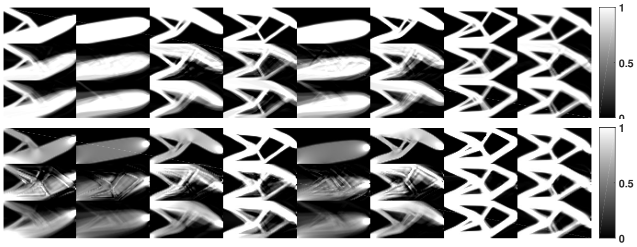
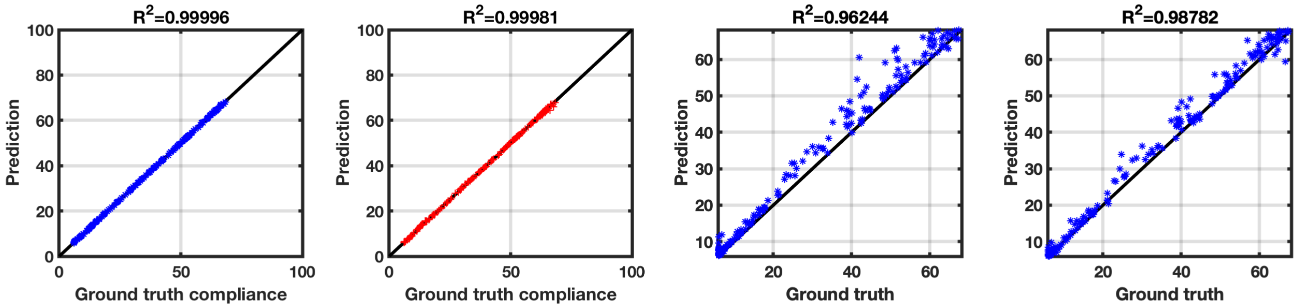


Fig. 2: **Prediction:** Density field (top) and stress field (bottom) predictions for cantilever beam (re-scaled to 0 and 1 for visualization) given simulation parameters. In each image, top: Ground truth; middle: Shared-GP; bottom: Pairwise-GPs.



(a) Pairwise-GPs (b) Shared-GP (c) Pairwise-GPs (d) Shared-GP

Fig. 3: **Prediction:** $R^2$ statistics of predicted compliance (a,b) and computed compliance of predicted density fields (c,d) given simulation parameters.

### 6.2 Topology optimization for cantilever beam design

In this example, we consider the topology optimization of a cantilever beam, shown in Fig. 1 (left). We took advantage of the fast implementation in [56] to perform density-based topology optimization by minimizing the compliance $C$ subject to volume constraints $V \leq \bar{V}$ ($\bar{V} = 0.5$ was used for this experiment). We used the SIMP method [57] to transform continuous density values to discrete optimized topologies with 1 and 0 indicating solid and void, respectively. We assumed three simulation parameters, namely, the location of point load $P_1$, the angle of point load $P_2$, and the filter radius $P_3$ [58]. We generated $N = 600$ data points associated with $\{P_i \rightarrow S_i \rightarrow C_i\}_{i=1}^{N}$, where $P = (P_1 \in [-20,20], P_2 \in [0,\pi], P_3 \in [1.1,2.5])$ is the simulation parameter triplet, $S$ is the optimized topology, and $C$ is the final compliance. We used a $40 \times 80$ regular mesh to solve this problem and the same mesh to present the field outputs. We performed five repeated random subsamplings to construct training (400 samples) / testing (200 samples) datasets.

**Prediction**: Given new (unseen) simulation parameters, we first demonstrate the performance of Shared-GP, compared to Pairwise-GPs (i.e., multiple traditional direct GPs [28, 59]) to model the mapping from simulation parameters to the density field, stress field, and compliance (Fig. 3). For Shared-GP, we set the latent dimension $L = 3$. For the density field predictions, the mean square error (MSE) is $6.52\text{e}{-4} \pm 4.7\text{e}{-3}$ and $3.8\text{e}{-4} \pm 2.5\text{e}{-3}$ for Shared-GP and Pairwise-GPs, respectively. For the stress field predictions, Shared-GP shows an MSE of $0.029 \pm 0.093$, whereas Pairwise-GPs $0.027 \pm 0.079$. Fig. 2 shows eight randomly selected predictive density and stress fields of Shared-GP and Pairwise-GPs. Both model predictions show different levels of artifacts. It has been reported in the literature that learning the output-correlation can improve the model performance when predicting a mutivariate output [8, 45, 60]. However, this is not the case here because we are dealing with noiseless simulation data, and the autokrigeability takes place (see Appendixes A and B for mathematical and empirical proofs). This artifact is mainly due to the specific difficulty when a surrogate model tries to capture the topology structure that contains sharp and fine details with limited training samples and inherent disjoint clusters in the solution (i.e., optimized topology) space. An effective way to reduce this artifact is to increase the training samples (see Appendix B for empirical results).

A surrogate model should be able to generate valid topological structures. We thus validate the density field prediction by computing its compliance, which is then compared with the compliance computed from a ground-truth design. The results are showed in Figs. 3(c) and 3(d), in which it is clear that the accuracy is not as good as that in Fig. 3(a,b), highlighting the challenges in learning high-dimensional field outputs compared to scalar-valued outputs. However, the predictive structures are still good approximations that generate similar compliance. In cases where errors are less acceptable, the predictive

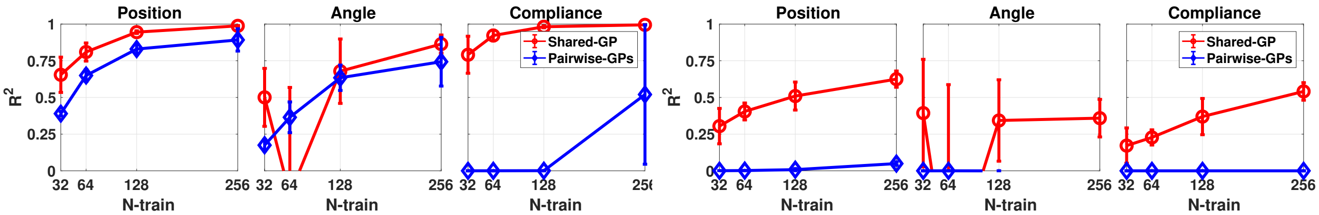


Fig. 4: **Inference:** $R^2$ statistics of predictive simulation parameters and compliance given stress fields (top row) and density fields (bottom row) with an increasing number of training samples.

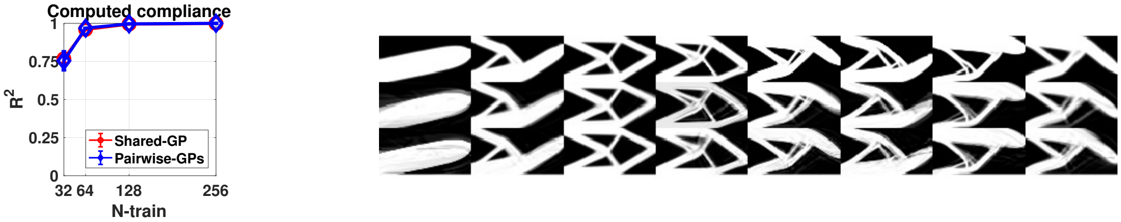
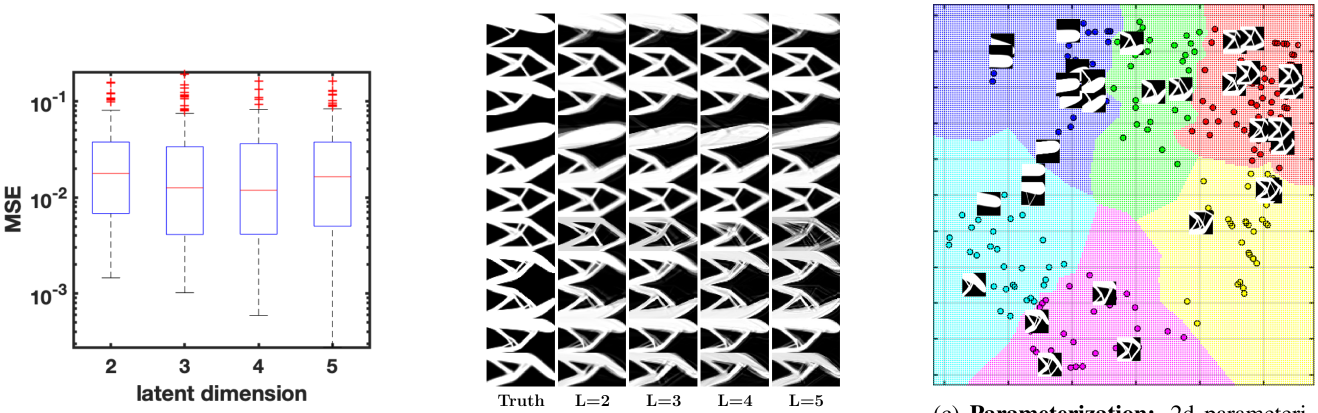


Fig. 5: **Inference:** Computed compliance based on predictive density fields (left) and eight predictive density field instances (right) with 256 training points. Top: Ground truth; middle: Shared-GP; bottom: Pairwise-GPs.

design can be used as an initialization for a simulation to provide a faster convergence. We therefore run the 200 test cases using the Shared-GP model predictions as initialization and record the number of iterations until convergence. Our method required 25.8 ± 17.3 iterations on average per test case, Pairwise-GPs 24.9 ± 21.5, and the original simulator 32.3 ± 13.5. The computational savings is insignificant (about 20%) because we used only a small amount of training data for demonstration purposes. In practical design space exploration, as we keep exploring the design space by executing the simulator, we will have a growing training dataset to improve our initializations, which can be generated with little extra computational cost using our model.

**Inference**: The standard simulation process follows a forward path, starting from simulation parameters to design solutions to derived QoIs (i.e., low- to high-dimensional spaces). It is sometimes desirable to reverse the direction to help designers build an understanding across these spaces, i.e., given some QoIs, predict the simulation parameters and other QoIs. In this experiment, we assess the Shared-GP's capacity, compared to Pairwise-GPs, to predict a scalar value from a high-dimensional field. Here, we used the density field or stress field to predict the simulation parameters and compliance. This model capacity is particularly important to accelerate simulation parameter space exploration when real simulation is not available or is costly to obtain. We varied the number of training samples from 32 to 256 and used 128 test samples to compute the coefficient of determination, i.e., $R^2$ statistics (closer to 1 is better). In Fig. 4, we show the mean and standard deviation of the $R^2$ statistics using five repeated random subsampled training/testing datasets. Both methods fail to give meaningful predictions for the filter size even with 256 training samples, and the results are thus not shown. Notice the performance drops for both methods when using the density field as the model input. This behavior can be attributed to the lack of data processing for the density fields whose pixel values show a dramatic change at the structure edge. One can observe that Shared-GP significantly outperforms Pairwise-GPs due to the *curse of dimensionality* of the high-dimensional input, making the training of Pairwise-GPs infeasible as the number of hyperparameters explodes. Our method, on the other hand, bypasses this issue by learning the reverse process (low-to-high mapping) and treating the learning problem as an inference problem. To validate the results of density field predictions, we report the $R^2$ statistics of the computed compliance based on the predictive density fields as a function of the training sample size and eight random samples using 256 training samples in Fig. 5.

**Dimension reduction and hidden latent space**: In simulation applications, some of the simulation parameters have a negligible impact on the characterization of design products. For instance, in the cantilever beam design problem, the filter size has significantly less impact on the overall layout of the optimized structure compared to the position and angle. In particular, changing the position and angle leads to significantly different layouts, whereas changing the filter size can impact only the thickness of the bars that appear in the design. Therefore, it is useful to identify these subtle parameters such that the designers (and customers) can concentrate on those of more importance to the resulting optimized structures. From the previous inference experiments, we already know that both Shared-GP and Pairwise-GPs fail in predicting the filter sizes, implying that the filter size may have negligible effects since both models cannot learn any useful patterns given the training data. To validate our assumption, we vary the latent dimension from 2 to 5 and train each Shared-GP with 256 training samples. Given 128 unseen, hold-out stress fields, the predictions of position, angle, and filter size are given



(a) Tukey box-plot of Shared-GP prediction MSE of density fields using different latent dimensions (L)

(b) Ten randomly selected density field predictions of Shared-GP with different latent dimensions (L)

(c) **Parameterization:** 2d parameterization of cantilever data with clusters by DPP and decision boundary by k-neighbor.

Fig. 6: Dimension reduction and hidden latent space

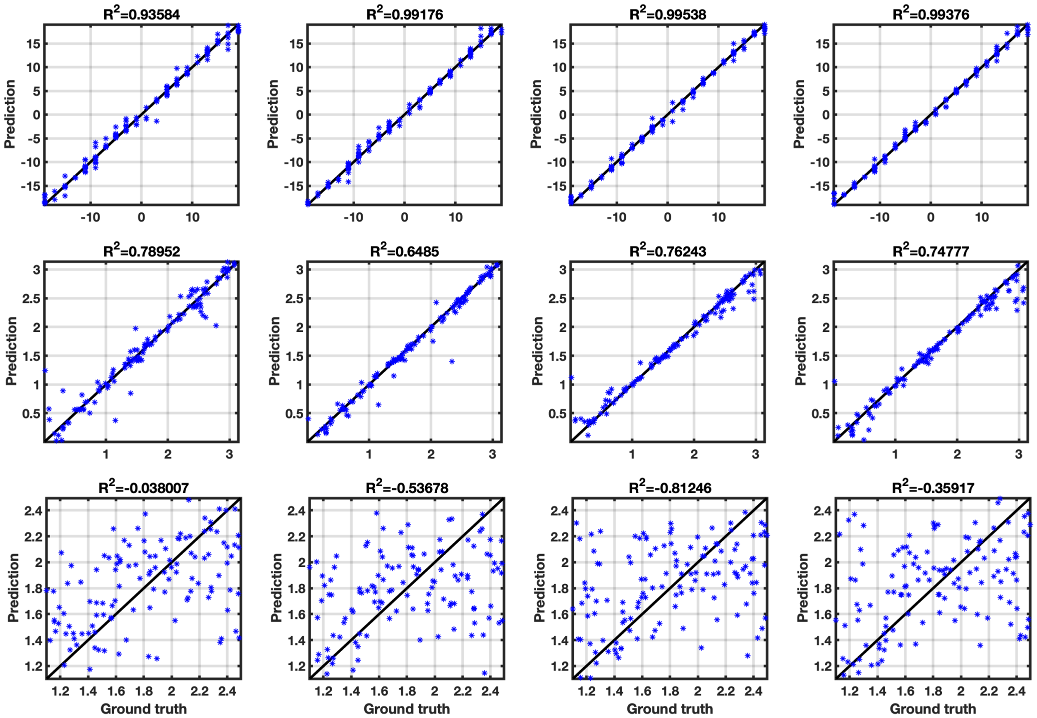


Fig. 7: Shared-GP predictions of the position (top row), angle (middle row), and filter size (bottom row) using latent dimension from 2 to 5 (left to right columns).

in Fig. 7. Prediction errors of the density fields using mean square error (MSE) are shown with a box plot in Fig. 6a, and 10 randomly selected density field predictions are given in Fig. 6b. We can see that the model has negligible improvement beyond $L = 2$ for predictions of position, angle, and density field, and filter size cannot be predicted accurately for all cases. Note that $R^2$ significantly reduces at $L = 3$ for the angle predictions. However, if we look at the individual predictions carefully, we conclude that the performance deterioration might be caused by one or two outlier predictions. Also note that the performance deteriorates slightly for $L = 5$, indicating that a more complex model is not necessarily a better model as it complicates the training and prediction process. We conclude that Shared-GP can automatically reveal the important factors by maximizing the joint likelihood of Eq. (5) with a given latent dimension. If the reconstruction of QoIs is sufficiently accurate and the latent dimension is smaller than the dimension of the simulation inputs, a lower-dimensional latent space exists, from which we can derive the important factors.

**Structured parameterization**: With the dimension reduction validated in the previous section, we now demonstrate the

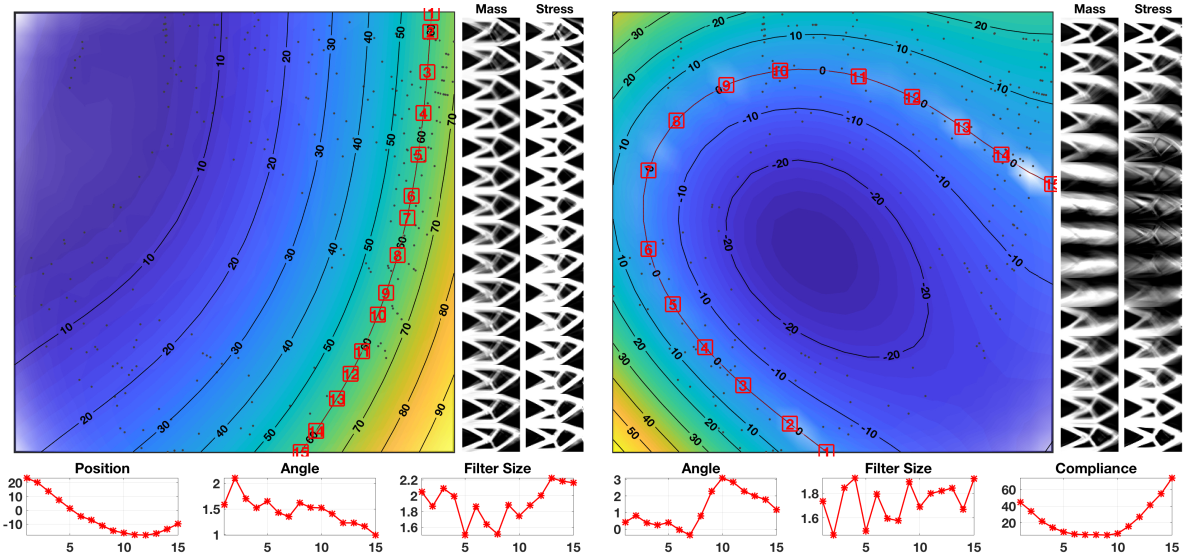


Fig. 8: **Exploration:** QoI interpolation over the latent space based on compliance (left) and position (right). The two image columns are stress fields and density fields interpolation along contour=0. Bottom subfigures show QoI interpolations.

core utilities of our model, i.e., deriving a meaningful structured hidden/latent parameterization that provides insight into the space of simulation parameters and useful explorations for the trade-off between QoIs. We used 256 training samples to train Shared-GP and derive a 2d ($L = 2$) latent space (see Fig. 6c). Topologies with different structures are automatically clustered, thanks to the DPP priors. This structured layout of the simulation parameter space can help designers focus on a particular cluster of topologies to further explore regions of simulation parameters (in relation to other QoIs) that were never visited/simulated. To show that this structured latent space is useful for simulation parameter space exploration, we show the contour plot of predicted compliances and positions in Fig. 8. To demonstrate associated uncertainties, we used the prediction variance of the GP to calculate the variance-to-mean ratio (VMR) to set the transparency of the contour plot (transparency increases with larger uncertainty). The top left image of Fig. 8 clearly shows the lack of confidence in the predictions. We then used the contour map to guide movements (as indicated by the sequential numbers) in the latent space to estimate the corresponding topology structures and predict the stress fields and simulation parameters. Notice that the filter size cannot be predicted accurately (as mentioned earlier) and its behavior seems random. Fig. 8 clearly shows that, for a fixed position of 0, i.e., of a load in the middle of the structure, the lowest compliance values are found to be associated with the angle = $\pi/2$. Accordingly, compliance increases as the angle approaches either 0 or $\pi$, which conforms to our understanding of the cantilever beam design problem.

### 6.3 Topology optimization for L-Bracket

We now test our approach on a more challenging problem, namely, the L-shape structure (see Fig. 1 (right)), which presumably yields more scattered optimized topologies with respect to variations in the simulation parameter space. We generated data similarly to the beam example, i.e., we performed compliance minimization subject to volume constraint with $\bar{V} = 0.4$ and generated $N = 500$ samples of $(P_i \rightarrow X_i \rightarrow C_i)$. In this example, we generated optimized topologies concerning scatter in the elastic modulus field of the structure. To that end, we consider a Karhunen-Loève (KL) expansion in the form of $E(x) = E_0 + \delta\sum_{i=1}^{n_M}\sqrt{\lambda_i}\gamma_i(x)\xi_i$ where $\xi_i \sim \mathcal{N}(0,1)$ are standard normal random variables. We use the explicit expressions $b_n^{1D} = A_n(\sin\omega_n x + \omega_n\cos(\omega_n x)); \lambda_n^{1D} = \frac{2}{1+\omega_n^2}$ for computing the eigenvalues and eigenvectors of the exponential kernel in two dimensions, $\gamma_n = b_{i_n}^{1D} b_{j_n}^{1D}; \lambda_n = \lambda_n^{1D}\lambda_n^{1D}$, where the constant $A_n$ is chosen such that $\|b_n^{1D}\| = 1$ [61]. We used $E_0 = 1, \delta = 0.02$ and assumed $n_M = 10$ eigenvalues with equally spaced frequencies $\omega_n \in [1, 2.8]$ to generate the elasticity field data. As part of this experiment, we used natural periods of the optimized structures as the QoIs. We first computed the mass matrix of the structure $M$ given the material layout. Having the stiffness matrix $K$, we then computed the three largest eigenvalues of $K^{-1}M$ as the square of natural period $T^2$. We also used a $300 \times 300$ mesh for elastic random field, stress field, and density field to demonstrate our model scalability for high-dimensional QoIs.

The task is to predict the compliance and the first, second, and third natural periods given an unseen elastic modulus field or stress field as the model input. This problem is challenging due to the very high-dimensional inputs (close to 1 million). To better reflect the accuracy on each predictions, we use a general metric, mean absolute percentage error (MAPE), computed as $mape = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|/y_i$, where $y_i$ is the ground-truth and $\hat{y}_i$ is the prediction. We fixed the latent dimension to 5 and ran each experiment three times with random shuffling training data and 200 fixed testing data points to compute the MAPE

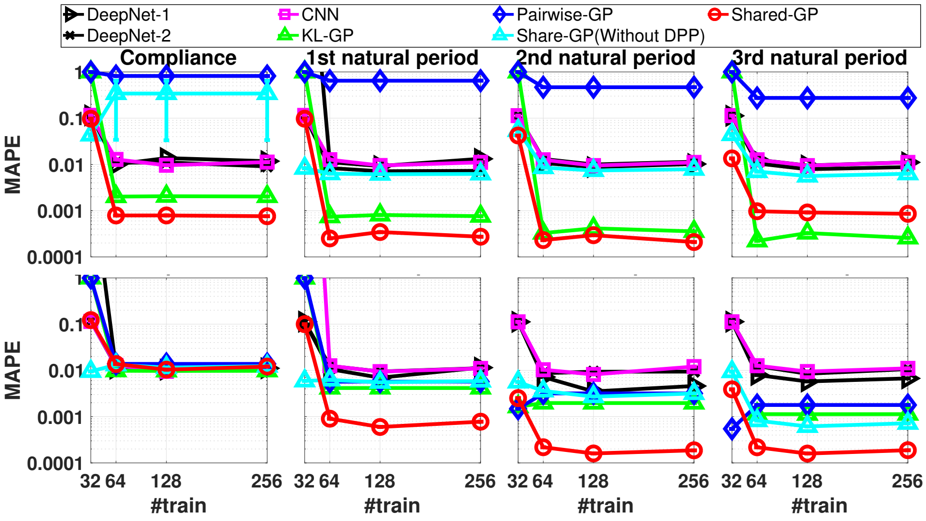


Fig. 9: MAPE statistics of QoI predictions with an increasing number of training samples given elastic modulus fields (top row) and stress fields (bottom row) as model inputs.

statistics as the final results.

To highlight the model accuracy improved by the DPP, we compared Share-GP with the exact same Shared-GP model without the DPP, which is equivalent to shared GPLVM. We also included KL-GP, which uses a discrete KL decomposition to first project the high-dimensional inputs to a low-dimensional subspace and then uses a GP to map the representations to the outputs. Finally, to compete with the powerful state-of-the-art deep learning models, we implemented three deep models. DeepNet-1 uses four fully connected hidden layers (300-100-5-5), DeepNet-2 uses six fully connected hidden layers (10000-1000-100-5-5-5), and CNN uses four convolution and pooling layers and two fully connected hidden layers (5-5). We limited the number of hidden units for the last layer to 5 so that these layers can be considered to provide low-dimensional representations equivalent to those of other methods. If a deep model is strong enough, it should be able to disentangle the complex high-dimensional inputs such that the low-dimensional outputs can be easily derived from the representations. All deep learning models use ReLu activation and are trained using stochastic gradient decent until convergence.

The results of using the elastic modulus fields and stress fields as inputs are shown in Fig. 9. Pairwise-GPs fail the task when using the elastic modulus fields as inputs due to the explosive number of hyperparameters. On the other hand, Pairwise-GPs perform fairly well when using stress fields as inputs. This good performance suggests that the stress field, as a simulation result of an elasticity field, shows a simpler pattern to learn. Nevertheless, obtaining the stress fields requires running costly simulations, a scenario that we want to avoid in the first place. When using elastic modulus fields as model inputs, KL-GPs perform well because the random fields are indeed generated using a KL expansion; not surprisingly, when using the stress fields as inputs, KL-GPs' performance decreases significantly due to the lack of the generality of the KL expansion. Shared-GP without DPP outperforms Shared-GP when the training samples are very limited (i.e., 32) because the DPP indeed requires more samples for the model training. However, Shared-GP without DPP struggles to improve with increasing training data. Shared-GP is the most stable method that works for all scenarios and outperforms other methods by a large margin when using the stress fields as the model inputs. When using the elastic modulus fields as inputs, shared-GP shows a better performance than KL-GP in most cases due to its capacity to capture the nonlinearity and multimode in the data. All deep models show similar results, stable for all cases and gradually improving with more training data. We believe the inferior performance of the deep models is due to the lack of training data and tuning tricks. We can exhaust different architectures, initialization methods, activation functions, and optimization methods to improve model performance with a large amount of computational resources (e.g., using a single core of a Intel i7 3.5GHz CPU, CNN took about 30s for 1 epoch with 128 training points whereas Shared-GP took approximately 0.05s.). However, such an approach can defeat the purpose of introducing a surrogate model in the first place. In contrast, Shared-GP provides an efficient solution (with many fewer model parameters) to the common multiple data space situation in a design process.

## 7 Conclusions

In this paper, we introduce a rigorous probabilistic model, Shared-GP, that finds a shared low-dimensional latent structure for design space analysis and exploration. To the best of our knowledge, this is the first work to introduce a shared latent space for modeling multiple QoIs, which are common in many design problems. Our results demonstrate the model capability for accurate QoI predictions, efficient inference across QoI spaces, dimension reduction, and intuitive design space visualization. Dimension reduction and visualization are particularly useful tools for both designers and customers to understand the trade-off among different QoIs; they provide an intuitive, less computationally expensive way to freely explore how the design changes when changing the QoIs, how the change of design influences the QoIs, and how the marginal utility decreases when we slightly alter the design. Note that our model has certain limitations. First, the inference across data spaces can fail when the underlying reverse mapping does not exist. Second, for dimension reduction, if all simulations are independent and significant for the designs/QoIs, any attempts at dimension reduction will fail, and the dimension of the latent representation should be not less than the dimensionality of the simulation inputs. Further improvements of this work include introducing advanced GP models (e.g., deep GP [62] and GP networks [63]), implementing advanced kernels (e.g., spectrum mixture kernels [60] and deep kernels [64]), and injecting physical constraints to the model [15].

## Acknowledgements

This work has been supported by DARPA TRADES Award HR0011-17-2-0016. The authors would like to thank Kyli McKay-Bishop for proofreading the manuscript.

## References

[1] Zitzler, E., 1999. *Evolutionary algorithms for multiobjective optimization: Methods and applications*, Vol. 63. Citeseer.

[2] Knowles, J., and Corne, D., 1999. "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation". In Congress on Evolutionary Computation (CEC99), Vol. 1, pp. 98–105.

[3] Chen, W., Fuge, M., and Chazan, J., 2017. "Design manifolds capture the intrinsic complexity and dimension of design spaces". *Journal of Mechanical Design,* **139**(5), p. 051102.

[4] Burnap, A., Liu, Y., Pan, Y., Lee, H., Gonzalez, R., and Papalambros, P. Y., 2016. "Estimating and exploring the product form design space using deep generative models". In ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A013–V02AT03A013.

[5] Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., and Möller, T., 2014. "Visual parameter space analysis: A conceptual framework". *Visualization and Computer Graphics, IEEE Transactions on*(99).

[6] Torsney-Weir, T., Saad, A., Moller, T., Hege, H.-C., Weber, B., Verbavatz, J.-M., and Bergner, S., 2011. "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration". *IEEE Transactions on Visualization and Computer Graphics,* **17**(12), pp. 1892–1901.

[7] Averkiou, M., Kim, V. G., Zheng, Y., and Mitra, N. J., 2014. "Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis". In Computer Graphics Forum, Vol. 33, Wiley Online Library, pp. 125–134.

[8] Xing, W., Triantafyllidis, V., Shah, A., Nair, P., and Zabaras, N., 2016. "Manifold learning for the emulation of spatial fields from computational models". *Journal of Computational Physics,* **326**, pp. 666–690.

[9] Apley, D. W., Liu, J., and Chen, W., 2006. "Understanding the effects of model uncertainty in robust design with computer experiments". *Journal of Mechanical Design,* **128**(4), pp. 945–958.

[10] Wang, G. G., and Shan, S., 2007. "Review of metamodeling techniques in support of engineering design optimization". *Journal of Mechanical design,* **129**(4), pp. 370–380.

[11] Jeong, S., Murayama, M., and Yamamoto, K., 2005. "Efficient optimization design method using kriging model". *Journal of aircraft,* **42**(2), pp. 413–420.

[12] Bessa, M., Bostanabad, R., Liu, Z., Hu, A., Apley, D. W., Brinson, C., Chen, W., and Liu, W., 2017. "A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality". *Computer Methods in Applied Mechanics and Engineering,* **320**, June, pp. 633–667.

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

[14] D'Agostino, D., Serani, A., Campana, E. F., and Diez, M., 2018. "Deep autoencoder for off-line design-space dimensionality reduction in shape optimization". In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 1648.

[15] Cang, R., Yao, H., and Ren, Y., 2019. "One-shot generation of near-optimal topology through theory-driven machine learning". *Computer-Aided Design,* **109**, pp. 12–21.

[16] Sosnovik, I., and Oseledets, I., 2019. "Neural networks for topology optimization". *Russian Journal of Numerical Analysis and Mathematical Modelling,* **34**(4), pp. 215–223.

[17] Gal, Y., Chen, Y., and Ghahramani, Z., 2015. "Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data". *arXiv:1503.02182 [stat]*, Mar. arXiv: 1503.02182.

[18] Oakley, J., and O'Hagan, A., 2002. "Bayesian inference for the uncertainty distribution of computer model outputs". *Biometrika,* **89**, pp. 769–784.

[19] Girard, A., Rasmussen, C. E., Candela, J. Q., and Murray-Smith, R., 2003. "Gaussian process priors with uncertain in-

puts application to multiple-step ahead time series forecasting". In Advances in neural information processing systems, pp. 545–552.

[20] Shon, A., Grochow, K., Hertzmann, A., and Rao, R. P., 2006. "Learning shared latent structure for image synthesis and robotic imitation". In Advances in neural information processing systems, pp. 1233–1240.

[21] El-Beltagy, M. A., and Keane, A. J., 2001. "Evolutionary optimization for computationally expensive problems using gaussian processes". In Proc. Int. Conf. on Artificial Intelligence, Vol. 1, Citeseer, pp. 708–714.

[22] Jiang, Z., Chen, W., Fu, Y., and Yang, R.-J., 2013. "Reliability-based design optimization with model bias and data uncertainty". *SAE International Journal of Materials and Manufacturing,* **6**(3), pp. 502–516.

[23] Eleftheriadis, S., Rudovic, O., and Pantic, M., 2013. "Shared gaussian process latent variable model for multi-view facial expression recognition". In International Symposium on Visual Computing, Springer, pp. 527–538.

[24] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E., 2015. "Multi-view convolutional neural networks for 3d shape recognition". In Proceedings of the IEEE international conference on computer vision, pp. 945–953.

[25] Ge, L., Liang, H., Yuan, J., and Thalmann, D., 2016. "Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3593–3601.

[26] Ek, C. H., Jaeckel, P., Campbell, N., Lawrence, N. D., and Melhuish, C., 2009. "Shared gaussian process latent variable models for handling ambiguous facial expressions". In AIP Conference Proceedings, Vol. 1107, AIP, pp. 147–153.

[27] Chen, W., and Fuge, M., 2019. "Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks". *Journal of Mechanical Design,* **141**(11).

[28] Conti, S., and OHagan, A., 2010. "Bayesian emulation of complex multi-output and dynamic computer models". *Journal of Statistical Planning and Inference,* **140**, pp. 640–651.

[29] Rasmussen, C. E., and Williams, C. K. I., 2006. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. OCLC: ocm61285753.

[30] Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P., 2015. "Deep Kernel Learning". *arXiv:1511.02222 [cs, stat]*, Nov. arXiv: 1511.02222.

[31] Wilson, A. G., 2014. "Covariance kernels for fast automatic pattern discovery and extrapolation with gaussian processes". *University of Cambridge*.

[32] Hinton, G. E., and Salakhutdinov, R. R. "Using Deep Belief Nets to Learn Covariance Kernels for Gaussian Processes". p. 8.

[33] Bishop, M., and Whitlock, P. A., 2007. "Monte carlo simulation of hard hyperspheres in six, seven and eight dimensions for low to moderate densities". *Journal of Statistical Physics,* **126**(2), pp. 299–314.

[34] Higdon, D., Gattiker, J., Williams, B., and Rightley, M., 2008. "Computer model calibration using high-dimensional output". *Journal of the American Statistical Association,* **103**(482), pp. 570–583.

[35] Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J., and Walsh, D., 2007. "Computer model validation with functional output". *The Annals of Statistics,* **35**(5), Oct., pp. 1874–1906. arXiv: 0711.3271.

[36] Ma, X., and Zabaras, N., 2011. "Kernel principal component analysis for stochastic input model generation". *J. Comput. Phys.,* **230**, pp. 7311–7331.

[37] Xing, W., Shah, A. A., and Nair, P. B., 2015. "Reduced dimensional gaussian process emulators of parametrized partial differential equations based on isomap". In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 471, The Royal Society, p. 20140697.

[38] Xing, W., Triantafyllidis, V., Shah, A., Nair, P., and Zabaras, N., 2016. "Manifold learning for the emulation of spatial fields from computational models". *Journal of Computational Physics,* **326**, pp. 666–690.

[39] Gadd, C., Xing, W., Nezhad, M. M., and Shah, A., 2018. "A surrogate modelling approach based on nonlinear dimension reduction for uncertainty quantification in groundwater flow models". *Transport in Porous Media*, pp. 1–39.

[40] Wackernagel, H., 1995. *Multivariate Geostatistics*. Springer, Berlin.

[41] Zhang, H., 2007. "Maximum-likelihood estimation for multivariate spatial linear coregionalization models". *Environmetrics,* **18**(2), Mar., pp. 125–139.

[42] Fricker, T., Oakley, J., and Urban, N., 2013. "Multivariate gaussian process emulators with nonseparable covariance structures". *Technometrics,* **55**, pp. 47–56.

[43] Konomi, B., Karagiannis, G., Sarkar, A., Sun, X., and Lin, G., 2014. "Bayesian treed multivariate gaussian process with adaptive design: Application to a carbon capture unit". *Technometrics,* **56**, pp. 145–158.

[44] Wilson, A. G., Gilboa, E., Nehorai, A., and Cunningham, J. P., 2014. "Fast kernel learning for multidimensional pattern extrapolation". In Advances in Neural Information Processing Systems, pp. 3626–3634.

[45] Zhe, S., Xing, W., and Kirby, M., 2019. "Scalable high-order gaussian process regression". In AISTAT.

[46] Wilson, A. G., and Nickisch, H. "Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP)". p. 10.

[47] Alvarez, M. A., Rosasco, L., and Lawrence, N. D., 2012. "Kernels for Vector-Valued Functions: A Review". *Foundations and Trends in Machine Learning,* **4**(3), pp. 195–266.

[48] Lawrence, N., 2005. "Probabilistic non-linear principal component analysis with Gaussian process latent variable models". *Journal of machine learning research,* **6**(Nov), pp. 1783–1816.

[49] Zhe, S., Xu, Z., Chu, X., Qi, Y., and Park, Y., 2015. "Scalable nonparametric multiway data analysis". In Artificial Intelligence and Statistics, pp. 1125–1134.

[50] Blei, D. M., Jordan, M. I., et al., 2006. "Variational inference for dirichlet process mixtures". *Bayesian analysis,* **1**(1), pp. 121–143.

[51] Eleftheriadis, S., Rudovic, O., and Pantic, M., 2013. "Shared gaussian process latent variable model for multi-view facial expression recognition". In International Symposium on Visual Computing, Springer, pp. 527–538.

[52] Sobol, I., 1976. "Uniformly distributed sequences with an addition uniform property". *USSR Comput. Maths. Math. Phys.,,* **16**, pp. 236–242.

[53] Jin, R., Chen, W., and Sudjianto, A., 2003. "An efficient algorithm for constructing optimal design of computer experiments". In ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 545–554.

[54] Sobol', I. M., 1967. "On the distribution of points in a cube and the approximate evaluation of integrals". *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki,* **7**(4), pp. 784–802.

[55] Bates, R., Buck, R., Riccomagno, E., and Wynn, H., 1996. "Experimental design and observation for large systems". *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 77–94.

[56] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011. "Efficient topology optimization in matlab using 88 lines of code". *Structural and Multidisciplinary Optimization,* **43**(1), Jan, pp. 1–16.

[57] Bendsoe, M. P., and Sigmund, O., 2004. "Topology optimization: Theory, methods and applications". *Springer*.

[58] Bruns, T. E., and Tortorelli, D. A., 2001. "Topology optimization of non-linear elastic structures and compliant mechanisms". *Computer Methods in Applied Mechanics and Engineering,* **190**(26), pp. 3443 – 3459.

[59] O'Hagan, A., and Kingman, J. F. C., 1978. "Curve Fitting and Optimal Design for Prediction". *Journal of the Royal Statistical Society. Series B (Methodological),* **40**(1), pp. 1–42.

[60] Wilson, A., and Adams, R., 2013. "Gaussian process kernels for pattern discovery and extrapolation". In International Conference on Machine Learning, pp. 1067–1075.

[61] Teckentrup, A. L., Jantsch, P., Webster, C. G., and Gunzburger, M., 2015. "A multilevel stochastic collocation method for partial differential equations with random input data". *SIAM/ASA Journal on Uncertainty Quantification,* **3**(1), pp. 1046–1074.

[62] Damianou, A., and Lawrence, N., 2013. "Deep gaussian processes". In Artificial Intelligence and Statistics, pp. 207–215.

[63] Friedman, N., and Nachman, I., 2000. "Gaussian process networks". In Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp. 211–219.

[64] Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P., 2016. "Deep kernel learning". In Artificial Intelligence and Statistics, pp. 370–378.

## List of Figures

## Appendix A: Autokrigeability

The LMC model assumes that the covariance of the ouput dimension and the covariance of different samples are separable, i.e., $\mathbb{C}\text{ov}(y^{(h)}_{ni}, y^{(h)}_{n'j}) = k^{(h)}(\mathbf{z}_n, \mathbf{z}'_n)Q^{(h)}_{ij}$, where $Q^{(h)}_{ij}$ indicates the covariance between the $i-th$ and $j-th$ dimensions. Assuming that the observations are noise free, we can write the joint distribution of all observations as a multivariate Gaussian distribution:

$$\mathbf{Y}^{(h)} \sim \mathcal{MN}_{D^{(h)}\times N}(\mathbf{0}, \mathbf{L}^{(h)}, \mathbf{K}^{(h)}) = \frac{\exp\left(-\frac{1}{2}\text{tr}[\mathbf{Q}^{(h)^{-1}}\mathbf{Y}^{(h)}\mathbf{K}^{(h)^{-1}}\mathbf{Y}^{(h)}]\right)}{(2\pi)^{O_h N/2}|\mathbf{L}^{(h)}|^{O_h/2}|\mathbf{K}^{(h)}|^{N/2}}. \tag{13}$$

The posterior distribution give $\mathbf{z}_*$ is also a Gaussian whose mean is

$$\begin{aligned}\mathbb{E}[\mathbf{y}^{(h)}_*] &= (\mathbf{Q}^T \otimes \mathbf{k}^{(h)})(\mathbf{Q}^T \otimes \mathbf{K}^{(h)})^{-1}\text{vec}(\mathbf{Y}^{(h)})\\ &= (\mathbf{Q}^T\mathbf{Q}^{-T}) \otimes (\mathbf{k}^{(h)}\mathbf{K}^{(h)})^{-1})\text{vec}(\mathbf{Y}^{(h)})\\ &= \mathbf{I} \otimes (\mathbf{k}^{(h)}\mathbf{K}^{(h)})^{-1})\text{vec}(\mathbf{Y}^{(h)}).\end{aligned} \tag{14}$$

It is clear that the actual structure and value of $\mathbf{Q}^{(h)}$ do not matter because they will always be canceled out when making the mean predictions.

## Appendix B: Autokrigeability in topology optimization

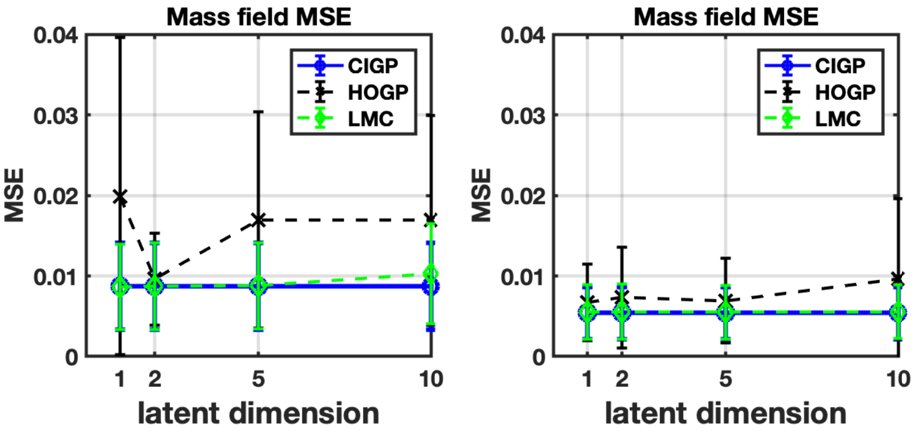
To validate the autokrigeability for our applications, we compare conditional independent GP (CIGP), high-order GP (HOGP), the state-of-the-art GP surrogate model for high-dimensional problems [49], and the linear model of coregionalization (LMC), the most popular GP framework for modeling multioutput problems [40]. Specifically, we predict the density fields given the design parameters for the topology optimization data of section 6.2. All models use MLE with a conjugate gradient method with maximum 300 iterations for model training. We first show the mean square error (MSE) with the standard deviation as an error bar of 128 predictive density fields using 256 training samples in Fig. Appendix 1a and Fig. Appendix 2. The latent dimension is the low-rank approximation of the correlation matrix ($Q$ in Eq. (14)) for HOGP and LMC and has no influence on CIGP. For noisy observations, the performance should gradually improve as we increase the number of latent dimensions for HOGP and LMC [49]. However, we see no performance improvements but, rather, deteriorations, mainly because of the numerical instability when LMC and HOGP invert and multiply the very large correlation matrix by itself ($Q^T Q^{-T}$ in Eq. (14)). As we use a higher latent dimension, we introduce more free parameters and complicate the model training, which eventually leads to inferior performance. In order to capture nonlinear correlations, HOGP introduces a nonlinear transformation, which, in this case, becomes a burden for model training and causes more numerical instability, as is shown in Fig. Appendix 1. Overall, we see no improvements when learning the output-correlations for our applications, which validates the autokrigeability for noiseless simulation problems. To improve the performance, we need a more appropriate kernel for this problem, or we can simply increase the number of samples for model training. The performance using 512 training samples is shown in Fig. Appendix 1b and Fig. Appendix 3. Comparing Fig. Appendix 2 and Fig. Appendix 3, it is clear that the artifacts in Fig. Appendix 2 are due to the lack of training samples considering the difficulty of the problems. Exploring different types of kernel is beyond the scope of this paper, and we leave it for future work.

## Appendix C: Truncated variational posterior for DPP

To have a DPP with finite support, we can use a truncated variational posterior [49] by setting a truncation level $T$ for each mode and set $q(v_{T=1}) = 1$ so that $q(w_m > T) = 0$. The DPP variational distributions are thus given by

$$\begin{aligned}q(w_n) &= \text{Multi}(w_n|\zeta_{n1}, \cdots, \zeta_{nT}),\\ q(v_m) &= \text{Beta}(v_m|\gamma_{m1}, \gamma_{m2}),\\ q(\boldsymbol{\eta}_m) &= \mathcal{N}(\boldsymbol{\eta}_m|\boldsymbol{\mu}_m, s_m\mathbf{I}),\end{aligned} \tag{15}$$

where



(a) N-train=256

(b) N-train=512

Fig. Appendix 1: **Prediction:** Density field prediction MSE with 256 and 512 training samples.

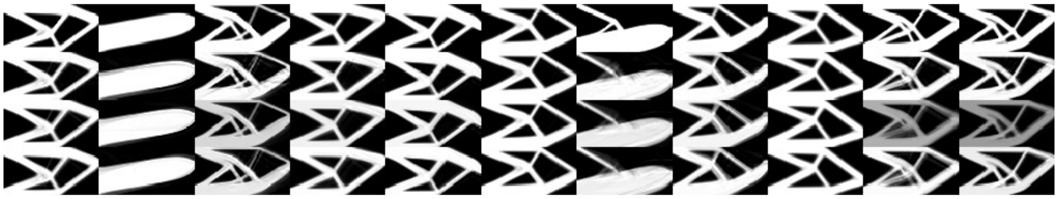


Fig. Appendix 2: **Prediction:** Density field predictions given simulation parameters using 256 training samples. Top row: Ground truth; top second: CIGP; top third: HOGP; bottom: LMC.

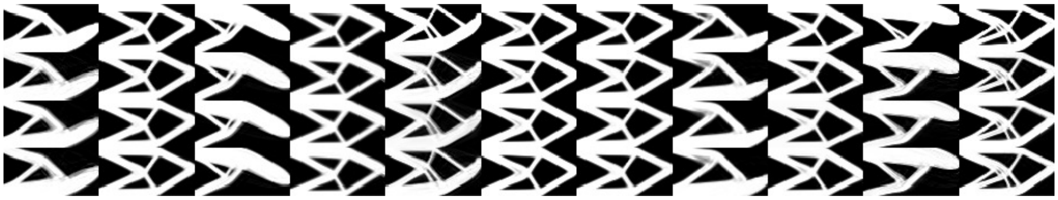


Fig. Appendix 3: **Prediction:** Density field predictions given simulation parameters using 512 training samples. Top row: Ground truth; top second: CIGP; top third: HOGP; bottom: LMC.

$$
\begin{aligned}
&\zeta_{nm} \propto \exp\left(\mathbb{E}_q\left[\log(v_m)\right] + \sum_{m'=1}^{m-1} \mathbb{E}_q\left[\log(1-v_{m'})\right] - \frac{1}{2\lambda}\mathbb{E}_q\left[||\boldsymbol{\eta}_m||^2\right] + \frac{1}{\lambda}\mathbf{z}_n{}^T\mathbb{E}_q\left[\boldsymbol{\eta}_m\right]\right),\\
&\gamma_{m1} = 1 + \sum_{n=1}^{N} \zeta_{nm}^{(k)}, \quad \gamma_{m2} = \alpha + \sum_{n=1}^{N}\sum_{m'=m+1}^{T} \zeta_{nm'},\\
&s_m = \frac{1}{1+\lambda^{-1}\sum_{n=1}^{N}\zeta_{nm}}, \quad \boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N}\zeta_{nm}\mathbf{z}_n}{\lambda_k + \sum_{n=1}^{N}\zeta_{nm}}.
\end{aligned}
\tag{16}
$$

The moments are computed as

$$
\begin{aligned}
\mathbb{E}_q[\log v_m] &= \psi(\gamma_{m1}) - \psi\left(\gamma_{m1}+\gamma_{m2}\right),\\
\mathbb{E}_q[\log(1-v_m)] &= \psi(\gamma_{m2}) - \psi\left(\gamma_{m1}+\gamma_{m2}\right),\\
\mathbb{E}_q[\boldsymbol{\eta}_m] &= \boldsymbol{\mu}_m,\\
\mathbb{E}_q[||\boldsymbol{\eta}_m||^2] &= ||\boldsymbol{\eta}_m||^2 + Ls_m,\\
\psi(x) &= \frac{d}{dx}\ln\Gamma(x).
\end{aligned}
\tag{17}
$$