Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P8724-cd

Latent space-based Bayesian approach to the NASA and DNV challenge 2025

Sangwon Lee
Department of Architecture, The University of Tokyo, JAPAN. E-mail: lee-sangwon@g.ecc.u-tokyo.ac.jp

Masaru Kitahara
Division of Civil Engineering, Hokkaido University, JAPAN. E-mail: kitahara@eng.hokudai.ac.jp

Taro Yaoyama
Department of Architecture, The University of Tokyo, JAPAN. E-mail: yaoyama@g.ecc.u-tokyo.ac.jp

Tatsuya Itoi
Department of Architecture, The University of Tokyo, JAPAN. E-mail: itoi@g.ecc.u-tokyo.ac.jp

This paper presents a latent space-based Bayesian methodology to tackle the NASA and DNV 2025 challenge on uncertainty quantification and design optimization. First, a Variational Autoencoder (VAE) is trained to investigate the distribution of the aleatory variables. From its latent-space representation, we conclude that a two-dimensional Gaussian distribution is suitable for modeling these uncertainties, thus enabling a data-driven calibration approach. Posterior estimates are obtained through a Bayesian updating procedure based on a multi-variational autoencoder (MVAE), effectively aligning simulation outcomes with observed data. Subsequently, tight prediction intervals are computed by extensive Monte Carlo simulations, demonstrating coverage of the calibration results. For the design optimization problem, a Bayesian optimization framework is employed to solve three separate tasks. In the performance-based design, the control variables are optimized to maximize an expected performance measure under the calibrated uncertainty model. In the reliability-based design, the worst-case failure probability across epistemic parameter variations is minimized using an adaptive Gaussian process modeling strategy. Finally, an ϵ-constrained design is performed using dual-GP surrogates for the objective and constraint functions, thereby balancing performance enhancement and failure probability control.

Keywords: Bayesian calibration, stochastic model updating, aleatory uncertainty, epistemic uncertainty, Bayesian optimization, latent space

1. Introduction

In this paper, authors assume that the reader is familiar with the NASA and DNV challenge on optimization under uncertainty Agrell et al. (2024). The notations for mathematical expressions follow the conventions established in the challenge document, and standard symbols are used unless additional clarification is necessary.

The challenge addresses complex systems that involve significant uncertainty, making it essential to handle both aleatory and epistemic types through a stochastic framework. Such problems frequently exhibit high-dimensional and sparse data, demanding methods that are robust to incomplete or noisy information.

In this challenge, authors primarily employ a latent space-based method Lee et al. (2024) for parameter calibration and use a Bayesian optimization approach for the optimization problems.

2. Problem 1: Uncertainty quantification

From this section onward, the components of each parameter are expressed as $\mathbf{X}_a = [a_1, a_2]$ and $\mathbf{X}_e = [e_1, e_2, e_3]$. For each simulation using local model, the seed parameter s is randomly selected as an integer in the range $[0, 1\,000\,000]$.

2.1. *Subproblem 1: uncertainty model of $\mathbf{X}_a$ and $\mathbf{X}_e$*

2.1.1. *Probabilistic model for $\mathbf{X}_a$*

In order to determine a suitable probabilistic model for the distribution f_a of the aleatory variables, a preliminary analysis is performed. A Vari-

ational Autoencoder (VAE) Kingma and Welling (2014) with a three-dimensional latent space is trained to encode both a validation dataset and the observed data for illustrative and visualization purposes. In addition, we use a VAE because it efficiently extracts latent features from high-dimensional data, capturing complex patterns without relying on restrictive assumptions. Fig. 1 shows the scatter plot of the latent space, where the colors represent the parameters a_1 and a_2, which were used during the dataset generation phase. The observed data points are plotted in black. We can see that the VAE effectively disentangles a_1 and a_2 in the latent space, suggesting that it successfully captures the characteristics of the aleatory variables.

We then conduct a naive estimation by collecting data points in the latent space those are closest to the observed data. The histograms of those samples are shown in Fig. 2. Because a_1 and a_2 appear to follow a truncated normal distribution near zero, a Gaussian distribution is adopted as the probabilistic model f_a for $\mathbf{X}_a$, as follows:

$$\mathbf{X}_a \sim f_a = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1}$$

Since $\mathbf{X}_a$ is two-dimensional, the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are expressed as:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}. \tag{2}$$

We define our calibration variable θ by combining the five hyperparameters of the aleatory variables and the three epistemic variables:

$$\theta = [\mu_1, \mu_2, \sigma_1, \sigma_2, \rho, e_1, e_2, e_3].$$

2.1.2. *Model updating with MVAE*

We use a dataset $\mathcal{D} = [\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(N_{\text{obs}})}]$ of 100 observations ($N_{\text{obs}} = 100$), which are directly taken from the example output file provided in the challenge, corresponding to the baseline design $\mathbf{X}_c = [0.533, 0.666, 0.5]$. We opted to use only the baseline design $\mathbf{X}_c$ because it provided sufficient sensitivity for estimating the aleatory variables $\mathbf{X}_a$. By contrast, the low sensitivity associated with the epistemic variables $\mathbf{X}_e$ meant that exploring alternative designs yielded no substantial improvements in $\mathbf{X}_e$ estimation or in the overall optimization process.

We address this calibration problem by employing a sequential extension of the Bayesian updating method with MVAE Suzuki and Mat-

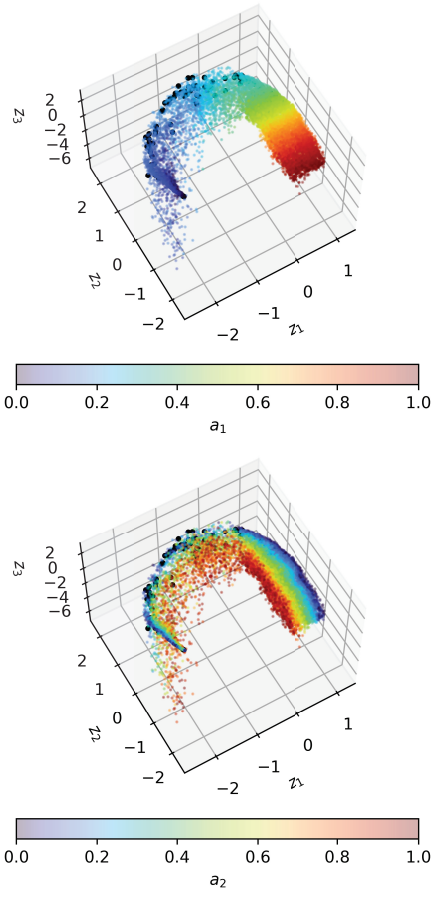

Fig. 1. Latent space of VAE

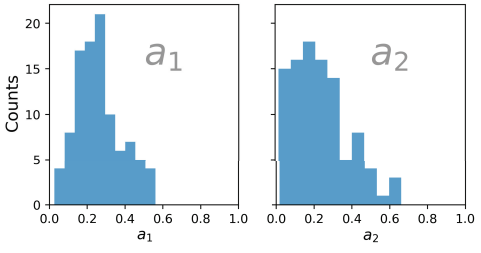

Fig. 2. Histogram of naively selected samples near the observed data in latent space

suo (2022). Fig. 3 illustrates the overall MVAE architecture, where two encoders are used: one for mapping the observed data $\mathbf{d}$ to a latent variable $\mathbf{z}$ and another for mapping the simulation-based parameters $\boldsymbol{\theta}$ to the same latent variable. Specifically, $q_d(\mathbf{z}|\mathbf{d})$ represents the approximate posterior from the data encoder, while $q_\theta(\mathbf{z}|\boldsymbol{\theta})$ represents the approximate posterior from the parameter encoder. In this framework, $\mathbf{z}$ is the shared latent variable capturing essential features for both observed data and parameter configurations.

We use the likelihood formulation from Itoi et al. (2024), which allows a nonparametric estimation of the likelihood for high-dimensional data, extended for a multi-observation multi-simulation scenario Lee et al. (2024), as shown in Eq. (3):

$$p(\mathcal{D}|\boldsymbol{\theta}) \simeq \prod_{i=1}^{N_{\text{obs}}} \sum_{j=1}^{N_{\text{sim}}} \int_{\mathcal{Z}} \frac{q_\theta(\mathbf{z}|\boldsymbol{\theta}^{(j)})\, q_d(\mathbf{z}|\mathbf{d}^{(i)})}{q_d(\mathbf{z})}\, d\mathbf{z}, \tag{3}$$

where $\mathbf{z}$ is a latent variable in a 5-dimensional latent space (chosen to match the total number of model parameters). Although we initially illustrate our approach in a 3D latent space for clarity in Section 2.1.1, we employ a 5D space in subsequent sections to ensure consistent calibration without undue computational overhead. Furthermore, $\boldsymbol{\theta}^{(j)}$ is the parameter vector that includes the randomly sampled aleatory variables derived from $\boldsymbol{\theta}$, and N_{sim} is the number of samples (set to 1000) used for the Monte Carlo integration. Finally, $\mathbf{d}^{(i)}$ denotes the i-th observation.

We employed a replica exchange Monte Carlo method Swendsen and Wang (1986) with a gradually increasing tempering parameter β. Specifically, the intermediate posterior distribution at t-th stage can be written as

$$p_t(\boldsymbol{\theta} \mid \mathcal{D}) \;\propto\; \left[p(\mathcal{D} \mid \boldsymbol{\theta})\right]^{\beta_t} p(\boldsymbol{\theta}).$$

At each stage, the MVAE is then trained on the intermediate posterior samples from the previous step, thereby iteratively refining the estimated parameter distribution.

The resulting posterior samples are subjected to kernel density estimation and normalized to a maximum value of 1, allowing them to be interpreted as fuzzy membership functions representing plausibility levelsBeer et al. (2013), as illustrated in Fig. 4. Table 1 presents the maximum a posteriori (MAP) estimates of the epistemic variables $\mathbf{X}_e$, and Table 2 shows the reduced ranges at an α-level of 0.6. In addition, Fig. 5 shows the contour of f_a derived from the MAP estimates of the aleatory hyperparameters.

Using the MAP estimates, we perform 1000 simulations in total. In Fig. 6, we display a ran-

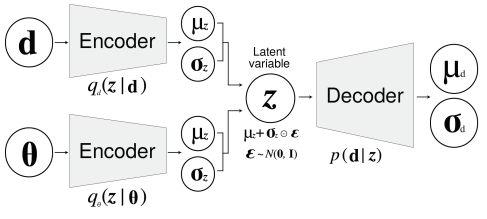

Fig. 3. Schematic of the MVAE architecture with two encoders for data $\mathbf{d}$ and parameters $\boldsymbol{\theta}$.

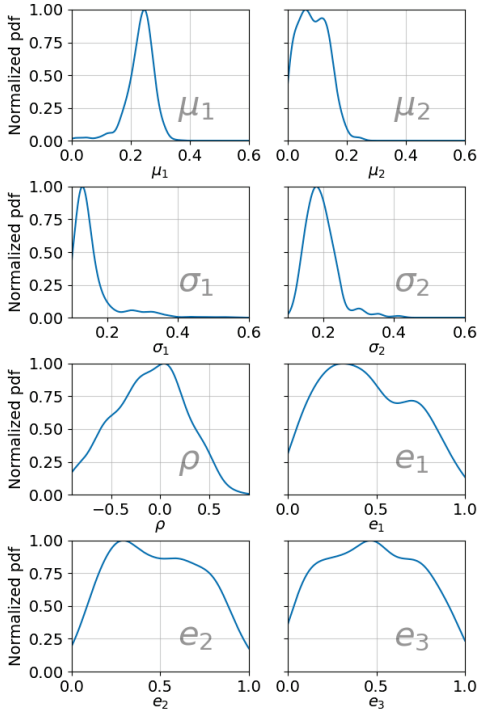

Fig. 4. Normalized pdf of posterior samples.

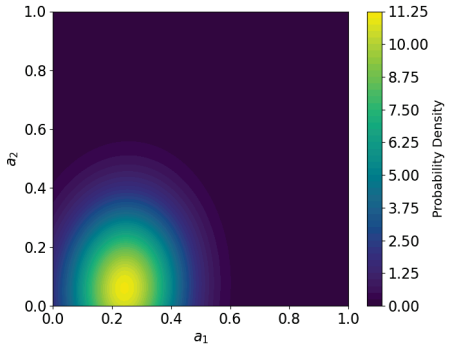

Fig. 5. Contour of aleatory variable $\mathbf{X}_a$ based on MAP estimates.

Table 1. Maximum a posteriori estimates of epistemic variables $\mathbf{X}_e$

e_1	e_2	e_3
0.3108	0.2911	0.4674

Table 2. Reduced ranges of epistemic variables $\mathbf{X}_e$

e_1	e_2	e_3
[0.0749, 0.8005]	[0.1118, 0.8498]	[0.0597, 0.8704]

dom subset of 100 of those simulations (in blue) to visually compare with the 100 observed data points (in red). n Fig. 7, we show histograms from 1000 "original" simulations (uniformly sampling $\mathbf{X}_a$ and $\mathbf{X}_e$ in $[0, 1]$) and 1000 "calibrated" simulations (using the distributions from Section 2.1.2), along with the observed data for comparison. Note that we only present the results for y_1 and y_4, as y_1–y_3 and y_4–y_6 exhibit similar trends. The results demonstrate that the Bayesian updating closely aligns the calibrated system sequences with the observed data.

2.2. *Subproblem 2: the tightest prediction intervals*

Using the f_a and E estimated above, we compute $\overline{y}$ and $\underline{y}$. Formally, they are given by:

$$\overline{y}\left(\mathbf{X}_c, \alpha\right) = \inf\left\{ u \mid \max_{\mathbf{X}_e \in E} p_u\left(\mathbf{X}_e, \mathbf{X}_c, u\right) \leq 1 - \alpha \right\} \tag{4}$$

$$\underline{y}\left(\mathbf{X}_c, \alpha\right) = \sup\left\{ l \mid \max_{\mathbf{X}_e \in E} p_l\left(\mathbf{X}_e, \mathbf{X}_c, u\right) \leq 1 - \alpha \right\} \tag{5}$$

where $p_u(\mathbf{X}_e, \mathbf{X}_c, u)$ and $p_l(\mathbf{X}_e, \mathbf{X}_c, l)$ are the probabilities that the system responses exceed the

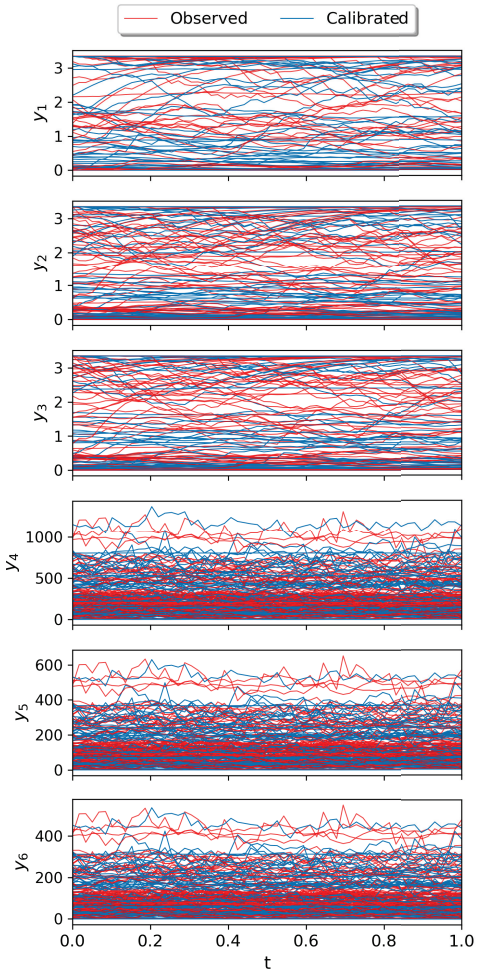

Fig. 6. Sequences of calibarated and observed data

upper limit u and fall below the lower limit l, respectively.

To reduce the computational burden, we randomly select a set E of 100 samples from $\mathbf{X}_e$ and draw 10,000 samples from the MAP distribution of f_a. We then generate Y from these parameters and extract $\overline{y}$ and $\underline{y}$. Table 3 summarizes the results.

Table 3. Tightest prediction intervals for each channel.

α	limit	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
0.95	$\overline{y}$	3.35	3.35	3.35	1078.37	546.41	529.23
	$\underline{y}$	0.00	0.00	0.00	7.03	3.54	3.49
0.999	$\overline{y}$	3.35	3.35	3.35	2184.17	1101.52	1025.04
	$\underline{y}$	0.00	0.00	0.00	0.00	0.00	0.00

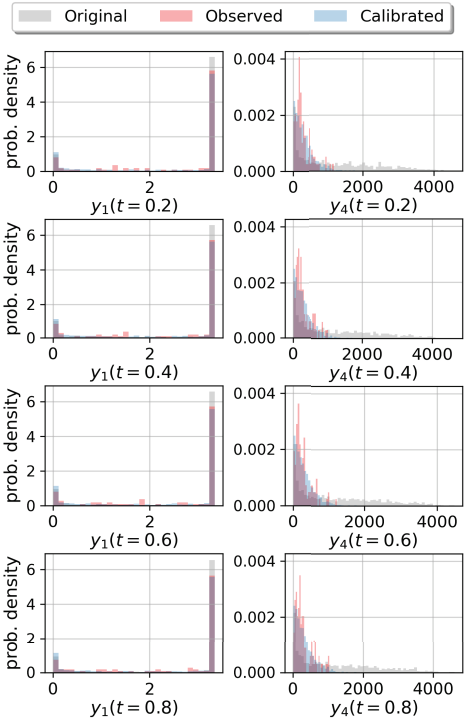

Fig. 7. Histograms of original, calibrated and observed data

3. Problem 2: Design optimization

3.1. *Performance-based design*

In this section, under the UM estimated (i.e., f_a and E), we seek a control parameter $\mathbf{X}_c$ that is optimal with respect to the following objective function:

$$h_1(\mathbf{X}_c) = \min_{\mathbf{X}_e \in E} J(\mathbf{X}_e, \mathbf{X}_c), \tag{6}$$

with

$$J(\mathbf{X}_e, \mathbf{X}_c) = \int_0^1 \mathbb{E}\left[y_i(\mathbf{X}_a,\ \mathbf{X}_e,\ \mathbf{X}_c,\ s,\ t)\right] dt \tag{7}$$

where $\mathbb{E}[\bullet]$ denotes the expected value operator with respect to $\mathbf{X}_a$ and s. This function is solved by the double-loop procedure, in which the outer loop generates 100 uniform samples of $\mathbf{X}_e$ in E, while the inner loop generates 100 samples $\mathbf{X}_a$ following f_a and s.

We use the Bayesian optimization algorithm to address the expensive-to-evaluate objective function by treating it as a black-box model that is characterized by a Gaussian process:

$$\hat{h}(\mathbf{X}_c) \sim \mathcal{GP}(m(\mathbf{X}_c),\ \kappa(\mathbf{X}_c, \mathbf{X}_c')) \tag{8}$$

where $m(\mathbf{X}_c)$ is the prior mean function, assumed to be either zero, constant or any polynomial; $\kappa(\mathbf{u}, \mathbf{u}')$ is the prior covariance function, which can be assumed to be e.g., Gaussian and Matérn kernel functions. Given a set of training data $\mathbf{D} = (\mathbf{X}_c^*, h_1(\mathbf{X}_c^*)) = \left\{\left[\mathbf{X}_c^{(j)}, h_1\left(\mathbf{X}_c^{(j)}\right)\right]\right\}_{j=1}^{n}$, the hyperparameters in the mean and covariance functions can be computed by maximizing the likelihood of the data. Then, the posterior, $\hat{h}_{\mathbf{D}}(\mathbf{X}_c)$, of $h_1(\mathbf{X}_c)$ is still Gaussian and the posterior mean $m_{\mathbf{D}}(\mathbf{X}_c)$ and covariance $\kappa_{\mathbf{D}}(\mathbf{X}_c, \mathbf{X}_c')$ are derived as Rasmussen (2003):

$$\begin{aligned} &{}_{\mathbf{D}}(\mathbf{X}_c) = m(\mathbf{X}_c) + \\ &\kappa(\mathbf{X}_c^*, \mathbf{X}_c)^T \mathbf{K}^{-1}\left(h_1(\mathbf{X}_c^*) - m(\mathbf{X}_c^*)\right) \end{aligned} \tag{9}$$

and

$$\begin{aligned} &\kappa_{\mathbf{D}}(\mathbf{X}_c, \mathbf{X}_c') = \kappa(\mathbf{X}_c, \mathbf{X}_c') - \\ &\kappa(\mathbf{X}_c^*, \mathbf{X}_c)^T \mathbf{K}^{-1} \kappa(\mathbf{X}_c^*, \mathbf{X}_c') + \\ &\mathbf{r}(\mathbf{X}_c)^T \left(\mathbf{f}(\mathbf{X}_c^*)\mathbf{K}^{-1}\mathbf{f}(\mathbf{X}_c^*)^T\right)^{-1} \mathbf{r}(\mathbf{X}_c) \end{aligned} \tag{10}$$

respectively, with $\mathbf{K} = \kappa\left(\mathbf{X}_c^*, \mathbf{X}_c^*\right)$ and $\mathbf{r}\left(\mathbf{X}_c\right) = \mathbf{f}\left(\mathbf{X}_c\right)\mathbf{K}^{-1}\kappa\left(\mathbf{X}_c^*, \mathbf{X}_c\right)$. The mean function $m_{\mathbf{D}}\left(\mathbf{X}_c^*\right)$ is a natural estimate of $h_1\left(\mathbf{X}_c\right)$, while the standard deviation function $\sigma_{\mathbf{D}}\left(\mathbf{X}_c\right) = \kappa_{\mathbf{D}}\left(\mathbf{X}_c, \mathbf{X}_c'\right)$ measures the prediction accuracy of the GP modeling.

To infer the maximum of the objective function using as few function evaluations as possible, our main concern is to effectively selects points where the objective function should be observed. Let $h_{max} = \max_{1\le j\le n} h_1\left(\mathbf{X}_c^{(j)}\right)$ be the current best solution observed so far. The improvement at the point $\mathbf{X}_c^+$ over h_{max} can be defined as Dang et al. (2022):

$$\begin{aligned} I_{max}\left(\mathbf{X}_c^+\right) &= max\left(m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) - h_{max}\ ,\ 0\right) \\ &= \begin{cases} m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) - h_{max}, & \text{if } m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) > h_{max} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

The so-called expected improvement (EI) criterion over the current maximum consists of taking expectation of $I_{max}\left(\mathbf{X}_c^+\right)$, and is derived as:

$$\begin{aligned} &EI_{max}\left(\mathbf{X}_c^+\right) = \\ &\left(m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) - h_{max}\right)\Phi\left(\frac{m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) - h_{max}}{\sigma_{\mathbf{D}}\left(\mathbf{X}_c^+\right)}\right) \\ &+ \sigma_{\mathbf{D}}\left(\mathbf{X}_c^+\right)\phi\left(\frac{m_{\mathbf{D}}\left(\mathbf{X}_c^+\right) - h_{max}}{\sigma_{\mathbf{D}}\left(\mathbf{X}_c^+\right)}\right) \end{aligned} \tag{12}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the standard normal cumulative density function and probability density function, respectively. This measures the improvement of the current best solution at the point $\mathbf{X}_c^+$, and the point that maximize $EI_{max}\left(\mathbf{X}_c^+\right)$ is added to the training data to update the GP model. This continues until our improvement becomes sufficiently small. In view of this, a stopping criterion can be given as:

$$\frac{max\left(EI_{max}\left(\mathbf{X}_c^+\right)\right)}{h_{max} - h_{min}} < \epsilon \tag{13}$$

where h_{min} is the current worst solution; ϵ is the chosen threshold. With the above procedure, a total of $n = 70$ function calls are required to obtain the optimal design variable $\mathbf{X}_c^\star$. The resulting $\mathbf{X}_c^\star$ is detailed in Table 4.

3.2. *Reliability-based design*

In Problem 2.2, we consider a reliability-based design optimization to minimize the following objective,

$$h_2(\mathbf{X}_c) := \max_{\mathbf{X}_e \in E} \text{pof}_{\text{sys}}(\mathbf{X}_e, \mathbf{X}_c) \tag{14}$$

where

$$\begin{aligned} &\text{pof}_{\text{sys}}(\mathbf{X}_e, \mathbf{X}_c) \\ &= \mathbb{P}\left[g_{\text{sys}}(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s) < 0\right] \\ &= \mathbb{P}\left[\min_{i\in I_2} g_i(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s) < 0\right]. \end{aligned} \tag{15}$$

The problem involves a double-loop procedure, wherein the outer-loop seeks the optimal $\mathbf{X}_c^\star$ and the inner-loop computes the failure probability in the RHS of Eq. (14). This incurs a prohibitive computational burden as the probability in Eq. (15) often takes a extremely small value (e.g., $< 10^{-6}$), especially around the optimal control point, requiring numerous calls to a simulator for accurate estimation. We thus adopt an adaptive GP metamodeling approach for both the outer- and inner-loop procedures. For the outer-loop, we use Bayesian optimization to approximate $h_2(\mathbf{X}_c)$ with a GP model, which is adaptively refined using the EI acquisition criterion. For the inner-loop, we replace the maximization in the RHS of Eq. (14) by the highest pof_{sys} among 50 samples of $\mathbf{X}_e$, randomly drawn within the reduced bounds obtained in Problem 1 , balancing representativeness and computational cost. The system limit state function g_{sys} under the current $\mathbf{X}_c$ is then approximated by a GP in the augumented space of $\{\mathbf{X}_a, \mathbf{X}_e\} \in \mathbb{R}^5$. The acquisition function is a so-called U function, defined as $U(\mathbf{x}) = |\widehat{g}(\mathbf{x})|/\sigma_{\widehat{g}}(\mathbf{x})$ Echard et al. (2011), where $\widehat{g}(\mathbf{x})$ and $\sigma_{\widehat{g}}(\mathbf{x})$ represent the mean and standard deviation of the current GP predictions. The failure probability in Eq. (15) is then computed by combining GP predictions and subset simulation (i.e., AK-SS algorithm Huang et al. (2016)). The sample size in each subset is set to $N_s = 50000$ and the conditional probability is $p_0 = 0.2$. We terminate the GP training when the following cri-

terion is satisfied Moustapha et al. (2022);

$$\frac{|\hat{\beta}^{(i)} - \hat{\beta}^{(i-1)}|}{\hat{\beta}^{(i)}} < \epsilon \tag{16}$$

where $\hat{\beta}^{(i)}$ is the estimated reliablity index that corresponds to the worst failure probablity at the ith step, i.e., $\max_{\mathbf{X}_e \in E} \text{pof}_{\text{sys}}^{(i)}$. We set the tolerance parameter as $\epsilon = 0.01$.

The acquired solution is detailed in Table 4 and the corresponding objective function value (the worst failure probability across varying $\mathbf{X}_e$) is 3.82×10^{-7}.

3.3. *ϵ-constrained design*

Next, we address the constrained optimization problem,

$$\begin{aligned} \mathbf{X}_c^\star &= \underset{\mathbf{X}_c}{\text{argmax}}\, h_1(\mathbf{X}_c) \\ \text{s.t.} \quad & c(\mathbf{X}_c) := \epsilon - h_2(\mathbf{X}_c) \geq 0 \end{aligned} \tag{17}$$

where $\epsilon \in [0, 1]$ denotes a predefined threshold for the failure probablity $\max_{\mathbf{X}_e \in E} \text{pof}_{\text{sys}}(\mathbf{X}_e, \mathbf{X}_c)$. In this study, we specifically consider two cases: $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$.

To solve the constrained optimization problem (17), we employ a constrained Bayesian optimization framework Gardner et al. (2014). We train two separate GPs: one for the objective function $h_1(\mathbf{X}_c)$ and the other for the constraint function $c(\mathbf{X}_c)$. These surrogate models are adaptively refined with the following acquisition function,

$$\begin{aligned} &cEI(\mathbf{X}_c) = \\ &\Phi\left(\frac{\widehat{c}(\mathbf{X}_c)}{\sigma_{\widehat{c}}(\mathbf{X}_c)}\right) \cdot \left\{ cI(\mathbf{X}_c)\Phi\left(\frac{cI(\mathbf{X}_c)}{\sigma_{\widehat{h}}(\mathbf{X}_c)}\right) + \right. \\ &\left. \sigma_{\widehat{h}}(\mathbf{X}_c)\phi\left(\frac{cI(\mathbf{X}_c)}{\sigma_{\widehat{h}}(\mathbf{X}_c)}\right) \right\} \end{aligned} \tag{18}$$

$$\begin{aligned} &cI(\mathbf{X}_c) = \\ &\mathbb{I}[c(\mathbf{X}_c) \geq 0] \cdot \max\left\{\mu_{\widehat{h}}(\mathbf{X}_c) - h_{\max}, 0\right\} \end{aligned} \tag{19}$$

where $\widehat{c}(\mathbf{X}_c)$ and $\sigma_{\widehat{c}}(\mathbf{X}_c)$ are the predictive mean and standard deviation of the GP surrogate for $c(\mathbf{X}_c)$; $\widehat{h}(\mathbf{X}_c)$ and $\sigma_{\widehat{h}}(\mathbf{X}_c)$ are the predictive mean and standard deviation of the GP surrogate for $h_1(\mathbf{X}_c)$; $\mathbb{I}[\cdot]$ is the indicator function, which equals one if the argument is true and zero otherwise; $h_{\max}$ is the best feasible value of the objective function so far.

To evaluate the constraint function $c(\mathbf{X}_c)$ efficently, we again apply the AK-SS algorithm for computing $\text{pof}_{\text{sys}}(\mathbf{X}_e, \mathbf{X}_c)$.

The optimal design for the three optimization problems—performance-based, reliability-based, and ϵ-constrained—are summarized in Table 4. Tables 5 and 6 provide the intervals of J and pof_{sys} calculated for each optimal design. These intervals were evaluated using the same set of 200×200 random combinations of $\mathbf{X}_a$ and $\mathbf{X}_e$ for all designs. The performance-based design exhibits the largest J among the three, while the reliability-based design yields the smallest pof_{sys}. Meanwhile, the ϵ-constrained design produces intermediate values for both J and pof_{sys}. These findings illustrate that each design method fulfills its respective objective in the optimization process.

Table 4. Optimal design

	c_1	c_2	c_3
Performance-based	0.752	0.274	0.734
Reliability-based	0.396	0.600	1.000
ϵ-constrained ($\epsilon = 10^{-3}$)	0.238	0.777	0.496
ϵ-constrained ($\epsilon = 10^{-4}$)	0.351	0.539	0.888

Table 5. J interval of optimal design

Performance-based	[6.010, 7.231]
Reliability-based	[5.407, 6.926]
ϵ-constrained ($\epsilon = 10^{-3}$)	[5.834, 7.212]
ϵ-constrained ($\epsilon = 10^{-4}$)	[5.503, 7.091]

4. Conclusion

A latent space-based Bayesian approach for uncertainty quantification and design optimization was presented to address the NASA and DNV 2025 challenge. Through a multi-variational Autoencoder and Bayesian updating, the model ef-

Table 6. $\mathrm{pof}_{\mathrm{sys}}$ interval of optimal design

Performance-based	[6.768e-4, 2.887e-2]
Reliability-based	[5.342e-12, 2.347e-6]
ϵ-constrained ($\epsilon = 10^{-3}$)	[3.822e-5, 8.229e-4]
ϵ-constrained ($\epsilon = 10^{-4}$)	[1.250e-8, 7.922e-6]

fectively accounted for both aleatory and epistemic uncertainties, and the calibrated model was then used for performance-based, reliability-based, and ϵ-constrained design optimization. Although this study employed a single baseline design $\mathbf{X}_c$, we plan to explore multiple $\mathbf{X}_c$ configurations to broaden the available data from real systems and further enhance $\mathbf{X}_e$-sensitivity. For example, designs that increase the range of failure probabilities or the variance of the objective function may improve both calibration and optimization. We will report these findings in subsequent work.

References

Agrell, C., L. Crespo, V. Flovik, E. Vanem, and S. Kenny (2024, 11). The nasa and dnv challenge on optimization under uncertainty. (https://uqtools.larc.nasa.gov/wp-content/uploads/sites/17/2025/01/The_NASA_and_DNV_Challenge_on_Optimization_Under_Uncertainty_10.01.pdf).

Beer, M., S. Ferson, and V. Kreinovich (2013, 5). Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing 37*, 4–29.

Dang, C., P. Wei, M. G. Faes, M. A. Valdebenito, and M. Beer (2022). Parallel adaptive bayesian quadrature for rare event estimation. *Reliability Engineering & System Safety 225*, 108621.

Echard, B., N. Gayton, and M. Lemaire (2011). Ak-mcs: An active learning reliability method combining kriging and monte carlo simulation. *Structural Safety 33*, 145–154.

Gardner, J. R., M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham (2014). Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, Beijing, China.

Huang, X., J. Chen, and H. Zhu (2016). Assessing small failure probabilities by ak–ss: An active learning method combining kriging and subset simulation. *Structural Safety 59*, 86–95.

Itoi, T., K. Amishiki, S. Lee, and T. Yaoyama (2024). Bayesian structural model updating with multimodal variational autoencoder. *Computer Methods in Applied Mechanics and Engineering 429*, 117148.

Kingma, D. P. and M. Welling (2014). Auto–encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings*.

Lee, S., T. Yaoyama, M. Kitahara, and T. Itoi (2024). Latent space-based stochastic model updating.

Lee, S., T. Yaoyama, Y. Matsumoto, T. Hida, and T. Itoi (2024). Latent space-based likelihood estimation using a single observation for bayesian updating of a nonlinear hysteretic model. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 10*(4), 04024072.

Moustapha, M., S. Marelli, and B. Sudret (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety 96*, 102174.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer.

Suzuki, M. and Y. Matsuo (2022). A survey of multimodal deep generative models. *Advanced Robotics 36*(5-6), 261–278.

Swendsen, R. H. and J. Wang (1986). Replica monte carlo simulation of spin–glasses. *Phys. Rev. Lett. 57*, 2607–2609.