The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3 – 7 June 2024, Lisboa, Portugal

# QUANTIFIED ACTIVE LEARNING KRIGING-BASED BAYESIAN UPDATING FOR HIGH-DIMENSIONAL MODELS USING CONVOLUTIONAL AUTOENCODERS

**Ioannis A. Prentzas and Michalis M. Fragiadakis**

National Technical University of Athens, 5, Iroon Polythechniou, Zografou 15773 Athens, Greece,
email: gprentzas@mail.ntua.gr, mfrag@mail.ntua.gr

**Key Words:** *Reliability analysis, Bayesian update, dimension reduction techniques, Deep learning, High-dimensional problems*

**Summary.** Bayesian Updating with Structural reliability methods (BUS) is a novel approach for calibrating models within uncertainty analysis, transforming Bayesian updating into a structural reliability problem. However, reliability analysis in the case of high-dimensional problems remains a significant challenge, especially due to the significant computational cost as a function of the desired level of accuracy. To overcome this issue, the paper introduces an innovative Bayesian updating framework based on convolutional autoencoders. This approach successfully overcomes the challenges associated with high dimensions. It first involves the training of a failure-informed convolutional autoencoder aiming to create a failure surface in a low-dimensional latent space. The dimensionality of the input space is reduced by the encoder, and the output is reconstructed using the decoder. The high-dimensional reliability problem can be handled by replacing the limit-state function in the latent space using a novel, highly efficient active learning Kriging model, known as qAK, that has been recently proposed by the authors. Therefore, an active learning technique is adopted for the training of the model in order to produce points in the vicinity of the limit state surface. This approach improves the accuracy and also expedites the model updating procedure. A high-dimensional structural example is used to demonstrate the effectiveness of the proposed method.

## 1 INTRODUCTION

Bayesian analysis is widely applied in model calibration, where uncertainties in model inputs are expressed through a probability distribution that influences the model output. Initial beliefs about input variables are captured by prior distributions, which are then updated with new data via a likelihood function, reducing uncertainties and defining the posterior distribution. In the past, the posterior distribution has commonly been estimated through the implementation of Markov chain Monte Carlo (MCMC) simulation [1]. However, MCMC sampling can struggle to reach a stable state if the acceptance rate or batch size is not properly defined. To overcome this limitation, Ching et al. [2] proposed the transitional Markov chain Monte Carlo simulation (TMCMC) that does not require a kernel density estimation in order to expand the application of Bayesian updating to high-dimensional problems. In addition, Cheung [3] et al.developed a hybrid MCMC method for high-dimensional Bayesian updating, considering a fictitious dynamic system. Additionally, Bayesian Network [4] offers another approach to model updating. Recently, Bayesian updating with structural reliability methods (BUS) [5] was introduced, linking Bayesian updating with reliability analysis and incorporating methods like subset simulation (SuS) and line sampling (LS) for enhanced efficiency.

The major advantage of BUS lies in using reliability methods to solve the Bayesian updating problem. In BUS, a performance function is used to transform Bayesian updating into a reliability analysis problem by introducing a uniformly distributed auxiliary random variable P and a constant c. An input sample X that satisfies the condition follows the posterior distribution. Furthermore, Betz et al. [6] have proposed a modified version of BUS in which the constant c is determined adaptively. A maximum likelihood-free variant of the original BUS was introduced by Betz et al. [7]. While these variants provide different ways to estimate c, they involve considerable computational cost. Thus, there is a need to develop a more efficient approach for determining c and to create a robust and effective computational framework for BUS, especially when c is not easy to determine.

Surrogate model-based reliability analysis, which uses the surrogate model as a substitute for the original performance function, has become popular in recent years due to its efficiency. Such methods can be combined with BUS to solve the reliability problem. In Bayesian model updating, Kriging has been integrated into the TMCMC [8] method to reduce the computational cost by replacing the full system simulations in likelihood evaluations. In the BUS framework, ANN-based [9] surrogate models of the likelihood function were used to reduce the number of simulations. Recently, Active learning Kriging (AK) methods [11 - 13] have been applied in the framework of BUS by calculating the constant c and adaptively selecting training samples. Moreover, AK methods encounter difficulties when handling high-dimensional problems since the size of the candidate sample increases significantly as the problem dimension also increases.

A novel framework that tackles the problem of updating in the case of high-dimensional models is proposed in this work. The paper introduces a novel Bayesian updating framework that uses convolutional autoencoders. This approach effectively overcomes issues related to high-dimensional problems by initially training a failure-informed convolutional autoencoder to construct a failure surface within the latent space. The encoder reduces the dimensionality of the input space, while the decoder is responsible for reconstructing the output. The framework is suitable for high-dimensional reliability problems by replacing the limit state function in the latent space with an innovative and highly efficient active learning Kriging model, for brevity termed as qAK (quantified Active learning Kriging) that was recently proposed by the authors [16]. The qAK method incorporates active learning to generate points near the limit state surface, thereby enhancing both the accuracy and the efficiency of the model updating process. The efficiency of this approach is demonstrated through an application to a structural dynamics problem.

The paper is organized as follows. Section 2 reviews the Bayesian updating problem and BUS. Section 3 describes the quantified active learning Kriging method. Section 4 presents the general framework of autoencoders and the convolutional autoencoders. Section 5 introduces the proposed framework for Bayesian updating for high-dimensional models and discusses implementation issues. Section 6 investigates the performance of the proposed approach through a structural dynamics problem. Finally, Section 7 summarizes the major findings of the paper.

## 2 BAYESIAN UPDATING WITH RELIABILITY METHODS

### 3.1 Bayesian updating

Bayesian model updating is an efficient tool for reducing model uncertainty by integrating new measurements that are used to update the prior distribution of input variables. Therefore, let $y = [y_1, y_2, \dots, y_m]^T$ represent the $m$ newly obtained output measurements, and $\boldsymbol{X} = [X_1, X_2, \dots, X_n]^T$ denote the $n$-dimensional input variables of the model. The prior probability density function (PDF) of $\boldsymbol{X}$ is $f_X(\boldsymbol{x})$. Using the measurements *y*, the likelihood function can be defined as follows:

$$L(\boldsymbol{x}|\boldsymbol{y}) \propto \Pr\,(\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x}) \tag{1}$$

The likelihood function $L(\boldsymbol{x}|\boldsymbol{y})$ is generally assumed to follow a Gaussian distribution. The likelihood function for the *i*-th measurement can be constructed based on the difference between the measurements and the corresponding model output.

$$L_i(\boldsymbol{x}|\boldsymbol{y}_i) = f_{\varepsilon_i}(y_i - g(\boldsymbol{x}_i)) \tag{2}$$

where $y_i$ is the measurements, $g(\boldsymbol{x}_i)$ is the model output, and $\varepsilon_i = y_i - g(\boldsymbol{x}_i)$ is commonly assumed to follow a Gaussian distribution, with $f_{\varepsilon_i}(\cdot)$ representing the PDF of the error $\varepsilon_i$ The likelihood function *L(x|y)* can be defined as *L(x|y)* = $\prod_{i=1}^{m} L_i(\boldsymbol{x}|y_i)$, where $m$ is the number of observations. The posterior PDF of $\boldsymbol{X}$ is obtained as follows:

$$f_{X|Y}\,(\boldsymbol{x}|\boldsymbol{y}) = c_E^{-1} L(\boldsymbol{x}|\boldsymbol{y}) f_{\boldsymbol{X}}(\boldsymbol{x}) \tag{3}$$

The model evidence parameter, $c_E$, is defined as follows:

$$c_E = \int_X L(\boldsymbol{x}|\boldsymbol{y}) f_{\boldsymbol{X}}(\boldsymbol{x})\, d\boldsymbol{x} \tag{4}$$

Thus, the essential aspect of Bayesian updating involves constructing the likelihood function $L(\boldsymbol{x}|\boldsymbol{y})$ using new measurements and then updating the prior distribution of the input variables in order to obtain their posterior distribution.

### 3.2 The basic principle of BUS

The BUS method shows significant potential in achieving a more accurate posterior distribution without requiring multi-dimensional (Eq. 4) integration. Moreover, the use of advanced reliability methods, such as SuS, to solve the model updating problem makes BUS highly efficient. First, a standard uniform auxiliary variable, P, is introduced. Next, BUS uses a simple rejection sampling algorithm to generate samples from the posterior distribution. The space of augmented variables in BUS is denoted as $[\boldsymbol{X}, P]$, and their joint PDF is as follows: $f_{X,P}(\boldsymbol{x}, p) = f_X(\boldsymbol{x}) f_P(p)$. The failure domain Ω is defined as:

$$\Omega = \{c \le p \cdot L(\boldsymbol{x}|\boldsymbol{y})\,\} \tag{5}$$

where $c$ is a constant satisfying $c{\cdot}L(\boldsymbol{x}|\boldsymbol{y}\ ) \leqslant 1$. The performance function can be constructed in BUS as follows:

$$h(\boldsymbol{x}, p) = p - c \cdot L(\boldsymbol{x}|\boldsymbol{y}) \tag{6}$$

According to reference [5], samples drawn from the prior distribution that fall within the domain Ω are consistent with the posterior distribution. The acceptance rate *Pacc* is the probability that the prior samples follow the posterior distribution, expressed as *Pacc* = *Pr*[*h*($\boldsymbol{X}$, *P*)⩽0].

## 3 QUANTIFIED ACTIVE LEARNING KRIGING METHOD

Let us assume a Design of Experiments (DoE), $\mathbf{S} = \left[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\mathrm{N}_0}\right]^{\mathrm{T}}$ with $\boldsymbol{x}_1 \in \mathbb{R}^n$ $(i = 1,2, \ldots, N_0)$ and $\mathbf{G} = \left[\mathbf{G}(\boldsymbol{x}_\mathbf{1}), \mathbf{G}(\boldsymbol{x}_\mathbf{2}), \ldots, \mathbf{G}\left(\boldsymbol{x}_{\boldsymbol{N}_\mathbf{0}}\right)\ \right]^{\mathbf{T}}$. $\mathbf{G}(\boldsymbol{x}_{\boldsymbol{i}}) \in \mathbb{R}^n$ are the values of the limit-state function $\boldsymbol{G}(\boldsymbol{x})$. The true response function $\boldsymbol{G}$ can be approximated by ∈ $\widehat{\boldsymbol{G}}(\boldsymbol{x})$ as follows:

$$G(\boldsymbol{x}) = \beta^T f(\mathbf{x}) + Z(\boldsymbol{x}) \tag{7}$$

where $\beta^T f(\mathbf{x})$ is the trend of a Gaussian Process (GP), and $\beta^T = \left[\beta_1, \ldots, \beta_p\right]^T$ is the regression coefficient of the basis function $f(\boldsymbol{x}) = \left[f(\boldsymbol{x}_1), \ldots, f\left(x_p\right)\right]^{\boldsymbol{T}}$ and $p$ is the number of basis functions. The term $\sigma_G(\boldsymbol{x})^2$ is the variance of $Z(\boldsymbol{x})$, which denotes a zero-mean stationary Gaussian Process that relates to a covariance matrix:

$$Cov(\boldsymbol{x}, \boldsymbol{x}') = \sigma_G(\boldsymbol{x})^2 R(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}) \tag{8}$$

where $R(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta})$ is the correlation function between points $\boldsymbol{x}$ and $\boldsymbol{x}'$and $\boldsymbol{\theta}$ is the vector with the parameters of the correlation function. A widely used autocorrelation function is the *anisotropic squared exponential model*:

$$R(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}) = exp\left[-\sum_{k=1}^{n}\left(\frac{x_k - x_k'}{\theta_k}\right)^2\right] \tag{9}$$

According to the principle of Kriging, $\mathbf{G}$ can be approximated with the PDF of a Gaussian distribution:

$$\widehat{\boldsymbol{G}}(\boldsymbol{x}) \sim N(\mu_{\widehat{\mathbf{G}}(x)}, \sigma_{\widehat{\mathbf{G}}(x)}) \tag{10}$$

In our work, a learning function that symmetrically quantifies a candidate support point is adopted. More specifically, the $\pi_q(\boldsymbol{x})$ learning function is called *the most probable misclassification function* [16], and it expresses the probability that the Kriging prediction $\widehat{G}(\boldsymbol{x})$ (Eq. 10) has been erroneously classified. For every sample $\boldsymbol{x}$ we calculate from the Kriging prediction, the mean $\mu_{\widehat{G}(x)}$ and the standard deviation $\sigma_{\widehat{G}(x)}$, and the most probable misclassification function is obtained as:

$$\pi_q(\boldsymbol{x}) = 2 \min\left(\pi(\boldsymbol{x}), 1 - \pi(\boldsymbol{x})\right) \tag{11}$$

In our work, a probabilistic-based stopping criterion [16] that is based on the learning function is adopted. First, we obtain the probability $p_q$ as:

$$p_q = \int_{R^n} \pi_q(x) f_X(\boldsymbol{x})\, d\boldsymbol{x} \tag{12}$$

The probability $p_q$ is used to identify, with a degree of approximation, the regions of integration where model uncertainties prevail. The proposed probabilistic-based stopping criterion is the ratio of $p_q$ over $p_{fe}$ (Eq. 14), thus:

$$\varepsilon_{stop} = p_q / p_{fe} \tag{13}$$

The tolerance for stopping the refinement procedure was empirically set equal to $\varepsilon = 5\%$, which offers a good balance between accuracy and efficiency. For the MCS method, the failure probability $p_{fe}$ is calculated as follows:

$$p_{fe} = \frac{1}{N}\sum_{i=1}^{N} I_F(x^{(i)}) \tag{14}$$

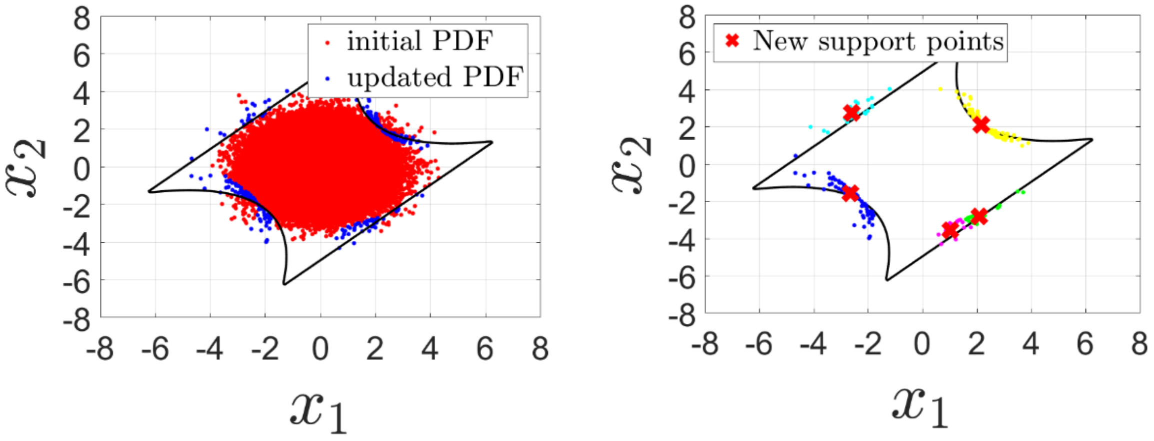
where $I_F$ is a boolean failure indicator function that is equal to one if $G(x^{(i)} \leq 0$ and zero otherwise.

The aim of the adopted strategy [16] is to increase the accuracy of the model, which is related to the capacity of the Kriging model to make accurate predictions in the vicinity of the limit-state surface $G(\boldsymbol{x}) = 0$. The metamodel will be built from the DoE population **S**, which is usually chosen with a Latin Hypercube scheme. The size of **S** depends on the number of random variables and has to be as small as possible. The algorithm initiates with the generation of a *Monte Carlo* population $\boldsymbol{X_{MCS}} = \left[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\right]^T$ in the design space. This sample will be used for selecting new training points and for calculating the failure probability. Thus, the function evaluations of the *most probable misclassification* function of Eq. 11 are equal to the size of this sample. The best candidate samples from $\boldsymbol{X_{MCS}}$ will be used to update the metamodel population **S**.

The active learning procedure that will update the vector of support points **S** is subsequently initiated. The selection of new support points requires calculating the *most probable misclassification* function $\pi_q(\boldsymbol{x})$. For sample $\boldsymbol{x}^{(i)}$, the PDF from which the candidate sample is created is updated as follows:

$$\hat{f}_X(\boldsymbol{x}) = H(w(x^{(i)}) - w_{\mathrm{lim}}\ ) \tag{15}$$

where H is the Heaviside function, **w** are weights, and $w_{lim}$ is a threshold, usually set equal to 0.1. According to Eq. 15, the updated PDF is formed by accepting points whose weights are greater than $w_{lim}$. For the MCS method, the weights are equal to the *most probable misclassification* function, i.e., $w_{lim} = \pi_q(\boldsymbol{x}^{(i)})$. The new PDF shifts the PDF to the areas that mostly affect the failure probability.



(a)

(b)

**Figure 1**: Four-branch limit-state function, (a) Sample from initial and updated PDF. (b) Data clustering and selection of new support points. The new points are shown with a red x.

Figure 1a shows the MCS candidate samples from both the initial PDF and the updated PDF. More specifically, in Fig. 1b, the red points were sampled according to the initial distribution of each random variable, while the blue points are obtained with the updated PDF. The blue points, which are close to the limit-state surface, are fewer than the red points because points with weights less than $w_{lim}$ have been rejected. Thus, points with large weight values are near the limit-state surface, while points with a weight less than $w_{lim}$ have been rejected due to their reduced proximity.

## 4 GENERAL FRAMEWORK OF AUTOENCODERS

The AE concept was introduced by Rumelhart et al. (1986), and it is regarded as a neural network that learns from an unlabeled data set in an unsupervised manner. An autoencoder (AE) is a type of neural network that is trained from unlabeled data in an unsupervised way, first introduced by Rumelhart et al. (1986). The primary goal of an AE [15] is to learn a compressed representation, known as encoding, for a given dataset, and subsequently to reconstruct the original input from this encoded version. The part of the AE responsible for this reconstruction is referred to as the decoder. The autoencoder processes an input $\boldsymbol{x}$ and maps it to $\boldsymbol{h}$, which can be expressed as $\boldsymbol{h} = \sigma(\boldsymbol{W}\ \boldsymbol{x} + \boldsymbol{b})$, where $\sigma$ is an activation function (e.g., $tanh$, $ReLU$, etc.), $\boldsymbol{W}$ is a weight matrix, and $\boldsymbol{b}$ is a bias vector. The vector $\boldsymbol{h}$ represents the encoded or latent representation of $\boldsymbol{x}$, with $\boldsymbol{H}$ being the latent or feature space. The decoder's role is to perform the inverse operation, $\psi$, to reconstruct the input $\boldsymbol{x}$ from its latent representation $\boldsymbol{h}$. The general structure and concept of an AE are illustrated in Fig. 2. AEs are typically trained using a backpropagation algorithm, which is the most widely used method for training neural networks.

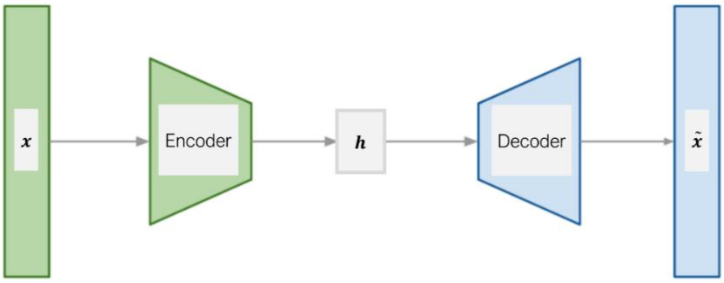


**Figure 2**: Schematic representation of an autoencoder.

Backpropagation efficiently calculates the gradient of the loss function with respect to a network's weights. Gradient-based optimization methods can be used to train multilayer neural networks by adjusting weights to minimize the loss function. For autoencoders, the loss function typically measures the reconstruction error between the input $\boldsymbol{x}_i$ and its corresponding output $\tilde{\boldsymbol{x}}_i$, often expressed as the mean squared error:

$$\mathcal{L} = \frac{1}{N}\sum_{I=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_i^{\sim}\|_2^2 \qquad (16)$$

with $\| \cdot \|^2$ denoting the $L_2$ norm and $N$ being the number of points in the training data set.

Although AEs are effective for dimensionality reduction, they encounter substantial difficulties with extremely high-dimensional inputs. To address these challenges, convolutional autoencoders (CAEs) have been developed [17]. In CAEs, the encoder is constructed using a combination of convolutional layers, fully connected layers, pooling layers, and normalization layers, while the decoder uses deconvolutional layers, unpooling layers, fully connected layers, and normalization layers.

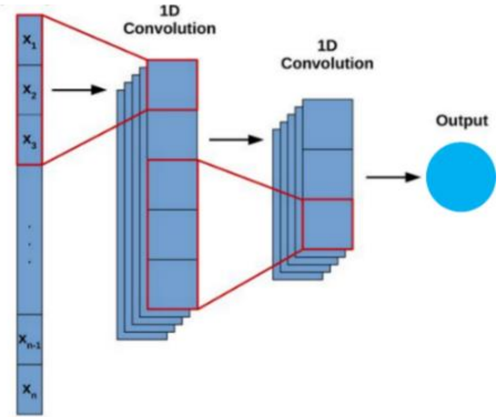


**Figure 3**: Schematic representation of a 1-D convolutional filter with stride $s$ =3.

Convolutional layers apply a filter $F$ of a defined size to the elements of array $\boldsymbol{M}$. The primary purpose of the convolution operation, as illustrated in Fig. 3, is to extract essential features from the input and utilize them for encoding. To clarify this process, consider a 1-D array $\boldsymbol{M}$ = $[m_i]$ and its encoded version $\boldsymbol{Menc}$ = $[\mu_i]$, referred to as the feature map. This feature map is generated by applying a filter $\boldsymbol{W}$ = $[w_i]$ of size f×1, moving horizontally with a stride $s$. The element $\boldsymbol{M}_{enc}$ is calculated as:

$$\mu_i = \sum_{u=1}^{f} m_{i'} \cdot w_u + b \quad with\{ i' = i \times s + u \tag{17}$$

## 5 PROPOSED FRAMEWORK

A novel framework (Fig. 4) for Bayesian updating of high-dimensional models is introduced using convolutional autoencoders. This framework reformulates the model updating problem into a structural reliability problem using BUS. Within this approach, the structural reliability problem is solved using the quantified active learning method [16], which directly influences the efficiency and precision of the updating process. Specifically, the limit-state function, initially defined through BUS, is replaced with an active learning Kriging model. The training dataset for the Kriging model includes the model's input parameters and the corresponding system outputs, which are derived from the values of the limit-state function. The constant $c$ is calculated using the Kriging model and the MCS candidate sample. The next step is to reduce the dimensionality of the input parameter vector using convolutional autoencoders. The Kriging model is then trained in the latent space. This framework enables a robust solution for updating high-dimensional models.

The process begins with defining the dimensionality of the latent space, which is a crucial step in determining how the data will be represented and compressed within the autoencoder. Next, the architecture of the autoencoder is established, defining the structure that will be used to encode and decode the data. The autoencoder is trained on the realizations of the input variables **X**, which represent the model parameters. This training enables the autoencoder to learn a compact representation of the input data in the latent space. Once the autoencoder is trained, it encodes the input data into this latent space, creating a lower-dimensional representation of the original input. Using this encoded data, a training dataset for the Kriging model is constructed. This dataset consists of inputs in the latent space and their corresponding outputs from the limit-state function.

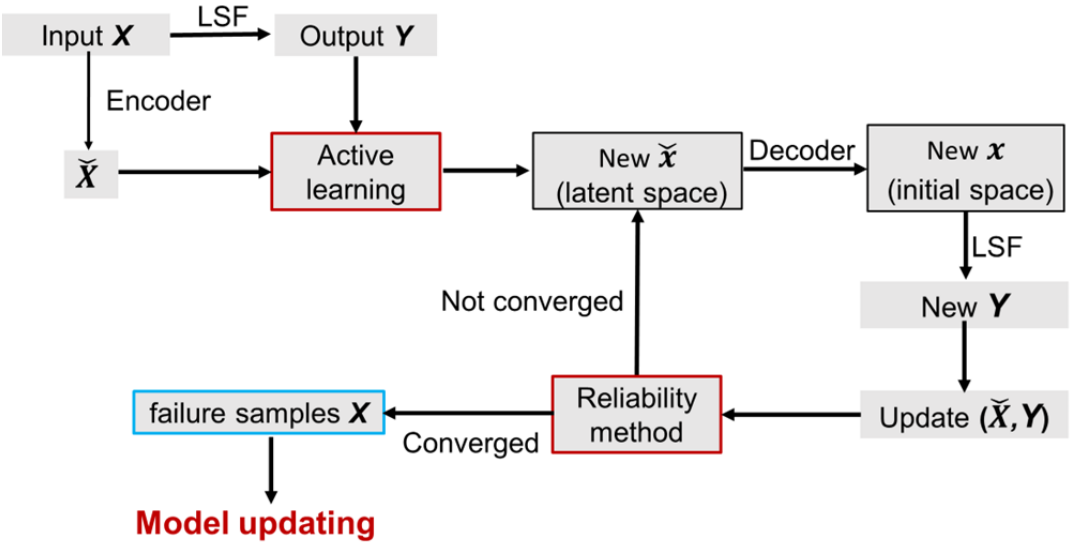


**Figure 4**: Flowchart of the proposed framework.

Active learning is then performed in the latent space to find new support points that are close to the limit-state surface. These newly identified support points are decoded back into the original input space using the decoder. For these new points, the limit-state function is evaluated to obtain the new output values for the system. The training dataset for the Kriging model is subsequently updated with these new outputs. The process of identifying new support points and updating the Kriging model continues iteratively until the probabilistic-based stopping criterion is satisfied. Using the MCS method, the failure probability is calculated, and the failure samples are defined. These failure samples are used to update the model's parameters. Finally, the mean value of these posterior samples is used to update the model. This approach allows for robust calibration of high-dimensional models.

## 6 NUMERICAL APPLICATION

A ten-story linear shear-building model (Fig. 5) has been borrowed from [3]. The identification process is based on simulated acceleration data, which are created using the El Centro ground-motion acceleration history as input. The measured response is simulated by first calculating the absolute acceleration response of the actual structure at the first and tenth floor (2 degrees of freedom) (Fig. 6), while a Gaussian discrete white noise sequence with standard deviation $\sigma$ equal to 10% of the root-mean-square value of the corresponding acceleration time histories is added. A nominal model is generated as the target model, and it can be found in [3].

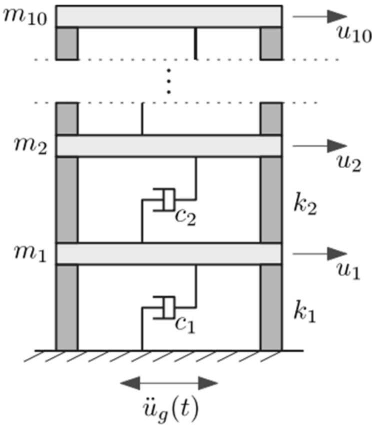


**Figure 5**: Ten-story shear building

For the model updating process, 10 model parameters are selected. They correspond to the stiffness parameters $k_i$, $i$ = 1, …, 10. The prior probability density functions for the model parameters $k_i$, where $i$ = 1, …, 10, are Gaussian distributions with mean values of $\bar{k} = 2 \times 10^7$ N/m and a coefficient of variation of 30%. The likelihood function selected is that proposed in reference [3].

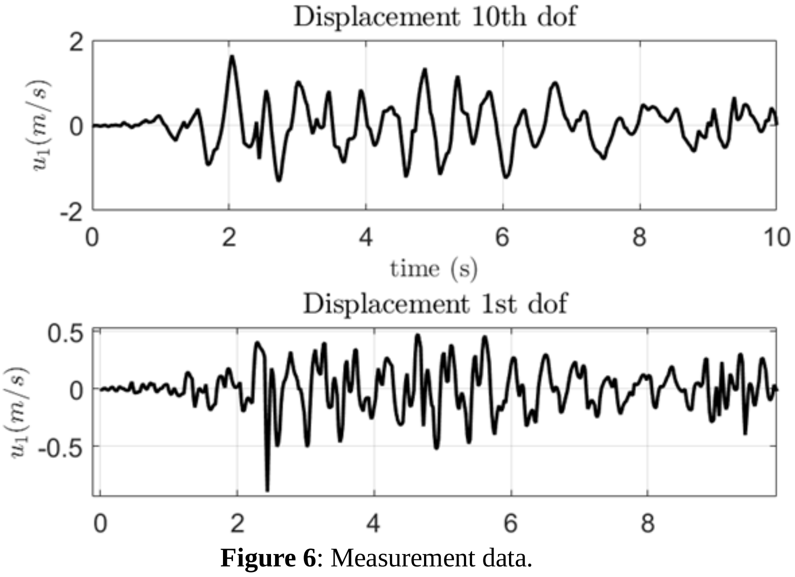


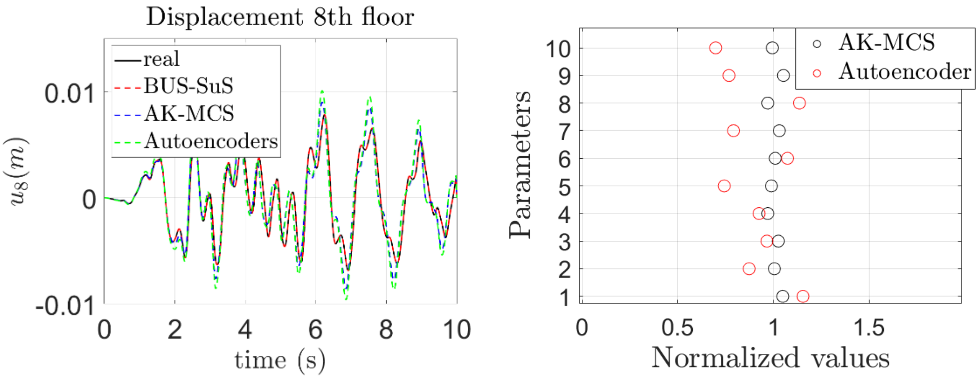
**Figure 6**: Measurement data.

Figure 7a presents the displacement time history of the 8th floor calculated with the finite element model, the active learning Kriging model, and the use of the convolutional autoencoders. The results show that responses from all the methods are very close to the real response. Moreover, using autoencoders, the number of candidate samples for active learning can be reduced by half compared to the sample size used in the active learning procedure (see

Table 1). In Fig. 7b, the normalized values (with respect to their target values, see [3]) of the model's parameters are presented for the qAK-MCS and the qAK-MCS plus autoencoders. Compared with the prior uncertainty in the structural model parameters, the posterior uncertainty is significantly reduced since the data provide relevant information about these parameters.

**Table 1**: Example of the construction of one table

| Method | No model Evaluations | Candidate Sample size |
|---|---|---|
| BUS-SuS | 5000 | - |
| qAK-MCS | 220 | 100.000 |
| Autoencoder | 220 | 50.000 |



(a) (b)

**Figure 7**: (a) displacement time-history of the 8th floor calculated with qAK-MCS and qAK-MCS plus autoencoders, (b) normalized values of the input parameters.

## 7 CONCLUSIONS

The Bayesian updating with structural reliability methods (BUS) transforms Bayesian updating into a structural reliability problem, addressing uncertainties in model calibration. High-dimensional reliability analysis remains challenging due to significant computational costs. To address this, the paper introduces a novel Bayesian updating framework using convolutional autoencoders, which reduce dimensionality and define the limit state function in a lower-dimensional latent space. This approach integrates a highly efficient active learning Kriging model (qAK) to enhance accuracy and model updating. The effectiveness of this method is demonstrated with a structural dynamics example.

## 8 ACKNOWLEDGEMENTS

The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number:04140).

## REFERENCES

[1] J.L. Beck., S.-K Au. Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation. *Journal of Engineering Mechanics*, 128 (4): 380 - 391, 2002, https://doi.org/10.1061/(ASCE)0733-9399(2002)128:4(380)

[2] J. Ching, Y.-C. Chen, Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging. *Journal of Engineering Mechanics*, 133 (7): 816 - 832, 2007, https://doi.org/10.1061/(ASCE)0733-9399(2007)133:7(816)

[3] S. H. Cheung, J. L. Beck, Bayesian Model Updating Using Hybrid Monte Carlo Simulation with Application to Structural Dynamic Models with Many Uncertain Parameters. *Journal of Engineering Mechanics*, 135 (4): 243-255. 2009, https://doi.org/10.1061/(ASCE)0733-9399(2009)135:4(243)

[4] D. Straub D., A Der Kiureghian, Bayesian network enhanced with structural reliability methods: Application. *Journal of Engineering Mechanics*, 136 (10): 1248 - 1258, 2010, https://doi.org/10.1061/(ASCE)EM.1943-7889.0000173

[5] D. Straub, I. Papaioannou, Bayesian Updating with Structural Reliability Methods. *Journal of Engineering Mechanics*, 141 (3) 04014134, 2015, https://doi.org/10.1061/(ASCE)EM.1943-7889.0000839.

[6] W. Betz, I. Papaioannou, J. L. Beck, D. Straub, Bayesian inference with Subset Simulation: Strategies and improvements. *Computer Methods in Applied Mechanics and Engineering*, 331: 72 - 93, 2018, https://doi.org/10.1016/j.cma.2017.11.021.

[7] W. Betz, J. L. Beck, I. Papaioannou, D. Straub, Bayesian inference with reliability methods without knowing the maximum of the likelihood function. *Probabilistic Engineering Mechanics*, 53: 14-22, 2018, https://doi.org/10.1016/j.probengmech.2018.03.004.

[8] P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, X-TMCMC: Adaptive kriging for Bayesian inverse modeling. *Computer Methods in Applied Mechanics and Engineering*, 289: 409-428, 2015, https://doi.org/10.1016/j.cma.2015.01.015.

[9] D. G. Giovanis, I. Papaioannou, D. Straub, V. Papadopoulos, Bayesian updating with subset simulation using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 319: 124-145, 2017, https://doi.org/10.1016/j.cma.2017.02.025.

[11] C. Song, Z. Wang, A. Shafieezadeh, R. Xiao, BUAK-AIS: Efficient Bayesian Updating with Active learning Kriging-based Adaptive Importance Sampling. *Computer Methods in Applied Mechanics and Engineering*, 391: 114578, 2022, https://doi.org/10.1016/j.cma.2022.114578.

[12]. X. Jiang, Z. Lu, Adaptive Kriging-based Bayesian updating of model and reliability,

*Structural Safety*, 104, 2023, 102362, https://doi.org/10.1016/j.strusafe.2023.102362.

[13] K. Feng, Z. Lu, J. Wang, P. He, Y. Dai, Efficient reliability updating methods based on Bayesian inference and sequential learning Kriging. *Structural Safety*, 104: 102366, 2023, https://doi.org/10.1016/j.strusafe.2023.102366.

[14] B. Echard, N. Gayton, M. Lemaire, AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, 33 (2): 145-154, 2011, https://doi.org/10.1016/j.strusafe.2011.01.002.

[15] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature*, 323: 533–536, 1986. https://doi.org/10.1038/323533a0

[16] I. Prentzas, M. Fragiadakis, Quantified Active Learning Kriging Model for Structural Reliability Analysis. *Available at http://dx.doi.org/10.2139/ssrn.4827814*

[17] Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., 2011. Stacked convolutional autoencoders for hierarchical feature extraction. *In: Proc. 21st International Conference On Artificial Neural Networks. pp. 52–59*. http://dx.doi.org/10.5555/2029556. 2029563