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

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P0223-cd

Axtreme: A package for Bayesian surrogate modeling and optimization of extreme response calculation

Sebastian Winter

Group Research and Development, DNV, Norway E-mail: sebastian.winter@dnv.com

Christian Agrell

Group Research and Development, DNV, Norway E-mail: christian.agrell@dnv.com

Engineers often need to understand long-term behavior of complex models in stochastic settings, such as when performing Ultimate Limit State (ULS) calculations. Models are expensive to run (in runtime or resources), meaning directly calculating the value of interest is often infeasible. Bayesian surrogate modeling with Design of Experiments (DOE) is one approach to this computational challenge and offers advantages over traditional methods such as environmental contours. However, despite its advantages, the adoption of Bayesian surrogate modeling with DOE in engineering has been limited, due in part to the technical expertise required to implement the methods.
To address this, we have developed Axtreme, an open-source Python package extending state-of-the-art Bayesian optimization frameworks for these engineering challenges. Axtreme enables engineers to build accurate surrogate models, compute quantities of interest, and conduct DOE to minimize uncertainty in these calculations efficiently. The package provides a flexible toolkit of ready-to-use functions, helpers, and tutorials, all built on top of robust, industrial-grade frameworks. By reducing the technical barriers to applying Bayesian surrogate modeling and DOE, this package makes advanced uncertainty quantification techniques more accessible, improving decision-making and design efficiency for engineers.
In this paper, we introduce the Axtreme package and demonstrate the application on a numerical example.

Keywords: Bayesian surrogate modeling, Gaussian Processes, Design of experiments (DOE), Active Learning, Uncertainty Quantification.

1. Introduction

Structural reliability analysis is an essential step in designing a wide variety of structures, such as ships, buildings, and offshore structures. The goal of structural reliability analysis is to determine if a given structure is sufficiently strong for the task and environment it is designed for. In other words, it determines if the structure can withstand the forces of the environment for the required length of time.

Structural reliability analysis is comprised of a variety of calculations and tests, but a common task is the need to understand the extreme (largest) responses a structure is likely to experience over a given time period (e.g., the structure's lifetime). A response is how the structure reacts to the load placed upon it. For example, the response could be the stress and strain, or the bending moment, experienced by a wind turbine (i.e., the structure) at a certain location due to wind and waves (i.e., the environmental conditions).

This can be formalized as follows[a]:

- The environmental conditions are modeled as a piece-wise stationary stochastic process, with each piece being of duration T_s. A single piece is parameterized by $\mathbf{x}$, and the behavior within the piece is refereed to as "short-term" behavior. $\mathbf{X}$ is the distribution of the parameters ($\mathbf{x} \in \mathbf{X}$), and is called the long-term environment. "Long-term" refers to behavior across pieces of the stochastic process. As a general example, $\mathbf{x}$ may be the average wind speed for one-hour, and $\mathbf{X}$ is the distribution of the one hour conditions that represents the environment the

[a] Random variables use upper case, fixed realizations use lower case, and variables in bold are vectors.

structure is in. $\mathbf{X}$ is a continuous random vector where $\mathbf{X} \in \mathbb{R}^N$ and has joint probability density $f_{\mathbf{x}}(\mathbf{x})$.

- $Y|\mathbf{x}$ is the short-term conditional response of the structure (also written as $Y(\mathbf{x})$). For a *given* long-term environmental realization $\mathbf{x}$, the maximum physical response within a time interval of length T_s is a random variable $Y|\mathbf{x}$ with probability density $g_{y|\mathbf{x}}(y|\mathbf{x})$.
- The marginal maximum physical response within a time interval of length T_s is denoted Y. For a *random* long-term environment $\mathbf{X}$, Y has density

$$g(y) = \int g_{y|\mathbf{x}}(y|\mathbf{x}) f_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x}. \quad (1)$$

- The long-term N_y-year extreme structural response is Y_{N_y}. If T_s is defined in terms of hours, then let $N = \lceil N_Y \cdot 365.25 \cdot 24/T_s \rceil$. The N_y-year extreme response is defined as the maximum

$$Y_{N_y} = \max\{Y_1, \ldots, Y_N\} \quad (2)$$

  where $Y_1, \ldots, Y_N$ are i.i.d. with probability density $g(y)$.
- $g_{N_y}(y)$ is the probability density of Y_{N_y}. This is referred to as the Extreme Response Distribution (ERD).
- The Quanitity of Interest (QOI) z is some statistic (e.g., median) of Y_{N_y}.

In summary, the relevant variables are:

- $\mathbf{X}$: Long-term environmental conditions.
- $Y|\mathbf{x}$: Short-term conditional structural response.
- $g(y)$: Short-term marginal response distribution
- $g_{N_y}(y)$: the Extreme Response Distribution (ERD).
- z: The Quantity of Interest (QOI).

Extreme response behavior is influenced by variability of long-term environmental conditions (i.e., an extreme $\mathbf{x}$ value) and short-term variability of the response (i.e., an extreme $Y|\mathbf{x}$ realization).

In the context of engineering, accurate but slow simulators (such as finite element models) can provide samples of $Y|\mathbf{x}$. The expense (time or resources) of these simulators precludes directly calculating QOIs using brute-force Monte Carlo simulation. Instead, approximate methods must be used. Approximate methods essentially make use of two different types of approximation, simulator approximation and environment approximation Wang et al. (2024). Environment approximation aims to establish extreme response estimates by using the simulator on a limited number of environment samples. Environmental contours is one such technique widely used within the industry, particularly in ocean engineering DNV (2021). A central assumption of this method is that extreme responses occur in the most extreme environments. It can be difficult to determine to what extent this assumption holds, so additional conservatism is often added based on domain expertise Wang et al. (2024). Alternatively, one may keep the full environmental distribution $\mathbf{X}$ and instead approximate the structural response, irrespective of whether the extreme responses are driven by long-term environment variability or short-term conditional response variability (or some combination of the two). By replacing the model $Y(\mathbf{x})$ with a computationally efficient alternative one can then make use of methods that need to evaluate $Y(\mathbf{x})$ a large number of times, such as sampling-based methods. Such approximations are often called response surface models, surrogate models, or emulators. Gaussian Process models are a popular alternative (Gramstad et al., 2020).

While surrogate approaches are robust to different underlying drivers of extreme responses, the use of a surrogate model introduces new challenges. Firstly, the quality of the surrogate fit must be determined, and standard quality of fit metrics such as Mean Squared Error are typically not sufficient. This is because small surrogate inaccuracies in regions important to the QOI calculation can have a far greater impact than large inaccuracies elsewhere. Secondly, data must be collected from the simulator to train the surrogate model. As the simulator is expensive, data should be collected from regions that contribute most to uncertainty in the QOI. As mentioned, these regions are not necessarily those where the surrogate has the greatest uncertainty.

Gaussian Process (GP) is a type of surrogate model that is well suited to the challenges outlined above. GPs are probabilistic, and provide estimates of the possible error in the surrogate prediction for any input. Additionally, GPs are widely used to optimally select additional training data through Design of Experiments (DOE). While GPs have been increasingly applied to extreme response problems (see Moustapha et al. (2022) for a recent survey), the majority of work has primarily been focused on deterministic simulators and responses. Treating $Y|\mathbf{x}$ as a random variable requires additional considerations to account for the stochasticity. Gond and Pan (2022) considers a simulator with heteroscedastic Gaussian noise, and heteroscedastic non-Gaussian noise is considered in Mohamad and Sapsis (2018); Wang et al. (2014); Gramstad et al. (2020); Wang et al. (2024).

The Axtreme package has been developed to facilitate the adoption of Bayesian surrogate modeling and DOE methods by industry. It provides an industrial-grade tool for performing QOI estimation and DOE, in a way that is applicable to extreme response estimation. The following section (Section 2) provides an overview of the package, while Section 3 demonstrates the use of the package on a toy problem, and finally in Section 4 we conclude and summarize the work.

2. Axtreme

The Axtreme (Ax for Extremes) package has been developed to lower the technical barrier of using GPs and DOE for extreme response calculations [b]. It aspires to simplify the process of applying these methods, empowering non-specialist engineers to benefit from the potential reduction in conservatism the methods offer. It is designed as a modular and extensible toolkit that can be run at an industrial scale, and can be adapted and extended to different types of extreme response problems. Conceptually Axtreme divides the calculation process into the following steps:

(1) Build a probabilistic surrogate model $\hat{Y}|\mathbf{x}$.
(2) Estimate the QOI $\hat{z}$ using $\hat{Y}|\mathbf{x}$.
(3) Reduce QOI uncertainty through DOE.

Axtreme's key contribution is that it facilitates the above using established industry-grade software. Axtreme is built on top of the Ax package and it extends Ax for use on extreme response calculations. Ax is an open source industry-grade adaptive experimentation platform which offers an established interface and approach for many components of DOE orchestration (Bakshy et al., 2018). While extensive, Ax is designed primarily for A/B testing and does not support extreme response calculations directly. In general, extreme response problems often use fundamentally different acquisition functions which are not part of standard Bayesian optimization packages.

2.1. *Build a probabilistic surrogate model*

The surrogate model should be probabilistic and capable of representing outputs with non-Gaussian error. Axtreme implements support for this following the approach detailed in Winter et al. (2025). This involves describing the surrogate model ($\hat{Y}|\mathbf{x}$) as a parametric distribution with parameters $\theta(\mathbf{x})$. For example, this distribution be a Gumbel, making $\theta(\mathbf{x}) = (\text{location}(\mathbf{x}), \text{scale}(\mathbf{x}))$. A GP $\hat{\theta}_k(\mathbf{x})$ is used to estimate $\theta(\mathbf{x})$, where k is the number of observations in the dataset D_k used to train the GP. Figure 5 from Section 3.2 demonstrates the use of the surrogate $\hat{Y}|\mathbf{x}$ to make a prediction at a point, and compares it to the true distribution of the simulator $Y|\mathbf{x}$.

While this is the default approach used by Axtreme, the package is flexible enough to be extended to alternate surrogate representation should they be required.

2.2. *Estimate the QOI*

Axtreme establishes the interface `QoIEstimator` for estimating the QOI using a surrogate model. It expects a surrogate model as input, and returns a list of estimates of the QOI, where each estimate is produced using a

[b]Code can be accessed through Winter et al. (2025) or https://github.com/dnv- opensource/axtreme

different posterior sample of the surrogate model. Algorithm 1 presents a high-level implementation of the interface [c].

The following implementations of this interface are provided:

- `GPBruteforce` uses the surrogate model in place of the simulator and runs a brute-force Monte-Carlo estimation of the QOI. This general method is expected to be a useful foundation for custom QOI functions.
- `MarginalCDFExtrapolation` implements the QOI method described in Winter et al. (2025). This can make use of importance sampling and unscented transforms to improve sample efficiency. The runtime of this method is independent of N (the number of time steps considered for a single extreme response) which allows fast estimation for large N.

It is expected that users may need to create custom implementations of `QoIEstimator` if their QOI is not supported by the methods provided. To facilitate this, modular helpers are provided to assist with common QOI calculation tasks such as integrating environment data, sampling posteriors, and performing importance sampling.

2.3. *Reduce QOI uncertainty through DOE*

Axtreme implements a generic acquisition function for extreme response calculation following the method detailed in Winter et al. (2025). This involves quantifying the uncertainty in the QOI, denoted as H_k, under the probability measure P_k [d].

$$H_k = var_{P_k}[\hat{z}]. \tag{3}$$

Here, the estimate $\hat{z}$ of z is uncertain because we are using a surrogate model $\hat{Y}(\mathbf{x})$ in place of $Y(\mathbf{x})$. As the surrogate model $\hat{Y}(\mathbf{x})$ is probabilistic, we can propagate uncertainty in $\hat{Y}(\mathbf{x})$ to the corresponding uncertainty in $\hat{z}$. H_k represents the variance of $\hat{z}$ when $\hat{Y}(\mathbf{x})$ is a surrogate model created using a dataset D_k of k observations (i.e., k experiments). One observation in D_k is obtained by running the simulator $Y(\mathbf{x})$ m times at an input $\mathbf{x}$ of our choice, and the sequence of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_k$ is determined by the DOE strategy. The DOE strategy is to let $\mathbf{x}_{k+1}$ be the input of the experiment that reduces H_{k+1} the most in expectation. The 'in expectation' is needed as H_{k+1} is uncertain before the new experiment number $k+1$ has been done. Formally, we write

$$s(\mathbf{x}) = E_{P_k}[H_{k+1}], \tag{4}$$

where H_{k+1} depends on $\mathbf{x}$ through D_{k+1}, and E_{P_k} is the probability measure obtained by the surrogate model after k experiments. We call $s(\mathbf{x})$ the *acquisition function*, and the DOE strategy is to select the next experimental input $\mathbf{x}_{k+1}$ such that

$$\mathbf{x}_{k+1} \in \arg\min s(\mathbf{x}) \tag{5}$$

Algorithm 2 presents an overview of the acquisition function, Algorithm 3 presents an overview of the full DOE process. For details on how to compute $s(\mathbf{x})$ we refer to Winter et al. (2025).

3. Example

The following example shows the use of the Axtreme package to solve a toy extreme response problem. The QOI calculated is a component used in ULS calculations. The complete example can be found in the tutorials provided with the package.

3.1. *Problem inputs*

3.1.1. *Environment samples*

The environment data used consists of samples from a two dimensional multivariate normal distribution (see Figure 1). We treat these samples as independent (as is commonly done in offshore engineering), but it is worth noting the helper functions provided in Axtreme also allow for fine-grain control of sampling. This can be used to

[c] The Python interface definition can be found at https://github.com/dnv-opensource/axtreme/blob/main/src/axtreme/qoi/qoi_estimator.py

[d] The surrogate model, through the GP $\hat{\theta}_k(x)$, serves as the probability measure P_k

Algorithm 1 Example QoIEstimator implementation

```
Require: GP: Gaussian process fitted to available simulator data
function EXAMPLEQOIESTIMATOR(GP)
    qoi_estimates ← empty list
    ▷ environment_data contains many data points ◁
    θ_posterior ← GP.PREDICT(environment_data)
    for all sample ∈ SAMPLE(θ_posterior) do
        ▷ a sample contains a θ for every environment data point ◁
        response_distributions ← DISTRIBUTION(sample)
        ▷ qoi_estimate defines how the QOI is calculated if the response distribution is known ◁
        estimate ← QOI_ESTIMATE(response_distributions)
        qoi_estimates.append(estimate)
    return qoi_estimates
```

Algorithm 2 Acquisition Function

```
Require: x: a candidate point (within the environment)
function ACQUISITIONFUNCTION(x)
    qoi_estimates ← empty list
    θ_ests ← ESTIMATE EXPERIMENT RESULT(GP, x)
    for all θ ∈ θ_ests do
        GP_new ← UPDATE_GP(GP, x, θ)
        estimate ← QOIESTIMATOR(GP_new)
        qoi_estimates.append(VARIANCE(estimate))
    estimated_variance ← MEAN(qoi_estimates)
    return estimated_variance
```

implement more advanced sampling regimes such as those that consider serial correlation.

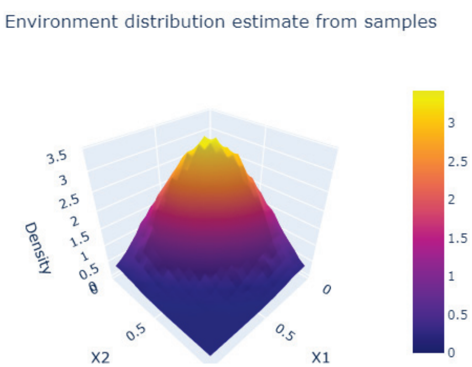

Fig. 1. KDE of environment samples.

3.1.2. *Simulator*

The toy simulator consists of two functions, which parameterize the location and scale argument of a Gumbel distribution. Running the simulator at a point **x** consists of running the location and scale function at this point, creating a Gumbel distribution from the output, and drawing a sample from this distribution. Figure 2 shows the underlying location function, Figure 3 shows the underling scale function, and 4 visualizes the resulting simulator.

3.2. *Build a probabilistic surrogate model*

The problem must first be defined as an Ax `Experiment` object. This involves providing the simulator, defining the search space, and defining the distribution that should be used to model the simulator's point-wise output.

Algorithm 3 Single DOE iteration

1: Train GP on available simulator data D_k
2: $qoi\_estimates \leftarrow$ QOIESTIMATOR(GP)
3: **if** VARIANCE$(qoi_estimates)$ is sufficiently small **then**
4: **return** $qoi\_estimates$
5: **else**
6: $\mathbf{x}_{k+1} \leftarrow \arg\min$ ACQUISITIONFUNCTION(x)
7: $\theta_{k+1} \leftarrow$ SIMULATOR$(\mathbf{x}_{k+1})$
8: $D_{k+1} \leftarrow D_k \cup (x_{k+1}, \theta_{k+1})$
9: Repeat from step 1 using the new dataset D_{k+1}

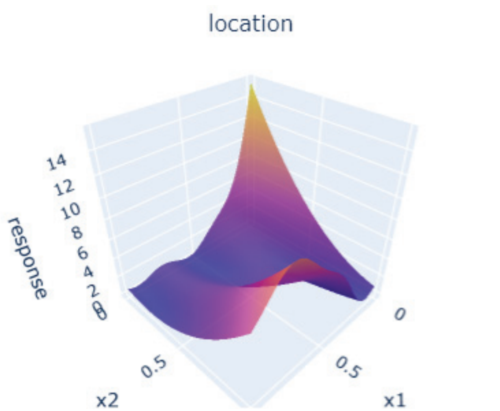

Fig. 2. Location function. This function determines the location parameter of the simulator's Gumbel distribution.

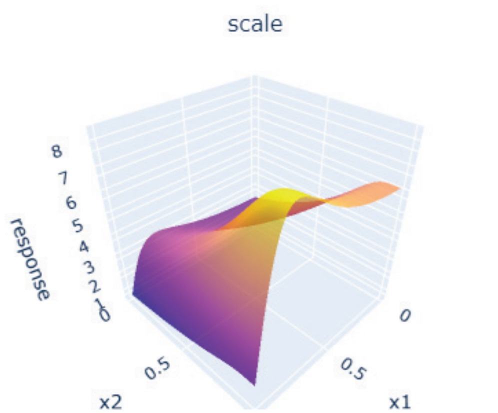

Fig. 3. Scale function. This function determines the scale parameter of the simulator's Gumbel distribution.

```
exp = make_experiment(
    simulator,
    search_space,
    output_distribution
)
```

Once the experiment has been defined a model can be generated using the following standard Ax code.

```
model = Models.BOTORCH_MODULAR(
    experiment=exp,
    data=exp.fetch_data()
)
```

Figure 5 demonstrates the use of the surrogate to make a prediction at a point, and compares it to the true distribution of the simulator.

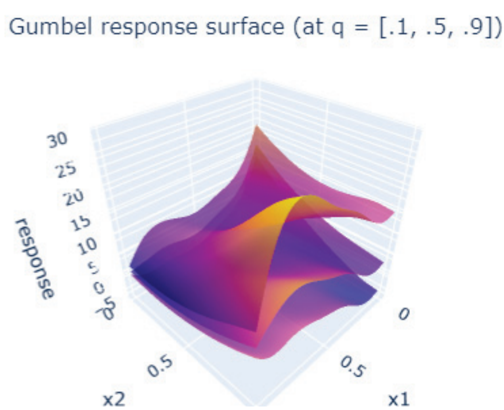

Fig. 4. Simulator behavior over environment space.

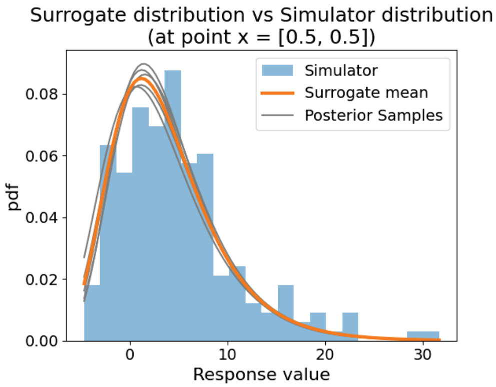

Fig. 5. Simulator and surrogate prediction at a point. The posterior samples produce other distributions the surrogate deems probable (i.e. they show the surrogate's uncertainty).

3.3. *Calculate the QOI*

The QOI is estimated using a provided QOI estimator. Once the environment samples and problem specifics are provided to the estimator it can calculate the QOI for any surrogate model.

```
estimator=MarginalCDFExtrapolation(
    # Environment samples
    env_iterable=dataloader,
    # number of samples N
    period_len= N ,
    # The QoI of the ERD
    quantile=torch.tensor(0.5),
)
estiamtes = estimator(model)
```

Figure 6 shows the QoI estimate distribution generated. The impact of increasing the amount of training data used by the model is clearly evident.

3.3.1. *Reduce QOI uncertainty through DOE*

Providing additional training data to the surrogate model reduces its uncertainty, which in turn reduce uncertainty in the QOI, as demonstrated in Figure 6.

The acquisition function for Section 2.3 is used to guide the DOE process, intelligently choosing new data points. Figure 7 shows the performance of this method (named "Look-ahead Acquisition") relative to a space-filling baseline ("Sobol Acquisition").

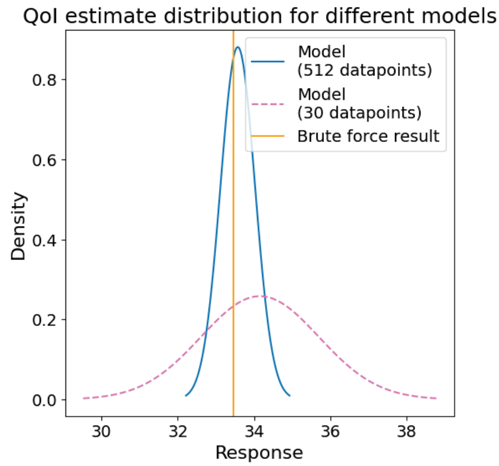

Fig. 6. QOI estimate distribution with additional training data.

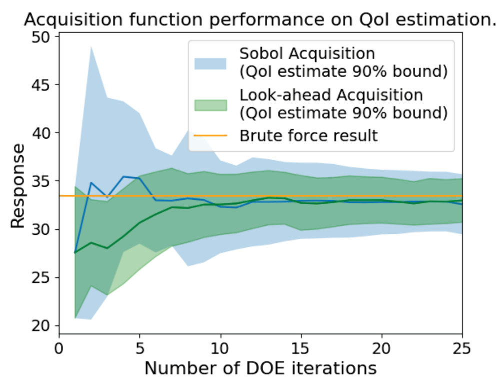

Fig. 7. Look-ahead acquisition function vs. Sobol baseline

4. Conclusion

Axtreme lowers the barrier for industry adoption of surrogate-based approaches to extreme response calculations and makes the associated benefits such as robustness and reduced conservatism more accessible. Axtreme facilitates adoption by integrating the unique requirements of extreme response calculations into the established, industry-grade, adaptive experimentation platform Ax. Specifically, this includes support for producing surrogates of stochastic black-box functions with non-Gaussian noise, and acquisition functions specific to extreme response problems.

Axtreme is expected to perform well with higher dimensional inputs, especially when compared to environmental contours (which is typically infeasible with more than 3 features). QOI calculations can use importance sampling to efficiently explore the input space, and GPs (with appropriate kernels) scale to higher dimensional inputs. We also expect a larger difference between the DOE method and space-filling baselines in high dimensions. Future work should validate this empirically by applying Axtreme to a variety of industry use cases.

Axtreme represents a concrete first step toward a larger aspiration of creating a conduit through which specialist knowledge of Bayesian optimization and extreme response calculation can scale to everyday industry impact.

The package can be accessed through Winter et al. (2025) or https://github.com/dnv- opensource/axtreme.

Acknowledgement

The Axtreme package has been developed as part of the NFR funded project RaPiD (Reciprocal Physics-based and Data-driven models, grant no. 313909).

References

Bakshy, E., L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh (2018). Ae: A domain-agnostic platform for adaptive experimentation. In *Proceedings of 32nd Conference on Neural Information Processing Systems (NIPS 2018)*, Montréal, Canada.

DNV (2021). *Environmental Conditions and Environmental Loads* (September 2019 ed.). DNV. DNV-RP-C205.

Gond, X. and Y. Pan (2022). Sequential Bayesian experimental design for estimation of extreme-event probability in stochastic input-to-response systems. *Computer methods in applied mechanics and engineering 395*, 114979.

Gramstad, O., C. Agrell, E. Bitner-Gregersen, B. Guo, E. Ruth, and E. Vanem (2020). Sequential sampling method using gaussian process regression for estimating extreme structural response. *Marine Structures 72*, 102780.

Mohamad, M. A. and T. P. Sapsis (2018). Sequential sampling strategy for extreme event statistics in nonlinear dynamical systems. *Proceedings of the National Academy of Science of the United States of America 115*(44), 11138–11143.

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

Wang, A., S. Huang, and N. Barltrop (2014, 06). Long term extreme analysis of fpso mooring systems based on kriging metamodel. Volume Volume 1B: Offshore Technology of *International Conference on Offshore Mechanics and Arctic Engineering*, pp. V01BT01A052.

Wang, H., O. Gramstad, S. Schär, S. Marelli, and E. Vanem (2024). Comparison of probabilistic structural reliability methods for ultimate limit state assessment of wind turbines. *Structural Safety 111*, 102502.

Winter, S., C. Agrell, J. C. G. Gómez, and E. Vanem (2025). Efficient long-term structural reliability estimation with non-gaussian stochastic models: A design of experiments approach.

Winter, S., K. Skare, and M. Kristiansen (2025). Axtreme. https://github.com/dnv-opensource/axtreme V0.1.1.