Probabilistic Metamodels for an Efficient Characterization of Complex Driving Scenarios

Max Winkelmann\textsuperscript{O}, Mike Kohlhoff, Hadj Hamma Tadjine, Senior Member, IEEE, and Steffen Müller\textsuperscript{O}

Abstract—To validate the safety of automated vehicles (AV), scenario-based testing aims to systematically describe driving scenarios an AV might encounter. In this process, continuous inputs such as velocities result in an infinite number of possible variations of a scenario. Thus, metamodels are used to perform analyses or to select specific variations for examination. However, despite the safety criticality of AV testing, metamodels are usually seen as a part of an overall approach, and their predictions are not questioned. This paper analyzes the predictive performance of Gaussian processes (GP), deep Gaussian processes, extra-trees, and Bayesian neural networks (BNN), considering four scenarios with 5 to 20 inputs. Building on this, an iterative approach is introduced and evaluated, which allows to efficiently select test cases for common analysis tasks. The results show that regarding predictive performance, the appropriate selection of test cases is more important than the choice of metamodels. However, the choice of metamodels remains crucial: Their great flexibility allows BNNs to benefit from large amounts of data and to model even the most complex scenarios. In contrast, less flexible models like GPs convince with higher reliability. Hence, relevant test cases are best explored using scalable virtual test setups and flexible models. Subsequently, more realistic test setups and more reliable models can be used for targeted testing and validation.

Index Terms—Automated driving, Bayesian neural networks, deep Gaussian processes, probabilistic metamodels, safety validation, scenario-based testing.

I. INTRODUCTION

The increasing automation of road transport promises increased sustainability, more cost-effective mobility, and, above all, greater safety. However, various accidents tragically demonstrated how automated vehicles (AV) can cause fatalities within seconds if they do not sufficiently understand their surroundings [1]. Thus, the validation of AVs’ safety is in public focus and paramount for their release.

Traditionally, driver assistance systems (SAE level 1-2) are tested using manually defined test cases. This is possible since the systems’ operational design domain (ODD) is clearly restricted, and the driver ensures that the system is only used within its ODD or takes control. However, for AV (SAE level 3-5), monitoring by the driver is omitted, and the ODD is less restricted. Hence, it is crucial that tests sufficiently cover the ODD’s conditions. A methodologically simple approach to this is to have an AV follow representative routes of human drivers until it is statistically proven that the AV causes fewer fatalities than human drivers. However, this approach not only endangers other road users but requires prohibitive amounts of resources [2]. This is mainly because representative driving is not informative since critical events (e.g., fatalities) are rare.

To enable sufficient coverage and a focus on critical events at the same time, scenario-based testing (SBT) [3] aims to systematically describe driving scenarios an AV might encounter. Scenarios are gradually specified: First, a functional scenario is described in natural language [4], e.g., ‘an AV crosses an intersection’. Then, a corresponding logical scenario specifies inputs and their ranges, e.g., the velocity of the AV $v_{AV}$ ∈ $[5 \text{ kmh}^{-1}, 50 \text{ kmh}^{-1}]$. Finally, concrete scenarios with fixed values represent executable test cases, e.g., $v_{AV} = 30 \text{ kmh}^{-1}$.

Despite the systematic description of scenarios, safety validation can become intractable. Even with virtual test setups (e.g., 3D simulations), detailed models or perception pipelines can be elaborate, thus limiting the number of concrete scenarios that can be examined. Combined with the rarity of critical events, it is hence necessary to intelligently scan the space of each logical scenario for relevant concrete scenarios [3]. For this purpose, techniques from the fields design of experiments and active learning are applied. Here, a limited number of concrete scenarios is often used to create a metamodel (also called surrogate model) that enables predicting the outcome of further concrete scenarios. Based on predictive analyses, relevant concrete scenarios are then selected for examination.

Using predictive analyses, metamodels become a crucial part of a safety validation procedure. Non-sufficient performance of a metamodel can lead to critical events being missed or biased risk estimates. Nevertheless, metamodels are usually considered mere components of the process they are embedded in, and their explicit capability to model scenarios has not yet been investigated in detail. To better understand metamodels in the context of SBT, this paper is organized along the following research questions and contributions:

1) Which demands does scenario-based testing place on metamodels and which metamodels can satisfy them? This question is answered through a survey of metamodel applications and an analysis of various metamodels’ characteristics.

2) How do the characteristics of probabilistic metamodels affect their predictive performance and reliability? This
question is answered through an experimental benchmark considering four probabilistic metamodels and four driving scenarios with 5 to 20 inputs.

3) How does a targeted selection of test cases influence the predictive performance of metamodels? To answer this question, we introduce an iterative approach to efficiently select concrete scenarios and benchmark the resulting metamodels.

Section II and III introduce the tasks and characteristics of metamodels. Section IV and V describe the benchmark and iterative approach. Results, conclusions, and future work are discussed in Section VI, VII, and VIII, respectively.

II. METAMODELS IN SCENARIO-BASED TESTING

In SBT, metamodels are typically created for a logical scenario in a test setup \( M \), involving the AV under test and its environment [5]. The logical scenario defines \( N_I \) internal and external influences on the AV’s behavior, which we denote by inputs \( x \in \mathbb{R}^{N_I} \). The ranges of the inputs define the input space \( I \), containing all possible concrete scenarios. In Fig. 1, the inputs are \( \Delta t \) and \( \Delta v \). For each concrete scenario, the AV shows a behavior \( \Phi(M, x) \), which is assessed by \( y \in \mathbb{R} \) (e.g., the minimum time to collision \( \text{TTC}_{\text{min}} \) [6], a common safety metric). Hence, the execution of each concrete scenario gives a sample \( (x, y) \) and \( N_S \) samples are a dataset \( x \in \mathbb{R}^{N_S \times N_I}, y \in \mathbb{R}^{N_S} \) that allows training a metamodel \( M \).

While various models can act as metamodels and benchmarks are common in fields like computer vision (see, e.g., [7]), only few works compare metamodels in SBT [8], [9], [10]; however, they assess their algorithms’ overall performance and not the metamodels themselves, which limits generalizability. To gain a deeper understanding, we first analyze the demands common analysis tasks in SBT place on metamodels.

A. The Task of Falsification

Falsification seeks a concrete scenario where safety requirements \( \psi \) are violated, i.e., \( \Phi(M, x) \not\models \psi \) [5]. This often corresponds to minimizing safety metrics below a threshold, as it is done with Gaussian processes (GP) in [11] and [12]. For a similar approach, [8] compares six metamodels, and extreme gradient boosting performs best. However, in [9] the authors question this result since critical events are frequent in [8]. In [11], minimization is supported by iteratively focusing the search space around a known minimum. Using local metamodels, which only model subspaces of \( I \), can improve the predictive performance since the local behavior tends to be more homogenous and is, therefore, easier to model. Correspondingly, [13] combines dimensionality reduction, regression trees, and polynomial regression. Reference [10] uses an ensemble of a GP, polynomial regression, and a radial basis function network for global search and compares the individual models for local search, observing insignificant differences. [14] combines a neural network with a genetic algorithm.

For falsification, metamodels need to model a small part of the input space well so that a violation of requirements can be demonstrated with a small number of evaluations of \( M \).

Fig. 1. Depiction of the Cut-In scenario. (a) shows the initial conditions: starting from a time gap \( \Delta t \) and relative velocity \( \Delta v \), the vehicle on the right lane cuts into the left lane. The AV on the left lane can brake to avoid collisions. The minimum time to collision \( \text{TTC}_{\text{min}} \) in longitudinal direction is used as the safety metric and shown in (b). Negative values of \( \text{TTC}_{\text{min}} \) indicate that the vehicle on the right cut in behind the AV.

Thereby, predictions do not have to be precise but only rank the criticality of potential samples. Nevertheless, metamodels should be able to deal with high \( N_I \) to avoid that inputs have to be excluded based on global analyses, even though they may be relevant to provoke critical concrete scenarios. Lastly, a statement regarding metamodel certainty can be essential to avoid falsely concluding that no falsification is possible.

B. The Task of Boundary Identification

A more comprehensive task is the identification of safety boundaries, i.e., the boundaries between the fulfillment and violation of requirements, \( \Phi(M, x) \models \psi \) and \( \Phi(M, x) \not\models \psi \) of requirements. Here, [15] and [16] use GP classification models and reason that high classification uncertainty indicates the safety boundaries. Since classification models require observations of the fulfillment and violation of requirements, [17] and [8] use GP regression models to select samples resulting in a specified \( y \).

For boundary identification, metamodels must perform well across a larger part of the input space than for falsification. To place samples on the safety boundary, predictions have to be precise. Additionally, predicted uncertainties are valuable to place samples where their classification is ambiguous.

C. The Task of Estimation

Estimation aims to estimate the probability of requirement violations given the likelihood of concrete scenarios, i.e., \( P(\Phi(M, x) \not\models \psi) \mid M, p(x), \psi \). In [18] and [19], GPs are used to assess potential samples within an importance sampling approach. For a similar approach, [9] compares six metamodels, and an inverse distance weighted model performs best. Reference [20] combines GP classification and regression to model dissimilarities between an AV and its model. [21] uses a GP to obtain gradients for the behavior of an examined system.

For estimation, metamodels must properly model multiple clusters of critical samples. Furthermore, it can be essential that the models’ uncertainties are well-calibrated, i.e., they not only rank the expected criticality of samples, but the expected criticality matches the actual frequency of critical events.
III. CHARACTERISTICS OF PROBABILISTIC METAMODELS

The varying results in [8], [9], and [10] do not allow generalizable insights regarding metamodelling. However, there are manifold demands on metamodelling: Large input spaces have to be dealt with, predicted uncertainties are valuable to guide exploration, and particularly for estimation and termination criteria, model calibration is crucial. Hence, probabilistic models such as GPs are suitable models since they explicitly consider uncertainties caused by limited training data and the stochastic outcomes of scenarios. In fact, most of the approaches discussed in Section II use GPs. However, most examined scenarios are simple with regard to the number of inputs \( N_I \leq 3 \), [21] being an exception with \( N_I = 24 \). Therefore, it is unclear how well GP scale with respect to \( N_I \) [22]. Classic alternatives to GPs are random forests (RF) and extra-trees (ET). Furthermore, recent approaches like Bayesian neural networks (BNN) and deep Gaussian processes (DGP) achieved promising results for many tasks [23] but have not been applied in SBT.

In the following, we discuss the properties of the mentioned metamodelling technologies that are relevant for use in SBT. Since our benchmark only investigates a limited number of logical scenarios, the theoretical analysis helps to generalize experimental insights. Generally, probabilistic metamodelling technologies have to consider two types of uncertainties. Aleatoric uncertainty (from Latin ‘alea’, dice) accounts for noise in the training data, e.g., caused by sensor noise, and does not reduce with more data [24]. Epistemic uncertainty (from Greek ‘episteme’, knowledge) accounts for the uncertainty about a model’s parameters and reduces with more data [24].

A. Gaussian Processes

The usage of GPs [25] (also called Kriging) is often motivated by their clear mathematical definition. GPs’ non-parametric nature scales the complexity to the amount of data, and choosing hyperparameters is relatively simple. Furthermore, kernel functions allow the introduction of prior knowledge [25]. However, the definition of kernel functions suitable for SBT is an open problem [20]. All approaches in Section II use stationary kernels, for which the epistemic uncertainty of a predicted sample is determined primarily by its distance to observed samples. This assumption can lead to erroneous conclusions: E.g., the \( \text{TTC}_{\text{min}} \) of a cut-in at \( \Delta t = 10 \) s and \( 8 \) s might not differ much, but it can differ greatly at \( \Delta t = 3 \) s and \( 1 \) s (see Fig. 1). Furthermore, GPs typically consider homoscedastic aleatoric uncertainty, i.e., it is assumed that all concrete scenarios have equal variance. However, concrete scenarios along an AV’s decision boundaries might be more stochastic than other concrete scenarios.

Thus, for GPs, ease of use contrasts with limited capabilities in modeling non-linear, inhomogeneous system behaviors.

B. Random Forests and Extra-Trees

Rather than trusting the prediction of a single complex model, the motivation behind ensemble models is to have many simple models vote on a prediction. Here, RFs [26] and ETs [27] are popular approaches since the individual decision trees can be trained within short time and allow understanding the influences of different inputs [13]. However, decision trees are not well-suited to model interactions between inputs.

Thus, transparency and speed face the constraint that system behaviors are explained mainly by individual inputs rather than their interaction. We investigate ETs since they were superior to RFs during hyperparameter tuning (see Section VI).

C. Bayesian Neural Networks

Popularized by the rise of neural networks (NN), the idea behind BNNs [28] is to consider all parameters not as deterministic values but as probability distributions. Substituting discrete outputs \( \hat{y} \) by a mean \( \mu \) and variance \( \sigma^2 \), the aleatoric uncertainty is heteroscedastic, i.e., it depends on \( x \) and can be explicitly learned by the BNN. To consider epistemic uncertainty, ensemble models can be used [29]. However, their training is very costly. Hence, we only train one BNN and use Monte Carlo (MC) dropout, where multiple predictions are generated with different neurons being deactivated [30].

Due to their high flexibility, BNNs can represent non-linear systems well. However, BNNs can be sensitive to hyperparameters, and their predictions are hardly traceable.

D. Deep Gaussian Processes

Combining the concepts of GPs and NNs, DGPs are a network of GPs [23], allowing for hierarchical learning. Since GPs themselves are capable models, DGPs only contain a few layers and GPs per layer. Furthermore, DGPs are based on variational GPs, allowing for training with vast quantities of data without the large memory requirements of traditional GPs.

Hence, DGPs can capture more complex relationships than conventional GPs while remaining relatively transparent.

IV. EXPERIMENTAL BENCHMARK OF METAMODELS

To investigate how the metamodelling characteristics affect their predictive performance, we examine them in a benchmark. Ultimately, the goal of the benchmark is to identify a metamodel suited for a wide range of scenario-based tests. For the purpose of generalization, we, therefore, consider five logical scenarios that differ in complexity and type, i.e., number of inputs, driving tasks, and safety metrics.

A. Examined Scenarios

To balance data expense and realism, the scenarios are modeled and simulated in a 2D environment (IAV Scene Suite), and sensor signals are overlaid with Gaussian noise. This enabled us to create tabular datasets with 10 Million concrete scenarios that can be used for future studies. In the following, we provide short descriptions of the scenarios. More details and references are given in the Appendix.

The longitudinal scenarios Emergency Braking \( (N_I = 5) \) and Following \( (N_I = 13) \) evaluate a combination of adaptive cruise control and automatic emergency braking. Here, a \( \text{TTC}_{\text{min}} < 0.1 \) s is considered critical. The scenario Lane Keeping \( (N_I = 15) \) evaluates a lane keeping controller based
on the maximum absolute deviation from a desired trajectory; \( p_{\text{err abs max}} > 3 \) m is considered critical. The scenario Parking \((N_f = 20)\) models a typical parking lot and uses a safety metric similar to the one in [11]: Positive values indicate the minimum distance, 0 is used for collisions after the AV stopped, and negative values indicate the velocity of the AV in the moment of a collision. Here, crashes at more than 10 kmh\(^{-1}\) are considered critical, i.e., Criterion \(< -10\). The dataset also includes the scenario Cut-In shown in Fig. 1.

**B. Experimental Setup**

In all experiments, a metamodel \( \tilde{M} \) for a logical scenario is built based on \( N_S \) concrete scenarios. Without prior knowledge or assumptions about the behavior, the whole input space \( I \) of the logical scenario is equally important, and we call it region of interest (ROI) (see Fig. 2). From the concrete scenarios within the ROI, a representative selection can be sampled using the Sobol sequence [31] which ensures reproducibility and samples of consistent quality, also in higher dimensions [32]. The generated concrete scenarios \( x \) are then carried out to obtain the results \( y \), and \( \tilde{M} \) is trained.

**C. Evaluation Criteria**

To assess the metamodel \( \tilde{M} \), a reference dataset is utilized, which, to avoid correlations between the training and reference data, consists of pseudo-randomly distributed samples. The true values \( y \) are compared to the predicted \( \mu(x) \) and \( \sigma(x) \). Since we consider probabilistic metamodels, our primary quality indicator is the likelihood in Eq. (1)

\[
p(y \mid x, \tilde{M}) = \frac{1}{N_S} \prod_{i=1}^{N_S} \frac{1}{\sigma(x_i)} \sqrt{\frac{2\pi}{2}} e^{-\frac{(y_i - \mu(x_i))^2}{2\sigma(x_i)^2}}
\]  

(1)

For the likelihood, deviations between \(\mu(x)\) and \( y \) are acceptable if they come with a suitable \( \sigma(x) \). The likelihood is calculated in log-space to increase numeric stability.

A secondary quality indicator is the root mean squared error (RMSE) in Eq. (2), which does not take into account \( \sigma(x) \).

\[
\text{RMSE}(y \mid x, \tilde{M}) = \sqrt{\frac{1}{N_S} \sum_{i=1}^{N_S} (y_i - \mu(x_i))^2}
\]  

(2)

To assess efficiency, we compare how many concrete scenarios are necessary to achieve a certain likelihood or RMSE. To assess the predictions in more depth, we utilize scatter plots, precision-recall curves, and reliability curves.

**V. TARGETED SELECTION OF SAMPLES**

While evenly distributed samples are suitable for the initial examination of unknown systems, the tasks discussed in Section II require efficient learning strategies. In part, the discussed approaches use techniques that do not scale well to scenarios modeled with many inputs, e.g., choosing samples from a full-factorial set of possible samples. In the following, we demonstrate how samples can be selected in a way that scales to scenarios with high \( N_f \). Our approach allows executing all tasks described in Section II. Hence, it can serve as a basis for a scalable testing framework and as a baseline for future work using our dataset (see Appendix).

As illustrated in Fig. 2, a trained metamodel can be combined with an acquisition function \( a(\tilde{M}, x) \) resulting in an updated ROI. Sampling from the updated ROI, the density of samples will correspond to the values of \( a(\tilde{M}, x) \). In the following, we discuss how the definition of \( a(\tilde{M}, x) \) allows performing the tasks discussed in Section II. Since falsification is not necessarily informative [33] and for estimation, the distribution of concrete scenarios \( p(x) \) has to be known, we evaluate our approach for the task of boundary identification.

**A. Boundary Identification**

A violation of requirements \( y \) can often be characterized as an output \( y \) lying within a critical interval \( M_c \). For example, in Fig. 1 we may define that a \( \text{TTC}_{\text{min}} \in [-5, 0.5] \) s corresponds to a violation. Probabilistic predictions allow calculating the probability that \( y \in M_c \) using the cumulative distribution function. As discussed in Section II-B, examining samples for which the classification is ambiguous should reveal the safety boundaries. Such samples can be generated using the acquisition function \( a_{\text{BI}} \) in Eq. (3), which is calculated in log-space to ensure numerical stability.

\[
a_{\text{BI}}(x, \tilde{M}, M_c) = 1 - 2 \mid p(y \in M_c \mid \mu(x), \sigma(x)) - 0.5 \mid
\]  

(3)

Since the resulting ROI can be arbitrarily complex, new samples are drawn using rejection sampling [34]. Here, using the same Sobol sequence across iterations ensures that new samples are placed between existing ones. The number of samples is freely selectable, allowing to balance the expenses of carrying out test cases and metamodel training. The generated samples are then executed, added to the dataset, and a new metamodel is trained. Here, the first 20 % of the samples are used as validation and the last 80 % as training data to preserve the distribution of the Sobol sequence. Even
Fig. 3. Achieved likelihoods for all test cases using evenly distributed samples. The likelihoods assess how well the metamodels can predict the concrete scenarios’ safety metrics of a reference set (see subsection IV-C). Dots represent the mean and bars the standard deviation over 50 repetitions. Generally, an increasing number of samples $N_S$ leads to higher (better) likelihoods. While for small $N_S$, the differences between the models can be marginal, some models show a logarithmic relationship between $N_S$ and their likelihood, indicating that they would require orders of magnitude more data to achieve the same likelihood as superior models. The BNN benefits from high $N_S$ and consistently reaches the highest likelihoods for 3 out of 4 logical scenarios.

Fig. 4. Achieved RMSEs (root mean squared errors) for all test cases using evenly distributed samples. The RMSEs assess how well the metamodels can predict the concrete scenarios’ safety metrics of a reference set, considering only the predicted mean and not the models’ predicted uncertainties (see subsection IV-C). Dots represent the mean and bars the standard deviation over 50 repetitions. While the RMSEs reduce (improve) with increasing $N_S$, the logarithmic slopes suggest that some models require orders of magnitude more data to reach the RMSEs of others.

without observing critical behavior, the continuous output of the regression models allows to progressively explore and model the safety boundaries over multiple iterations.

B. Falsification and Estimation

Falsification can be realized using acquisition functions from Bayesian optimization [11], [12]. For estimation, metamodel-based importance sampling [35] can be applied, as shown in [19]. Here, the acquisition function $a_{IS}$ considers the distribution of concrete scenarios $p(x)$ as shown in Eq. (4).

$$a_{IS}(x, \tilde{M}, M_c, p) = P(y \in M_c | \mu(x), \sigma(x)) \cdot p(x)$$  \hspace{1cm} (4)

VI. EXPERIMENTAL RESULTS

To fully exploit the models’ potentials and at the same time investigate models that generalize well to future applications, we used Bayesian optimization to obtain one set of hyperparameters per model (see Table I). For a given hyperparameter set, the corresponding model had to model all five logical scenarios using evenly distributed samples for the initial and final $N_S$. Then, the likelihoods of the reference datasets were multiplied, and the product was maximized. Due to its low $N_I$, we omit the scenario Cut-In in the following examinations.

A. Global Model Quality Using Evenly Distributed Samples

First, we examine the metamodels without a targeted selection of samples and evaluate the global quality, i.e., the predictive performance for all test cases. We generate evenly distributed samples using a constant acquisition value $a_{ES} = 1$. The achieved likelihoods are shown in Fig. 3. For 3 out of 4 logical scenarios, the BNN reaches the highest likelihoods for all $N_S$. Interestingly, the RMSEs in Fig. 4 give a more balanced impression. Despite lower likelihoods, the GP and DGP can often achieve the same or lower RMSEs than the BNN and ET. As discussed in Section IV-C, this indicates that the errors of the ET’s and BNN’s predictions are accompanied by suitable uncertainties. This can be supported considering Fig. 5, especially for the samples where crashes occurred, i.e., Criterion $\leq 0$. For the GP and DGP, the predictions’ likelihoods are highly correlated to their absolute error (the
vertical distance to the diagonal). According to Eq. (1), this indicates that the predicted uncertainties are homogeneous, which, based on the discussion in Section III-A can be attributed to how GPs model uncertainty. In contrast, the ET and BNN potentially predict higher uncertainties for samples with higher errors, i.e., their uncertainties are heterogeneous.

The potential drawbacks of heterogeneous uncertainties are evident for the BNN in the scenario Lane Keeping. For specific concrete scenarios, the lane keeping controller gets unstable, resulting in high $p_{err \, abs \, max}$. The BNN is flexible enough to interpret these rare events as noise. This way, the large predicted $\sigma(x)$ result in competitive likelihoods. However, the predicted $\mu(x)$ do not follow $y$, resulting in poor RMSEs.

In terms of global likelihoods and RMSEs, the BNN achieves the best performance overall. However, the high fraction of unlikely predictions in Fig. 5 indicates that none of the models is well-calibrated, which is discussed in Section VI-C. It is also striking that critical samples are predicted poorly by all models. However, considering that critical samples are rare (for Fig. 5, the models observe about 10), the poor predictions are plausible and confirm that for the tasks discussed in Section II, samples must be selected efficiently.

**B. Model Quality for Critical Concrete Scenarios**

We now investigate how the acquisition function for boundary identification $a_{BI}$ (see Eq. (3)) influences the predictions of critical samples; Fig. 6 gives a visual impression. In all following figures, the results for evenly (quasi-randomly) distributed samples are shown in gray and serve as a baseline. Since precise predictions of the critical samples are essential for all tasks discussed in Section II, we evaluate the achieved likelihoods in Fig. 7. Applying the acquisition function can greatly increase the likelihood of critical samples. This is also the case for the RMSEs shown in Fig. 8.

Overall, it shows that often the best metamodel trained with evenly distributed samples cannot keep up with the worst one using the acquisition function. Thus, the influence of the acquisition function appears to be greater than that of the metamodel. Still, the BNN achieves the highest likelihoods and lowest RMSEs for 3 out of the 4 logical scenarios.

**C. Classification Performance**

For boundary identification (see Section II-B), it is crucial that the metamodels can distinguish critical and non-critical samples. To conduct such binary classification, a threshold for $P(y \in M_c \mid \mu(x), \sigma(x))$ can be defined, above which a sample is considered critical; a high threshold reduces false positives, while a low threshold helps to identify more critical samples. The precision-recall curves in Fig. 9 show that the application of the acquisition function $a_{BI}$ mostly increases the precision and therefore improves the classification performance. Considering a recall of 90 %, the GP, DGP, and BNN achieve precisions that increase the frequency of critical samples by two orders of magnitudes or more for all logical scenarios.
However, to identify the remaining 10% of critical samples as such, the precision sometimes converges to the natural frequency of critical samples. In rare cases, the application of the acquisition function reduces the precision.

The results above confirm that our approach is suitable to characterize safety boundaries. However, while identifying most of the safety boundaries is done very efficiently, a complete identification can not always be achieved.

D. Model Calibration

Examining scenarios with aleatoric uncertainty and limited data, a complete characterization can not be expected. It is, therefore, essential to know the degree to which a scenario is characterized. To this end, the classification calibration is particularly relevant in practice and examined in Fig. 10. The acquisition function can mostly improve the calibration. The ET did not perform well with previous performance indicators but achieves good calibrations. In contrast, the BNN’s calibration is often worse than that of the other models. Here, it becomes visible that due to the varying degrees of flexibility and the associated bias-variance tradeoff, different models shine with respect to different performance indicators.

E. Running Times of Model Training and Sampling

For all models, the training duration $t_{\text{train}}$ depends mainly on $N_S$. Hence, in Table I we list $t_{\text{train}}$ for $N_S = 500$ (averaged over the scenarios Cut-In and Emergency Brake) and $N_S = 10^4$ (averaged over the remaining scenarios). The ET is the fastest model (without a GPU). All other models use an NVIDIA GeForce RTX 2080 Ti. Here, the GP is slower than the ET but still much faster than the BNN and DGP.

The duration of the sampling process for the boundary identification task $t_{\text{sampling}}$ is stated per sample. $t_{\text{sampling}}$ depends mainly on the frequency of critical samples. If critical samples are rare, more potential samples have to be generated until suitable samples are identified. Hence, $t_{\text{sampling}}$ is stated for the scenarios Lane Keeping and Parking (where critical samples are rare) and averaged over all iterations. For sampling, the differences are smaller than for training. Sampling with the ET is the fastest, followed by the BNN, GP, and DGP.

To conclude, the more time-consuming or expensive individual test cases are, the more likely the effort of model training and sampling will be negligible compared to the effort of additional test cases. For applications with small batches (e.g., for iterative proving ground tests), fast models like ET or GP may have to be used since long running times are prohibitive.

VII. Conclusion

This paper provided an in-depth analysis of metamodels for the characterization of driving scenarios. Regarding global predictive performance, some models require orders of magnitude more samples than others to reach a certain likelihood or RMSE. The BNN seems most capable as a global metamodel due to its low RMSEs and heterogeneous uncertainties resulting in the highest likelihoods. However, using evenly distributed samples, none of the models performed well on the metrics used to evaluate the performance for typical tasks in scenario-based testing, such as boundary identification.
To build models that perform well for these tasks, we introduced a scalable, intuitive approach to select samples. As a result, the task-specific performance can be greatly improved to a degree where it seems evident that the appropriate selection of samples is more important than the choice of metamodels. In general, the predictive performance seems to depend more on a system’s complexity than the number of inputs it is modeled with. This implies that it is not necessary to rigorously limit the number of inputs during the modeling stage, which reduces the risk of omitting possibly crucial influences of the environment on an examined system.

While the acquisition function strongly influences the final model quality, for many applications, choosing an appropriate metamodel is still crucial. From a macroscopic view, the BNN seems to be the best metamodel indicated by high likelihoods and low RMSEs. However, there often is a small fraction of samples where the BNN’s predictions are rather poor. The less flexible models are more conservative, i.e., they are not especially good for most test cases but achieve reliable performance for almost all test cases. Among these models, the GP seems to be a very good choice due to its widespread use and solid performance for all performance indicators.

Overall, metamodels are well-suited for prioritizing test cases. However, metamodels’ predicted risks should be treated with caution. The choice of metamodels and exploration strategies always entails a tradeoff between efficiency and reliability. This tradeoff needs to be discussed not only methodologically but also ethically to provide society with a deeper understanding of AV safety validation and its limits.

We conclude that in early development stages, where data is easily available and which may be quite explorative because the system and the operational design domain may not be fully defined, flexible models like BNNs can significantly contribute to characterizing AVs as efficiently and completely as possible. In advanced stages, where less data is available and model calibration is more important, more reliable models like GPs can create evidence for the validation and release of AVs.

VIII. FUTURE WORK

There are many directions for future work. First, considering prior assumptions or the likelihood of concrete scenarios could speed up our iterative approach to some extent. However, it also introduces the risk of introducing a bias towards the experience of human drivers and overseeing scenarios that are challenging for AVs. For the task of boundary identification, considering the uncertainty of the predictions in the acquisition function (e.g., as done in [17]) could help to keep the focus of samples that lie exactly on the safety boundary and shift it towards samples with uncertain outcomes.

Furthermore, common safety metrics are created for human intuition and not for the purpose of building metamodels. Metrics created with a focus on properties such as continuity would allow modeling a system’s behavior more easily,
especially for less flexible metamodels. Here, the likelihoods used for predictions play a role as well. For example, Gaussian especially for less flexible metamodels. Here, the likelihoods used for predictions play a role as well. For example, Gaussian

| Model | Parameter | Value |
|-------|-----------|-------|
| GP    | Implementation | GPyTorch [37] |
|       | Kernel     | Matern\(_{\eta=0.5}\) / per input |
|       | Learning Rate | 0.1 |
|       | Early Stopping | 200 epochs with \(\Delta = 0.01\) |
|       | \(t_{\text{train}}\) / sample | 5.8 s (\(N_S = 500\)), 129 s (\(N_S = 10^5\)) |
|       | \(t_{\text{sampling}}\) / sample | 80 ms (Lanekeeping), 36 ms (Parking) |
| DGP\(_2\) | Implementation | GPyTorch [37] |
|       | Kernel | Ration Quadratic, \(\eta\) / per input |
|       | Shape Inducing Points | \([N_J, N_D]\) \(100\) |
|       | Mixture Components | 9 |
|       | Learning Rate | 0.01 |
|       | Early Stopping | 200 epochs with \(\Delta = 0.01\) |
|       | MC Samples | 50 |
|       | \(t_{\text{train}}\) / sample | 71 s (\(N_S = 500\)), 1259 s (\(N_S = 10^5\)) |
|       | \(t_{\text{sampling}}\) / sample | 162 ms (Lanekeeping), 72 ms (Parking) |
| ET    | Implementation | scikit-learn 0.24.0 [38] |
|       | Estimators | 100 |
|       | Min Samples Split | 4 |
|       | Max Features | \(N_J\) |
|       | \(t_{\text{train}}\) / sample | 0.27 s (\(N_S = 500\)), 0.54 s (\(N_S = 10^5\)) |
|       | \(t_{\text{sampling}}\) / sample | 29 ms (Lanekeeping), 26 ms (Parking) |
| BNN   | Implementation | Concrete Dropout [30] |
|       | Shape Activation Concrete Dropout | \([N_J, 180, 110, 430, 830, 160, 2N_D]\) \(\tau = 7, \eta = 0.0005\) |
|       | Minibatch Size | 8 |
|       | Learning Rate Early Stopping | 0.001 (\(0.2\) each 10k minibatches) 20k minibatches with \(\Delta = 0.1\) |
|       | MC Samples | 50 |
|       | \(t_{\text{train}}\) / sample | 1699 s (\(N_S = 500\)), 1765 s (\(N_S = 10^5\)) |
|       | \(t_{\text{sampling}}\) / sample | 43 ms (Lanekeeping), 54 ms (Parking) |

**APPENDIX**

**DRIVING SCENARIO INPUT OUTPUT DATASET**

Tabular results, descriptions, and videos of the investigated scenarios are available at https://github.com/wnklmx/DSIOD. We invite researchers to add similar datasets to this repository.

**ACKNOWLEDGMENT**

The authors would like to thank Mike Harttrumpf, Constantin Vasconi, David Seidel, Roland Kallweit, and Matthias Butenuth of IAV GmbH for their feedback and support.

**REFERENCES**

[1] P. Pennmetsa, P. Sheinidashegol, A. Musaev, E. K. Adanu, and M. Hudnall, “Effects of the autonomous vehicle crashes on public perception of the technology,” IATSS Res., vol. 45, no. 4, pp. 485–492, Dec. 2021.

[2] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” Transp. Res. A, Policy Pract., vol. 94, pp. 182–193, Dec. 2016.

[3] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on scenario-based safety assessment of automated vehicles,” IEEE Access, vol. 8, pp. 87456–87477, 2020.

[4] T. Menzel, G. Bagschik, and M. Maurer, “Scenarios for development, test and validation of automated vehicles,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018, pp. 1821–1827.

[5] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, “Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques,” IEEE Control Syst., vol. 36, no. 6, pp. 45–64, Dec. 2016.

[6] J. C. Hayward, “Near-Miss Determination through use of a scale of danger,” in Traffic Records (Highway Research Record), vol. 384. Washington, DC, USA: The Pennsylvania State Univ, Pennsylvania Transportation and Traffic Safety Center, 1972, pp. 24–34. [Online]. Available: https://www.worldcat.org/title/near-miss-determination-through-use-of-a-scale-of-danger/oclc/122690289

[7] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, “A review and comparative study on probabilistic object detection in autonomous driving,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 9961–9980, Aug. 2022.

[8] J. Sun, H. Zhou, H. Xi, H. Zhang, and Y. Tian, “Adaptive design of experiments for safety evaluation of automated vehicles,” IEEE Trans. Intell. Transp. Syst., early access, Dec. 3, 2021, doi: 10.1109/TITS.2021.3130040.

[9] H. Zhang, H. Zhou, J. Sun, and Y. Tian, “Risk assessment of highly automated vehicles with naturalistic driving data: A surrogate-based optimization method,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2022, pp. 580–585.

[10] F. U. Haq, D. Shin, and L. Briand, “Efficient online testing for DNN-enabled systems using surrogate-assisted and many-objective optimization,” in Proc. 44th Int. Conf. Softw. Eng., May 2022, pp. 811–822.

[11] H. Beglecoro, M. Stolz, and M. Horn, “Testing of autonomous vehicles using surrogate models and stochastic optimization,” in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC), Oct. 2017, pp. 1–6.

[12] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, “Identification of test cases for automated driving systems using Bayesian optimization,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 1961–1967.

[13] R. Matinnejad, S. Nejati, L. Briand, and T. Brcukmann, “MiL testing and sensitivity analysis for safe automated driving using Gaussian process classification,” in Proc. 29th ACM/IEEE Int. Conf. Automated Softw. Eng., Aug. 2016, pp. 63–74.

[14] J. Zhou and L. del Re, “Safety validation of ADAS by collision-free boundary searching of a parameterized catalog,” in Proc. Annu. Amer. Control Conf. (ACC), Jun. 2018, pp. 4790–4795.

[15] F. Batsch, A. Daneshkhhah, M. Cheah, S. Kanarachos, and A. Baxendale, “Performance boundary identification for the evaluation of automated vehicles using Gaussian process classification,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 419–424.

[16] F. Batsch, A. Daneshkhhah, V. Palade, and M. Cheah, “Scenario optimization and sensitivity analysis for safe automated driving using Gaussian processes,” Appl. Sci., vol. 11, no. 2, p. 775, Jan. 2021.

[17] Z. Huang, H. Lam, and D. Zhao, “Towards affordable on-track testing for autonomous vehicle—A Kriging-based statistical approach,” in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC), Oct. 2017, pp. 1–6.

[18] M. Winkelmann, C. Vasconi, and S. Müller, “Transfer importance sampling—How testing automated vehicles in multiple test setups helps with the bias-variance tradeoff,” Apr. 2022, arXiv:2204.07619.
