City Research Online

City St George's, University of London

Citation: Zhao, X., Aghazadeh-Chakherlou, R., Cheng, C-H., Popov, P. & Strigini, L. (2025). On the Need for a Statistical Foundation in Scenario-Based Testing of Autonomous Vehicles. 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC), pp. 3998-4005. doi: 10.1109/itsc60802.2025.11423546 ISSN 2153-0009 doi: 10.1109/itsc60802.2025.11423546

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/38054/

Link to published version: https://doi.org/10.1109/itsc60802.2025.11423546

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

On the Need for a Statistical Foundation in Scenario-Based Testing of Autonomous Vehicles

Xingyu Zhao[1], Robab Aghazadeh-Chakherlou[1], Chih-Hong Cheng[2], Peter Popov[3], Lorenzo Strigini[3]
[1]*WMG, University of Warwick, Coventry, U.K.*
[2]*Department of Computer Science, Chalmers University of Technology & University of Gothenburg, Gothenburg, Sweden*
[3]*Centre for Software Reliability, City St George's, University of London, London, U.K.*
{xingyu.zhao,robab.aghazadeh-chakherlou}@warwick.ac.uk, chihhong@chalmers.se, {p.t.popov,l.strigini}@city.ac.uk

***Abstract*—Scenario-based testing has emerged as a common method for autonomous vehicles (AVs) safety, offering a more efficient alternative to mile-based testing by focusing on high-risk scenarios. However, fundamental questions persist regarding its stopping rules, residual risk estimation, debug effectiveness, and the impact of simulation fidelity on safety claims. This paper argues that a rigorous statistical foundation is essential to address these challenges and enable rigorous safety assurance. By drawing parallels between AV testing and traditional software testing methodologies, we identify shared research gaps and reusable solutions. We propose proof-of-concept models to quantify the probability of failure per scenario (*pfs*) and evaluate testing effectiveness under varying conditions. Our analysis reveals that neither scenario-based nor mile-based testing universally outperforms the other. Furthermore, we introduce *Risk Estimation Fidelity* (REF), a novel metric to certify the alignment of synthetic and real-world testing outcomes, ensuring simulation-based safety claims are statistically defensible.**

***Index Terms*—Statistical modelling, safety assurance, residual risk, operational profile, simulation fidelity, software reliability.**

I. Introduction

Autonomous vehicles (AVs) are transitioning rapidly from research environments to public roads. Waymo initiated this shift by launching its first commercial AV taxi in December 2018. More recently, Tesla has announced plans to deploy a robotaxi service in Austin, Texas, by June 2025, leveraging its Full Self-Driving system. These developments underscore the urgent need for comprehensive and rigorous safety assessment to satisfy regulatory standards and public safety expectations.

While many efforts have been made to address the AV safety challenges covering various aspects like design, implementation, regulation and legal issues [1]–[5], AV testing remains an indispensable method for generating the verification and validation evidence necessary for safety assurance. In general, there are two common types of AV testing, *mile-based* and *scenario-based*. The former involves accumulating a large number of real-world/simulated miles to statistically validate the vehicle's safety. The premise is that, as the vehicle encounters a wider variety of driving conditions and edge cases, statistical confidence in its safety increases. Given the required rarity of incidents, demonstrating safety through this method would require millions/billions of miles [6]–[8]. The industry largely resists such mile-based "driving to safety" strategy due to the prohibitive cost and risk. Instead, they favour scenario-based testing [9]–[12] as a complementing approach, which focuses on using controlled, high-risk scenarios to infer the AV's safety. This method avoids the need for extensive driven miles; instead, it tests the vehicle's response to predefined, potentially dangerous situations through simulated or field-tested extreme scenarios.

However, current scenario-based testing practices cannot yet answer several pressing questions:

i) What is the stopping rule of scenario-based testing? The number of possible parameter combinations to describe test scenarios is exploding, so exhaustive testing of all possible scenarios for a given operational design domain (ODD) is infeasible [13]. While coverage criteria were studied (e.g. in [13]–[15]) for scenario-based testing, borrowing ideas from traditional software testing where coverage criteria are used to decide when to stop [16], problems like the arbitrariness of discretising continuous parameters and redundancy/bias in scenario generation limit their effectiveness [13].

ii) What is the residual safety risk after extensive scenario-based testing? Even after testing a vast number of scenarios (say achieved 100% coverage of some criteria), the residual risk (measured by, e.g., the probability of crash per random scenario/mile) remains unknown. This uncertainty arises because coverage criteria typically measure which parts of the input space have been exercised but do not translate directly into an estimate of overall system safety delivered [17], [18] (a well-known problem in software engineering).

iii) Is scenario-based testing always more effective at improving safety of AVs? Scenario-based testing "smartly" selects conditions that are hypothesised to stress the AV, thus potentially detecting more bugs than mile-based testing. However, "more bugs does not necessarily imply less reliable" [19]. Consider an extreme example: if scenario-based testing uncovers only "5,000-year bugs"[1], fixing them may have little improvement on AVs' operational safety, compared to fixing fewer "5-year bugs" discovered through mile-based testing. Essentially, *not all bugs contribute equally* to operational

Supported by the UK EPSRC New Investigator Award [EP/Z536568/1].

[1]E. Adams studied fault occurrence rates in large IBM software systems and found that over 60% were "5,000-year bugs", meaning each fault manifests as failures, on average, once every 5,000 years in normal operation [20].

safety risk. Scenario-based testing, e.g., using optimisation algorithms to achieve a required level of coverage, often overlooks this fact in its objectives, while mile-based testing naturally detects more "common bugs".

iv) How does the fidelity[2] *of simulated scenarios affect safety estimates?* Given that scenario-based testing focuses on high-risk conditions, synthetic data generated by simulation is typically employed to execute test scenarios. However, the Sim2Real gap introduces an additional layer of uncertainty, implying that safety claims based on synthetic scenario testing evidence may not fully capture real-world performance [10]. Despite great efforts having been made to bridge the gap [21]–[24], there remains a lack of rigorous methods to quantify simulation fidelity [23] and translate varying fidelity levels into *measurable* safety risk.

The aforementioned questions are not independent but rather highly interrelated. Without rigorous answers to them, the usefulness of scenario-based testing in *formal safety arguments for regulatory justification* may be limited. While there is no simple solution, we believe that *statistical analysis and probabilistic modelling can provide a solid theoretical foundation* for addressing these challenges.

In this paper, we begin by reviewing the state-of-the-art in statistical modelling for AV safety assessment. We observe that most existing work focuses on mile-based testing, while statistical foundations for scenario-based testing remain largely under-explored, despite its growing prominence in industry practice. Furthermore, we notice that many of the open questions in scenario-based testing, e.g., effectiveness in debugging and residual risk estimation, *closely parallel longstanding challenges studied in the traditional software testing community*. We present an initial model, as a proof-of-concept, to demonstrate the potential of applying statistical approaches to rigorously reason about scenario-based safety evidence, and to highlight the potential for AV testing to benefit from software testing, with existing models potentially being reusable. Our goal is to lay a foundation that future work can build upon to develop rigorous safety arguments for AVs.

In summary, the main contributions of this paper are:

- A survey on statistical modelling for AV safety assessment, and a comparison study with software testing to map shared research questions and identify reusable solutions.
- Proof-of-concept models using scenario-based testing evidence to assess AV safety and evaluate their effectiveness in reducing operational risk.
- A novel notion of *Risk Estimation Fidelity* of synthetic data, with formal definitions and example use cases.

II. AV Testing vs Software Testing

A. AV Testing

The main idea behind scenario-based testing for AVs is to generate *relevant scenarios* intentionally in simulations or on proving grounds, rather than waiting until they *randomly* occur in distance-based testing. Relevant scenarios are all scenarios that can occur within the ODD and are challenging for the AV and, therefore, might cause failures[3].

Definition 1 (ODD [11])**.** *An ODD is a set of operating conditions under which a given driving automation system or feature thereof is specifically designed to function.*

Definition 2 (Scenario [9])**.** *A scenario describes the temporal development between several scenes in a sequence of scenes, where a scene is a snapshot of the environment, including the scenery and dynamic elements, and all actors' and observers' self-representations and the relationships among those entities.*

Definition 3 (Scenario abstraction levels [25])**.** *Scenarios are commonly organised into four levels of abstraction: 1) Functional scenarios use natural language to describe high-level situations. 2) Abstract scenarios formalise natural language descriptions into machine-readable formats, often supported by ontologies. 3) Logical scenarios define parametrised versions of abstract scenarios, specifying variable ranges (e.g., speed, position, weather). 4) Concrete scenarios assign specific values to each parameter, representing fully specified test cases derived from logical scenarios.*

Note, such levels of abstraction can also be viewed as levels of partitioning in the parameter space that describes scenarios.

Definition 4 (Mile-based testing)**.** *Mile-based testing evaluates AV safety by accumulating real/simulated miles within the ODD, aiming to sample scenarios as they naturally occur. Safety is inferred statistically by analysing failure rates (e.g., crashes, disengagements) per unit distance. It assumes an unbiased estimate and exposure to diverse scenarios, potentially offering some benefits similar to scenario-based testing.*

Similar to [13], the following remark attempts to link the two types of AV testing, with a relaxation of rigour in details, e.g., the overlapping of scenarios on driven miles.

Remark 1 (Mile-based testing vs scenario-based testing)**.** *Mile-based testing can be viewed as a form of scenario-based testing, where driven miles can be modelled as a stochastic sequence of concrete scenarios sampled from the ODD according to an "operational scenario distribution".*

B. Software Testing

There is a substantial body of work in the software engineering community on safety and reliability modelling based on testing evidence. While software testing methods can be classified from various perspectives, e.g., functional vs non-functional, black-box vs white-box, and static vs dynamic, we focus on the following two dimensions that facilitate mapping concepts from AV testing to software testing.

Broadly, software testing falls into two categories: *directed* testing and *operational* testing; they are also often called

[2]While specific definitions of fidelity vary across studies, in general, fidelity refers to how closely synthetic data replicates real-world conditions, thereby supporting reliable simulation-based testing of AVs.

[3]In this paper, "failure" is used as a general term in the context of AV safety. Depending on the assessor's focus in specific models, it may refer to events such as a crash, fatality, near-miss, or human-initiated disengagement.

debug testing and *acceptance* testing [18], [26] based on their intended purposes. The former aims to find and fix defects, while the latter assesses existing quality to build confidence for deployment. The key difference is whether test cases are sampled from the operational profile (OP). Debug testing typically ignores the OP, whereas acceptance testing depends on it [18]. However, research shows operational testing may outperform directed testing in bug detection, and some acceptance tests rely solely on directed methods. We show this using our proof-of-concept models later.

Definition 5 (OP [27])**.** *An OP is a probability distribution over the input space of a software system, representing the relative frequencies with which different inputs are expected to occur during actual operation.*

From the perspective of "*how we generate tests*" in the input parameter space, software testing can be broadly categorised into *partition-based testing* and *random testing* [28], [29].

Definition 6 (Partition-based testing)**.** *Partition-based testing partitions the input domain of a software system into distinct subsets. Test cases are then selected from each partition to ensure comprehensive coverage of the input space.*

This approach aims to maximise testing efficiency by "smartly" partitioning the input space and selecting representative test cases from each partition, thereby reducing the total number of test cases while maintaining effective coverage.

Definition 7 (Random testing)**.** *Random testing generates test cases randomly from the input domain, according to a predefined probability distribution, e.g., the OP.*

C. Mapping Concepts Between AV and Software Testing

Let us consider the parameter space used to describe scenarios (within a given ODD) as analogous to the input parameter space in software testing. Then we may conclude:

Remark 2 (ODD vs OP)**.** *The ODD can be regarded as an OP without explicitly defining the distribution over it. If an "operational scenario distribution" can be defined over the ODD, conceptually, it aligns with the OP in software testing.*

Remark 3 (Scenario-based testing vs partition-based testing)**.** *Scenario-based testing can be viewed as a form of partition-based testing, where logical scenarios partition the ODD, and representative concrete scenarios are tested in each partition to ensure coverage. Both aim to save tests by assuming failures cluster in certain "smartly partitioned" subdomains, allowing testers to focus effort where it matters most.*

Remark 4 (Mile-based testing vs random testing)**.** *Continuing Remark 1, since each concrete scenario arises randomly from an operational scenario distribution (i.e., the OP), mile-based testing resembles repeated bursts of random testing across the scenario parameter space.*

Remark 5 (Align AV testing with testing purpose)**.** *Given the motivation, assumptions, and design of partition-based and random-based testing, they naturally align with debug testing and acceptance testing (where an OP is used), respectively. Thus, following Remarks 3 and 4, scenario-based testing aligns more closely with the goals of debugging, whereas mile-based testing is better suited for acceptance testing.*

That said, there are exceptional cases that we learnt from studies of software testing, e.g., [8], [17]–[19], [26], [28], [29]: 1) Partition/scenario-based testing is often thought-to-be good at debugging, but there is no formal proof and it may be less effective under certain, not especially unusual, conditions. 2) Partition/scenario-based testing can still be used for estimating reliability (i.e., residual safety risk), if distributional information is explicitly incorporated within and across logical scenario partitions. 3) Random/mile-based testing can still discover bugs, whether they may or may not be less efficient in improving safety. All of these points are context-dependent and cannot be answered universally. This highlights *the need for a theoretical model with statistical foundation capable of analysing different AV testing methods across conditions*.

III. Statistical Modelling for AV Safety

There are, indeed, many existing works on AV safety applying statistical reasoning to estimate failure rates. However, they typically focus on specific abstract or logical scenarios (e.g., [30]–[34]), rather than modelling across the entire ODD using an operational distribution—which is the focus of our work. We also exclude papers that rely solely on descriptive statistics for AV safety, and instead focus on those that employ statistical inference where an underlying stochastic failure process is assumed. To the best of our knowledge, the selected publications are summarised in Table I, from which the following key insights can be drawn.

Remark 6 (Metrics)**.** *The nature of the metric, whether it represents a probability, e.g., probability of failure per mile (pfm), or a rate, e.g., disengagements per mile (dpm), is crucial in determining the appropriate modelling approach. Although probability and rate are sometimes used interchangeably, they are fundamentally different. Probability represents the likelihood of an event occurring and ranges from 0 to 1. In contrast, rate quantifies the number of events per selected unit (e.g., time or distance), and ranges from 0 to ∞. It determines the choice of the underlying stochastic process (e.g., Bernoulli for probabilities or Poisson for rates) and informs whether a discrete or continuous model is more appropriate. The metric also guides how data should be collected in practice, and whether the data realistically supports the assumptions of the chosen stochastic process.*

Several other works [42]–[46] study the probability of failure per demand (*pfd*), a general metric applicable across domains such as AVs and nuclear power. In the context of AV safety, a "demand" can be abstracted as a trip, a mission, or a scenario, as long as the modelling assumptions can be justified at the defined "demand" level, e.g., AV trips are independently and identically distributed (i.i.d.).

TABLE I: Summary of AV safety studies employing statistical inference with assumed stochastic failure processes across operational domains.

Paper	Metric	Stoc. Proc.	Estimator	Data form and Source	Main Message
[6]	*pfm*	Bernoulli	Reliability model from [35]	Failure-free miles, illustrative data	Extensive mileage requirements. Necessity for alternative methods. Adaptive regulatory frameworks.
[6]	*fpm*	Poisson	Significance test	Miles with failure, illustrative data	Extensive mileage requirements. Necessity for alternative methods. Adaptive regulatory frameworks.
[7] [36]	*pfm*	Bernoulli	CBI	Failure-free & rare failure illustrative data	In addition to [6], prior knowledge may bring advantages, while avoiding optimistic biases.
[7]	*dpm*	Poisson	SRGM	Waymo's disengagement data (up to 2019)	Sensitive to model used. CBI preferred for high-reliability claims.
[37]	*dpm*	NHPP	Monotonic splines (non-parametric model)	CDMV's disengagement data	AVs are becoming more reliable by decreasing disengagement trends Statistical tools useful for monitoring.
[38]	*dpm*	NHPP	Bayesian inference with SRGM	CDMV's disengagement data	Providing a statistically grounded structured approach to AV reliability test design
[38]	*dpm*	HPP	Bayesian inference without SRGM	CDMV's disengagement data	Providing a statistically grounded structured approach to AV reliability test design
[39]	Bounded reachability probability	Poisson	Model checker on SAN	HighD dataset [40], Lyft [41]	Probabilistic framework to understand and mitigate AV safety risks.

Remark 7 (Stochastic process)**.** *For mathematical convenience, common stochastic processes are used, e.g., Bernoulli (for pfm) or Poisson (for dpm). However, as discussed in the papers listed in Table. I, those stochastic processes impose assumptions—e.g., a constant failure probability/rate per mile and independence between miles/events—that may not always hold true in real-world applications due to factors like software updating of the AV or changing environments. That said, they can still offer a useful approximation in practice.*

After defining the stochastic process, likelihood functions of the failure events of interest can be derived. Then either Bayesian or frequentist estimators can be applied. As shown in Table I, statistical significance testing, CBI (Conservative Bayesian Inference), and SRGM (Software Reliability Growth Models) are the most commonly used estimation methods. CBI uses evidence-based *partial prior knowledge* about the metric (e.g., *pfm*) to determine a set of prior distributions[4]. This set is then updated using Bayesian inference based on statistical operational evidence, and the most conservative posteriors can be derived by optimisation. SRGMs, originating from software reliability engineering, are meant to extrapolate observed trends as a software product becomes more reliable as bugs are found and corrected. The models themselves may for instance represent the failure process as a non-Homogeneous Poisson Process (NHPP).

A Stochastic Activity Network (SAN) is a probabilistic modelling framework which is used to represent complex, stochastic systems. It can be used to assess the safety of AVs under various hazardous conditions, e.g., road hazards, cyber-attacks, etc. There are tools (like Möbius) that help analyse and simulate SAN models.

Remark 8 (Estimator)**.** *Frequentist methods, such as statistical significance tests, rely only on observed data, which can make them less reliable when dealing with small datasets. In the case of AVs, this can be challenging as a large amount of data is often needed for safety demonstration. They also cannot incorporate prior knowledge, an advantage offered by Bayesian inference methods (e.g., CBI). However, Bayesian estimators are normally sensitive to prior knowledge and how it is translated into mathematical form [46].*

[4]Classical Bayesian inference requires a fully specified prior distribution, which is often difficult to justify in practice.

Remark 9 (Estimator)**.** *SRGMs help track and predict failures over time, but they cannot be trusted a priori. They are especially problematic for highly reliable software and, even when generally accurate for a system, can occasionally produce seriously incorrect predictions. For SAN models, accurately mapping real-world components and data into SAN structures and transition rates between states is crucial and challenging.*

IV. Proof-of-Concept Models

A. The Metric, Stochastic Failure Processes, and Estimators

As stated in Remark 6, the choice of metric is crucial in statistical inference. As a first attempt, we align with existing metrics and define the *probability of failure per randomly selected scenario* (drawn from the operational scenario distribution over the ODD), abbreviated as *pfs*:

Definition 8 (*pfs*)**.** *Let $x \in D$ be a concrete scenario in the scenario parameter space D of the ODD, and $Op : D \to [0, 1]$ is the operational scenario distribution.*

$$pfs := \int_{x \in D} I_{\{x \text{ causes a failure}\}}(x) Op(x) dx = \theta \qquad (1)$$

where $I_{\{S\}}$ is an indicator function—it is equal to 1 when S is true and 0 otherwise. We then denote pfs as θ.

Similar to the "frequentist" interpretation of *pfd* in [47], *pfs* can be regarded as a *limiting relative frequency of concrete scenarios for which the AV fails in an infinite sequence of independently selected concrete scenarios from the OP*. Note, as stated in footnote 3, the "failure" in *pfs* is a general term that can be replaced by any safety event of the assessor's interest, e.g., crash and near-miss.

1) Scenario-based testing conducted in a mile-based/random testing way: The Bernoulli process normally is a good approximation of the stochastic failure process for this case [6], [7], thus the likelihood of seeing k failed scenarios in a sequence of t random test (concrete) scenarios is:

$$Pr(k, t \mid \theta) = \theta^k (1-\theta)^{t-k}. \tag{2}$$

Given the likelihood function, many statistical inference models can be better used for estimating θ, including those summarised in Table I, depending on the observed data (e.g., failure-free), prior knowledge available, and desired posterior estimation forms (e.g., mean or confidence bounds of *pfs*). E.g., a Bayesian estimator for the posterior mean of *pfs* is:

$$\hat{\theta} = \mathbb{E}[\theta \mid k, t] = \frac{\int_0^1 \theta \cdot \theta^k (1-\theta)^{t-k} f(\theta) d\theta}{\int_0^1 \theta^k (1-\theta)^{t-k} f(\theta) d\theta} \tag{3}$$

where $f(\theta)$ is the prior distribution of *pfs*. Thanks to the set of CBI models summarised earlier, a complete f is not required; rather, partial knowledge of f would suffice for the reasoning.

2) Scenario-based testing conducted in a partition-based testing way: Let the scenario parameter space D partitioned into subdomains $D_1, \ldots, D_n$, i.e., n logical scenarios. Then, by the law of total probability

$$\theta = \sum_{i=1}^{n} \theta_i Op_i, \text{ where } Op_i := \sum_{x \in D_i} Op(x) \tag{4}$$

Thus, to estimate θ, we need to know the *conditional pfs* of each logical scenario θ_i and the OP. Normally, inside a given logical scenario D_i, a new and (thought-to-be) more efficient distribution is used to generate concrete test scenarios (e.g., by an optimisation algorithm) rather than the (local) OP. Usually, the estimator using *importance sampling* can be used where the proposal distribution is either known or implicitly approximated [30]–[33].

B. Under Which Conditions Will Scenario-based Testing Deliver Better Safety than Mile-based Testing?

We continue to investigate the above question by retrofitting the model from [18] for AV testing. To model the fix behaviour after detecting failures, [18] introduces the concept of "failure regions (F)"—sets of inputs in D that cause failure and are eliminated by fixes. It assumes[5] a fix removes a specific failure region and does not introduce new failure regions. While [18] considers the more general case of multiple (disjoint) failure regions, we only present the simplified *single failure region* case to better illustrate the conceptual mapping between AV and software testing, and the potential for reusing existing research insights.

Consider an AV under testing with *pfs*, $\theta = q$. After doing mile-based testing that included t random concrete scenarios (cf. Remarks 1 & 4). Depending on whether the failure region has been detected or not, the expected *pfs* after fixing is stated in Eq. (5), where the "0" reflects the fact that after fixing, there is no bug residing.

$$\mathbb{E}[\theta] = 0 \cdot Pr(\theta = 0) + q \cdot Pr(\theta = q) = q(1-q)^t \tag{5}$$

Now consider scenario-based testing and let the scenario parameter space D be partitioned into subdomains $D_1, \ldots, D_n$, i.e., n logical scenarios. According to some specific scenario-based testing methods, t_i test cases are generated independently for each D_i. Let d_i be the bug detection rate for logical scenario D_i, i.e., the probability that a concrete scenario generated for D_i reveals a failure, then the probabilities of seeing at least 1 failure and no failures after scenario-based testing are, respectively:

$$1 - \prod_{i=1}^{n} (1-d_i)^{t_i}, \quad \prod_{i=1}^{n} (1-d_i)^{t_i}. \tag{6}$$

For the former case, after fixing the failure detected, $\theta = 0$; for the latter case, the AV remains as it is with $\theta = q$, thus the expected *pfs* after scenario-based testing and fixing is:

$$\mathbb{E}[\theta] = 0 \cdot Pr(\theta = 0) + q \cdot Pr(\theta = q) = q \prod_{i=1}^{n} (1-d_i)^{t_i} \tag{7}$$

For a fair comparison, we normally take $\sum_{i=1}^{n} t_i = t$. Then a comprehensive analytical study can be conducted to compare Eq. (5) and Eq. (7). For simplicity, we only present examples to show that there is *no universal conclusion* on the superiority of scenario-based testing over mile-based testing.

Consider two possible cases: *1)* The single failure region F is spread across all possible D_i, e.g., a misperception failure that may happen in many logical scenarios; *2)* F is strictly contained by a single D_i, e.g., a rare failure can only happen in a specific logical scenario. We show examples for each case.

Example 1. *When the failure region is uniformly "spread out" over all the subdomains, implying that the chance of finding a failure is the same in all subdomains (a constant detection rate $d_i = \bar{d}$). Then the case reduces to scenario-based testing without subdomains, and Eq.* (7) *becomes:*

$$\mathbb{E}[\theta] = q(1-\bar{d})^t \tag{8}$$

Comparing Eq. (8) *to Eq.* (5) *shows that scenario-based testing is superior if and only if $\bar{d} > q$.*

Example 2. *Consider the failure region is a strict subset of a single subdomain k, $F \subset D_k$, and with some points in D_k not being failure points, $D_k \not\subset F$, and no other subdomains overlapping with F: $F \cap D_i = \emptyset, \forall i \neq k$. Additionally, we assume that within D_k, both scenario-based and mile-based testing (on average) have an equal probability to encounter F. That is, the detection probability d_k is just the fraction of the operational distribution scenarios in D_k that encounter F: $d_k = \frac{\sum_{x \in F} Op(x)}{\sum_{x \in D_k} Op(x)}$, For scenario-based testing, assume test budgets are distributed equally across n subdomains, so $t_k = t/n$, where t is the total number of tests.*

- ***Situations when scenario-based testing is superior:*** *This case occurs when the failure-prone subdomain is relatively*

[5]The original paper [18] discusses the validity of some assumptions and model extensions to relax them, e.g., "partial fix". For simplicity, we omit these discussions here.

rare in real operation, so that the mile-based testing gives it fewer tests than the scenario-based testing. Mathematically, this is expressed as $t\sum_{x\in D_k} Op(x) \ll t_k$. *After substituting* $t_k = t/n$, *we will have:*

$$\sum_{x\in D_k} Op(x) \ll \frac{1}{n} \tag{9}$$

Then, the expected pfs for scenario-based testing is:

$$\mathbb{E}[\theta] = q\prod_{i=1}^{n}(1-d_i)^{t_i} = q(1-d_k)^{t_k} \tag{10}$$

$$= q\left(1-\frac{\sum_{x\in F} Op(x)}{\sum_{x\in D_k} Op(x)}\right)^{\frac{t}{n}} < q\left(1-\frac{\sum_{x\in F} Op(x)}{1/n}\right)^{\frac{t}{n}} \tag{11}$$

$$\approx q\left(1-t\sum_{x\in F} Op(x)\right) \approx q(1-q)^t \tag{12}$$

where the last term $q(1-q)^t$ *is the pfs for mile-based testing. The two approximations in Eq.s* (12) *require that* d_k *and* q *are small, using* $(1+x)^y \approx 1+yx$ *for small* x.
Intuitively, this happens because the scenario-based testing allocates more tests to the only "failure prone" logical scenario (subdomain D_k*), compared to mile-based testing.*

- ***Situations when mile-based testing is superior:*** *A similar argument leads to the opposite conclusion when mile-based testing heavily samples* D_k *(many mile-based test scenarios fall within* D_k*). If there are many other subdomains, scenario-based testing ends up "wasting" most of its effort on them (still assuming* $t_k = t/n$*), rather than the "failure prone" subdomain. Mathematically, mile-based testing provides more tests,* $t\sum_{x\in D_k} Op(x) \gg t_k$, *substituting* $t_k = t/n$, *we have:*

$$\sum_{x\in D_k} Op(x) \gg \frac{1}{n} \tag{13}$$

Substituting Eq. (13) *in place of Eq.* (9) *in Eq.* (11) *reverses the inequality, showing that scenario-based testing results in a lower expected pfs than mile-based testing.*

C. Modelling Fidelity of Synthetic Data

In the scope of using synthetic data for testing AV, substantial work has been done [21]–[24]. Those studies define fidelity in different ways, e.g.,: 1) distance between individual real and synthetic data points, 2) distance between (safety-relevant) outputs when processing real vs synthetic inputs [23], and 3) statistical distribution similarity of real vs synthetic datasets [24]. In this section, from the perspective of sameness in doing statistical inference for safety risk, we introduce *Risk Estimation Fidelity* (REF) as:

Definition 9 ((ϵ,α)-REF)**.** *Let* $\hat{\theta}^r$ *and* $\hat{\theta}^s$ *be the pfs estimates by an estimator*[6]*(e.g., Eq.* (3)*) using real and synthetic data, respectively, then the simulator achieves* (ϵ,α)*-REF if*

$$Pr(|\hat{\theta}^s - \hat{\theta}^r| \le \epsilon) \ge 1-\alpha \tag{14}$$

[6]The same estimator should be used for both real and synthetic data to ensure the estimation differences reflect data bias, not estimator bias.

where (ϵ,α) *is a specified tuple of two small positive constants.*

Eq. (14) is a classical confidence statement[7]: with confidence at least $1-\alpha$, the synthetic-data estimate deviates[8] from the real-data estimate by no more than ϵ. Indeed, even if the estimates using real and synthetic data are close enough, they may still detect very different failures. Thus, our goal is *not to replace* existing fidelity definitions and studies like [23], but rather to *complement* them from an additional perspective.

Let θ^r and θ^s be the (unknown) ground truth *pfs* of the AV deployed in real-world and simulators, respectively. The estimation difference can be decomposed into:

$$|\hat{\theta}^s - \hat{\theta}^r| = |\underbrace{(\hat{\theta}^s - \theta^s)}_{\text{syn. sampling var.}} + \underbrace{(\theta^s - \theta^r)}_{\text{syn. bias }\Delta} + \underbrace{(\theta^r - \hat{\theta}^r)}_{\text{real-w. sampling var.}}| \tag{15}$$

The first and third terms representing sampling uncertainty can be reduced given more real and synthetic data collected. The second term Δ captures the simulator's *inherent quality*, which is irreducible unless the simulator is reconfigured to be a better one. By understanding those three uncertainties, we propose a REF use case with the following steps:

1) **Collect real-world data** Execute a fixed number of real-world scenario tests within a given testing budget. Estimate $\hat{\theta}^r$ using a chosen estimator.
2) **Generate synthetic data** Use the current simulator configuration to run a number of synthetic scenario tests. Estimate $\hat{\theta}^s$ using the same estimator.
3) **Certify REF** Empirically check if the simulator satisfies (ϵ,α)-REF for predefined ϵ and confidence level $1-\alpha$.
4) **If REF certification fails, increase synthetic data** Increase the number of simulator tests to reduce the variance in $\hat{\theta}^s$. Recompute REF. Repeat until: REF is satisfied, or additional samples yield no significant improvement.
5) **If REF certification still fails, reconfigure simulator** Modify the simulator to reduce the gap between real and synthetic performance (i.e., reduce bias $\Delta = \theta^s - \theta^r$). Repeat from Step 2.
6) **If REF remains unsatisfied, quantify fidelity limit** Given the current data, determine the *smallest* ϵ (and/or *largest* α) for which REF is certified. The new (ϵ,α) reflects *the best fidelity we can certify with the available data.*
7) **Perform extended testing with certified simulator** If REF is satisfied, scale up simulator testing to estimate θ^r. Use certified (ϵ,α) to bound uncertainty in final safety claims.

Example 3. *We illustrate the REF workflow:*

1) *Step 1: Run* $t_r = 500$ *on-road scenarios sampled from the OP, observing* $k_r = 17$ *near-miss events (defined as the "failure" of our interest in this example). Using the Maximum Likelihood Estimator (MLE):* $\hat{\theta}^r = \frac{17}{500} = 0.034$.
2) *Step 2: In simulation, run* $t_s = 2000$ *comparable scenarios with* $k_s = 45$ *near-misses. By MLE,* $\hat{\theta}^s = \frac{45}{2000} = 0.0225$.

[7]This is not a Bayesian probability distribution of the actual error, but rather a frequentist confidence bound: if you repeated the process of testing and deriving the estimator, many times, the estimator would fall within the error range $(-\epsilon,+\epsilon)$ around the true value with at least $1-\alpha$ probability.

[8]In real safety cases, relative error often matters more than absolute error.

3) *Step 3: Presuming we first set error tolerance $\epsilon = 0.02$ and confidence level $1-\alpha = 0.95$. For large t_r and t_s, by the Central Limit Theorem, the two sample mean $\hat{\theta}^r$ and $\hat{\theta}^s$ follow Gaussian distributions with*

$$\mathrm{Var}(\hat{\theta}^r) \approx \frac{\hat{\theta}^r(1-\hat{\theta}^r)}{t_r} = \frac{0.034 \cdot 0.966}{500} = 6.57 \times 10^{-5},$$

$$\mathrm{Var}(\hat{\theta}^s) \approx \frac{\hat{\theta}^s(1-\hat{\theta}^s)}{t_s} = \frac{0.0225 \cdot 0.9775}{2000} = 1.1 \times 10^{-5}.$$

Thus the difference $\hat{\Delta} = \hat{\theta}^s - \hat{\theta}^r$ is also a Gaussian[9] *with*

$$\mu_{\hat{\Delta}} = -0.0115, \quad \sigma_{\hat{\Delta}} = \sqrt{\mathrm{Var}(\hat{\theta}^s) + \mathrm{Var}(\hat{\theta}^r)} \approx 0.0088.$$

$$\begin{aligned}\Pr\big(|\hat{\Delta}| \le 0.02\big) &= \Phi\Big(\frac{0.02 - \mu_{\hat{\Delta}}}{\sigma_{\hat{\Delta}}}\Big) - \Phi\Big(\frac{-0.02 - \mu_{\hat{\Delta}}}{\sigma_{\hat{\Delta}}}\Big) \\ &\approx 0.83 < 0.95,\end{aligned}$$

so the simulator fails $(0.02, 0.05)$-REF with the samples.

4) *Step 4: Doubling t_s to 4000 with $k_s = 102$ yields $\Pr(|\hat{\Delta}| \le 0.02) \approx 0.91 < 0.95$; $(0.02, 0.05)$-REF still uncertified.*
5) *Step 5: Improving the simulator, collect new data of $t_s = 2000$ and $k_s = 58$, and may get*

$$\hat{\Delta} = -0.005, \ \sigma_{\hat{\Delta}} \approx 0.0089, \ \Pr(|\hat{\Delta}| \le 0.02) > 0.95,$$

*so $(0.02, 0.05)$-**REF is certified**.*
6) *Step 6: Skipped, as the ideal, predefined (ϵ, α) tuple for REF is certified with the available data.*
7) *Step 7: Scale up the $(0.02, 0.05)$-REF simulator testing with $t_s = 50{,}000$ and $k_s = 1415$, we compute $\hat{\theta}^s = 0.0283$ and its standard error as $\sigma_{\hat{\theta}^s} = 0.000741$. The 95% confidence interval (CI) for θ^s is:*

$$\hat{\theta}^s \pm 1.96 \times \sigma_{\hat{\theta}^s} = [0.02685, \ 0.02975].$$

*Applying the REF bias bound $\epsilon = 0.02$, the **adjusted 90% CI** for θ^r becomes*[10]*:*

$$[0.02685 - 0.02, \ 0.02975 + 0.02] = [0.00685, \ 0.04975].$$

While the adjusted CI for θ_r may be (one-side) tighter than the CI directly calculated from real-world data along, it is certainly more actionable by certifying a smaller ϵ and/or adding more synthetic data. Enhanced statistical models are needed for Step 7 to capture more complexities, which is our future work. Furthermore, there are limitations like the failures are so rare that the sample means can no longer be approximated by normal distributions. We omit such cases, as our intent is to demonstrate the feasibility of REF, rather than provide a comprehensive formal treatment.

Another future direction is to define a Bayesian version of REF, where beliefs about the actual θ^r and θ^s are modeled, and being able to reason under rare failures. Once such a Bayesian REF is certified (e.g., with 95% probability that θ^r is no worse than θ^s), existing models for "No Worse Than" reasoning [36], [42], [43], [45], [46] can be applied to support safety claims based on both real and simulated evidence.

V. Discussion and Conclusion

The four high-level questions posed in the introduction are tightly interrelated. The stopping rule for scenario-based testing should depend on the level of residual risk that assessors are willing to accept. Estimating this residual risk, in turn, must account for the extra uncertainty introduced by synthetic data fidelity. Minimising such risk requires a clear understanding of the debug effectiveness of the test method within the available budget. So, *centring around the residual safety risk*, these questions can be jointly addressed by the proof-of-concept models proposed in this paper, with more careful designs to tackle practical challenges in future.

a) The metric: While we introduce *pfs* in our proof-of-concept models for scenario testing evidence, is *pfs* truly the best metric? Or should we consider higher abstraction level of "per trip" or "per mission"? What constitutes an acceptable *pfs* in AVs? Can we derive such thresholds from human-driven incident statistics at "per scenario" level? Answer to those question depends on how data are collected in practice for the specific AV under study, considering, e.g., if scenarios are i.i.d. and if *pfs* is constant for the data collected.

b) Dynamic OP: How can we accurately estimate the OP, i.e., the operational scenario distribution over the given ODD? More importantly, how can we cope with the dynamic OP over time (due to, e.g., new roads and roadworks)? While this is an established problem [48]–[50], models require retrofit to cope with new challenges imposed by AV, e.g., scalability.

c) Asymmetric fidelity: In practice, false positives (simulated failures that never occur on-road) may be tolerable, whereas false negatives (missed hazards in simulation) are catastrophic. Our REF definition can be improved to cope with such practical consideration to ease the REF certification.

d) Ultra-high reliability: For AVs, demonstrating ultra-high reliability—a level unachievable through testing alone—is a core challenge [8]. Bayesian methods (e.g., CBI) enable combining test data with prior knowledge, but identifying suitable AV-specific priors and formally translating them into beliefs about *pfs* remains an open, case-dependent problem.

Our goal in building a statistical foundation for scenario-based testing is not to replace existing approaches to AV safety, but to offer a complementary perspective: one that is explicitly driven by reasoning about *residual operational risk*. By mapping new challenges in AV testing to known problems in software reliability, we aim to bridge methodological gaps and enable more rigorous safety claims for AVs.

[9]The sum of two Gaussian random variables is also a Gaussian one.

[10]Proof sketch: We aim to construct a CI for θ^r, accounting for the following two sources of uncertainty. **Event A** (with probability α_A): the simulation CI misses θ^s, $\theta^s \notin [I_l, I_u]$, i.e., the $1-\alpha_A = 0.95$ CI for $\hat{\theta}^s$ does not contain the true θ^s; **Event B** (with probability α_B): REF fails to bound the bias, $|\theta^s - \theta^r| > \epsilon$, i.e., the REF certification fails to ensure $|\theta^s - \theta^r| \le \epsilon$. By the union bound, $P(A \cup B) \le \alpha_A + \alpha_B$, it is evident that the probability that either the simulation CI misses θ^s or the REF certification fails is at most $\alpha_a + \alpha_B$. Thus, the probability that both the simulation CI and the REF certification are correct is at least $1 - (\alpha_A + \alpha_B)$. So, if neither A nor B fails (the simulation CI contains θ^s and the REF ensures $|\theta^s - \theta^r| \le \epsilon$), then the adjusted CI (simulation CI expanded by $[I_l - \epsilon, I_u + \epsilon]$) contains θ_r.

REFERENCES

[1] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, "Autonomous vehicle technology: A guide for policymakers," Rand Corporation, Tech. Rep. RR-443-2-RC, 2016.

[2] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[3] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[4] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[5] H. Wang, W. Shao, C. Sun, K. Yang, D. Cao, and J. Li, "A Survey on an Emerging Safety Challenge for Autonomous Vehicles: Safety of the Intended Functionality," *Engineering*, vol. 33, pp. 17–34, 2024.

[6] N. Kalra and S. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transp. Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[7] X. Zhao, V. Robu, D. Flynn, K. Salako, and L. Strigini, "Assessing the safety and reliability of autonomous vehicles from road testing," in *Int. Symp. on Software Reliability Engineering (ISSRE)*, 2019, pp. 13–23.

[8] B. Littlewood and L. Strigini, "Validation of ultra-high dependability for software-based systems," *Comm. of the ACM*, vol. 36, pp. 69–80, 1993.

[9] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *ITSC'15*. IEEE, 2015, pp. 982–988.

[10] ISO 34502, "Road vehicles - test scenarios for automated driving systems - scenario based safety evaluation framework," 2022.

[11] BSI Flex 1889, "Formalised natural language description of scenarios for automated driving systems. Specification," 2023.

[12] 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. 87 456–87 477, 2020.

[13] C. Amersbach and H. Winner, "Defining required and feasible test coverage for scenario-based validation of highly automated vehicles," in *ITSC'19*. IEEE, 2019, pp. 425–430.

[14] F. Gao, J. Duan, Z. Han, and Y. He, "Automatic virtual test technology for intelligent driving systems considering both coverage and efficiency," *IEEE Tran. on Vehicular Technology*, vol. 69, pp. 14 365–14 376, 2020.

[15] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, "Automated scenario generation for regression testing of autonomous vehicles," in *Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2017, pp. 476–483.

[16] Z. Tahir and R. Alexander, "Coverage based testing for v&v and safety assurance of self-driving autonomous vehicles: A systematic literature review," in *IEEE AITest'20*, 2020, pp. 23–30.

[17] D. Hamlet and R. Taylor, "Partition testing does not inspire confidence," *IEEE Tran. on Softw. Engin.*, vol. 16, no. 12, pp. 1402–1411, 1990.

[18] P. G. Frankl, R. G. Hamlet, B. Littlewood, and L. Strigini, "Evaluating testing methods by delivered reliability [software]," *IEEE Tran. on Software Engineering*, vol. 24, no. 8, pp. 586–601, 1998.

[19] B. Littlewood, "Limits to dependability assurance—a controversy revisited," in *Keynote Paper presented at ICSE'07*, 2007.

[20] E. Adams, *Minimizing cost impact of software defects*. IBM Thomas J. Watson Research Division, 1980.

[21] A. Stocco, B. Pulfer, and P. Tonella, "Mind the gap! a study on the transferability of virtual versus physical-world testing of autonomous driving systems," *IEEE Tran. on Softw. Engin.*, vol. 49, no. 4, pp. 1928–1940, 2022.

[22] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen, "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence," *IEEE Tran. on Intelligent Vehicles*, vol. 9, no. 1, pp. 593–612, 2023.

[23] C.-H. Cheng, P. Stöckel, and X. Zhao, "Instance-level safety-aware fidelity of synthetic data and its calibration," in *27th Int. Conf. on Intelligent Transportation Systems*. IEEE, 2024, pp. 2354–2361.

[24] X. Yan, Z. Zou, S. Feng, H. Zhu, H. Sun, and H. X. Liu, "Learning naturalistic driving environment with statistical realism," *Nature communications*, vol. 14, no. 1, p. 2037, 2023.

[25] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *IV'18*, 2018.

[26] D. Cotroneo, R. Pietrantuono, and S. Russo, "Relai testing: a technique to assess and improve software reliability," *IEEE Tran. on Software Engineering*, vol. 42, no. 5, pp. 452–475, 2015.

[27] J. D. Musa, "Operational profiles in software-reliability engineering," *IEEE Software*, vol. 10, no. 02, pp. 14–32, 1993.

[28] S. Ntafos, "On random and partition testing," in *Proc. of the ACM SIGSOFT Int. Symp. on Software Testing and Analysis*, 1998, pp. 42–48.

[29] T. Chen and Y. Yu, "On the relationship between partition and random testing," *IEEE Tran. on Softw. Engi.*, vol. 20, no. 12, pp. 977–980, 1994.

[30] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Tran. on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2016.

[31] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Tran. on Intelligent Transp. Syst.*, vol. 19, no. 3, pp. 733–744, 2017.

[32] K. Ren, J. Yang, Q. Lu, Y. Zhang, J. Hu, and S. Feng, "Intelligent testing environment generation for autonomous vehicles with implicit distributions of traffic behaviors," *Transportation Research Part C: Emerging Technologies*, vol. 174, p. 105106, 2025.

[33] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature communications*, vol. 12, no. 1, p. 748, 2021.

[34] W. Ding, B. Chen, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 2243–2250.

[35] P. D. O'connor and A. V. Kleyner, *Practical reliability engineering*. john wiley & sons, 2011.

[36] X. Zhao, K. Salako, L. Strigini, V. Robu, and D. Flynn, "Assessing safety-critical systems from operational testing: A study on autonomous vehicles," *Information and Software Techn.*, vol. 128, p. 106393, 2020.

[37] J. Min, Y. Hong, C. B. King, and W. Q. Meeker, "Reliability analysis of artificial intelligence systems using recurrent events data from autonomous vehicles," *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 71, no. 4, pp. 987–1013, 04 2022.

[38] S. Zheng, L. Lu, Y. Hong, and J. Liu, "Planning reliability assurance tests for autonomous vehicles based on disengagement events data," *IISE Transactions*, vol. 0, no. ja, pp. 1–25, 2025.

[39] C. Buerkle, F. Oboril, M. Paulitsch, P. Popov, and L. Strigini, "Modelling road hazards and the effect on av safety of hazardous failures," in *Int. Conf. on Intelligent Transportation Systems*, 2022, pp. 1886–1893.

[40] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *Int. Conf. on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[41] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska *et al.*, "Level 5 perception dataset 2020," *Woven Planet Holdings, Tokyo, Japan*, 2019.

[42] B. Littlewood, K. Salako, L. Strigini, and X. Zhao, "On reliability assessment when a software-based system is replaced by a thought-to-be-better one," *Reliability Engineering & System Safety*, vol. 197, p. 106752, 2020.

[43] K. Salako, L. Strigini, and X. Zhao, "Conservative confidence bounds in safety, from generalised claims of improvement & statistical evidence," in *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, 2021.

[44] P. Bishop, A. Povyakalo, and L. Strigini, "Bootstrapping confidence in future safety from past safe operation," in *33rd Int. Symp. on Software Reliability Engineering (ISSRE)*. IEEE, 2022, pp. 97–108.

[45] R. Aghazadeh Chakherlou, L. Strigini, "Using pre change operational evidence for predicting post change reliability, given prior confidence in fault-freeness," *34th European Safety and Reliability Conf.*, 2024.

[46] R. A. Chakherlou and L. Strigini, "Impact of prior beliefs on dependability prediction for a changed system using pre-change operational evidence," in *QRS'24*. IEEE, 2024, pp. 572–583.

[47] X. Zhao, B. Littlewood, A. Povyakalo, L. Strigini, and D. Wright, "Modeling the probability of failure on demand (pfd) of a 1-out-of-2 system in which one channel is "quasi-perfect"," *Reliability Engineering & System Safety*, vol. 158, pp. 230–245, 2017.

[48] T. Adams, "Total variance approach to software reliability estimation," *IEEE Trans. on Software Engineering*, vol. 22, no. 9, pp. 687–688, 1996.

[49] P. Bishop and A. Povyakalo, "Deriving a frequentist conservative confidence bound for probability of failure per demand for systems with different operational and test profiles," *Reliability Engineering & System Safety*, vol. 158, pp. 246–253, 2017.

[50] R. Pietrantuono, P. Popov, and S. Russo, "Reliability assessment of service-based software under operational profile uncertainty," *Reliability Engineering & System Safety*, vol. 204, p. 107193, 2020.