Confidence Composition for Monitors of Verification Assumptions

Ivan Ruchkin
University of Pennsylvania
Philadelphia, Pennsylvania

Matthew Cleaveland
University of Pennsylvania
Philadelphia, Pennsylvania

Radoslav Ivanov
Rensselaer Polytechnic Inst.
Troy, New York

Pengyuan Lu
University of Pennsylvania
Philadelphia, Pennsylvania

Taylor Carpenter
University of Pennsylvania
Philadelphia, Pennsylvania

Oleg Sokolsky
University of Pennsylvania
Philadelphia, Pennsylvania

Insup Lee
University of Pennsylvania
Philadelphia, Pennsylvania

ABSTRACT

Closed-loop verification of cyber-physical systems with neural network controllers offers strong safety guarantees under certain assumptions. It is, however, difficult to determine whether these guarantees apply at run time because verification assumptions may be violated. To predict safety violations in a verified system, we propose a three-step confidence composition (CoCo) framework for monitoring verification assumptions. First, we represent the sufficient condition for verified safety with a propositional logical formula over assumptions. Second, we build calibrated confidence monitors that evaluate the probability that each assumption holds. Third, we obtain the confidence in the verification guarantees by composing the assumption monitors using a composition function suitable for the logical formula. Our CoCo framework provides theoretical bounds on the calibration and conservatism of compositional monitors. Two case studies show that compositional monitors are calibrated better than their constituents and successfully predict safety violations.

KEYWORDS

confidence composition, run-time monitoring, calibration error, closed-loop verification of neural networks, cyber-physical systems

1 INTRODUCTION

Autonomous cyber-physical systems, such as self-driving cars and service robots, are increasingly deployed in complex and safety-critical environments in our society [10, 40, 48]. Recently, the breakthrough capabilities in handling such environments came from the use of learning components, which may behave unpredictably. To consistently rely on such capabilities, one needs to ensure that the system would not to endanger the lives and property around it, or at least that an early enough warning is given to avert the disaster.

When assuring a complex cyber-physical system, one can obtain strong safety guarantees from closed-loop reachability verification, recently extended to explicitly check neural network (NN) controllers [18, 45]. To provide its guarantees, the verification relies on assumptions about system's dynamics, perception, and environment. Should the system find itself in circumstances not matching these assumptions, the verification guarantees are void — and remarkably difficult to re-obtain at run time due to limited scalability.

On another front, many run-time monitoring techniques were developed to detect anomalies, such as model inconsistencies and out-of-distribution samples [3, 5, 28]. These tools can provide valuable situational insights, but their outputs often lack a direct connection to the verification guarantees or system-level safety. For example, it is not clear to which extent an out-of-distribution image of a stop sign invalidates a collision-safety guarantee for an autonomous car.

Thus, it is both challenging and important to quantify and monitor the trust in design-time verification guarantees at run time. In particular, it is vital to know when the guarantees no longer apply, so as to switch to a backup controller, execute a recovery maneuver, or ask for human assistance. The monitoring of verification guarantees has the potential to predict otherwise unforeseen failures in situations for which the system was not trained or designed.

To monitor verification guarantees, we propose quantifying the *confidence* in the *assumptions* of verification. By confidence we mean an estimate of the probability that the assumption holds. Although an assumption may not be directly observable, its monitor would over time accumulate confidence, which, if properly calibrated, would be close to the true chance of satisfying the assumption given the observations. If all the assumptions are satisfied, our verification retroactively guarantees safety. Such assumptions can be monitored with off-the-shelf techniques [5, 7, 32, 39, 47], and their confidences would be combined into a single confidence in the guarantees. In a safety-critical system, this confidence should not be over-estimated.

This paper introduces the CoCo *framework for composing confidences* from monitors of verification assumptions, consisting of three steps: (i) *verify* the system under explicit assumptions, such that a propositional formula over these assumptions entails the system's safety, (ii) build a well-calibrated *confidence monitor* for each assumption, (iii) use a *composition function* informed by the formula from the first step to combine the monitor outputs into a composed confidence. This confidence quantifies the chance that the verification guarantees apply at that moment.

We develop the theoretical conditions under which the composed confidence is calibrated and conservative, up to a bounded error, with respect to the true probability of safety. These conditions are that (a) the system model under verification can explain most safe behaviors, (b) a violation of assumptions would likely lead to a failure, and (c) the composition function is calibrated/conservative with respect to the assumptions. We also prove calibration error bounds for two composition functions — product and weighted average — and a conservatism bound for the product.

We evaluate CoCo on two systems with NN controllers: a mountain car and an underwater vehicle. Experiments show that our compositional monitors are useful for safety prediction, outperform the individual monitors, and can be tuned for conservatism. Our data-driven composition functions improve the performance further if provided the data relating the monitors and the assumptions.

To summarize, this paper makes four contributions:

- The CoCo framework for composing confidence monitors of verification assumptions with five composition functions.
- Sufficient conditions for bounded calibration of composite confidence to the safety chance, with the expectations from models, assumptions, monitors, and composition functions.
- Upper bounds on the expected calibration error of two composition functions, and the conservatism error of one.
- Two case studies that demonstrate the utility of the framework and the trade-offs between composition functions.

The rest of the paper proceeds as follows. The necessary background on verification and monitoring is given in the next section. Section 3 surveys the existing research. Section 4 presents the key contributions: the framework, the end-to-end calibration conditions, and bounds on the errors of composition functions. Section 5 describes two case studies and the experimental results. The paper wraps up with a brief discussion in Section 6.

2 BACKGROUND

Here we describe the preliminaries of verification and monitoring.

2.1 Verification

Definition 1 (System). *A system $s = (X, X_0, Y, U, h, F_d, F_m)$ consists of the following elements:*

- State space X: continuous, unbounded, finite-dimensional, containing states x (which include the discrete time)
- Initial states $X_0 \subset X$
- Observation space Y, containing observations y
- Action space U
- Controller $h : Y \longrightarrow U$, implemented with a neural network
- Dynamics models F_d: a set of functions $f_d : X, U \longrightarrow X$
- Measurement models F_m: a set of functions $f_m : X \longrightarrow Y$

A system determines a set of state traces $\boldsymbol{X}(s)$ and a set of observation traces $\boldsymbol{Y}(s)$ resulting from executing every combination of functions from $F_d \times F_m$ on every initial state in X_0 indefinitely. A particular realization of a system is a pair of state and observation vectors $\boldsymbol{x}, \boldsymbol{y}$ that occur for specific $x_0 \in X_0, f_d \in F_d$, and $f_m \in F_m$.

A *safety property* φ is a Boolean predicate over traces: $\varphi(\boldsymbol{x}) \in \{\mathsf{T}, \mathsf{F}\}$. A property φ is satisfied on system s, denoted $s \models \varphi$, iff every trace from that system satisfies φ: $\forall \boldsymbol{x} \in \boldsymbol{X}(s), \varphi(\boldsymbol{x}) = \mathsf{T}$. Thus, this paper considers arbitrary deterministic temporal safety properties.

A *verification assumption* A is a restriction on the system's "unknowns" — the states and models of the system — so, $A \subseteq X \times F_d \times F_m$. The assumption holds on a trace $(\boldsymbol{x}, \boldsymbol{y})$ if for any combination of x_0, f_d, and f_m that realizes this trace it is true that $(x_0, f_d, f_m) \in A$. When assumptions $A_1 \dots A_n$ are combined with a propositional logical formula ψ, $A_\psi = \psi(A_1 \dots A_n)$ is also an assumption. The meaning of propositional operators $(\wedge, \vee, \neg, \implies)$ is defined by the corresponding set operations (intersection for $\wedge$, union for $\vee$, etc). A system $s = (X, X_0, Y, U, h, F_d, F_m)$ *under assumption* $A = (X'_0, F'_d, F'_m)$, denoted as s_A, is an intersection of the initial states and respective models: $s_A = (X, X_0 \cap X'_0, Y, U, h, F_d \cap F'_d, F_m \cap F'_m)$.

For a given system s, a verification result of property φ, denoted as $V_{s,\varphi}$, is a function that maps any assumption A to $\{\mathsf{T}, \mathsf{F}\}$. It represents the outcome of a verification effort under that assumption, regardless of the exact method. Value T is assigned only if φ was guaranteed by the verification algorithm, whereas F is assigned in all the other cases (counterexample exists, uncertainty too high, time limit reached, etc). Since verification is over-approximate and exhaustive, it never produces a false safety outcome: $V_{s,\varphi}(A) = \mathsf{T} \implies s_A \models \varphi$. Such an assumption A is called *sufficient* for φ.

2.2 Confidence Monitoring

Intuitively, we want to compute the *confidence* in (i.e., an estimate of the probability of) the system satisfying a safety property φ in the future given a prefix of observations $\boldsymbol{y}$. Confidences are computed by CPS monitors in uncertain conditions, when the exact state, dynamics, and measurement model are not known. Therefore, we represent the selection of the actual system as a random sampling of the system's unknowns — the initial state x_0, dynamics f_d, and measurement function f_m — from some unknown distribution $\mathcal{D}$ over X, F_d, and F_m.

Once we fix the distribution $\mathcal{D}$, it induces the distribution $\mathcal{D}_{\boldsymbol{x},\boldsymbol{y}}$ on the system's realization $(\boldsymbol{x}, \boldsymbol{y})$. Therefore, our monitoring goal is to estimate the probability of safety given observations up to the current moment, namely $\mathsf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_{(\boldsymbol{x},\boldsymbol{y})}}\big(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{y}_{1..n}\big)$, where $\boldsymbol{y}_{1..n}$ means the first n elements of $\boldsymbol{y}$. We pursue this goal by monitoring confidence in assumptions. Since an assumption A can be seen as a predicate over random $(x_0, f_d, f_m) \sim \mathcal{D}$, its satisfaction is also random: $A \sim \mathcal{D}_A$, where $\mathcal{D}_A$ is induced by $\mathcal{D}$.

A confidence monitor $M : Y \longrightarrow [0, 1]$ for assumption A takes $\boldsymbol{y}_{1..n}$ and outputs its estimate of $\mathsf{Pr}_{\boldsymbol{y}\sim\mathcal{D}_{\boldsymbol{y}}}(\boldsymbol{y}_{1..n} \in Y(s_A))$, that is, its degree of belief that the observations came from a system where A holds. The monitor's output, $M(\boldsymbol{y})$, is stochastic because it depends on $\boldsymbol{y}$. Since monitors estimate probabilities, we measure the quality of monitors using three types of calibration error with respect to A:

- *Expected calibration error* (*ECE*):

$$ECE(M, A) := \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_{\boldsymbol{y}}} \left[\left|\mathsf{Pr}_{A\sim\mathcal{D}_A}\big(A \mid M(\boldsymbol{y})\big) - M(\boldsymbol{y})\right|\right]$$

- *Maximum calibration error* (*MCE*):

$$MCE(M, A) := \max_{p\in[0,1]} \left[\left|\mathsf{Pr}_{A\sim\mathcal{D}_A,\boldsymbol{y}\sim\mathcal{D}_{\boldsymbol{y}}}\big(A \mid M(\boldsymbol{y}) = p\big) - p\right|\right]$$

- *Conservative calibration error (CCE):*

$$CCE(M, A) := \max_{p\in[0,1]} \left[p - \mathsf{Pr}_{A\sim\mathcal{D}_A,\boldsymbol{y}\sim\mathcal{D}_{\boldsymbol{y}}}\big(A \mid M(\boldsymbol{y}) = p\big)\right]$$

ECE and *MCE* are widely used measures of calibration [13, 25]. The concept of *CCE* is novel — we introduce it to asymmetrically quantify safety in critical systems: false alarms are safe, but missed alarms are not. *MCE* is the strictest measure of monitor quality because $MCE \geq ECE$ and $MCE \geq CCE$. When $ECE = 0$, the monitor is *calibrated in expectation.* When $MCE = 0$, the monitor is *perfectly calibrated.* When $CCE \leq 0$, the monitor is *conservative.*

For brevity, we will omit the distributions when they are clear from the context and refer to monitor output $M(\boldsymbol{y})$ as just M. We assume that this output is characterized by a continuous probability density $\mathsf{Pr}(M)$ with finite expectation $\mathbb{E}[M]$ and variance $\mathrm{Var}[M]$.

3 RELATED WORK

The research related to this paper spans several areas: detection and estimation, aggregation of probabilities, run-time monitoring and assurance, and assumption monitoring.

Anomaly detection is a well-studied problem in control theory and signal processing. In particular, there are multiple well-written books on sequential detection and estimation [32, 39, 47]. We rely on two groups of such methods. First, filtering and parameter estimation can be used to implement the monitors that estimate states and noise parameters. Specifically, we implemented monitors using standard Monte Carlo and particle filtering [7]. Second, monitoring model validity is related to classical change detection [49] as well as more recent computational model validation methods [5, 27, 33], one of which we use to monitor assumptions on our model. Unlike classical detection methods, our work focuses on calibration [13], i.e., the faithfulness of the detector to the true frequency of the underlying event. While classical model-based detectors [49] can be considered "calibrated" by design under distributional assumptions (since probabilities can be explicitly computed), recent computational approaches [5] and classical detectors under unknown distributions are in essence "black-box" and need to be calibrated post-hoc in order to provide performance guarantees.

Combining probability estimates is well-studied in statistics and artificial intelligence [34, 37, 50]. In a typical setting, such as forecast aggregation or ensemble learning, the combined methods estimate the probability of the *same* underlying phenomenon. In contrast, our monitors predict fundamentally *different* assumptions combined with logical operators. This setting can be interpreted as probabilistic graphical models with calibration constraints (as opposed to factor weights or conditional probabilities) [19], and our product composition corresponds to the noisy-OR graphical model [29]. Copulas [26] encode low-dimensional joint distributions with given marginals and can be used to model dependencies between verification assumptions, which we have so far assumed conditionally independent. Broadly, the literature on combining probabilities inspires the functions we use for confidence composition.

Confidence is emerging as a key concept for expressing uncertainty in learning-enabled systems in such scenarios as detecting objects [2] and anticipating human motion [10]. Confidences can be endowed with distributions to enable effective and general inference [41]. However, the poor calibration of confidences remains a major issue, especially for neural networks [13, 46]. We study the calibration of black-box monitors in a safety-critical compositional setting without detailed assumptions on confidence distributions.

Run-time monitoring is increasingly important in assurance of cyber-physical systems, with multiple run-time assurance frameworks proposed recently [1, 3, 8, 22, 40, 44]. Some of them focus on safety-preserving decision-making rather than accurate monitoring [4, 8, 40]. Others focus on specifying and monitoring safety properties in Linear/Metric/Signal Temporal logic with uncertainty [6, 43, 48], whereas we indirectly predict the satisfaction of safety properties. Yet others provide well-calibrated confidence with non-compositional techniques such as conformal prediction [3] and dynamic Bayesian networks [1] — and thus can be incorporated into our framework as individual monitors. A closely related recent framework is ReSonAte [14], based on representing risks with bowtie diagrams (a counterpart of our propositional formulas) and estimating risk by using conditional distributions between system states and failures. ReSonAte's approach is analogous to our Bayesian composition, which learns a joint distribution of monitors conditioned on assumptions — and naturally requires joint monitor samples or additional, strong independence assumptions.

Assumptions have long been considered a potential cause of failures in safety-critical systems [11, 30, 38]. The probabilistic and compositional formalization of assumptions is most common in frameworks for assume-guarantee reasoning and compositional verification [9, 20, 36]. Our paper investigates a complementary direction of connecting the guarantees of closed-loop neural network verification [17] with uncertain and imprecisely modeled run-time environments by using assumptions as an "interface" between the two. Prior works have pioneered assumption monitoring in model-based, non-deterministic settings: explicitly specified monitors [42] and monitors of proof obligations with partially observable variables [6, 23]. These model-based approaches have the advantage of verifying monitors within the semantics of their respective models. Pursuing our vision of compositional confidence-based assurance [35], this paper extends confidence monitoring of assumptions to a setting where monitors do not conform to any given semantics and can exhibit unknown stochastic behavior.

4 CONFIDENCE COMPOSITION FRAMEWORK

Our CoCo framework uses the following intuition. Suppose that we have monitors $M_1 \dots M_n$ for assumptions $A_1 \dots A_n$, some combination of which, A_ψ, is sufficient for, and thus predictive of, safety. From $M_1 \dots M_n$, we can build a compositional monitor M_C of A_ψ that will estimate $\Pr(A_\psi)$, which is used as an indirect estimate of the chance of safety. Our composite monitor $M_C : [0,1]^n \to [0,1]$ has form $C(M_1 \dots M_n)$, where C is a *composition function* selected depending on ψ. Our framework formalizes the argument that if safety depends on the assumptions that have monitors with bounded *ECE* (or *CCE*), then an appropriate compositional monitor will have bounded *ECE* (*CCE* resp.) error with respect to the safety chance. This argument needs to account for model inaccuracies, overly conservative assumptions, and imperfect monitors.

Our framework imposes certain requirements on the models, explained in the next subsection, and proceeds in three steps:

(1) Perform verification and elicit the assumptions sufficient for safety (Section 4.2)
(2) Build and calibrate a confidence monitor for each assumption (Section 4.3)
(3) Compose monitors using a composition function with desirable bounds on the calibration error (Section 4.4)

In each step, we identify the framework's requirements and briefly outline how they can be achieved. Section 4.5 capitalizes on these requirements by providing end-to-end bounds that link the composed confidence and the true chance of safety.

4.1 Model Requirements

From the modeling standpoint, we distinguish two subsystems of the overarching system s: the unknown, true, "real" subsystem s^* and the modeled, known subsystem $\hat{s}$ that will undergo verification. Systems s^* and $\hat{s}$ are *compatible*, i.e., they share the same X, Y, U, and h. The latter can be shared because we explicitly encode and verify the NN controller in our model. We fix some safety property φ and verification method $V_{\hat{s},\varphi}$ and introduce the notion of *safety relevance* between two compatible models.

Definition 2 (Safety-relevant model). *A model s_1 is* safety-relevant *up to a bound e for a compatible model s_2 if it accounts for the safe behaviors of s_2 most of the time. Formally, for a random state trace $\boldsymbol{x}$, safety property φ, and some small $e \in [0,1]$,*

$$\Pr\big(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \in \boldsymbol{X}(s_2) \wedge \boldsymbol{x} \notin \boldsymbol{X}(s_1)\big) \leq e.$$

We expect the system model $\hat{s}$ to be safety-relevant for the real system s^* and in that case just say it is safety-relevant. We also expect $\hat{s}$ to be *verifiable.*

Definition 3 (Verifiable model). *A model $\hat{s}$ is* verifiable *if there is a non-trivial assumption A (i.e., containing many states and/or models) such that all traces of $\hat{s}_A$ are safe:*

$$\exists A, |A| > 0 \wedge V_{\hat{s},\varphi}(A) = \mathsf{T}.$$

Safety relevance intuitively means that we are unlikely to get a safe trace not represented by our model. This gives verification an opportunity to verify a system that "explains" a large part of truly safe behaviors. Then, failing the verification would correspond to a low true chance of safety. Without safety relevance, whether verification holds may be orthogonal to whether the system is safe.

Verifiability of a sizeable set of assumptions is important because if the model is safe only under trivial assumptions, few observed traces would satisfy them. Then, the monitors would be forced to alarm perpetually and, hence, poorly predict safety. For instance, if we verified a system model only with zero measurement noise, a monitor would almost always invalidate this model on a real system. Verifiability is challenging to achieve due to the scalability and uncertainty limits of over-approximating reachability algorithms.

There is a trade-off between safety relevance and verifiability: expanding the set of explained behaviors leads to more parameters and a larger scope of the model, making it harder to verify. In the case studies, we negotiated this trade-off by starting with simple verifiable low-dimensional models and iteratively extending their relevance while preserving their verifiability.

4.2 Verification Assumptions

Assumptions are made to constrain the modeled system $\hat{s}$ and pass safety verification. For assumption monitoring to be useful, we expect our assumptions to be *sufficient* and *safety-relevant.*

Definition 4 (Sufficient assumption). *An assumption A is* sufficient *for property φ if the verification of φ on $\hat{s}$ succeeds for A:*

$$V_{\hat{s},\varphi}(A) = \mathsf{T}$$

Definition 5 (Safety-relevant assumption). *An assumption A applied to $\hat{s}$ is* safety-relevant *up to some bound $e \in [0,1]$ if the subsystem $\hat{s}_A$ is safety-relevant up to e for $\hat{s}$ as per Definition 2.*

The sufficiency enables the system's verification and is typically straightforward to achieve via grid search. We partition the joint space of initial states and model parameters into hypercubes, optionally simulate the model to quickly rule out the unsafe regions, and then verify each remaining hypercube in parallel. The union of hypercubes where $V_{\hat{s},\varphi} = \mathsf{T}$ then becomes our sufficient assumption A.

Notice that the safety-relevance of assumptions is defined analogously to the safety-relevance of the models, and for the same reason: we want assumption failures to correlate with safety failures. The combined safety-relevance of the models and its assumptions gives us a bound on the chance of true safety:

Theorem 1 (Bounded safety under failed assumptions). *If model $\hat{s}$ is safety-relevant up to e_1 and assumption A is sufficient and safety-relevant for $\hat{s}$ up to e_2, then the safety chance under violated assumptions is bounded by $e_1 + e_2$. I.e., for a random trace $\boldsymbol{x} \in \boldsymbol{X}(s^*)$,*

$$\Pr\big(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A)\big) \leq e_1 + e_2.$$

Proof. See Appendix A. □

We also expect that the sufficient and safety-relevant assumption can be monitored with the available monitors. Specifically, we expect that we can find a *monitorable decomposition* of A into sub-assumptions $A_1 \dots A_n$ such that (i) a propositional formula ψ ensures the original assumption: $\psi(A_1 \dots A_n) \implies A$, and (ii) for each sub-assumption A_i, there exists a monitor M_i with a (preferably tight) bound on *ECE* and *CCE*.

The monitorable decomposition is a key idea behind this paper: while it is difficult to build a monolithic monitor for the exact and complete verification assumptions, it is possible to isolate monitorable sub-assumptions. We choose to decompose the assumption using propositional logic because the logical operators directly correspond to the set operations on the states and possible models. Notice that $\psi(A_1 \dots A_n)$ is required to imply A — not the other way around — to ensure conservative monitoring:

$$\Pr\big(\psi(A_1 \dots A_n)\big) \leq \Pr(A)$$

In practice, the choice of how to decompose assumptions depends on the available information in the observations, the available monitoring techniques, and the scalability of monitors at run time. Often, the logical structure arises from the hazard analysis of the system. For example, if at least one of the redundant sensors functions correctly, the system can guarantee performance. This corresponds to a disjunction of the assumptions.

In our case studies, closed-loop reachability verification of hybrid systems with NN controllers relies on three categories of assumptions, suggested by Definition 1. First, verification assumes that a system starts in *initial states* from where it can avoid safety violations. Second, verification assumes that the reality is approximately described by the *dynamics equations*, which are used to propagate the reachable sets over time. Third, verification assumes that control inputs are related to the true state by a constrained set of *observation models* that capture known sensor uncertainties. An assumption can span more than one category, and the particular decomposition depends on the specifics of the model and available monitors.

4.3 Confidence Monitors

The goal is, for each sub-assumption A_i, to obtain a confidence monitor M_i that estimates $\Pr(A_i \mid \boldsymbol{y})$ with bounded errors *ECE* and *CCE*. In our case studies, we built a state estimation-based monitor to determine whether the state at $t = 0$ (or the current

t) was part of the verification assumptions. Another monitor determines if the latest observations were consistent (up to some error bound) with the dynamical model under bounded observation noise. Our monitors were based on the existing detection and estimation techniques [5, 7]: Monte Carlo estimation using the dynamical model, particle filtering based on the dynamical model, and statistical model invalidation.

The produced monitors are often miscalibrated. There is a tradeoff between reducing *ECE* and *CCE* of a monitor: a monitor with small *ECE* may sometimes overestimate the probability of an assumption holding, leading to a sizeable *CCE*; on the other hand, a conservative monitor may significantly underestimate the probability and, hence, have large *ECE*.

Since we treat monitors as black boxes, we reduce their calibration errors with post-hoc calibration, which requires a validation dataset. Ideally, each monitor can use its own dataset without samples from other monitors, enabling independent development and tuning of the monitors.

We calibrate each monitor M_i with Platt scaling [31], a popular calibration technique, to produce a calibrated monitor M_i'. This technique is based on a linear transformation of the monitor's log-odds (LO). For every monitor output m, we compute the calibrated value m' as follows:

$$m' = \frac{1}{1 + \exp(c\,\mathrm{LO}(m) + d)}, \qquad \mathrm{LO}(m) = \log(\frac{m}{1-m}),$$

where c and d are calibration parameters to be determined.

To negotiate the tradeoff between *ECE* and *CCE*, we fit calibration parameters c and d using weighted cross-entropy loss. The weight $\lambda \in [0, 1]$ sets the relative importance of *CCE* over *ECE* by penalizing overconfidence ($\lambda = 0.5$ in standard Platt scaling). The fitting is done on a validation dataset containing pairs of monitoring outputs m_j and indicators a_j of the assumption holding at the time: $\{(m_j, a_j)\}$, and we fit over the calibrated scores m_j':

$$\arg\min_{c,d} - \sum_j (1-\lambda) a_j \log(m_j') + \lambda(1 - a_j)\log(1 - m_j') \quad (1)$$

4.4 Composition of Confidence Monitors

Our goal here is to build a compositional monitor M_C for A_ψ given $M_1 \dots M_n$ for $A_1 \dots A_n$ and provide bounds on its calibration error.

Problem (Composition of Calibration Errors).
Given A_ψ and $M_1 \dots M_n$ calibrated to $A_1 \dots A_n$ with known bounds, find function C with bounds on $ECE(M_C, A_\psi)$ and $CCE(M_C, A_\psi)$.

To solve this problem in full generality, one would need to know the joint distribution of random variables in what we call the *monitoring probability space* induced by $\mathcal{D}$:

- Bernoulli variable Φ indicating the "safety event" $\varphi(\boldsymbol{x}) = \mathsf{T}$. We use Φ as an equivalent shorthand.
- Bernoulli variables $A_1 \dots A_n$, and A_ψ corresponding to the satisfaction of assumptions and formula $\psi(A_1 \dots A_n)$.
- Continuous variables $M_1 \dots M_n$ and M_C. corresponding to the outputs of the monitors and their composition.

We will not assume knowing that joint distribution, nor shall we try to estimate it. Instead, this paper takes an early step to solving that problem by constraining it with judicious simplifications:

- We know the propositional formula ψ.
- Assumption monitors have bounded *MCE* and *CCE*.
- Assumptions are conditionally independent given composite confidence, e.g., $A_i \perp A_j \mid M_C$.
- Given their monitor, assumptions are independent of composition: $A_i \perp M_C \mid M_i$.
- Conditioning monitor variances on composition does not increase them: $\mathrm{Var}(M_i \mid M_C) \leq \mathrm{Var}(M_i)$.

The first two simplifications mirror the process of independent, modular development of monitors. The conditional independence of assumptions has been approximately true in our case studies and is conservative unless the assumptions share a cause of violations. Future work can investigate other assumption dependencies. (Without any information about assumption dependencies, formula ψ is of limited use.) The fourth simplification indicates that a monitor provides all the relevant information about its assumption, and composition has none to add. The last simplification, informally, states the composite confidence is "informative" to monitors: knowing its value limits the likely values of individual monitors. Although this statement is difficult to prove, it has always held in our experiments.

The above enables the rest of the section to proceed in three steps:

1. Identify plausible families of functions C (Section 4.4.1)
2. Provide bounds on $ECE(M_C, A_\psi)$ and $CCE(M_C, A_\psi)$ given calibration bounds of individual monitor (Section 4.4.2)
3. Provide bounds on $ECE(M_C, \Phi)$ and $CCE(M_C, \Phi)$ given bounds on $ECE(M_C, A_\psi)$ and $CCE(M_C, A_\psi)$ respectively (Section 4.5)

In the first two steps, we focus on the monitoring probability sub-space without Φ:

$$\Pr(A_1 \dots A_n, A_\psi, M_1 \dots M_n, M_C)$$

Given the complexity of this sub-space, this paper analyzes bounds on $ECE(M_C, A_\psi)$ and $CCE(M_C, A_\psi)$ only for the scenario with two assumptions (which proves sufficient in the case studies):

$$\Pr(A_1, A_2, A_\psi, M_1, M_2, M_C)$$

In the third step, we focus on the sub-space without the individual assumptions and monitors:

$$\Pr(\Phi, A_\psi, M_C)$$

4.4.1 Composition Functions.

We identify composition functions in two steps: equivalently simplifying the expression $\Pr(A_\psi)$ and proposing plausible functions for conjunctions of assumptions.

The structure of C comes from the structure ψ. To determine C for any $\psi(M_1 \dots M_n)$, we simplify the expression $\Pr(\psi(M_1 \dots M_n))$ by converting $\psi(M_1 \dots M_n)$ to DNF and advancing[1] the probability operator until all probability operators are either over individual assumptions or conjunctions of assumptions. We replace marginal probabilities $\Pr(A_i)$ with M_i, so it is sufficient to determine C for conjunctions of assumptions — and the rest of the expression is determined by the simplified version of $\Pr(A_\psi)$.

In the rest of this section, we will take initial steps towards addressing a key compositional sub-problem — providing functions C and their calibration bounds for $\Pr(A_1 \wedge A_2)$.

[1] Using standard identities such as the inclusion-exclusion principle, e.g., $\Pr(A_1 \vee A_2) = \Pr(A_1) + \Pr(A_2) - \Pr(A_1 \wedge A_2)$ and $\Pr(\neg A_1) = 1 - \Pr(A)$.

Problem (Binary Conjunctive Composition of *ECE*/*CCE*). *Given M_1 and M_2 calibrated to A_1 and A_2 respectively, find function C with bounds on $ECE(M_C, A_1 \wedge A_2)$ and $CCE(M_C, A_1 \wedge A_2)$.*

First, we obtain candidate functions C via plausible restrictions of the probability space. Since these functions are difficult to compare theoretically, we leave the task of finding the best composition function for an arbitrary conjunction of assumptions for future work. We do, however, compare them experimentally in Section 5.

Product: If one presumes that the assumptions are mutually independent and monitors are perfectly calibrated, then:

$$\Pr(A_1 \wedge \cdots \wedge A_n) = \prod_{i=1..n} \Pr(A_i) = \prod_{i=1..n} M_i$$

Thus, here $C(M_1 \dots M_n) = \prod_{i=1..n} M_i$.

Weighted averaging: If we presume that the monitors independently predict the combination of assumptions, then inverse-variance weighting minimizes the variance of the estimate [15]:

$$\Pr(A_1 \wedge \cdots \wedge A_n) = \sum_{i=1..n} w_i M_i, \quad w_i = \frac{1/Var(M_i)}{\sum_{j=1..n} 1/Var(M_j)}$$

Thus, here $C(M_1 \dots M_n) = \sum_{i=1..n} w_i M_i, \sum_{i=1..n} w_i = 1$.

The above two functions are modular and driven by theoretical considerations: they do not require information about the joint distribution of monitors. We also propose two data-driven functions that are not modular: they require samples from the joint distribution of the monitors and assumptions. Due to their sample-dependent performance, we do not derive the error bounds for them.

Logistic regression: This standard method treats the monitor values as arbitrary features (not probabilities) that linearly predict the log-odds of $\Pr(A_1 \dots A_n)$.

$$C(M_1 \dots M_n) = \frac{1}{1 + e^{-w_0 - \sum_{i=1..n} w_i M_i}}$$

Parameters $w_0 \dots w_n$ are fit on data from $M_1 \dots M_n$ and A_ψ using λ-weighted cross-entropy loss from Equation (1).

Sequential Bayes: if we know the joint density $p(M_1 \dots M_n)$ and its conditioning on $A_1 \wedge \cdots \wedge A_n$, our sequential Bayesian estimator starts with a uniform prior C_0 at $t = 0$, and at time $t + 1$ is updated as follows:

$$C_{t+1}(M_1 \dots M_n) = C_t \cdot \frac{\Pr\big(M_1 \dots M_n \mid A_1 \wedge \cdots \wedge A_n\big)}{\Pr(M_1 \dots M_n)}$$

4.4.2 Error bounds for composition functions. First, we note a useful lemma that links assumption probabilities to monitor expectations. Then we bound *ECE* for the product and weighted averaging composition, and finally bound *CCE* for the product composition.

Lemma 1 (Conditional Assumption Bounds). *For any composition function C, if:*

$$MCE(M_1, A_1) \le e_1, \quad MCE(M_2, A_2) \le e_2,$$
$$A_1 \perp M_C \mid M_1, \quad A_2 \perp M_C \mid M_2$$

Then:

$$\mathbb{E}[M_1 \mid M_C] - e_1 \le \Pr(A_1 \mid M_C) \le \mathbb{E}[M_1 \mid M_C] + e_1$$
$$\mathbb{E}[M_2 \mid M_C] - e_2 \le \Pr(A_2 \mid M_C) \le \mathbb{E}[M_2 \mid M_C] + e_2$$

Proof. See Appendix B. □

Theorem 2 (*ECE* bound for product composition). *If:*

$$M_C = M_1 M_2,\ MCE(M_1, A_1) \le e_1,\ MCE(M_2, A_2) \le e_2,$$
$$A_1 \perp A_2 \mid M_C,\ A_1 \perp M_C \mid M_1,\ A_2 \perp M_C \mid M_2,$$
$$\mathrm{Var}(M_1 \mid M_C) \le \mathrm{Var}(M_1),\ \mathrm{Var}(M_2 \mid M_C) \le \mathrm{Var}(M_2)$$

Then:

$$ECE(M_1 M_2, A_1 \wedge A_2)$$
$$\le \max[4e_1 e_2, \sqrt{\mathrm{Var}[M_1]\,\mathrm{Var}[M_2]} + e_1 + e_2 + e_1 e_2]$$

Proof. See Appendix C. □

The above bound is fairly narrow if the monitors are well-calibrated and have low variance. This suggests that product composition may perform well in practice.

Theorem 3 (*ECE* bound for weighted averaging comp.). *If:*

$$M_C = w_1 M_1 + w_2 M_2, \quad w_1 + w_2 = 1,$$
$$MCE(M_1, A_1) \le e_1, \quad MCE(M_2, A_2) \le e_2,$$
$$A_1 \perp A_2 \mid M_C, \quad A_1 \perp M_C \mid M_1, \quad A_2 \perp M_C \mid M_2$$

Then:

$$ECE(w_1 M_1 + w_2 M_2, A_1 \wedge A_2)$$
$$\le \max[e_1 + e_2 + e_1 e_2,\ \max[w_1, w_2] + e_1 + e_2 - e_1 e_2]$$

Proof. See Appendix D. □

Notice that the above *ECE* bound for the averaging composition is lower-bounded by 0.5. This reflects that under our precondition, the proof is required to bound $w_1 M_1 + w_2 M_2 - M_1 M_2$, which can reach values at least as great as 0.5. This reflects the mismatch between additive composition and conditionally independent assumptions, and we do not expect it to perform well in our case studies. Finding different conditions under which this *ECE* bound can be tightened remains for future work.

Theorem 4 (Conservatism bound for product comp.). *If:*

$$M_C = M_1 M_2, \quad A_1 \perp A_2 \mid M_C, \quad A_1 \perp M_C \mid M_1, \quad A_2 \perp M_C \mid M_2,$$
$$MCE(M_1, A_1) \le e_1, \quad MCE(M_2, A_2) \le e_2,$$

Then:

$$\Pr(A_1, A_2 \mid M_C = x) \ge \max[0, x - e_1] \max[0, x - e_2]$$

Proof. See Appendix E. □

This theorem shows that low-confidence outputs of product composition should not be trusted to be conservative. It also motivates another composition function that we expect to be more conservative:

Power Product:

$$\Pr(A_1 \wedge \cdots \wedge A_n) = \prod_{i=1..n} (M_i)^n$$

Corollary 5 (*CCE* bound for product composition). *Under conditions of Theorem 4, $CCE(M_1 M_2, A_1 \wedge A_2)$ is bounded by*

$$\begin{cases} \left(e_1^2 - 2e_1(-1 + e_2) + (1 + e_2)^2\right)/4, & \textit{if } (1 + e_1 + e_2)/2 \in [0, 1] \\ e_1 + e_2 - e_1 e_2, & \textit{otherwise} \end{cases}$$

Proof.

$$CCE(M_1 M_2) = \max_{x \in [0,1]} [x - \Pr(A_1 \wedge A_2 \mid M_C = x)]$$
$$\leq \max_{x \in [0,1]} [x - \max[0, x - e_1] \max[0, x - e_2]]$$

On interval $x \in [0, \max[e_1, e_2]]$, the maximum is achieved on $x = \max[e_1, e_2]$ and equal to $\max[e_1, e_2]$.

On interval $x \in [\max[e_1, e_2], 1]$, we have $f(x) = x - (x - e_1)(x - e_2) = -x^2 + (1 + e_1 + e_2)x - e_1 e_2$, which is a quadratic function with $f''(x) < 0$. Therefore, its maximum on $[0, 1]$ may be achieved only in $x = \max[e_1, e_2]$, $x = 1$, or when $f'(x) = 0$. The former case is above, leaving the other two. $f(1) = e_1 + e_2 - e_1 e_2$, which can be shown to be greater than $\max[e_1, e_2]$, eliminating that bound.

Solving $f'(x_0) = 0$ for x_0, we get:

$$x_0 = ((1 + e_1 + e_2)/2; f(x_0) = \left(e_1^2 - 2e_1(-1 + e_2) + (1 + e2)^2\right)/4$$

Thus, if $(1 + e_1 + e_2)/2 \in [0, 1]$, we get the above bound, and $e_1 + e_2 - e_1 e_2$ otherwise.

□

When proving Theorem 4, we took advantage of a property of product composition: the composed confidence is a lower bound of each monitor confidence. This property is not present in the sum-based compositions (averaging, logistic regression): given a composition value, one of the monitors may be arbitrarily small, and so the probability of the (conditionally independent) assumptions can be arbitrarily small as well. Hence, no general conservatism bound can be provided for such compositions under our preconditions. It may be possible provide it in special cases of assumption dependencies or monitor distributions, which remain for future work.

4.5 End-to-End Bound on Calibration Error

Finally, we get to tie the CoCo framework together with the results from Sections 4.1 to 4.4 and prove a *key result of this paper*: when the requirements of our framework are satisfied, compositional monitors have guaranteed upper bounds on *ECE* and *CCE*. The proofs can be found in Appendices F and G.

Theorem 6 (End-to-end bound on *ECE*). *If the model is safety-relevant up to* e_1*, assumptions* A_ψ *are sufficient and safety-relevant up to* e_2*, and monitor* M_C *is calibrated to* A_ψ *with* $ECE(M_C, A_\psi) \leq e_3$*, then it is calibrated to* Φ *with bounded* $ECE(M_C, \Phi)$*:*

$$ECE(M_C, \Phi) \leq e_1 + e_2 + e_3$$

Theorem 7 (End-to-end bound on *CCE*). *If* A_ψ *is a sufficient assumption and* $CCE(M_C, A_\psi) \leq e$*, then* $CCE(M_C, \Phi) \leq e$*.*

Notice that the preconditions for the *CCE* bound are weaker due to relying on formal verification with strong guarantees.

5 EVALUATION

The goal of our case studies is to evaluate the usefulness of the CoCo framework as a whole ("does assumption monitoring predict safety violations?"), the usefulness of confidence composition ("do compositions outperform their constituents?"), and our ability to improve conservatism ("how to reduce the *CCE* of compositions?").

We perform two case studies: a mountain car getting up a hill and an underwater vehicle tracking a pipeline. Each system has two verification assumptions and two monitors. The studies differ in several ways to show the flexibility of CoCo: the safety properties are somewhat different (eventually vs always), the true model is unknown to us in the second case study, initial-state assumptions are combined differently with measurement assumptions, the state assumption is evaluated at different times ($t = 0$ and current t), and the state estimation monitors use different techniques.

Our plan is to execute each system and collect, for each monitor, a dataset of N monitor outputs, $\widehat{M} = \{m_1 \dots m_N\}$, binary satisfactions of the respective monitored assumption, $\widehat{A} = \{a_1 \dots a_N\}$, and true eventual binary safety outcomes (the chance of which the monitor predicts indirectly), $\widehat{\Phi} = \{\phi_1 \dots \phi_N\}$.

Our analysis will measure the binned approximations of the calibration errors from Section 2.2 on a uniform binning $B_1 \dots B_K$ of $[0, 1]$ into $K = 10$ confidence bins. We compare the average confidence within each bin, $\mathrm{conf}(B_k) := \frac{1}{|B_k|} \sum_{i \in B_k} m_i$, with the rate of assumption occurrence in that bin, $\mathrm{occ}(B_k) := \frac{1}{|B_k|} \sum_{i \in B_k} a_i$.

- *Estimated expected calibration error:* ($\widehat{ECE}$)
$$\widehat{ECE}(\widehat{M}, \widehat{A}) := \sum_{k=1}^{K} \frac{|B_k|}{N} |\,\mathrm{occ}(B_k) - \mathrm{conf}(B_k)|$$
- *Estimated maximum calibration error:* ($\widehat{MCE}$)
$$\widehat{MCE}(\widehat{M}, \widehat{A}) := \max_{k \in K} |\,\mathrm{occ}(B_k) - \mathrm{conf}(B_k)|$$
- *Estimated conservatism error:* ($\widehat{CCE}$)
$$\widehat{CCE}(\widehat{M}, \widehat{A}) := \max_{k \in K} [\mathrm{conf}(B_k) - \mathrm{occ}(B_k)]$$

To evaluate how calibrated $\widehat{M}$ is to safety, $\widehat{A}$ is replaced with $\widehat{\Phi}$ in the above definitions.

Calibration should not be evaluated in isolation from accuracy-related measures; otherwise, a monitor could "cheat" by always outputting an estimate of average probability — and thus give up its ability to discriminate the outcomes. So, in addition to calibration measures, we will provide two measures of accuracy:

- *Estimated Brier Score* ($\widehat{Brier}$) is a classic scoring rule for probability predictions [34, 50], a.k.a. the mean squared error:
$$\widehat{Brier}(\widehat{M}, \widehat{A}) := \frac{1}{N} \sum_{i=1}^{N} (m_i - a_i)^2$$
- *Area under Curve* ($\widehat{AuC}$) of the trade-off (ROC) curve between the true positive and false positive rates, which manifests when $\widehat{M}$ is thresholded by every number between 0 to 1. It is used to measure a classifier's discrimination ability.

The case study data and the source code for its analysis are available at https://github.com/bisc/coco-case-studies.

5.1 Mountain Car

The *mountain car* (MC) [24] is a standard reinforcement learning benchmark where the task is to drive an underpowered car up a hill from a valley. The controller needs to first drive the car up the opposite hill so as to gather enough speed. Formally, the car has two continuous states, position and velocity, both one-dimensional in the horizontal direction, $\boldsymbol{x}$:= (position p, velocity v).

		Mountain Car Case Study					UUV Case Study				
Monitor	Predicts	$\widehat{ECE}$	$\widehat{MCE}$	$\widehat{CCE}$	$\widehat{Brier}$	$\widehat{AuC}$	$\widehat{ECE}$	$\widehat{MCE}$	$\widehat{CCE}$	$\widehat{Brier}$	$\widehat{AuC}$
M_1	A_1	0.021 ± 0.004	0.157 ± 0.04	0.152 ± 0.04	0.043 ± 0.002	0.987 ± 0.001	0.1 ± 0.02	0.41 ± 0.15	0.26 ± 0.15	0.176 ± 0.01	0.829 ± 0.009
	Φ	0.285 ± 0.01	0.456 ± 0.02	0.456 ± 0.02	0.313 ± 0.01	0.699 ± 0.01	0.111 ± 0.04	0.399 ± 0.14	0.316 ± 0.12	0.202 ± 0.02	0.796 ± 0.01
M_2	A_2	0.157 ± 0.02	0.322 ± 0.04	0.278 ± 0.02	0.225 ± 0.004	0.764 ± 0.002	0.197 ± 0.06	0.317 ± 0.1	0.296 ± 0.11	0.224 ± 0.04	0.676 ± 0.02
	Φ	0.241 ± 0.02	0.436 ± 0.01	0.436 ± 0.01	0.307 ± 0.004	0.674 ± 0.007	0.239 ± 0.06	0.327 ± 0.1	0.317 ± 0.1	0.261 ± 0.04	0.659 ± 0.02
M_1M_2	A_ψ	0.087 ± 0.01	**0.207 ± 0.01**	0.207 ± 0.01	**0.132 ± 0.003**	**0.887 ± 0.004**	0.109 ± 0.02	0.343 ± 0.18	0.196 ± 0.12	0.184 ± 0.008	0.82 ± 0.02
	Φ	**0.129 ± 0.007**	**0.280 ± 0.04**	0.18 ± 0.01	**0.202 ± 0.004**	**0.784 ± 0.007**	0.107 ± 0.03	**0.343 ± 0.18**	0.184 ± 0.12	0.182 ± 0.007	0.821 ± 0.009
$w_1M_1 + w_2M_2$	A_ψ	0.349 ± 0.02	0.659 ± 0.02	0.659 ± 0.02	0.266 ± 0.01	0.811 ± 0.003	0.212 ± 0.06	0.428 ± 0.1	0.428 ± 0.1	0.238 ± 0.03	0.813 ± 0.009
	Φ	0.223 ± 0.01	0.467 ± 0.02	0.467 ± 0.02	0.244 ± 0.006	0.742 ± 0.004	0.188 ± 0.06	0.417 ± 0.1	0.417 ± 0.1	0.227 ± 0.03	0.807 ± 0.009
$(M_1M_2)^2$	A_ψ	0.092 ± 0.009	0.210 ± 0.02	0.204 ± 0.02	**0.132 ± 0.004**	**0.887 ± 0.004**	0.218 ± 0.07	0.342 ± 0.05	**−0.038 ± 0.08**	0.226 ± 0.03	0.82 ± 0.009
	Φ	0.213 ± 0.01	0.428 ± 0.05	0.175 ± 0.01	0.234 ± 0.008	**0.784 ± 0.007**	0.239 ± 0.07	0.386 ± 0.06	**−0.044 ± 0.08**	0.235 ± 0.03	0.821 ± 0.009
LogReg(M_1, M_2)	A_ψ	**0.049 ± 0.006**	0.245 ± 0.04	**0.108 ± 0.01**	**0.13 ± 0.003**	0.867 ± 0.006	**0.07 ± 0.03**	**0.331 ± 0.18**	0.155 ± 0.08	0.173 ± 0.006	0.829 ± 0.008
	Φ	**0.129 ± 0.02**	0.294 ± 0.03	**−0.018 ± 0.03**	0.212 ± 0.007	0.764 ± 0.008	**0.079 ± 0.03**	0.402 ± 0.18	0.143 ± 0.08	0.173 ± 0.006	0.829 ± 0.009
Bayes(M_1, M_2)	A_ψ	0.285 ± 0.02	0.679 ± 0.07	0.679 ± 0.07	0.286 ± 0.02	**0.886 ± 0.02**	0.141 ± 0.02	0.438 ± 0.07	0.27 ± 0.11	**0.155 ± 0.02**	**0.914 ± 0.01**
	Φ	0.328 ± 0.01	0.572 ± 0.06	0.572 ± 0.06	0.331 ± 0.01	0.76 ± 0.02	0.14 ± 0.02	0.445 ± 0.07	0.22 ± 0.09	**0.153 ± 0.02**	**0.917 ± 0.02**

Table 1: Average monitor performance across 20 cross-validation runs with neutrally-weighed calibration ($\lambda = 0.5$)

		Mountain Car Case Study					UUV Case Study				
Monitor	Predicts	$\widehat{ECE}$	$\widehat{MCE}$	$\widehat{CCE}$	$\widehat{Brier}$	$\widehat{AuC}$	$\widehat{ECE}$	$\widehat{MCE}$	$\widehat{CCE}$	$\widehat{Brier}$	$\widehat{AuC}$
M_1	A_1	0.058 ± 0.006	0.405 ± 0.03	−0.002 ± 0.002	0.057 ± 0.004	0.987 ± 0.001	0.251 ± 0.05	0.506 ± 0.16	−0.039 ± 0.02	0.236 ± 0.02	0.825 ± 0.01
	Φ	0.315 ± 0.009	0.465 ± 0.01	0.465 ± 0.01	0.329 ± 0.009	0.693 ± 0.01	0.191 ± 0.05	0.449 ± 0.18	0.002 ± 0.06	0.228 ± 0.02	0.793 ± 0.02
M_2	A_2	0.195 ± 0.008	0.548 ± 0.07	0.241 ± 0.009	0.236 ± 0.002	0.764 ± 0.002	0.096 ± 0.03	0.363 ± 0.28	0.099 ± 0.03	0.193 ± 0.006	0.671 ± 0.01
	Φ	0.274 ± 0.008	0.475 ± 0.05	0.437 ± 0.01	0.316 ± 0.005	0.67 ± 0.007	0.125 ± 0.03	0.337 ± 0.2	0.162 ± 0.03	0.22 ± 0.006	0.653 ± 0.02
M_1M_2	A_ψ	**0.102 ± 0.008**	**0.231 ± 0.02**	0.203 ± 0.009	**0.137 ± 0.004**	**0.881 ± 0.004**	0.287 ± 0.05	0.429 ± 0.05	−0.076 ± 0.02	0.277 ± 0.03	0.817 ± 0.01
	Φ	0.223 ± 0.008	0.486 ± 0.04	0.176 ± 0.008	0.242 ± 0.006	0.779 ± 0.008	0.31 ± 0.05	0.46 ± 0.07	−0.08 ± 0.02	0.277 ± 0.03	0.818 ± 0.01
$w_1M_1 + w_2M_2$	A_ψ	0.229 ± 0.008	0.363 ± 0.009	0.363 ± 0.009	0.197 ± 0.002	0.826 ± 0.01	**0.109 ± 0.03**	**0.272 ± 0.11**	0.072 ± 0.09	**0.185 ± 0.007**	**0.825 ± 0.01**
	Φ	**0.172 ± 0.006**	**0.308 ± 0.04**	0.228 ± 0.01	**0.22 ± 0.003**	0.749 ± 0.01	**0.12 ± 0.04**	**0.281 ± 0.11**	0.051 ± 0.08	**0.186 ± 0.009**	**0.827 ± 0.01**
$(M_1M_2)^2$	A_ψ	0.151 ± 0.006	0.431 ± 0.04	0.197 ± 0.01	0.157 ± 0.005	**0.881 ± 0.004**	0.443 ± 0.05	0.668 ± 0.05	**−0.162 ± 0.03**	0.394 ± 0.04	0.817 ± 0.01
	Φ	0.273 ± 0.007	0.614 ± 0.02	0.169 ± 0.009	0.276 ± 0.006	0.779 ± 0.008	0.466 ± 0.05	0.67 ± 0.05	**−0.164 ± 0.03**	0.415 ± 0.04	0.818 ± 0.01
LogReg(M_1, M_2)	A_ψ	0.144 ± 0.01	0.447 ± 0.02	**−0.059 ± 0.008**	0.16 ± 0.006	0.868 ± 0.005	0.235 ± 0.05	0.472 ± 0.13	−0.044 ± 0.04	0.231 ± 0.02	**0.826 ± 0.009**
	Φ	0.276 ± 0.009	0.481 ± 0.02	**−0.237 ± 0.01**	0.275 ± 0.007	0.761 ± 0.008	0.258 ± 0.05	0.569 ± 0.1	−0.048 ± 0.04	0.243 ± 0.02	**0.827 ± 0.01**

Table 2: Average monitor performance across 20 cross-validation runs with conservatively-weighed calibration ($\lambda = 0.8$)

The car is considered safe if it gets to the top of the hill in 110 steps: $\varphi := t \geq 110 \implies p \geq 0.45$. The dynamics F_d is as follows:

$$p_{k+1} = p_k + v_k, \qquad v_{k+1} = v_k + 0.0015u_k - z * cos(3p_k),$$

where $u_k \in [-1, 1]$ is the controller's output, and z is the hill steepness, sampled uniformly from two values: $\{0.0025, 0.0035\}$. Initial position p_0 is sampled uniformly from $[-0.6, -0.4]$, and $v_0 = 0$.

In our extension of the classic mountain car, noisy measurements $\boldsymbol{y}$:= (estimated position $\hat{p}$, estimated velocity $\hat{v}$) are obtained from measurement models F_m in which driving faster makes localization more difficult, and being on a hillside biases the velocity estimates. This model uses noise parameters c and d chosen uniformly from $[-1, 1]$ and $[-0.01, 0.02]$, respectively:

$$\hat{p}_k = p_k + cv_k, \qquad \hat{v}_k = v_k + dp_k$$

We use a NN controller that was trained and verified in the related work [18]. We extended its verification under two assumptions: A_1 encodes the relation between the initial states and noise parameters where the verification can guarantee as a predicate over p_0, c, and d; A_2 expects the execution to follow F_d above with $z = 0.025$ (less steep hill) and noise c and d from the intervals above, up to a certain error bound. The car may fail due to violating either assumption, leading to $A_\psi = A_1 \wedge A_2$.

Monitoring A_1 is performed by an initial Monte-Carlo sample of triples (p_0, c, d) and gradually inferring their weights based on F_d and the observations. M_1 outputs the weight fraction of the samples that satisfy the predicate from A_1. Monitoring A_2 is performed by statistically testing the consistency between F_d, F_m, and a trace of last 6 observations using an existing tool ModelGuard [5]. The confidence M_2 is either 1 when the model matches the execution or the percentage of the model parameter space that was explored and found to be inconsistent with the execution.

In data collection, we uniformly sample initial states, noise parameters, hill steepness z, and add white process noise ($\mathcal{N}(0, 0.001)$, $\mathcal{N}(0, 0.0001)$) to introduce a slight mismatch between our model $\hat{s}$ and the "real" system s^*. We collected 2002 MC executions with $N = 196449$ samples total.

5.2 Unmanned Underwater Vehicle

The second case study, is an *unmanned underwater vehicle* (UUV) based on a challenge problem from the DARPA Assured Autonomy program. The UUV follows an underwater pipeline and inspects it for cracks. Here, the "real" system s^* is implemented with a high-fidelity UUV simulator based on the Robot Operating System [21]. We use a linearized identified dynamics model F_d of the UUV. The states are $\boldsymbol{x}$:= (two-dimensional position p_x, p_y, heading θ, velocity v, and 4 digital variables from system identification), and the pipe coincides with x-axis. The measurement $\boldsymbol{y}$:= (heading to the pipe $\hat{\theta}$, range to the pipe $\hat{r}$) contains a sensor estimate of the heading θ, which is the angle formed between the UUV's direction and the positive x-axis. It also contains the range measurement $\hat{r}$ which is the distance to the horizontal pipeline at $y = 0$, perpendicular to the heading. The measurements are computed as:

$$\hat{r}_k = \frac{p_{y,k}}{cos(\theta_k)}, \qquad \hat{\theta}_k = \theta_k + Du_k,$$

where the coefficient matrix $D = \begin{bmatrix} 0 & 0 & \dots & d \end{bmatrix}^T$ extracts the controlled turn angle and multiplies with a noise parameter $d \in$

$[-0.1, 1.5]$. Intuitively, d approximates heading estimation delays during turns, improving our model's safety-relevance.

We consider the UUV safe at any time $\hat{t}$ if for the next 30 seconds its distance to pipe is within the side-looking sonar range (between 10m and 50m): $\varphi := t \leq \hat{t} + 30 \implies 10 \leq p_y \leq 50$. Given these measurements, we train a NN controller running at 0.5 Hz to follow the pipe using the TD3 reinforcement learning algorithm [12].

Assumption A_1 is a predicate over p_y and θ where the verification succeeded. We monitor it with a particle filter, which propagates particles (p_y, θ) over time. M_1 outputs the weight fraction of the current particles within the A_1. A_2 expects the system to behave consistently with our F_d and F_m with $d \in [-0.1, 1.5]$ based on 4 latest observation steps, and M_2 is analogous to M_2 for MC. A conjunction of these assumptions proved sufficient for safety: $A_\psi = A_1 \wedge A_2$.

Our scenario is the UUV heading towards the pipe and needing to make a sharp left turn before $p_y < 10$. We sample $p_{y,0}$ between 12m and 22m and θ_0 between 5 and 25 degrees, some of which violate A_1. We also introduce a 33% chance of a fin getting stuck and limiting the turn rate, which violates A_2 and makes the UUV less likely to maintain safety. We collected 194 UUV executions and evaluated the safety predictions in the first 20 seconds of each ($N = 3880$).

5.3 Results

We experiment with two calibration settings: neutral ($\lambda = 0.5$) and conservative ($\lambda = 0.8$). In each, the monitors are evaluated with 50-50 cross-validation: tuned on a randomly chosen half of the data and tested on the rest. On the validation set, the monitors $\widehat{M}_1$, $\widehat{M}_2$ are *individually* calibrated with Platt scaling and composed into product, weighted average (weights set inverse to the post-calibration variance), and the product-squared (the two-monitor version of power product). The other two methods use the *joint* monitor-assumption data: logistic regression fits monitor outputs to A_ψ, and Bayes estimates $\Pr(M_1, M_2 \mid A_\psi)$ with histograms.

We present the monitor evaluations in Table 1 ($\lambda = 0.5$) and Table 2 ($\lambda = 0.8$). Each table contains the means and standard deviations of monitor errors and scores after 20 cross-validations. Notice two caveats: (i) M_1 and M_2 predict A_1 and A_2, not A_ψ, (ii) since Bayes is not affected by λ, it is omitted from Table 2. What follows is our observations and interpretations, mostly based on Table 1.

Framework predicts safety: compositional monitors show $\widehat{AuC}$ above 0.7 in the MC case and above 0.8 in the UUV case. $\widehat{ECE}$ stays within $0.1 - 0.2$, explained in part by the difficulty of calibrating M_2, which tends to take extreme values. The good composite calibration is supported by the high safety relevance of our assumptions: on traces with $\neg A_\psi$, the safety chances are 18% for the MC and 6% for the UUV. Brier scores are relatively high because our "true probabilities" are 0s and 1s, which penalized the predictions in the middle of $[0, 1]$ — even when well-calibrated and discriminative.

Compositions outperform constituents: when predicting safety (Φ), all compositional monitors have a higher $\widehat{AuC}$ than M_1 and M_2, and most have a higher $\widehat{ECE}$. We note that compositions have higher uncertainty, which drives up their $\widehat{Brier}$ scores.

Data-driven compositions outperform the non-data-driven: logistic regression shows the best $\widehat{ECE}$ on both case studies, and Bayes dominates the $\widehat{AuC}$ in most cases. This outcome is expected given their information advantage. Among the non-data-driven compositions, as predicted by our theorems, the product outperforms the weighted average across all metrics in Table 1.

Compositions are tunable for conservatism: as per Table 2, the product and product-squared compositions improved their $\widehat{CCE}$ with $\lambda = 0.8$ in the UUV case, as expected. Logistic regression got more conservative in both cases studies, which motivates trying to mimic its benefits without joint data. Surprisingly, the product and product-squared did not respond to conservative tuning in the MC case. Our investigation showed that although M_2 got more conservative on average, its predictions with ≥90% confidence got *less* conservative. This suggests future work in calibration techniques with conservative guarantees and using confidence monitoring as an early warning rather than a final arbiter of safety. Product-squared failed to improve conservatism for the MC due to a similar reason: the overconfidence occurred in the bin $[0.9, 1]$. Generally, we observe a trade-off between conservatism and calibration, and given that weighted average is the least conservative composition, it predictably improves its performance from $\lambda = 0.5$ to $\lambda = 0.8$.

6 DISCUSSION AND CONCLUSION

Our theoretical and empirical investigation highlighted the distinct advantages of the proposed CoCo framework. First, unlike many assurance methods, it does not require a detailed or complete enumeration/model of hazards and failure modes. As long as the models and assumptions are safety-relevant, and the monitors are well-calibrated and accurate, the composite monitor should detect any potential violation of safety. Second, it is not tied to specific closed-form distributions — at most, our bounds only require the knowledge of the calibration bounds and monitor variance. Third, the non-data-driven compositions support independent development of monitors without the need for combined tuning. However, if joint monitor data is available, compositions based on logistic regression and Bayesian estimation can take advantage of it.

This paper is but an early step in compositional confidence monitoring, opening several exciting research directions. First, richer models of assumption dependencies (e.g., copulas [26]) may enable new composition functions (e.g., based on known odds or correlations) and tighten the bounds on the existing ones. Second, extending the composition bounds to 3+ monitors and new composition functions is mathematically challenging, but it can be facilitated by detailed models of monitor dependencies. Another fruitful direction is empirical validation and refinement of the simplifications made in Section 4.4. Since it is difficult to decompose assumptions, it would be useful to create verification techniques that make granular assumptions bottom-up. Finally, a longer-term goal would be to extend the scope of a confidence monitor by considering its temporal behavior and monitoring the assumptions of other monitors [16].

To conclude, this paper presented an approach for run-time safety prediction by monitoring confidence in the assumptions of formal verification. The proposed CoCo framework introduced theoretical requirements for compositional bounds on calibration errors, and instantiated these bounds for two composition functions. Two case studies demonstrated the practical usefulness of CoCo. Furthermore, we observed that the compositions can adjust their conservatism, as well as improve their performance given synchronized data about multiple monitors and assumptions.

ACKNOWLEDGMENTS

The authors thank James Weimer, Sooyong Jang, and Northrop Grumman Corporation for helping develop the UUV simulator. This work was supported in part by ARO W911NF-20-1-0080, AFRL and DARPA FA8750-18-C-0090, and ONR N00014-20-1-2744. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Research Laboratory (AFRL), the Army Research Office (ARO), the Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR) or the Department of Defense, or the United States Government.

REFERENCES

[1] Erfan Asaadi, Ewen Denney, and Ganesh Pai. 2020. Quantifying Assurance in Learning-Enabled Systems. In *Computer Safety, Reliability, and Security*. Springer International Publishing, Cham, 270–286.

[2] Dimitrios Boursinos and Xenofon Koutsoukos. 2020. Improving Prediction Confidence in Learning-Enabled Autonomous Systems. In *Dynamic Data Driven Applications Systems*. Springer International Publishing, Cham, 217–224.

[3] Dimitrios Boursinos and Xenofon Koutsoukos. 2021. Assurance monitoring of learning-enabled cyber-physical systems using inductive conformal prediction based on distance learning. *AI EDAM* 35, 2 (May 2021), 251–264.

[4] Antoine Cailliau and Axel Van Lamsweerde. 2019. Runtime Monitoring and Resolution of Probabilistic Obstacles to System Goals. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* (2019).

[5] Taylor J. Carpenter, Radoslav Ivanov, Insup Lee, and James Weimer. 2021. ModelGuard: Runtime Validation of Lipschitz-continuous Models. In *7th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'21)*. arXiv: 2104.15006.

[6] Alessandro Cimatti, Chun Tian, and Stefano Tonetta. 2019. Assumption-Based Runtime Verification with Partial Observability and Resets. In *Runtime Verification*. 165–184.

[7] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. 1999. Monte Carlo localization for mobile robots. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, Vol. 2. 1322–1328 vol.2.

[8] Ankush Desai, S. Ghosh, S. Seshia, N. Shankar, and A. Tiwari. 2019. SOTER: A Runtime Assurance Framework for Programming Safe Robotics Systems. *IEEE/IFIP International Conference on Dependable Systems and Networks* (2019).

[9] Karam Abd Elkader, Orna Grumberg, Corina S. Păsăreanu, and Sharon Shoham. 2015. Automated Circular Assume-Guarantee Reasoning. In *FM 2015: Formal Methods*. Springer, Cham, 23–39.

[10] Jaime F. Fisac, Andrea Bajcsy, Sylvia L. Herbert, David Fridovich-Keil, Steven Wang, Claire J. Tomlin, and Anca Dragan. 2018. Probabilistically Safe Robot Planning with Confidence-Based Human Predictions. In *Robotics: Science and Systems XIV*.

[11] Zhicheng Fu, Chunhui Guo, Zhenyu Zhang, Shangping Ren, and Lui Sha. 2020. UACFinder: Mining Syntactic Carriers of Unspecified Assumptions in Medical Cyber-Physical System Design Models. *ACM Transactions on Cyber-Physical Systems* 4, 3 (March 2020), 24:1–24:25.

[12] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*. PMLR, 1587–1596.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

[14] Charles Hartsell, Shreyas Ramakrishna, Abhishek Dubey, Daniel Stojcsics, Nagabhushan Mahadevan, and Gabor Karsai. 2021. ReSonAte: A Runtime Risk Assessment Framework for Autonomous Systems. IEEE Computer Society, 118–129.

[15] Larry V. Hedges and Ingram Olkin. 1985. *Statistical Methods for Meta-Analysis* (1st ed.). Academic Press, Orlando.

[16] T. Henzinger and N. Saraç. 2020. Monitorability Under Assumptions. In *RV*.

[17] Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. 2021. Verisig 2.0: Verification of Neural Network Controllers Using Taylor Model Preconditioning. In *Computer Aided Verification (CAV)*.

[18] Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. 2019. Verisig: Verifying Safety Properties of Hybrid Systems with Neural Network Controllers. In *Proc. of the International Conference on Hybrid Systems: Computation and Control (HSCC)* (Montreal, Quebec, Canada).

[19] Daphne Koller, Nir Friedman, and Francis Bach. 2009. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA.

[20] Marta Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. 2013. Compositional probabilistic verification through multi-objective model checking. *Information and Computation* 232 (Nov. 2013), 38–65.

[21] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. 2016. UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*.

[22] Stefan Mitsch and Andre Platzer. 2014. ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models. In *Proc. of the International Conference on Runtime Verification (RV)*. Springer International.

[23] Stefan Mitsch and André Platzer. 2018. Verified Runtime Validation for Partially Observable Hybrid Systems. *arXiv:1811.06502 [cs]* (Nov. 2018).

[24] Andrew W. Moore. 1991. Efficient memory based learning for robot control. *PhD Thesis, Computer Laboratory, University of Cambridge* (1991).

[25] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the AAAI Conference on Artificial Intelligence* 2015 (Jan. 2015), 2901–2907.

[26] Roger B. Nelsen. 2006. *An Introduction to Copulas* (2 ed.). Springer-Verlag, NY.

[27] Necmiye Ozay, Mario Sznaier, and Constantino Lagoa. 2014. Convex certificates for model (in) validation of switched affine systems with unknown switches. *IEEE Trans. Automat. Control* 59, 11 (2014), 2921–2932.

[28] Sangdon Park, Shuo Li, Osbert Bastani, and I. Lee. 2021. PAC Confidence Predictions for Deep Neural Network Classifiers. *ICLR* (2021).

[29] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[30] Henry Petroski. 1992. *To Engineer Is Human: The Role of Failure in Successful Design*. Vintage, New York.

[31] John C. Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*. MIT Press, 61–74.

[32] H Vincent Poor. 2013. *An introduction to signal detection and estimation*. Springer Science & Business Media.

[33] Stephen Prajna. 2006. Barrier certificates for nonlinear model validation. *Automatica* 42, 1 (2006), 117–126.

[34] Roopesh Ranjan and Tilmann Gneiting. 2010. Combining probability forecasts. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72, 1 (2010), 71–91.

[35] Ivan Ruchkin, Matthew Cleaveland, Oleg Sokolsky, and Insup Lee. 2021. Confidence Monitoring and Composition for Dynamic Assurance of Learning-Enabled Autonomous Systems. In *Formal Methods in Outer Space: Essays Dedicated to Klaus Havelund on the Occasion of His 65th Birthday*. Springer International.

[36] Ivan Ruchkin, Oleg Sokolsky, James Weimer, Tushar Hedaoo, and Insup Lee. 2020. Compositional Probabilistic Analysis of Temporal Properties Over Stochastic Detectors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020).

[37] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining Knowl Discov* 8, 4 (2018), e1249.

[38] P. D. Saqui-Sannes and L. Apvrille. 2016. Making Modeling Assumptions an Explicit Part of Real-Time Systems Models. In *Proc. of 8th ERTS*.

[39] Louis L Scharf and Cédric Demeure. 1991. *Statistical signal processing: detection, estimation, and time series analysis*. Prentice Hall.

[40] Johann Martin Schumann, Nagabhushan Mahadevan, Michael Lowry, and Gabor Karsai. 2019. Model-based On-board Decision Making for Autonomous Aircraft. *Annual Conference of the PHM Society* 11, 1 (Sept. 2019).

[41] Jieli Shen, Regina Y. Liu, and Min-ge Xie. 2018. Prediction with confidence - A general framework for predictive inference. *Journal of Statistical Planning and Inference* 195 (May 2018), 126–140.

[42] Oleg Sokolsky, Teng Zhang, Insup Lee, and Michael McDougall. 2016. Monitoring Assumptions in Assume-Guarantee Contracts. *Electronic Proceedings in Theoretical Computer Science* 208 (May 2016), 46–53.

[43] Scott D. Stoller, Ezio Bartocci, Justin Seyster, Radu Grosu, Klaus Havelund, Scott A. Smolka, and Erez Zadok. 2012. Runtime Verification with State Estimation. In *Runtime Verification*. Springer Berlin Heidelberg, 193–207.

[44] Hoang-Dung Tran, Luan Viet Nguyen, Patrick Musau, Weiming Xiang, and Taylor T. Johnson. 2019. Decentralized Real-Time Safety Verification for Distributed Cyber-Physical Systems. In *Formal Techniques for Distributed Objects, Components, and Systems*. Springer International Publishing, Cham, 261–277.

[45] Hoang-Dung Tran, Xiaodong Yang, Diego Manzanas Lopez, Patrick Musau, Luan Viet Nguyen, Weiming Xiang, Stanley Bak, and Taylor T. Johnson. 2020. NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems. In *Computer Aided Verification*.

[46] Ben Van Calster and et al. 2019. Calibration: the Achilles heel of predictive analytics. *BMC Medicine* 17, 1 (Dec. 2019), 230.

[47] Abraham Wald. 2004. *Sequential analysis*. Courier Corporation.

[48] Kosuke Watanabe, Eunsuk Kang, Chung-Wei Lin, and Shinichi Shiraishi. 2018. Runtime Monitoring for Safety of Intelligent Vehicles. In *Proceedings of the 55th Annual Design Automation Conference*. ACM, New York, NY, USA, 31:1–31:6.

[49] Alan S Willsky. 1976. A survey of design methods for failure detection in dynamic systems. *Automatica* 12, 6 (1976), 601–611.

[50] R. L. Winkler, Y. Grushka-Cockayne, K. C. Lichtendahl, and V. R. Jose. 2018. Averaging Probability Forecasts: Back to the Future.

APPENDIX

This appendix contains the proofs of our theorems.

A PROOF OF THEOREM 1

Proof. Throughout this proof, we use the fact that $\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}) \implies \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A)$, hence $\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s})) \leq \mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))$.

$$\begin{aligned}&\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\\ &\quad= \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \in \boldsymbol{X}(\hat{s}) \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\\ &\qquad\qquad + \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}) \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\end{aligned}$$

The first summand is bounded by e_2 because A is safety-relevant:

$$\begin{aligned}&\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \in \boldsymbol{X}(\hat{s}) \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\\ &\quad= \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \in \boldsymbol{X}(\hat{s}) \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\ \mathsf{Pr}(\boldsymbol{x} \in \boldsymbol{X}(\hat{s}) \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\\ &\quad\leq \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \in \boldsymbol{X}(\hat{s}) \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A)) \leq e_2\end{aligned}$$

For the second summand, we apply Bayes' theorem:

$$\begin{aligned}&\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}) \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))\\ &\quad= \frac{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A) \mid \varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))\ \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))}{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))}\\ &\quad= \frac{\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \wedge \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))}{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))}\end{aligned}$$

Recall that $\boldsymbol{x} \in \boldsymbol{X}(s^*)$ by the precondition of the theorem and thus $\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s})) = \mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}), \boldsymbol{x} \in \boldsymbol{X}(s^*))$ Then, by the safety relevance of $\hat{s}$, the above fraction is bounded:

$$\frac{\mathsf{Pr}(\varphi(\boldsymbol{x}) = \mathsf{T} \mid \boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))}{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))} \leq e_1 \frac{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}))}{\mathsf{Pr}(\boldsymbol{x} \notin \boldsymbol{X}(\hat{s}_A))} \leq e_1$$

□

B PROOF OF LEMMA 1

Proof. We will build up the bounds for the expression under the expectation. Suppose monitors M_1 and M_2 have probability densities $\mathsf{p}_1(x)$ and $\mathsf{p}_2(y)$. From *MCE* bounds, integration, and our conditional independence of assumptions and monitors, for M_1 we get:

$$\begin{aligned}x - e_1 &\leq \mathsf{Pr}(A_1 \mid M_1 = x) \leq x + e_1,\\ \int_0^1 (x - e_1)\mathsf{p}_1(x \mid M_C)dx &\leq \int_0^1 \mathsf{Pr}(A_1 \mid M_1 = x)\mathsf{p}_1(x \mid M_C)dx\\ &\leq \int_0^1 (x + e_1)\mathsf{p}_1(x \mid M_C)dx,\\ \mathbb{E}[M_1 \mid M_C] - e_1 &\leq \mathsf{Pr}(A_1 \mid M_C) \leq \mathbb{E}[M_1 \mid M_C] + e_1 \qquad (2)\end{aligned}$$

Analogously, for M_2:

$$\mathbb{E}[M_2 \mid M_C] - e_2 \leq \mathsf{Pr}(A_2 \mid M_C) \leq \mathbb{E}[M_2 \mid M_C] + e_2 \qquad (3)$$

□

C PROOF OF THEOREM 2

Proof. From conditional independence:

$$\begin{aligned}ECE(M_C, A_1 \wedge A_2) &= \mathbb{E}[|\mathsf{Pr}(A_1 \wedge A_2 \mid M_C) - M_C|])\\ &= \mathbb{E}[|\mathsf{Pr}(A_1 \mid M_C)\mathsf{Pr}(A_2 \mid M_C) - M_C|])\end{aligned}$$

We split the proof into three cases:

(1) Event H_1: $\mathbb{E}[M_1 \mid M_C] \geq e_1$ and $\mathbb{E}[M_2 \mid M_C] \geq e_2$
(2) Event H_2: $\mathbb{E}[M_1 \mid M_C] < e_1$ and $\mathbb{E}[M_2 \mid M_C] < e_2$
(3) Event H_3: $\mathbb{E}[M_1 \mid M_C] < e_1$ xor $\mathbb{E}[M_2 \mid M_C] < e_2$

Then the expectation can be split accordingly:

$$\begin{aligned}&\mathbb{E}[|\mathsf{Pr}(A_1 \wedge A_2 \mid M_C) - M_C|]) \qquad (4)\\ &\quad= \sum_{i=1}^{3} \mathsf{Pr}(H_i)\,\mathbb{E}[|\mathsf{Pr}(A_1 \wedge A_2 \mid M_C) - M_C| \mid H_i]\end{aligned}$$

To complete the proof, with the help of Lemma 1, we need to show that we can bound the conditional expectation by at least one term in the max under each case, i.e., for each i

$$\begin{aligned}&\mathbb{E}[|\mathsf{Pr}(A_1 \wedge A_2 \mid M_C) - M_C| \mid H_i] \qquad (5)\\ &\quad= \max[4e_1e_2, \sqrt{\mathrm{Var}[M_1]\,\mathrm{Var}[M_2]} + e_1 + e_2 + e_1e_2]\end{aligned}$$

For the sake of brevity, let $E_1 := \mathbb{E}[M_1 \mid M_C]$ and $E_2 := \mathbb{E}[M_2 \mid M_C]$.

Case H_1:

Our H_1 restrictions allows multiplying inequalities (2) and (3) because all sides are non-negative:

$$(E_1 - e_1)(E_2 - e_2) \leq \mathsf{Pr}(A_1 \mid M_C)\mathsf{Pr}(A_2 \mid M_C) \leq (E_1 + e_1)(E_2 + e_2),$$

and then subtract M_C:

$$\begin{aligned}(E_1 - e_1)(E_2 - e_2) - M_C &\leq \mathsf{Pr}(A_1 \mid M_C)\mathsf{Pr}(A_2 \mid M_C) - M_C\\ &\leq (E_1 + e_1)(E_2 + e_2) - M_C\end{aligned}$$

Taking the absolute value, using max inequality, and triangle inequality we get a bound on the expression under the expectation:

$$\begin{aligned}&|\mathsf{Pr}(A_1 \mid M_C)\mathsf{Pr}(A_2 \mid M_C) - M_C|\\ &\quad\leq \max\Big[|(E_1 - e_1)(E_2 - e_2) - M_C|, |(E_1 + e_1)(E_2 + e_2) - M_C|\Big]\end{aligned}$$

Now we use two facts: $M_C = M_1M_2$ and $\mathbb{E}[M_1]\,\mathbb{E}[M_2] = \mathbb{E}[M_1M_2] - \mathrm{Cov}[M_1, M_2]$. Then, we can proceed with the triangle inequality:

$$\begin{aligned}&\max\Big[|(E_1 - e_1)(E_2 - e_2) - M_C|, |(E_1 + e_1)(E_2 + e_2) - M_C|\Big]\\ &\quad= \max\Big[|\,\mathrm{Cov}[M_1, M_2|M_C] - e_1e_2 + e_1E_2 + e_2E_1\,|,\\ &\qquad\qquad |\,\mathrm{Cov}[M_1, M_2|M_C] - e_1e_2 - e_1E_2 - e_2E_1\,|\Big]\\ &\quad\leq |\,\mathrm{Cov}[M_1, M_2|M_C]\,| + e_1E_2 + e_2E_1 + e_1e_2\end{aligned}$$

Using the Cauchy-Schwarz inequality, our assumption on variances, and the fact that $E_1 \leq 1, E_2 \leq 1$ we get the final bound under Case H_1:

$$\begin{aligned}&|\,\mathrm{Cov}[M_1, M_2|M_C]\,| + e_1E_2 + e_2E_1 + e_1e_2\\ &\qquad\leq \sqrt{\mathrm{Var}[M_1]\,\mathrm{Var}[M_2]} + e_1 + e_2 + e_1e_2\end{aligned}$$

Case H_2**:** Recalling that $M_C = M_1M_2$, note that

$$M_C \leq E_1 < e_1, \qquad (6)$$

since $M_1 \geq M_C$ everywhere. Also note that (2) now becomes

$$0 \leq \mathsf{Pr}(A_1 \mid M_C) \leq 2e_1. \qquad (7)$$

Similarly, (6) becomes

$$\begin{aligned}|\mathsf{Pr}(A_1 \mid M_C)\mathsf{Pr}(A_2 \mid M_C) - M_C| &\leq \max[|-M_C|, |4e_1e_2 - M_C|]\\ &\leq \max[e_1, e_2, 4e_1e_2].\end{aligned}$$

Note that both e_1 and e_2 are smaller than the bound under H_1, hence we only keep $4e_1e_2$ in the final bound.

Case H_3: Without loss of generality, consider the case when $E_1 < e_1$ and $E_2 \geq e_2$. Then once again (2) becomes

$$0 \leq \mathsf{Pr}(A_1 \mid M_C) \leq 2e_1. \qquad (8)$$

The bound in Equation (6) is now simplified only on one side:

$$\begin{aligned}&|\Pr(A_1 \mid M_C)\Pr(A_2 \mid M_C) - M_C| \\ &\qquad\le \max[|-M_C|, |(E_1+e_1)(E_2+e_2) - M_C|] \\ &\qquad\le \max[e_1, \sqrt{\mathrm{Var}[M_1]\,\mathrm{Var}[M_2]} + e_1 + e_2 + e_1e_2].\end{aligned}$$

The case $E_1 \ge e_1$ and $E_2 < e_2$ is symmetric. □

D PROOF OF THEOREM 3

Proof. Following the same structure as Theorem 2, the proof is split into three cases:

(1) Event H_1: $\mathbb{E}[M_1 \mid M_C] \ge e_1$ and $\mathbb{E}[M_2 \mid M_C] \ge e_2$
(2) Event H_2: $\mathbb{E}[M_1 \mid M_C] < e_1$ and $\mathbb{E}[M_2 \mid M_C] < e_2$
(3) Event H_3: $\mathbb{E}[M_1 \mid M_C] < e_1$ xor $\mathbb{E}[M_2 \mid M_C] < e_2$

Then we again split the expectation into (4). To complete the proof, with the help of Lemma 1, we need to show that we can bound the conditional expectation by at least one term in the max under each case. For the sake of brevity, let $E_1 := \mathbb{E}[M_1 \mid M_C]$ and $E_2 := \mathbb{E}[M_2 \mid M_C]$.

Case H_1 for averaging:

$$\begin{aligned}&\max\left[|(E_1-e_1)(E_2-e_2) - M_C|, |(E_1+e_1)(E_2+e_2) - M_C|\right] \\ &\quad= \max\left[|(E_1-e_1)(E_2-e_2) - w_1E_1 - w_2E_2|,\right. \\ &\qquad\qquad\left.|(E_1+e_1)(E_2+e_2) - w_1E_1 - w_2E_2|\right]\end{aligned}$$

Now note that max[a, |b|] = max[a, b, -b], and also note that in our case, a > -b:

$$\begin{aligned}&\max\left[w_1E_1 + w_2E_1 - (E_1-e_1)(E_2-e_2),\right. \\ &\qquad\qquad\left.|(E_1+e_1)(E_2+e_2) - w_1E_1 - w_2E_2|\right] \\ &\quad= \max\left[w_1E_1 + w_2E_2 - (E_1-e_1)(E_2-e_2),\right. \\ &\qquad\qquad\left.(E_1+e_1)(E_2+e_2) - w_1E_1 - w_2E_2\right] \\ &\quad= \max\left[w_1E_1 + w_2E_2 - E_1E_2 + e_2E_1 + e_1E_2 - e_1e_2,\right. \\ &\qquad\qquad\left.E_1E_2 - w_1E_1 - w_2E_2 + e_2E_1 + e_1E_2 + e_1e_2\right]\end{aligned}$$

The final step is to recognize that $w_1a + w_2b - ab \in [0, \max[w_1, w_2]]$. This can be revealed by solving an optimization problem. Thus, we can bound the first expression above with:

$$\max[\max[w_1, w_2] + e_1 + e_2 - e_1e_2, e_1 + e_2 + e_1e_2]$$

Case H_2 for averaging: Here, we are bounded by

$$\max\left[|M_C|, |(E_1+e_1)(E_2+e_2) - M_C|\right],$$

which is itself bounded by M_C since the values in the second argument are both positive. Hence, $M_C = w_1E_1 + w_2E_2 \le w_1e_1 + w_2e_2$. This bound is dominated by $e_1 + e_2 + e_1e_2$ from case H_1.

Case H_3 for averaging: Suppose $E_1 \le e_1, E_2 \ge e_2$. Then we need to bound:

$$\max\left[|M_C|, |(E_1+e_1)(E_2+e_2) - M_C|\right] = M_C = w_1E_1 + w_2E_2 \le w_1e_1 + e_2$$

Analogously, in the alternative case the bound is $e_1 + w_2e_2$. Both of these bounds are lower than $e_1 + e_2 + e_1e_2$ from case H_1.

Similar to the product bound proof, the ultimate bound is:

$$\max[e_1 + e_2 + e_1e_2, \max[w_1, w_2] + e_1 + e_2 - e_1e_2]$$

□

E PROOF OF THEOREM 4

Proof. When $M_C = x = M_1M_2$, it is clear that $M_1 \ge x, M_2 \ge x$ because monitors take values between 0 and 1. It follows that:

$$\mathbb{E}[M_1 \mid M_1M_2 = x] \ge x, \qquad \mathbb{E}[M_2 \mid M_1M_2 = x] \ge x$$

Following Lemma 1, we can see that:

$$\begin{aligned}\Pr(A_1 \mid M_C = x) &\ge \mathbb{E}[M_1 \mid M_1M_2 = x] - e_1 \ge x - e_1 \\ \Pr(A_2 \mid M_C = x) &\ge \mathbb{E}[M_1 \mid M_1M_2 = x] - e_2 \ge x - e_2\end{aligned}$$

To ensure multiplicability:

$$\begin{aligned}\Pr(A_1 \mid M_C = x) &\ge \max[0, x - e_1] \\ \Pr(A_2 \mid M_C = x) &\ge \max[0, x - e_2]\end{aligned}$$

Therefore:

$$\Pr(A_1 \land A_2 \mid M_C = x) \ge \max[0, x - e_1]\max[0, x - e_2]$$

□

F PROOF OF THEOREM 6

Proof.

$$\begin{aligned}&ECE(M_C, \Phi) = \mathbb{E}[|\Pr(\Phi \mid M_C) - M_C|] \\ &\quad= \mathbb{E}[|\Pr(\Phi \mid M_C) - M_C|] - ECE(M_C, A_\psi) + ECE(M_C, A_\psi) \\ &\quad= \mathbb{E}[|\Pr(\Phi \mid M_C) - M_C|] - \mathbb{E}[|\Pr(A_\psi \mid M_C) - M_C|] + ECE(M_C, A_\psi) \\ &\quad\le \mathbb{E}[|\Pr(\Phi \mid M_C) - \Pr(A_\psi \mid M_C)|] + ECE(M_C, A_\psi)\end{aligned}$$

The second summand is bounded by e_3 by the condition of the theorem. Consider now the first summand:

$$\begin{aligned}&\mathbb{E}[|\Pr(\Phi \mid M_C) - \Pr(A_\psi \mid M_C)|] \\ &\qquad= \mathbb{E}[|\Pr(\Phi \land A_\psi \mid M_C) + \Pr(\Phi \land \neg A_\psi \mid M_C) \\ &\qquad\qquad- \Pr(A_\psi \land \Phi \mid M_C) - \Pr(A_\psi \land \neg\Phi \mid M_C)|] \\ &\qquad= \mathbb{E}[|\Pr(\Phi \land \neg A_\psi \mid M_C) - \Pr(A_\psi \land \neg\Phi \mid M_C)|]\end{aligned}$$

Note that $\Pr(A_\psi \land \neg\Phi \mid M_C) = 0$ because A_ψ is sufficient for proving safety and thus A_ψ and $\neg\Phi$ are mutually exclusive, leaving us with:

$$\begin{aligned}\mathbb{E}[|\Pr(\Phi \land \neg A_\psi \mid M_C)|] = \Pr(\Phi \land \neg A_\psi) &= \Pr(\Phi \mid \neg A_\psi)\Pr(\neg A_\psi) \\ &\le \Pr(\Phi \mid \neg A_\psi) \le e_1 + e_2,\end{aligned}$$

where the last bound $e_1 + e_2$ is a result of Theorem 1. □

G PROOF OF THEOREM 7

Proof.

$$\begin{aligned}&\max_{x\in[0,1]}[x - \Pr(\Phi \mid M_C = x)] \\ &\quad= \max_{x\in[0,1]}[x - \Pr(A \mid M_C = x)\Pr(\Phi \mid A, M_C = x) \\ &\qquad\qquad- \Pr(\neg A \mid M_C = x)\Pr(\Phi \mid \neg A, M_C = x)] \\ &\quad\le \max_{x\in[0,1]}[x - \Pr(A \mid M_C = x)] \le e\end{aligned}$$

We are justified in using the upper bound $\max_{x\in[0,1]}[x - \Pr(A \mid M_C = x)]$ as a conservative approximation because (i) A is sufficient, so $\Pr(\Phi \mid A, M_C = x) = 1$, and (ii) the last summand $\Pr(\neg A \mid M_C = x) \cdot \Pr(\Phi \mid \neg A, M_C = x)$ is non-negative. □