

Learning Heteroscedastic Uncertainty in Deterministic Simulations for Probabilistic Abductive Reasoning

Avvai Chandrasekaran^[0009-0006-3550-5561], Adrienne Raglin^[0000-0002-5489-1633], Andrew S. Gordon

1 Georgia State University, Atlanta GA 30302, USA

2 DEVCOM Army Research Laboratory, Aberdeen Proving Ground MD 21005, USA

3 University of Southern California, Institute for Creative Technologies, Los Angeles CA 90094, USA

achandrasekaran2@student.gsu.edu

adrienne.raglin2.civ@army.mil

gordon@ict.usc.edu

Abstract. We present a framework for enhancing decision-making in deterministic event simulations by incorporating probabilistic reasoning. This approach combines Enhanced Tactical Inferencing (ETI) and Etcetera Abduction (Etc-Abduction) reasoning frameworks to generate informed decisions at critical simulation points. This research addresses a fundamental limitation: deterministic simulators lack event probabilities needed for robust reasoning. To solve this, we employ a Neural Network that estimates uncertainty by learning from bootstrapped samples created through systematic input perturbation. The network predicts heteroscedastic standard deviations, which are combined with simulator outputs to model outcomes as Gaussian distributions. Event probabilities are then computed using cumulative distribution functions. Large Language Models facilitate the process by converting simulation data into natural language descriptions and representational logic, allowing for matching between observations and knowledge base rules for Etc-Abduction inference. We hypothesize that integrating probabilistic forecasts with ETI and Etc-Abduction frameworks will improve reasoning capabilities in deterministic simulation environments.

Keywords: Artificial Reasoning, Neural Networks, Theory of Mind, Natural Language Processing, Probabilistic Inference.

1 Introduction

Discrete event simulation (DES) has proven to be a useful tool for modeling complex domains that include decisions made by (simulated) human actors, including manufacturing, logistics, and military operations. DES models may be described as either stochastic or deterministic, where the former includes some degree of randomness and probabilistic simulation rules to capture uncertainties in real-world processes. Accordingly, they are well-suited for quantifying uncertainty in complex scenarios, e.g., estimating the likelihood of an unfavorable outcome through repeated simulation runs. However, for some application domains only deterministic DES models are available due to the lack of real-world data needed to craft probabilistic

simulation rules. In these cases, useful measures of uncertainty and the probabilities of various simulation outcomes need to be gathered using novel methods.

In this research, we investigate methods for extracting quantifiable measure of uncertainty and probability from an unspecified deterministic DES model used to simulate outcomes of missions involving teams of adversarial agents that involve complex decision-making. In our application domain, quantified uncertainty and probabilities would aid in improving the utility of them model by enabling richer interpretations of scenario state and the generation of recommendations to enhance tactical decision-making.

To reason about the simulated scenarios, we will be using two reasoning frameworks, Enhanced Tactical Inferencing (ETI) and Etcetera Abduction (Etc-Abduction), to enhance decision making (Richardson et al., 2023; Gordon 2016, 2018).

We hypothesize that leveraging the ETI and Etcetera-Abduction reasoning frameworks along with probabilistic forecasts will allow for enhanced reasoning over scenarios produced by deterministic simulators.

1.1 Enhanced Tactical Inferencing

The Enhanced Tactical Inferencing (ETI) framework is a modular reasoning system designed to support decision-making in multi-agent environments by explicitly accounting for uncertainty in information sources. Unlike traditional systems that treat all incoming data as equally reliable, ETI employs specialized reasoning modules that evaluate the quality and uncertainty of information before generating recommendations.

At the core of ETI is the Uncertainty of Information (UoI) module, which quantifies and categorizes uncertainty in data sources using a taxonomy derived from Gershon’s characterization of imperfect information. The UoI module evaluates information along seven dimensions: inconsistent, corrupt, disjoint, incomplete, imprecise, complicated, and questionable. These assessments are combined using a weighted sum to produce a scalar uncertainty score ranging from 0 to 10, which is then partitioned into low, medium, and high uncertainty levels. This quantitative score is paired with qualitative justifications that explain the sources and nature of the uncertainty, providing decision-makers with both numerical assessments and contextual understanding.

The ETI framework generates decision recommendations by reasoning over the uncertainty attributes of input data used for decision-making. For each decision point, ETI computes a UoI value based on the characteristics of available information, produces an actionable recommendation (such as “continue”, “request additional intelligence”, or “await report”), and outputs justifications for both the uncertainty assessment and the recommended course of action. For example, when ETI detects medium or high uncertainty due to sensor malfunctions or conflicting intelligence reports, it may recommend acquiring additional data sources rather than proceeding with high-risk actions, thereby helping commanders make more informed decisions despite imperfect information.

1.2 Etcetera Abduction

Etcetera Abduction is a probabilistic reasoning framework proposed that extends classical logical abduction to handle uncertain, real-world knowledge (Gordon 2016, 2018). Unlike traditional logic, which treats rules as universally true, Etcetera Abduction recognizes that real-world rules are inherently uncertain and may fail to hold under certain conditions.

The core mechanism of Etcetera Abduction is the *etcetera literal*, which represents the implicit conditions required for a rule to apply. For example, consider the rule “infected individuals test positive”. In reality, this rule only holds when certain conditions are met, such as proper test administration, no interfering substances, and so on. Etcetera Abduction captures this by rewriting the rule as $\forall p (Infected(p) \wedge Etc(0.94, p) \rightarrow TestsPositive(p))$. The etcetera literal $Etc(0.94, p)$ represents all unmodeled factors. The probability value (0.94) reflects how often the rule holds in practice, specifically the conditional probability $P(C|A)$ where C is the consequent and A represents the other antecedents.

The reasoning process operates through iterative backchaining from observations through the knowledge base to find explanations. Starting with a set of observed facts O , the algorithm recursively applies rules in reverse to generate potential explanations. A hypothesis H is complete when it consists entirely of etcetera literals and other assumable predicates. The joint probability of hypothesis H is computed as $P(H) = \prod_{\{h \in H\}} P(h)$, assuming the components are conditionally independent.

Critically, the algorithm is able to unify shared assumptions across multiple observations. When different pieces of evidence can be explained by the same underlying cause, the algorithm uses first-order unification to instantiate that cause only once. This reduces redundancy in the explanation and increases its overall probability, naturally favoring coherent explanations that connect multiple observations through common causes. The system ultimately returns the hypothesis H^* , which represents the most probable explanation given the available evidence and encoded background knowledge.

1.3 Probabilistic Forecasting from Deterministic Models

Populating Etcetera Abduction’s knowledge base requires event probabilities, a fundamental limitation when working with deterministic simulators that produce only point estimates.

We address this by adapting the uncertainty quantification approach of Camporeale et al. (2019). The authors propose a method for converting deterministic model outputs into probabilistic forecasts by learning how prediction uncertainty varies with input conditions. Rather than running computationally expensive ensemble simulations, their approach trains a neural network to predict the uncertainty (standard deviation) of a model’s output based on input features. This uncertainty is learned from historical data by comparing deterministic model predictions against observed outcomes, enabling the construction of full predictive distributions from single deterministic forecasts.

However, to adapt this method to our purposes, we propose the following techniques. Since our simulator lacks stochastic outputs, we create bootstrapped ground truth by systematically perturbing inputs and measuring output variability.

These probabilities populate our Etcetera Abduction knowledge base and inform ETI's uncertainty assessments. Large Language Models convert simulation data into natural language and logic for automated knowledge base matching. We implement this technique in two phases: validation and deployment.

We hypothesize that integrating probabilistic forecasts with ETI and Etcetera Abduction frameworks will enable robust reasoning in deterministic simulation environments.

2 Methodology

2.1 Problem

We consider a deterministic simulator that maps scenario inputs $x = (x_1, x_2, x_3) \in [0,1]^3$ to a scalar mission completion time $\mu(x)$. The simulator produces a single point prediction for each specific scenario without any probabilistic information.

Unlike the simulator used in Camporeale et al. (2019), uncertainty cannot be learned from historical observation-prediction pairs as no such repository of real-world observational data exists. As a result, true prediction errors – calculated as the difference between the observation and the prediction cannot be computed.

Our objective here is to learn an input-dependent uncertainty function $\sigma(x)$ without access to ground-truth uncertainty labels or real observational error data. Thus, to address this problem, we employ an alternate strategy to quantify uncertainty: we generate synthetic error estimates through the simulation itself rather than extracting them from historical comparisons.

Two validation settings are employed to isolate different aspects of the framework. Validation A evaluates whether the custom Accuracy-Reliability (AR) loss function and neural network architecture as put forth by Camporeale et al. (2019) can recover known heteroscedastic uncertainty when true observational noise is available; this is used to confirm whether the learning mechanism itself is correct. Validation B evaluates whether uncertainty as defined via input perturbations can be learned and generalized by the framework, ultimately determining the pseudo-error approach to be used in the absence of real-world observations.

2.2 Validation A: Synthetic Observations and Recovering Uncertainty

To first verify that the AR framework can recover heteroscedastic uncertainty in principle, we construct synthetic data with known ground-truth uncertainty. We construct a total of $M = 125$ scenarios by evaluating all combinations on a $5 \times 5 \times 5$ grid over the input space.

For each configuration x , we define a deterministic prediction function $\mu(x)$ mapping scenario inputs to predicted mission completion time (in minutes):

$$\mu(x) = 10.0 + 10x^1 - 5x^2 + 15x^3.$$

This deterministic function reflects operational intuition about mission duration drivers. Higher uncertainty is assigned to more complex, rougher, and higher-threat scenarios. This ground truth is used only for observation generation and validation, not training. The prediction model is fully deterministic and produces a single point estimate for each input. Next, to introduce heteroscedastic uncertainty, we define a ground-truth uncertainty function:

$$\sigma_{true}(x) = 2.0 + 3x_1 + 2(1 - x_2) + 1x_3.$$

This function assigns higher uncertainty to scenarios characterized by greater task complexity, rougher terrain, and elevated threat levels. The ground-truth uncertainty is used only to generate synthetic observations and is not provided to the model during training. Observed mission durations are generated by sampling from a Gaussian distribution:

$$y_{observed} \sim \mathcal{N}(\mu(x), \sigma_{true}(x)^2),$$

and the prediction errors are computed as:

$$\varepsilon = y_{observed} - \mu(x).$$

This produces simulated stochastic real-world outcomes in which identical inputs can result in variable results due to latent factors, despite the use of a deterministic predictive model. Then, the resulting dataset is

$$D = \{ (x^{(i)}, \varepsilon^{(i)}, \sigma_{true}^{(i)}) \}_{i=1}^M$$

During training, the neural network is provided only with $x^{(i)}$ and $\varepsilon^{(i)}$; the true uncertainty σ_{true} is withheld, forcing the model to infer the uncertainty structure indirectly by recognizing that scenarios with higher σ_{true} produce errors with larger magnitude and variance. This validation step confirms that the custom Accuracy-Reliability loss function as defined by Camporeale et al. (2019) can recover input-dependent uncertainty when observational noise follows a known heteroscedastic structure.

2.3 Validation with Input Perturbations

Next, to address the absence of ground-truth observational data, we utilize a pseudo-error-based approach based on input perturbation. The method is executed as follows: for each nominal scenario $X_{nominal}$ that we wish to characterize, we generate $N = 100$ perturbed inputs

$$x_i = x_{nominal} + \varepsilon_i, \varepsilon_i \sim \mathcal{N}(0, \Sigma(x_{nominal})).$$

And evaluate the simulator to obtain corresponding outputs $y=f(x)$. Rather than defining errors relative to unavailable observations, we then define pseudo-errors relative to the nominal simulator output:

$$e_i = y_i - \mu(x_{nominal}).$$

Empirical uncertainty is then computed as:

$$\sigma_{\text{empirical}(x)} = \text{std}(e_1, \dots, e_N).$$

The neural network is then trained to approximate the mapping $x \mapsto \sigma_{\text{empirical}(x)}$. This is designed to replace repeated Monte Carlo estimation using perturbation with a single forward pass. Thus, Validation B evaluates whether this uncertainty proxy is both learnable and internally consistent.

2.4 Neural Network Training with Accuracy Reliability Cost Function

The neural network learns to predict $\sigma(x)$ by minimizing the AR Cost function. A feed forward neural network is implemented which takes the scenario configuration vector $x = (x_1, x_2, x_3) \in [0, 1]^3$ as input, and outputs a single scalar corresponding to the logarithm of the predicted standard deviation. The predicted uncertainty, then, is obtained by exponentiating this output to ensure positivity.

$$\sigma(x) = \exp(g(x)),$$

where $g(x)$ represents the raw network output.

The training dataset is split into 70% training and 30% validation subsets. Model performance is evaluated using three criteria: (1) the Pearson correlation coefficient between $\sigma(x)$ and $\sigma_{\text{true}}(x)$, used as a diagnostic measure; (2) alignment of reliability diagrams with the diagonal, and (3) minimization of AR cost on the validation set.

Training is performed using the AR cost function, as it penalizes both inaccurate and miscalibrated probabilistic forecasts (Camporeale et al. (2019)). The network is trained via gradient descent, receiving only x^i and ε^i and no ground-truth σ values. It must infer that scenarios with larger, more variable errors require higher $\sigma(x)$ for calibration. The resulting uncertainty estimates are then combined with the deterministic simulator's outputs to enable probabilistic forecasting and eventual reasoning within the ETI and Etcetera Abduction frameworks. Train-test splits were performed at the scenario level to prevent leakage across repeated perturbations of the same nominal input, ensuring that generalization is evaluated on unseen scenarios.

2.5 Integration with Enhanced Tactical Inferencing

At simulation decision points, our probabilistic forecasts integrate with both frameworks. Uncertainty estimates $\sigma(x)$ inform ETI's UoI assessments, which enables decision recommendations grounded in quantified prediction confidence. ETI outputs both action recommendations and justifications based on assessed uncertainty levels. The same forecasts also configure Etcetera Abduction rules (Section 2.4), which operates independently to generate ranked explanations for observed events.

During the abductive inference, the Etcetera Abduction algorithm backchains from observed events through the knowledge base, constructing candidate explanations constructed from assumable predicates and etcetera literals. The

probability assigned to each hypothesis is then computed as the product of the probabilities of the etcetera literals it is composed of. Hypotheses reliant on events with a greater probability are then ranked higher, which allows the system to prefer explanations that are consistent with both the observed data and the learned uncertainty structure of the simulator.

Ultimately, these frameworks complement each other: ETI provides forward-facing decisions under uncertainty, while Etcetera Abduction provides backward-looking explanations, resulting in decision-makers obtaining both actionable recommendations and situational understanding.

3 Results and Discussion

We evaluate whether the proposed uncertainty estimation approach produces probabilistic forecasts that operate in a manner consistent with the requirements of ETI and Etcetera Abduction frameworks. Specifically, we assess calibration, recovery of the input-dependent uncertainty structure, and comparison against fixed-uncertainty baselines.

3.1 Comparison Against Fixed-Uncertainty Baselines

To show the necessity of input-dependent uncertainty estimation, we compared the AR-trained neural network against fixed-uncertainty baselines in which a constant standard deviation σ is applied uniformly across all scenarios. Table 1 displays performance for several constant σ values.

Table 1. AR cost and component metrics for fixed- σ baseline models. Lower values indicate better probabilistic performance.

Fixed σ	CRPS	RS	AR Cost
1.0	3.292	0.0417	1.667
1.5	3.120	0.0265	1.573
2.0	2.986	0.0161	1.501
2.5	2.885	0.0093	1.447
3.0	2.810	0.0048	1.408
5.0	2.722	0.0005	1.361

The best fixed baseline ($\sigma = 5.0$) achieves an AR cost of 1.3161 by assigning the same uncertainty to all scenarios uniformly. While larger fixed σ values reduce reliability error by inflating uncertainty, they do so at the expense of accuracy, resulting in uniformly conservative forecasts that fail to distinguish between low- and high-uncertainty scenarios.

3.2 Validation A and B Results

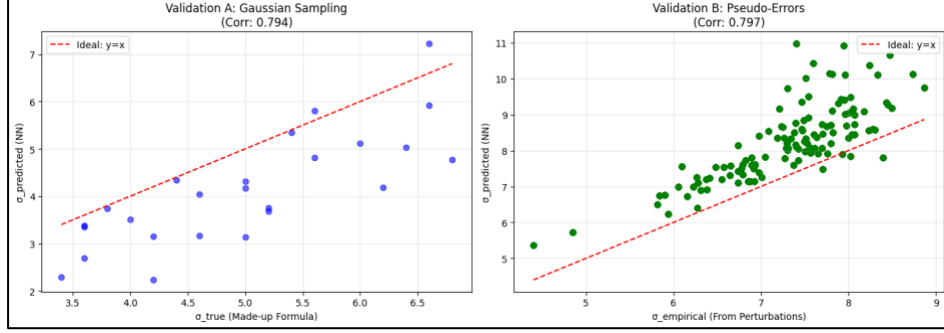


Figure 1: Predicted versus ground-truth uncertainty for Validation A (left) and predicted versus empirical perturbation-based uncertainty for Validation B (right). Alignment with the identity line indicates successful recovery of input-dependent uncertainty structure.

In Validation A, on the synthetic validation dataset, the model achieves a Pearson correlation of 0.794 between predicted uncertainty $\sigma(x)$ and ground truth uncertainty $\sigma_{true}(x)$. Figure 1 (left) shows predicted versus true uncertainty values on the held-out test scenarios. Points cluster around the identity line, indicating that the model successfully recovered the heteroscedastic uncertainty.

In Validation B, the model achieves a Pearson correlation of 0.797 between the predicted uncertainty $\sigma(x)$ and ground truth uncertainty $\sigma_{empirical}(x)$. Figure 1 (right) shows predicted versus empirical uncertainty. The scatter exhibits a strong linear relationship with moderate dispersion, indicating that the network successfully approximates the perturbation-based Monte Carlo estimator using a single forward pass.

4 Conclusion

We presented a framework for enhancing decision-making in deterministic event simulations through probabilistic reasoning using ETI and Etcetera Abduction. The critical challenge was that deterministic simulators lack event probabilities required for robust reasoning; this was addressed by training neural networks to estimate heteroscedastic uncertainty from systematic input perturbations.

Using synthetic data, we demonstrated that a neural network trained with an Accuracy-Reliability loss function can recover meaningful heteroscedastic uncertainty from prediction errors alone. The resulting forecasts are both accurate and well calibrated, as the predicted probabilities align closely with observed outcomes across validation scenarios, indicating that uncertainty estimations in this method are suitable for reasoning systems that rely on probabilistic inputs.

Moreover, we showed that these probabilistic forecasts can be converted into event probabilities at simulation decision points and be used to configure existing abductive reasoning frameworks.

Ultimately, we present a practical and computationally efficient method to make deterministic simulations compatible with probabilistic reasoning systems, while remaining simulator-agnostic. Future work will include applying this approach to operational simulation environments and demonstrating end-to-end reasoning in realistic scenarios, as validation here is limited to synthetic data and simplified uncertainty assumptions.

Acknowledgments. We thank Mark Mittrick and John Richardson for their support in this research program. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Numbers W911NF-23-2-0224, W911NF-26-2-A004, and W911NF-25-2-0040. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Camporeale, E., Chu, X., Agapitov, O.V., Bortnik, J.: On the generation of probabilistic forecasts from deterministic models. *Space Weather* 17(3), 455–475 (2019).
2. Gordon, A.S.: Interpretation of the Heider-Simmel film using incremental etcetera abduction. In: *Advances in Cognitive Systems*, 7, pp. 23–38 (2018).
3. Gordon, A.S.: Commonsense interpretation of triangle behavior. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 3719–3725, Phoenix, AZ (2016).
4. Richardson, John, et al. ‘Enhanced Tactical Inferencing Framework: Future Visualization and Collaboration Demonstration’. DEVCOM Army Research Laboratory Technical Report ARL-TR-9746. (2023).