LLM-Assisted Credibility Scoring for Simulation Models from Raw Time-Series via Automatic Visualization and Interpretable Indicators

Ameya Shrikant[0009-0007-0887-860X], Philipp Andelfinger[0000-0002-0211-7136], and Wentong Cai[0000-0002-0183-3835]

Nanyang Technological University, Singapore
{ameya.shrikant,philipp.andelfinger,aswtcai}@ntu.edu.sg

Abstract. Simulation credibility assessment traditionally relies on substantial human expert involvement, making the process time-consuming and difficult to scale. We propose AutoCred, an LLM-assisted framework that automates credibility assessment while preserving transparency and methodological rigor. The framework takes simulation time-series data and a textual problem description as input. From these inputs, the LLM constructs a structured conceptual model, generates credibility indicators with normalized weights, and produces deterministic scripts for diagnostic visualizations tailored to the simulation model. Evaluation is centered on these visualizations rather than raw time-series data. Structured insights derived from the plots are used to score indicators, which are then aggregated into a final credibility measure. A meta-validation module iteratively checks internal consistency among assumptions, indicators, weights, and scores. Experiments across multiple simulation domains show that AutoCred assigns high credibility to valid configurations and reliably detects inconsistencies with the stated model purpose. The approach enables scalable and interpretable simulation credibility assessment with reduced reliance on continuous human oversight.

Keywords: Simulation model validation · Model credibility · LLM-assisted reasoning · Time-series data

1 Introduction

Simulation models have become key tools in scientific understanding, engineering design, and policy decisions. Since simulation outputs frequently inform real-world action, the trustworthiness of a model for its intended purpose is a central concern. Establishing such credibility remains a demanding and largely expert-driven task [15,8]. Analysts typically inspect diagnostic plots, examine modeling assumptions, compare observed patterns with expected behaviors, and combine these observations into a final credibility judgment. While effective, this process requires substantial expert effort, scales poorly to large parameter studies, and may yield different conclusions across evaluators due to subjective interpretation.

Recent work aiming to reduce subjectivity and the required expert effort has explored partial automation. For example, quantitative learning approaches attempt to learn mappings from simulation outputs to credibility scores [19]. Although such methods reduce manual intervention, they rely on predefined indicators and domain-specific design choices, which can make adaptation to new simulation settings or revised modeling assumptions challenging.

In parallel, large language models (LLMs) have demonstrated the ability to summarize domain knowledge, generate structured explanations, and explicate their reasoning steps. They are pretrained on diverse text corpora that include descriptions of dynamical systems, statistical models, and domain conventions. This exposure suggests that they can reason about models of system evolution over time by relating observed behavior to stated assumptions and objectives, rather than relying solely on predefined numerical mappings. This observation motivates the central question of this paper: *Can the structured reasoning process used by experts in simulation credibility assessment be operationalized through LLM-generated intermediate artifacts, without task-specific training?*

To address this question, we propose AutoCred, an LLM-assisted framework that implements credibility assessment as a sequence of reasoning steps.

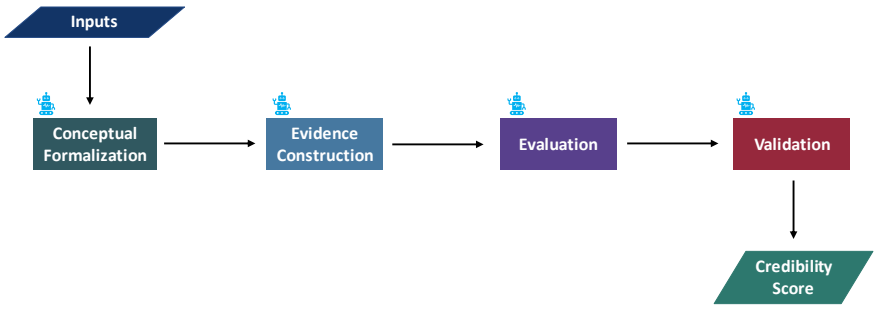

Fig. 1. Overview of the AutoCred workflow.

Taking simulation time-series data and a textual problem description as input, AutoCred structures credibility assessment into four stages (Fig. 1): conceptual formalization, evidence construction, evaluation, and validation. Each stage is executed using a large language model (represented by the robot icon) and produces inspectable intermediate artifacts, allowing the final credibility score to be traced back to observable patterns and documented reasoning steps.

AutoCred is implemented as a structured prompting workflow. In this context, the main contributions of this work are:

- An LLM-driven simulation credibility assessment framework implemented using structured prompting without additional training.
- A workflow that places diagnostic plots at the center of evaluation, using them as direct evidence for credibility scoring.

– An iterative meta-validation mechanism that enforces internal consistency in the credibility assessment process.

2 Background and Related Work

Simulation credibility assessment lies at the core of modeling and simulation practice, aiming to determine whether a model is sufficiently trustworthy for a specific context of use [15,8]. Within the verification and validation (V&V) literature, credibility is established from accumulated evidence rather than a single numerical metric [15,8]. Credibility requires both the adequacy of the model for its purpose and the alignment of its implementation with its specification.

2.1 Simulation Credibility as Structured Evidence

In classical V&V frameworks, credibility is established by iteratively carrying out the following steps [15]: checking the conceptual model, verifying the implementation, assessing operational behavior, and examining data validity. Evidence collected across these steps is combined into an overall expert judgment.

This process relies heavily on human experts for examining diagnostic plots, assessing modeling assumptions, and interpreting behavioral patterns in light of domain expectations. Formal standards such as NASA-STD-7009 and its Credibility Assessment Scale (CAS) decompose credibility into structured factors, emphasizing documentation and context [12]. CAS does not compute a single credibility value; instead, it structures evidence for evaluation by decision-makers.

2.2 Stylized Facts as Validation Targets

One common way to structure validation is through the use of *stylized facts*, which describe recurring patterns in a system's behavior while abstracting from minor variations [11]. They serve as behavioral targets when models aim to reproduce general tendencies rather than specific datasets, e.g., predator–prey oscillations in ecological systems or volatility clustering in financial time series [11].

Traditionally, stylized facts are expressed in natural language and assessed through expert interpretation supported by visualization, limiting reproducibility. Recent work aimed to address this limitation by formalizing stylized facts in a domain-specific language to enable rigorous statistical testing [18]. A challenge of this approach is the requirement for experts to agree on shared definitions of patterns and thresholds.

In contrast, AutoCred operates on natural language descriptions of stylized facts and employs structured LLM-driven reasoning to interpret them. Expected behaviors described in the simulation specification are evaluated through diagnostic visualizations and intermediate reasoning artifacts (cf. Fig. 1). This preserves interpretability while allowing systematic assessment.

2.3 Quantitative and Automated Credibility Assessment

To reduce manual effort, a nascent research thrust has begun exploring quantitative and learning-based approaches that map simulation outputs to credibility indicators or scores. The Quantitative Learning Method (QLM) [19] learns structured mappings from simulation outputs to credibility values from small numbers of expert-provided scores. Cheng et al. [3] extend this direction with an LLM-assisted framework that integrates metrics, visualizations, and expert preference signals to construct evaluation criteria and weights.

These approaches depend on predefined indicators, labeled data, or domain-specific metrics. Such dependencies can make transfer difficult when modeling assumptions, evaluation criteria, or domains change. AutoCred does not train domain-specific mappings. Instead, it uses a pretrained LLM to arrive at documented credibility assessments without requiring task-specific retraining.

2.4 Visualization-Centered Model Evaluation

Visualization plays a central role in simulation analysis. Diagnostic plots support validity checks, identification of behavioral patterns, and anomaly detection [15]. In Bayesian workflows, posterior predictive checks treat visualization as primary evidence for diagnosing model misfit and guiding refinement [5]. More recent work frames diagnostic visualizations as tools for explaining why models fail under certain conditions [6]. Similar perspectives appear in visual analysis of time-dependent observables in complex simulations, where visualization supports interpretation of dynamic behavior [4].

AutoCred builds on this perspective by placing diagnostic plots at the center of the credibility assessment (cf. Fig. 1). Visualizations to assess the model's credibility are generated automatically and serve as structured evidence from which insights are extracted and linked to credibility indicators.

2.5 LLMs and Structured Judgment

Besides producing end-to-end predictions, LLMs are capable of structured judgment tasks by identifying relevant features, generating intermediate representations, and articulating reasoning steps. For instance, work on modeling human moral judgment shows that LLMs can extract meaningful attributes and aggregate them through reasoning mechanisms [7].

A recent work made use of these capabilities to infer conceptual models from simulation code [13] to facilitate V&V efforts. In contrast, our work infers a conceptual model from natural language and uses it as an intermediate step, with the objective of determining overall credibility judgments automatically.

A well-known drawback of LLMs is their sensitivity to the prompting and the possibility of generating hallucinations stemming from the probabilistic nature of text generation and the diversity of the pretraining data. Self-reflection and refinement of previous outputs has been shown to ameliorate these issues [10]. AutoCred employs this idea through iterative consistency checks and potential correction of its credibility assessments.

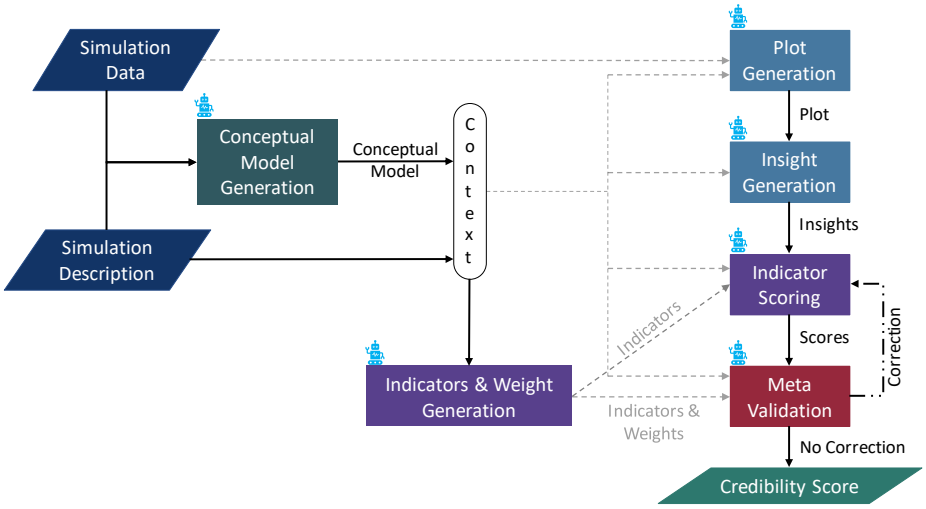

Fig. 2. AutoCred workflow. Parallelograms denote information entering or leaving the system. Solid arrows indicate the main forward reasoning flow. Dashed arrows denote contextual conditioning. The dotted feedback loop represents meta-validation refinement. The robot icon denotes LLM-driven steps. Colors correspond to Fig. 1.

3 Methodology

We designed AutoCred based on three key requirements:

1. **Context awareness.** The credibility evaluation must remain tied to the model's objectives, assumptions, and intended use.
2. **Evidence-based reasoning.** The evaluation must provide a transparent link between observable behaviors and the criteria used for assessment.
3. **Internal coherence.** Indicators, weights, and assigned scores must be logically consistent with one another and with the stated modeling assumptions.

Figure 2 presents the AutoCred workflow. AutoCred operates on two external inputs: (i) a simulation description, and (ii) multi-run simulation outputs.

The simulation description defines objectives, assumptions, and expected behaviors. The simulation time-series data provide observable realizations of the modeled system. From these inputs, AutoCred constructs a sequence of reasoning steps that lead to a credibility score.

The framework satisfies the above properties as follows: **1.** The conceptual model and description remain active conditioning inputs throughout the pipeline. **2.** Scores are derived exclusively from structured insights grounded in diagnostic visualizations. **3.** The meta-validation loop evaluates consistency among assumptions, indicators, weights, and scores before aggregation.

3.1 Input Preparation

AutoCred operates on two inputs: simulation description and simulation time-series data. The data consist of R independent runs, denoted by $\{X^{(r)}\}_{r=1}^{R}$, each containing T time steps. A short excerpt of the first run (e.g., the first 100 time steps) is provided during conceptual model construction to anchor variable semantics and numerical scale. The excerpt is not scored.

To account for potential stochasticity of the simulation model, an averaged reference run is constructed by taking the pointwise mean across runs. This averaged run is appended to the dataset and treated as run $R + 1$.

3.2 Reasoning Steps

AutoCred structures the credibility assessment into seven reasoning steps, all of which are carried out using a large language model, specifically OpenAI's ChatGPT-5.1 [14].

Step 1: Conceptual Model Generation
Inputs: simulation description and data excerpt.
Output: structured conceptual model (JSON schema).

The LLM converts the narrative description into a structured representation of entities, state variables, parameters, mechanisms, assumptions, and expected behaviors. This representation forms the conceptual model. The conceptual model and the simulation description are combined into a shared *model context* (see Fig. 2), which conditions all subsequent modules.

Step 2: Indicator and Weight Generation
Inputs: model context
Output: indicator set with normalized weights.

Indicators translate the model's stated expectations into measurable evaluation criteria. For example, in a flocking model, an indicator such as *collective alignment formation* captures whether agents converge toward a common heading. Each indicator corresponds to a clearly defined behavioral property derived from the model description. Associated weights reflect the declared importances.

Step 3: Visualization Generation
Inputs: model context and data excerpt.
Output: deterministic plotting module and standardized plots.

AutoCred constructs evidence through diagnostic visualization. The LLM generates a deterministic plotting script tailored to the simulation model. The visualization structure is fixed across runs, while the underlying data vary. Prompt templates steer the LLM towards plotting of model-relevant statistics rather than raw time-series columns. Figure 3 shows example plots from Section 4.4.

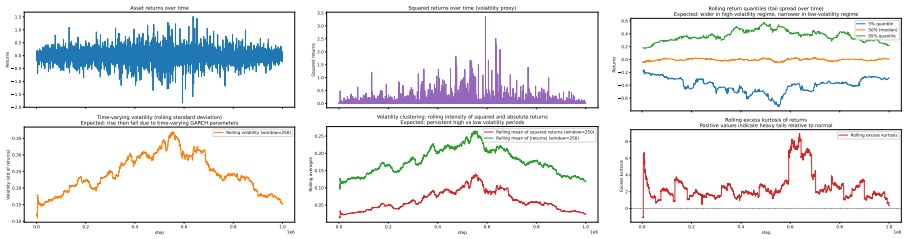

Fig. 3. Diagnostic plots from the GARCH(1,1) experiment. The plots were generated automatically using LLM-produced Python scripts.

Step 4: Insight Generation
Inputs: model context and run-specific plots.
Output: structured insight set per run.

Visual patterns are translated into explicit observations and resulting insights. The insights identify relevant patterns and explain their relevance with respect to the simulation description. They are directly tied to observations from the plots. All subsequent scoring decisions refer to these insights.

Step 5: Indicator Scoring
Inputs: *model context*, indicators, and run-specific insights.
Output: per-run indicator scores $s_k^{(r)} \in [0, 1]$.

For each run, the LLM assigns scores to the fixed set of indicators based on the structured insights derived from the plots. Scores vary across runs, while the indicator definitions and weights remain unchanged. Each assigned value is justified by reference to the corresponding insights, ensuring traceability from score to observed evidence.

Step 6: Meta-Validation
Inputs: model context, indicators, weights, and indicator scores
Output: confirmation or correction directive.

Meta-validation checks internal consistency by evaluating whether indicators match the conceptual model, whether weights reflect declared priorities, and whether scores align with observed evidence.

If inconsistencies are detected, corrections are applied and all runs are rescored until no further correction is required. The dotted feedback loop in Fig. 2 represents this process, after which the pipeline proceeds to aggregation.

Step 7: Credibility Aggregation
Inputs: indicator weights and per-run scores.
Output: overall credibility score.

Aggregation combines run-level scores into a final credibility measure. For each run,

$$A^{(r)} = \sum_{k=1}^{K} w_k \, s_k^{(r)}. \tag{1}$$

The overall credibility score is

$$C = \frac{1}{R} \sum_{r=1}^{R} A^{(r)}. \tag{2}$$

Here, $w_k \in [0, 1]$ is the normalized weight of indicator k, with $\sum_{k=1}^{K} w_k = 1$ and $s_k^{(r)} \in [0, 1]$ is the score assigned to indicator k for run r.

The aggregation step is deterministic. Each scalar value can be traced back to the corresponding indicators, insights, plots, and conceptual model elements. The credibility score therefore reflects the accumulated results of the preceding reasoning steps.

From a computational perspective, the cost of AutoCred is dominated by language model inference and increases linearly with the number of runs, at approximately 14.65 seconds per run, while remaining largely insensitive to time-series length.

4 Experimental Evaluation

This section evaluates AutoCred with respect to three specific questions:

1. *Can AutoCred replicate ground truth labels derived from human judgments on model behavior?*
2. *Can AutoCred distinguish intended model behavior from deliberately unrealistic parameter regimes?*
3. *Does the meta-validation loop detect and correct inconsistent indicator weights or scoring rules?*

All experiments were executed using the same pipeline configuration. No domain-specific tuning or prompt modifications were made. For each configuration, AutoCred was executed seven independent times. Repeated executions under identical configurations yield stable credibility ranges across runs, as reflected in Tables 3, 5, and 6.

4.1 Swarm Behavior Classification Against Ground Truth Labels

Objective. As our first experiment, we test whether AutoCred produces judgments consistent with ground truth labels assigned by human annotators.

Dataset. We use the *Swarm Behavior* dataset from the UCI Machine Learning Repository [1]. The dataset contains approximately 25,000 simulation-derived instances generated from a Boids model. Each instance represents a single timestep snapshot of agent positions and velocities, with no temporal sequence across snapshots.

Each snapshot is labeled independently by human annotators under three binary categories: *aligned*, *flocking*, and *grouping*. These labels were generated outside AutoCred and therefore provide an external reference.

Evaluation Procedure. Fifty instances were randomly sampled for evaluation. For each behavioral category, AutoCred outputs a probability estimate $\hat{p}$ indicating confidence that the snapshot exhibits the corresponding behavior.

Predictions are compared with ground-truth labels $y \in \{0, 1\}$ using binary cross-entropy (BCE):

$$\mathrm{BCE}(y, \hat{p}) = -\left[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\right]. \tag{3}$$

BCE measures the divergence between predicted probabilities and true binary outcomes. A BCE of 0 indicates perfect agreement. In a balanced binary task, random guessing yields an expected BCE of approximately 0.69. Lower values indicate stronger probabilistic alignment.

Table 1. Binary cross-entropy loss and random baseline on Swarm Behavior dataset

Class	BCE Loss	Random Baseline
Aligned	0.25	0.692
Flocking	0.15	0.686
Grouping	0.27	0.686

Results. All observed losses are substantially below the random baseline across the three independent behavioral dimensions. This indicates that AutoCred's evaluation of aligned motion, flocking, and grouping from snapshot-based evidence is consistent with externally curated labels.

This experiment also demonstrates adaptability. Although the Swarm Behavior dataset is structured around individual snapshots of simulation states, only minor preprocessing adjustments were required. The core reasoning steps of AutoCred remained unchanged, indicating that the framework does not require strictly time-series-structured input data to generate meaningful judgments.

4.2 Wolf-Sheep Predator-Prey Model

Objective. This experiment tests whether AutoCred distinguishes sustained predator-prey coexistence from parameter settings that lead to rapid ecological collapse.

Model Description. The NetLogo Wolf–Sheep model [16] simulates interacting predator and prey populations. The simulation description specifies sustained predator–prey coexistence as the primary credibility criterion.

Experimental Setup. Two parameter regimes were constructed with 30 runs and 300 time-steps per run: In the realistic regime, reproduction rates, energy transfer, and grass regrowth allow sustained coexistence over time. In the unrealistic regime, grass regrowth is set to zero and wolf energy gain is reduced, leading to early predator extinction and prey overpopulation.

Table 2. Wolf-Sheep parameter configurations

Parameter	Realistic	Unrealistic
Initial number of sheep	100	100
Initial number of wolves	50	80
Sheep reproduce (%)	4	4
Wolf reproduce (%)	5	5
Sheep gain from food	4	4
Wolf gain from food	20	6
Grass regrowth time	30	0

Table 3. Credibility score ranges for the Wolf-Sheep experiment.

Configuration	Credibility range (%, [min, max])
Realistic regime	[93.8, 95.2]
Unrealistic regime	[9.1, 11.0]

Results. Runs exhibiting sustained predator–prey oscillations received high credibility, whereas runs leading to early extinction received low credibility. The separation between regimes reflects the ecological criterion specified in the simulation description. This outcome confirms that AutoCred evaluates the model relative to the stated ecological objective.

4.3 Flocking Model (NetLogo)

Objective. This experiment tests whether AutoCred distinguishes stable flock formation from parameter settings that disrupt local interaction rules.

Model Description. The NetLogo Flocking model [17] implements three local rules: alignment (steering toward neighbors' headings), cohesion (steering toward neighbors' center of mass), and separation (avoiding crowding).

The simulation description defines credible behavior as the formation of at least one dominant locally aligned and cohesive group.

Experimental Setup. Five regimes were tested with 15 runs and 250 time-steps per run:

The realistic configuration balances alignment, cohesion, and separation to produce stable flock formation. The distorted regimes isolate or exaggerate individual mechanisms: cohesion-only produces clustering without spacing, jitter produces unstable turning, no-interaction prevents collective formation, and separation-only prevents cohesive grouping.

Table 4. Flocking parameter configurations.

Parameter	Realistic	Cohesion-only	Jitter	No-interaction	Separation-only
Vision	5	7	7	0.5	5
Max separate turn	1.5	0.5	30	3	8
Minimum separation	1	0.2	1	1	5
Max align turn	5	0.44	30	5	1
Max cohere turn	3	8.41	32	3	1

Table 5. Credibility score ranges for the Flocking experiment.

Configuration	Credibility range (%, [min, max])
Realistic flocking	[93.8, 96.1]
Cohesion-only	[29.0, 31.3]
Jitter regime	[21.4, 22.9]
No-interaction regime	[24.7, 26.3]
Separation-only regime	[9.9, 11.5]

Results. The realistic regime is clearly distinguished from the altered regimes. Configurations that prevent stable flock formation or produce unstable collective motion receive substantially lower credibility. The results are consistent across runs and reflect the interaction rules specified in the simulation description, confirming that AutoCred's evaluation conforms with the simulation description.

In addition, we conducted an ablation experiment for the realistic flocking configuration by removing the conceptual model from the model context. Under this modification, the generated credibility score decreased from 94.4% to 87.9%. This reduction demonstrates the benefit of explicitly formalizing expectations about model behavior, as the conceptual model provides structured guidance for interpreting observable patterns.

4.4 GARCH(1,1) Financial Time-Series Model

Objective. The GARCH model of asset volatility is defined by a simple recurrence relation. It allows us to study AutoCred's ability to generalize beyond commonly studied agent-based models.

Model Description. The GARCH(1,1) model [2] generates a time series of asset returns with time-varying volatility:

$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2.$$

Here, σ_t^2 denotes the conditional variance (volatility) of returns at time t. The term ϵ_{t-1} represents the shock (unexpected return component) at time $t-1$, and ϵ_{t-1}^2 captures the magnitude of that shock. The term σ_{t-1}^2 is the previous period's conditional variance. The parameter ω controls the baseline variance level, α determines how strongly recent shocks affect current volatility, and β governs the

persistence of past volatility over time. Our baseline configuration uses $\omega = 0.01$ with $\alpha, \beta \in [0.01, 0.09]$ with a linear increase to 0.09 and subsequent decrease to 0.01 over the course of the simulation.

In financial time series, an important modeling consideration is the *leverage effect*, according to which negative returns lead to larger increases in future volatility than positive returns of similar magnitude. GARCH(1,1) does not distinguish between positive and negative shocks, allowing us to test AutoCred's ability to capture the absence of this effect.

Experimental Variants. Four configurations were evaluated using 20 runs and one million time-steps per run:

1. *Default configuration:* Volatility evolves according to standard GARCH(1,1), producing volatility clustering without leverage asymmetry.
2. *Constant volatility:* The volatility recursion is removed, yielding fixed-variance returns.
3. *Imposed leverage requirement with explicit disclosure:* The simulation description states that the data were generated by a symmetric GARCH(1,1) model. At the same time, the evaluation demanded the presence of a leverage effect, which is incompatible with GARCH(1,1).
4. *Imposed leverage requirement without disclosure:* The same leverage requirement was imposed, but the simulation description did not state that the model was GARCH(1,1) or that it was symmetric. In this case, AutoCred must rely only on observed data patterns and the stated expectations.

Table 6. Credibility score ranges for GARCH(1,1).

Configuration	**Credibility range (%, [min, max])**
Default configuration	[96.1, 97.5]
Constant volatility	[20.6, 22.3]
Imposed leverage (disclosed)	[90.5, 92.6]
Imposed leverage (undisclosed)	[83.8, 86.9]

Results. The GARCH experiment demonstrates that AutoCred evaluates the model in accordance with its stated assumptions. The default configuration, which exhibits time-varying volatility, received high credibility. The constant-volatility variant was penalized because it removes volatility clustering, a central property of the model.

More importantly, the two leverage variants illustrate how declared assumptions influence the evaluation. When the simulation description clearly stated that the data were generated by a symmetric GARCH(1,1) model, the imposed leverage requirement was recognized as incompatible with the model specification and did not substantially reduce credibility. When the simulation description did not specify that the data were generated by a symmetric GARCH(1,1) model, the missing leverage effect led to a lower credibility score. As in the previous experiments, AutoCred correctly evaluated the simulations according to the assumptions stated in the simulation description.

4.5 Meta-Validation Consistency Test

Objective. This experiment tests whether the meta-validation loop detects and corrects inconsistent evaluation rules.

Setup. In the Flocking experiment (realistic configuration), the indicator weights were artificially set equal regardless of declared importance. Extreme scoring constraints were introduced (e.g., forcing scores above 0.96 or below 0.15 under arbitrary conditions).

Result. AutoCred correctly identified inconsistencies between weights, scores, and the simulation description. It proposed targeted corrective adjustments, restoring differentiated weights and coherent scoring behavior. For example, the system flagged that equal indicator weights contradicted the declared importance hierarchy in the simulation description and recommended restoring differentiated weights before rescoring.

Beyond this specific setup, meta-validation is triggered infrequently in practice, occurring in approximately one out of every twenty trials under identical data and configuration, indicating that the reasoning pipeline is largely stable. This further confirms that the meta-validation loop corrects inconsistent indicator weights and scoring rules before final aggregation.

4.6 Qualitative Analysis

In addition to aggregate scores, AutoCred produces structured reasoning artifacts. For the realistic flocking configuration, a representative run yielded the following LLM-generated indicators and scores:

> **collective-alignment-formation (w=0.35, s=0.96):** `Alignment rises monotonically from near zero to ∼0.9 by step 250 and then stabilizes, matching the expectation that moderate/large vision and balanced turns yield highly aligned, coherent flock motion after transient reorganization.`
>
> **spatial-cohesion-and-spacing (w=0.30, s=0.94):** `Cohesion remains in a mid-high, bounded band with smooth oscillations and mean NN distance declines from ∼2.6 to ∼1.1 while stabilizing, which aligns with expectations of moderate, stable compactness without collapse or full dispersion under successful flocking.`
>
> **collision-and-overlap-control (w=0.20, s=0.93):** `Minimum NN distance stays low but nonzero (∼0.05-0.3) and noisy without evidence of collapse, indicating rare and brief close approaches but no persistent overlaps, consistent with effective minimum-separation enforcement.`
>
> **turn-rate-dynamics-consistency (w=0.15, s=0.95):** `Mean turn rate shows a high spike at initialization then decays from ∼4.5 to ∼1.5 with residual variability, matching the expectation of elevated turning during transient reorganization followed by lower, stable values during established flocking.`

These excerpts illustrate how indicator scores are directly grounded in observable patterns from the diagnostic plots. In one additional trial, a high BCE of 0.61 was observed for the flocking class. Our inspection of the data revealed that in these cases, the ground truth labels were erroneous and the classification produced by AutoCred was in fact correct [9]. This finding suggests an alternative use of AutoCred for auditing human-labeled simulation data.

Taken together, the results address the three evaluation questions: AutoCred aligns with external labels, distinguishes intended behavior from altered regimes, and meta-validation successfully detects and corrects inconsistencies.

5 Conclusion

We presented AutoCred, a framework that structures simulation credibility assessment as a reasoning process supported by a large language model. The framework processes a simulation description and simulation time-series data through a sequence of structured steps, producing inspectable artifacts that link credibility scores to observable evidence and stated modeling assumptions.

Across the experiments, several high-level insights emerge. First, AutoCred replicated human-labeled behavioral categories in the swarm dataset, demonstrating alignment with external ground truth. Second, it consistently distinguished realistic model configurations from unrealistic parameter combinations in both predator–prey and flocking simulations. Third, the GARCH experiment showed that contradictory and unsatisfied assumptions are identified correctly. Finally, the meta-validation experiments confirmed that inconsistencies in indicator weights or scoring rules are detected and corrected before aggregation.

Our framework and experiments represent a step toward a higher degree of automation in the validation of simulation models. Although AutoCred cannot replace human expertise and oversight, it provides a structured and transparent environment in which expert supervision can focus on reviewing indicators, weights, and meta-validation decisions. In this way, the framework reduces manual validation efforts while preserving interpretability and methodological rigor. Future work will focus on extending the framework to provide recommendations for repairing identified model defects.

Acknowledgements

This research was supported by the RIE2025 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

1. Swarm Behaviour. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5N02J.
2. Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.
3. Qi Cheng, Licheng Liu, Qing Zhu, Runlong Yu, Zhenong Jin, Yiqun Xie, and Xiaowei Jia. Llm-based evaluation policy extraction for ecological modeling. *arXiv preprint arXiv:2505.13794*, 2025.

4. Lena Cibulski, Fiete Haack, Adelinde Uhrmacher, and Stefan Bruckner. Visual analysis of time-dependent observables in cell signaling simulations. *arXiv preprint arXiv:2509.08589*, 2025.
5. Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in bayesian workflow. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 182(2):389–402, 01 2019.
6. Alex Kale, Ziyang Guo, Xiao Li Qiao, Jeffrey Heer, and Jessica Hullman. Evm: Incorporating model checking into exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):208–218, January 2024.
7. Joe Kwon, Sydney Levine, and Joshua B Tenenbaum. Neuro-symbolic models of human moral judgment: Llms as automatic feature extractors. 2023.
8. Averill M. Law. How to build valid and credible simulation models. In *2022 Winter Simulation Conference (WSC)*, pages 1283–1295, 2022.
9. Zihao Liu, Ryan McArdle, and Gianni Orlando. Classifying the behaviors of boid swarms. *Institute for Artificial Intelligence*, 2020.
10. Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc., 2023.
11. Matthias Meyer. How to use and derive stylized facts for validating simulation models. In *Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*, pages 383–403. Springer, 2019.
12. National Aeronautics and Space Administration. Nasa standard for models and simulations. Technical Report NASA-STD-7009, NASA, 2008.
13. Rumyana Neykova and Derek Groen. Reversed model verification by inferring conceptual models from simulation code. In Michael H. Lees, Wentong Cai, Siew Ann Cheong, Yi Su, David Abramson, Jack J. Dongarra, and Peter M. A. Sloot, editors, *Computational Science – ICCS 2025*, pages 361–368, Cham, 2025. Springer Nature Switzerland.
14. OpenAI. ChatGPT-5.1. `https://openai.com`, 2025. Large language model.
15. Robert G. Sargent. Verification and validation of simulation models. *Journal of Simulation*, 7(1):12–24, 2013.
16. Uri Wilensky. Netlogo wolf sheep predation model. `http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation`, 1997. Center for Connected Learning and Computer-Based Modeling, Northwestern University.
17. Uri Wilensky. Netlogo flocking model. `http://ccl.northwestern.edu/netlogo/models/Flocking`, 1998. Center for Connected Learning and Computer-Based Modeling, Northwestern University.
18. Pia Wilsdorf, Marian Zuska, Philipp Andelfinger, Adelinde M Uhrmacher, and Florian Peters. Validation without data-formalizing stylized facts of time series. In *2023 Winter Simulation Conference (WSC)*, pages 2674–2685. IEEE, 2023.
19. Jiayi Zhang, Yuanjun Laili, Jiabei Gong, Lin Zhang, and Lei Ren. A quantitative learning method for simulation model evaluation using l-shade optimized structured regression. In *Asia Simulation Conference*, pages 3–15. Springer, 2025.