# NNV3: Expanding Neural Network Verification to New Architectures and Domains*

Anne M. Tumlin[1], Samuel Sasaki[1], Ben Wooding[1], Diego Manzanas Lopez[1], Muhammad Usama Zubair[2], Navid Hashemi[1], Hongchao Zhang[1], Waseem Abbas[2], Ipek Oguz[1], Meiyi Ma[1], and Taylor T. Johnson[1]

[1] Vanderbilt University, USA
[2] The University of Texas at Dallas, USA

**Abstract.** We present NNV3, the latest version of the Neural Network Verification (NNV) tool, a MATLAB framework for formal verification of deep learning models and learning-enabled cyber-physical systems. Building on the set-based reachability foundation of NNV 1.0 (FFNNs, CNNs, NNCS) and NNV 2.0 (RNNs, SSNNs, neural ODEs), NNV3 introduces new members of the Star-set family: ModelStar for verifying networks under weight perturbation, VolumeStar for video and 3D volumetric inputs, and GraphStar for graph neural networks. A conformal-inference-based probabilistic reachability mode complements sound analysis for problems where exact verification is intractable, while FairNNV certifies counterfactual and individual fairness properties over continuous input regions. NNV3 introduces new benchmarks for malware detection, graph-based power-system models, medical imaging, variable-length time series data, and action recognition. NNV3 also incorporates tutorials and developer guides through a unified documentation site. This paper details these major updates, demonstrating NNV's maturation into a comprehensive, robust, and accessible verification tool for a diverse range of AI systems.

## 1 Introduction

Deep neural networks (DNNs) have become integral to solving complex problems across various domains, from image classification to autonomous control. However, their deployment in safety-critical applications is hindered by their opaque nature and susceptibility to adversarial perturbations. Formal verification provides a means to analyze and rigorously guarantee the behavior of these models, which is essential for establishing trust in AI-powered systems.

The Neural Network Verification (NNV)[1] tool [72] was introduced as a comprehensive, open-source MATLAB toolbox to tackle this challenge. NNV is built around a powerful computation engine that performs set-based reachability analysis, computing the set of all possible outputs for a given set of inputs. The initial

* A.Tumlin and S.Sasaki are co-first authors.
[1] https://github.com/verivital/nnv/

release of NNV focused on providing exact and over-approximate reachability for feed-forward neural networks (FFNNs), convolutional neural networks (CNNs), and neural network control systems (NNCS) using a variety of set representations like polyhedra, zonotopes, and the novel star set.

Building on this foundation, NNV 2.0 [47] expanded the tool's scope to handle a wider array of complex and dynamic network architectures. It introduced verification support for neural ordinary differential equations (neural ODEs), recurrent neural networks (RNNs), and semantic segmentation neural networks (SSNNs). This version also improved scalability with new relaxed reachability methods [71] and enhanced usability by supporting standard community formats like ONNX [50] and VNNLIB [54].

This paper presents NNV3, an evolution of the NNV tool that addresses emerging challenges in AI verification and broadens the tool's applicability to new domains and data types. While previous versions focused on expanding architectural support, NNV3 introduces verification techniques for new classes of properties and extends reachability analysis to previously intractable data modalities. The contributions span three categories. **Algorithmic and model improvements** introduce reachability-based verification for parameter perturbations (ModelStar [90, 91]), spatio-temporal data (VolumeStar [55]), graph-structured models (GNNV with GraphStar [75]), probabilistic guarantees via conformal inference [27], fairness properties (FairNNV [74]), and variable-length time-dependent networks [51]. **New application domains** unlocked by these algorithms include video classification [55], medical-image segmentation [26], power-system analysis [75], malware detection [53], ethical decision-making [74], and deployment-uncertainty robustness [90, 91]. **System upgrades** include a unified documentation site[2] that consolidates the user guide, developer guide, API reference, *etc.*; per-feature details and the supporting infrastructure changes are presented in Section 3. These advancements solidify NNV's position as one of the most comprehensive and versatile verification frameworks available to the research community.

## 2 NNV3 vs. NNV 2.0

The core architecture of NNV, illustrated in Fig. 1, is composed of two primary modules: the **Computation Engine** and the **Analyzer**. The **Computation Engine** parses neural network and system models (ONNX and MATLAB formats) and performs layer-by-layer reachability analysis using set-based abstractions, including Star sets [70], ImageStar [65], and newly introduced representations such as VolumeStar (*a.k.a* VideoStar) [55], ModelStar [90, 91], and GraphStar [75]. In NNV3, the computation engine supports both sound reachability and a probabilistic reachability mode [27], the latter enabling scalable verification via sampling-based uncertainty quantification. The **Analyzer** consumes the resulting reachable sets or evaluation traces to verify system-level properties, including robustness, safety specifications expressed in VNNLIB, and

[2] https://verivital.github.io/nnv/

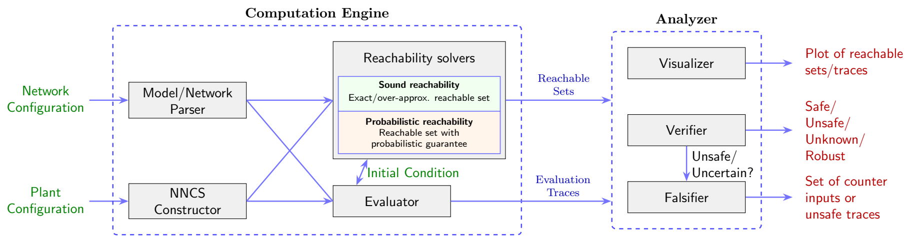


Fig. 1: Updated NNV verification pipeline, composed of a computation engine and analyzer, and extended to support both sound and probabilistic reachability analysis.

fairness constraints, and supports visualization, verification, and counterexample generation. Together, these extensions broaden the scope of verifiable models while preserving the modular structure of the NNV framework.

Practitioners working with heterogeneous, real-world AI systems require a unified and extensible verification framework. This drives the continued evolution of NNV. Table 1 illustrates that NNV and its prior iterations support a broad spectrum of architectures and applications, including feedforward and convolutional networks (FFNNs and CNNs), recurrent models (RNNs), semantic segmentation (SSNNs), neural ODEs, and neural network control systems (NNCSs). With the release of NNV3, we expand the tool to emerging and increasingly important model classes, including graph neural networks (GNNs) and video classification architectures (VolumeStar). NNV3 addresses a broader class of deployment-relevant properties, including algorithmic fairness, parameter and weight perturbations, and probabilistic guarantees. These capabilities enable verification beyond conventional robustness analysis and reflect practical concerns encountered in real-world deployment.

NNV has been evaluated consistently in community benchmarks such as VNN-COMP [39] and ARCH-COMP [46, 56], and has served as the foundation for tutorials at DESTION [67], EMSOFT [69], DSN [36], SPIE, IAVVC, AAAI, *etc.*, underscoring its maturity and continued relevance to the verification community (see the user guide[3] and the tutorials index[4] for further details).

The remainder of the paper is organized as follows. Section 3 summarizes major new features and capabilities. Section 4 introduces the extended domain applications. We then validate new features in Section 5 and refresh the comparison with the MathWorks AI Verification Library. Finally, we discuss related works in Section 6, and conclude the paper in Section 7.

[3] `https://verivital.github.io/nnv/user-guide/index.html`
[4] `https://verivital.github.io/nnv/examples/index.html`

Table 1: Overview of major features available in NNV. Items in regular weight are NNV 1.0 baseline; *blue italics* denote additions introduced in NNV 2.0; **purple bold** denote new capabilities introduced in NNV3.

| **Feature** | **Supported** (NNV 1.0, *NNV 2.0*, **NNV3**) |
|---|---|
| Neural Network Type | FFNN, CNN, *NeuralODE*, *SSNN*, *RNN*, **GNN**, **TDNN**, **3D CNN** |
| Layers | MaxPool, Conv, BN, AvgPool, FC, *MaxUnpool*, *TC*, *DC*, *NODE*, **GCN**, **GINE**, **Conv3D** |
| Activation functions | ReLU, Satlin, Sigmoid, Tanh, *Leaky ReLU*, *Satlins* |
| Plant dynamics (NNCS) | Linear ODE, Nonlinear ODE, Continuous & Discrete Time, *HA* |
| Set Representation | Polyhedron, Zonotope, Star, ImageStar, **VolumeStar**, **ModelStar**, **GraphStar** |
| Star Reach methods | exact, approx, abs-dom, *relax-** |
| Reachable set visualization | exact and over-approximation |
| Verification | Safety, Robustness, *VNNLIB*, **Fairness**, **Weight Perturbation**, **Probabilistic** |
| Miscellaneous | Parallel computing, counterexample generation, *ONNX*, **CI/CD** |

## 3 Overview and Features

This section summarizes the major features introduced in NNV3 and in Table 1. These additions extend the reachability-based verification capabilities of prior versions of NNV to new model classes, data modalities, and verification objectives, while preserving soundness guarantees and compatibility with existing analysis pipelines. Section 3.1 presents the per-feature highlights; Section 3.2 describes the underlying engine and extensibility upgrades.

### 3.1 Feature Highlights

**ModelStar [90, 91]:** Neural networks deployed in practice are subject to parameter uncertainty arising from quantization, numerical imprecision, and hardware faults [24, 61, 88]. To address this, NNV3 introduces ModelStar, a star-set–based representation for interval-bounded weight perturbations. ModelStar enables reachability analysis under simultaneous input and parameter uncertainty. For networks with a single perturbed layer and singleton inputs, the reachable set of the perturbed layer is computed exactly; for multi-layer perturbations, sound over-approximations are constructed using Star sets. ModelStar is currently implemented for fully-connected and 2D convolutional layers and can be readily implemented for verification against weight perturbations in any linear layer. Rounding errors introduced by quantized compression can be modeled as interval-bounded specifications for ModelStar–as demonstrated in our experiments in Section 5–but fixed-point inference arithmetic (discrete operations) is not yet modeled.[5]

**VolumeStar (VideoStar) [55]:** NNV can formally verify the robustness of video classifiers. Verification of neural networks operating on spatio-temporal data, such as videos and volumetric medical images, presents significant scalability challenges due to input dimensionality. NNV3 extends the Star and ImageStar representations [65, 70] to VolumeStar, a set abstraction for spatio-temporal inputs, *e.g.*, an image at each time step. VolumeStar supports reachability analysis of architectures with 3D convolutional and pooling layers, including video

[5] https://verivital.github.io/nnv/theory/weight-perturbation.html

classification networks (such as C3D [64] and I3D [12]), as well as 3D medical imaging models [89]. Using VolumeStar reachability, NNV can formally certify classification robustness for all admissible spatio-temporal perturbations within a specified input set. In addition, NNV3 extends star-based reachability analysis to time-dependent neural networks (TDNNs) by allowing the analysis horizon to vary over a bounded temporal range, generalizing prior support for fixed-length recurrent models.[6]

**GraphStar [75]:** Graph neural networks (GNNs) are increasingly used as surrogates for numerical solvers in domains such as molecular modeling, traffic forecasting, and power-system analysis [85]. NNV3 introduces GNNV, extending reachability-based verification to graph-structured learning models. GNNV implements GraphStar, a generalization of Star sets that captures both graph connectivity and uncertainty in node and edge features. This abstraction enables exact propagation of affine message-passing operations and sound over-approximation of ReLU nonlinearities in common GNN architectures, including graph convolutional networks (GCNs) [40] and graph isomorphism networks with edge features (GINE) [32]. GraphStar reachability is demonstrated on power-system case studies, including power flow, optimal power flow, and cascading failure analysis [77].[7]

**Probabilistic Verification [27]:** Exact reachability analysis can become intractable for large networks due to exponential complexity in the number of unstable nonlinear activations. To address this limitation, NNV3 integrates a probabilistic, model-agnostic verification approach based on conformal inference. Given an input set, a neural network, and a specification, verification is performed via sampling rather than exhaustive propagation. This approach scales with inference cost and is independent of network architecture, providing probabilistic coverage guarantees for models that are beyond the practical reach of exact methods.[8]

**Fairness Verification [74]:** As machine learning systems are deployed in high-stakes decision-making settings, verification of fairness properties has become increasingly important [48]. NNV3 integrates FairNNV, a reachability-based framework for formally validating fairness specifications over continuous input regions. FairNNV verifies specifications corresponding to counterfactual fairness [42], which requires predictions to remain invariant under changes to sensitive attributes, and individual fairness, which enforces similar outcomes for inputs within a bounded neighborhood. Fairness is quantified using the Verified Fairness (VF) score, defined as the proportion of inputs for which fairness properties are formally certified.[9]

[6] https://verivital.github.io/nnv/theory/imagestar-volumestar.html
[7] https://verivital.github.io/nnv/theory/gnn-reachability.html
[8] https://verivital.github.io/nnv/theory/probabilistic.html
[9] https://verivital.github.io/nnv/theory/fairness.html

### 3.2 Core Upgrades & Extensibility

**Star-set family unification.** VolumeStar, GraphStar, and ModelStar are not disjoint abstractions but instances of a common Star-set template, parameterized by an anchor, a generator basis, and a polyhedral constraint on the generator coefficients. Variants differ in the rank and indexing of the underlying tensor (vector for Star, 3D for ImageStar, 4D for VolumeStar) or in auxiliary structure (graph adjacency for GraphStar, perturbed-parameter dimensions for ModelStar). Extending NNV to a new modality therefore reduces to selecting these parameters and implementing layer-specific dispatch over the shared propagation kernel.

**Engine-level changes.** Several core-engine upgrades support the new abstractions: a GNN wrapper that maps message-passing layers onto Star-set affine operations and routes adjacency through GraphStar; a 4D-tensor extension of the ImageStar reach kernels that lets VolumeStar inherit existing 3D-conv and pooling implementations; perturbation tracking in ModelStar that maintains symbolic dependencies between input generators and weight-perturbation generators across linear layers; and a probabilistic reachability mode that composes with sound reachability so that any feature implemented for the sound mode is automatically usable in the probabilistic mode.

**Continuous Integration and Deployment (CI/CD).** To improve reliability and maintainability, NNV3 adopts a continuous integration and deployment pipeline based on GitHub Actions. The pipeline automatically builds and tests the codebase for each commit and pull request, including unit tests for core reachability methods and regression tests on established benchmarks. This ensures backward compatibility and supports sustained development of NNV as an open-source verification tool.

**Extensibility and developer API.** NNV3's engine and analyzer modules are documented as a stable developer-facing API[10] with guidance for adding new layer types, set representations, and verification properties[11]. The unified documentation site now provides the user guide, developer guide, API reference, theory background, and the tutorials index as a single entry point. Together, these resources lower the barrier for contributors extending NNV to architectures and properties beyond those listed in Table 1.

## 4 New Domains

NNV3's new abstractions and probabilistic mode unlock application domains previously out of reach for reachability-based methods, characterized by high-dimensional inputs, structured representations, or deployment-driven threat models that exceed traditional $L_p$-bounded robustness analysis. Below we highlight four such domains.

**Malware Detection [53]:** DNN malware classifiers are vulnerable to adversarial evasion, where attackers apply small functionality-preserving modifications to bypass detection. NNV3 introduces a benchmark for verifying robustness of

[10] https://verivital.github.io/nnv/api/index.html
[11] https://verivital.github.io/nnv/developer/index.html

static-feature classifiers under realistic feature-space perturbations (e.g., modifying non-executable sections, appending benign bytes), enabling formal assessment of classifier resilience under a well-defined threat model.

**Power-System Analysis [75]:** Power flow, optimal power flow, and cascading failure analysis are naturally graph-structured and increasingly use GNN surrogates [77], yet prediction errors carry severe operational consequences. Through GNN verification, NNV3 provides the first general-purpose framework supporting reachability analysis of such topology-aware models with both node- and edge-feature uncertainty.

**Medical Imaging Classification [26, 27]:** High-dimensional semantic segmentation defeats exact verification. NNV3's probabilistic pipeline analyzes large segmentation networks on lung X-ray datasets [11,35] with dense pixel-level outputs, providing coverage guarantees while substantially reducing conservatism relative to exact methods [27].

**Financial Predictions [74]:** FairNNV extends NNV3 to consequential decision making systems such as credit approval and loan risk assessment [7,30,49]. Unlike statistical auditing over finite datasets, reachability-based fairness analysis certifies properties over continuous input regions, formally reasoning about bias under input perturbations and counterfactual scenarios.

## 5 Evaluation

NNV3's primary novelty is breadth: most new features target architectures and threat models without a direct competitor, summarized in Table 8. Per-feature comparisons against the closest-related approaches appear in the underlying papers (GNNV vs. CORA [75], conformal pipeline [27]); we summarize the per-feature results below. We also refresh the NNV 2.0 head-to-head [47] against the MathWorks AI Verification Library (AIVL) [63] on FFNN/CNN models where the two tools overlap. All runs were conducted using the online MATLAB 2025b docker container on a machine with `Intel 24 Core i9-285K` CPU with 64GB RAM, and `NVIDIA RTX 5090` GPU with 32GB VRAM.

**Verification Under Weight and Parameter Perturbations.** We evaluate ModelStar [90, 91] on an MNIST MLP (5 hidden layers: 1024, 512, 256, 256, 256) against Certificated-Robust [81] and Formal-Robust [73] (Fig. 2). Perturbations are $L_\infty$ bounds on each layer's weight range, with magnitudes 0.05%/0.1%/0.2%/0.4% corresponding to 10-/9-/8-/7-bit quantization rounding error. Prior bounds only support uniform row-wide [81] or matrix-wide [73] perturbations; for a fair comparison, we perturb every weight in the chosen layer simultaneously with one independent scalar variable per weight [91]. On 100 MNIST test images, ModelStar consistently matches or exceeds prior bounds: at 0.2% perturbation on `fc_4`, it verifies 69/100 images versus 13 for Certificated-Robust (a 56% improvement). This certifies network robustness under quantization or hardware-induced weight uncertainty, addressing a deployment threat model beyond the reach of input-only verifiers.

**Verification of Graph Neural Networks.** We evaluate GraphStar [75] on AC power flow (PF) for the IEEE-24 bus system, verifying 10 graph instances

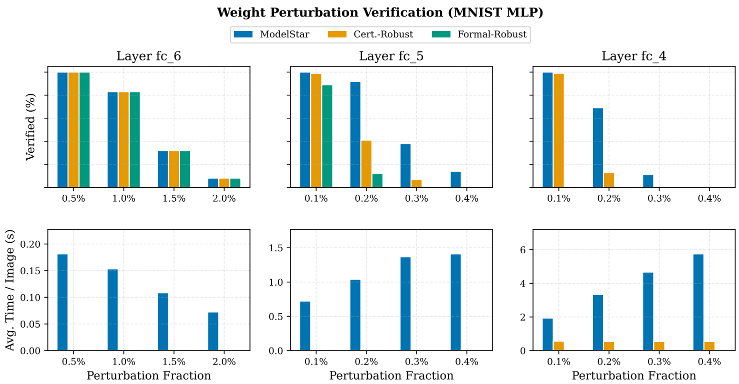


Fig. 2: Single-layer weight-perturbation verification on the MNIST MLP. (Top) fraction of images verified safe; (bottom) average execution time per image, both vs. the $L_\infty$-norm perturbation magnitude.

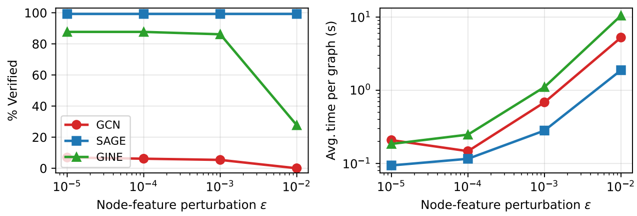


Fig. 3: GNNV verification on IEEE-24 power flow across three architectures (GCN, SAGE, GINE-Conv) vs. node-feature perturbation $\epsilon$ (log axis). (Left) percentage of voltage-magnitude nodes verified. (Right) average verification time per graph instance (log scale).

each for GCN, SAGE, and GINE-Conv models under the ML4ACOPF perturbation scheme [39]: $L_\infty$ perturbations on active and reactive power node features for $\epsilon \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, with voltage magnitude as the safety constraint. Figure 3 reports the percentage of voltage-magnitude nodes verified across the four perturbation levels: SAGE is the most robust (99% across all $\epsilon$), GCN is least robust and degrades fastest, and GINE-Conv falls in between. This delivers reachability-based verification of GNN surrogates, supporting formal verification of GNN-based estimators.

**Verification of Spatio-Temporal and Volumetric Data.** We evaluate VolumeStar [55] on the ZoomIn-4f benchmark, a 4-frame MNIST-video classifier, under $L_\infty$ perturbations $\epsilon \in \{1/255, 2/255, 3/255\}$ with a 30-minute timeout per sample. VolumeStar verifies 7/10 (70%) of cases at every $\epsilon$ tested (Table 2); the remaining three are unknown due to over-approximation. This delivers the first reachability-based robustness certification for 3D-convolutional video classifiers, a modality where ImageStar-based propagation is intractable.

Table 2: VolumeStar verification on ZoomIn-4f (10 samples per $\epsilon$, 30-min timeout).

| $\epsilon$ | Ver. | Unk. | Avg. Time (s) |
|---|---|---|---|
| 1/255 | 7 | 3 | 35.54 |
| 2/255 | 7 | 3 | 37.88 |
| 3/255 | 7 | 3 | 36.49 |

Table 3: Probabilistic verification on TinyYOLO (CP-Star, local Docker build with GPU).

| Property | $\epsilon$ | Time (s) | Result |
|---|---|---|---|
| Prop 101 | 1/255 | 131.86 | UNSAT |
| Prop 277 | 1/255 | 111.92 | UNSAT |
| Prop 356 | 1/255 | 111.26 | UNSAT |

Table 4: FairNNV verification on Adult Census: Verified Fairness (VF, %) and per-sample verification time (s). *Counterfactual fairness* (CF) perturbs only the sensitive attribute ($\epsilon$=0); *individual fairness* (IF) additionally perturbs non-sensitive features at radius $\epsilon$.

| | | CF | IF ($\epsilon$) | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | metric | ($\epsilon$=0) | 0.01 | 0.02 | 0.03 | 0.05 | 0.07 | 0.10 |
| Small | VF (%) | 89 | 87 | 84 | 81 | 69 | 50 | 22 |
| | Time (s) | 0.78 | 0.89 | 1.06 | 1.40 | 2.17 | 3.41 | 5.53 |
| Medium | VF (%) | 87 | 86 | 84 | 82 | 71 | 50 | 27 |
| | Time (s) | 0.72 | 2.64 | 5.76 | 10.10 | 21.75 | 39.73 | 98.70 |

**Probabilistic Verification.** We evaluate NNV3's probabilistic verification pipeline [27] on the TinyYOLO object detector from VNN-COMP 2023 [9]. The approach combines randomized falsification with conformal prediction–based reachability analysis to provide statistical guarantees on property satisfaction. A surrogate model is trained to estimate output distributions and construct confidence bounds, achieving 99.9% coverage. We evaluate three randomly selected VNNLIB property specifications from the benchmark's 71 instances. All three properties are verified as UNSAT (specification holds), with falsification completing in under one second per instance with the use of GPU acceleration in Table 3. This extends NNV to perception-scale models for which exact reachability is intractable, providing statistical coverage guarantees in place of soundness.

**Fairness Verification.** We evaluate FairNNV [74] on the Adult Census dataset [7] using two FFNN classifiers, Small (16–8) and Medium (50) hidden-layer. Under counterfactual fairness, both reach Verified Fairness (VF) $\in [87, 89]\%$ with average verification time below 0.01 s per sample; under individual fairness, VF degrades as $\epsilon$ grows, and the medium model shows both a steeper VF decline and substantially higher verification cost (Table 4). This delivers per-sample fairness certificates over continuous input regions.

Table 5: Tool comparison on fully-connected VNNLIB benchmarks. Per cell: **V** verified, **X** violated, **U** unknown, **T** timeout, **S** mean per-instance time (seconds). Timeout cap: 900 s. AIVL uses `est-bnds`; `r-50` is NNV3's `relax-star-range-50`. Bold marks NNV3's strongest verification count per benchmark.

| | ACAS p3 ($N$=20) | | | | | ACAS p4 ($N$=20) | | | | | RL ($N$=50) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V | X | U | T | S | V | X | U | T | S | V | X | U | T | S |
| AIVL | 0 | 3 | 17 | 0 | 0.06 | 0 | 1 | 19 | 0 | 0.07 | 20 | 11 | 19 | 0 | 0.04 |
| exact | 8 | 3 | 0 | 9 | 58.23 | 9 | 3 | 0 | 8 | 83.99 | 32 | 15 | 1 | 2 | 5.36 |
| approx | **10** | 3 | 7 | 0 | 0.69 | 9 | 3 | 8 | 0 | 0.90 | 32 | 14 | 4 | 0 | 0.14 |
| r-50 | 1 | 2 | 17 | 0 | 0.30 | 1 | 2 | 17 | 0 | 0.30 | 32 | 14 | 4 | 0 | 0.08 |

Table 6: Tool comparison on convolutional VNNLIB benchmarks. Per cell: **V** verified, **X** violated, **U** unknown, **T** timeout, **S** mean per-instance time (seconds). Timeout cap: 900 s. AIVL uses `est-bnds`; NNV3 uses `approx-star`.

| | OVAL21 ($N$=30) | | | | | Collins RUL ($N$=62) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | V | X | U | T | S | V | X | U | T | S |
| AIVL | 0 | 9 | 21 | 0 | 0.28 | 10 | 47 | 5 | 0 | 0.65 |
| approx | 0 | 10 | 14 | 6 | 33.47 | 10 | 47 | 5 | 0 | 0.29 |

Table 7: MNIST-ResNet-8 robustness verification at perturbation magnitude $\varepsilon$ (25 test images per row). Both tools verify all 25 instances per row; the comparison reduces to runtime. AIVL uses `verifyNetworkRobustness` (DeepPoly, residual-network support added in R2024b); NNV3 uses `relax-star-area-50`.

| Tool / $\varepsilon$ | 1/255 | 2/255 | 4/255 | 8/255 |
|---|---|---|---|---|
| AIVL Time (s) | 10.29 | 10.32 | 10.28 | 10.41 |
| NNV3 Time (s) | 6.13 | 8.80 | 13.94 | 34.94 |

**Tool Comparison.** We extend the NNV 2.0 head-to-head [47] against the MathWorks AI Verification Library (AIVL) [63] on R2025b, evaluating its interval-based output bounds for VNNLIB output specifications and its DeepPoly implementation for argmax robustness on residual architectures, and incorporate two new VNN-COMP-derived benchmarks (OVAL21 [5], Collins RUL CNN [41]) alongside a natively-trained MNIST-ResNet-8 (Tables 5, 6, 7). On ACAS Xu (Table 5), AIVL's interval-based bounds are too coarse for the half-space output specifications and yield no verifications; NNV3 validates roughly half the instances under either `exact-star` or `approx-star`, with the latter completing every instance in sub-second time. On RL, NNV3 verifies more instances against AIVL and identifies three additional counterexamples; the `relax-star-range-50`

| | NNV3 | AIVL [63] | $\alpha,\beta$-Crown [79,86] | CORA [2,3,43] | FastBATLLNN [19] | JuliaReach [8,58] | Marabou [38,82,83] | NeuralSAT [16,17] | NeVer2 [15] | mnenum [4] | ReachNN [18,33] | Reluplex [37] | SobolBox [14] | StarV [66] | Verinet [28] | Verisig [34] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **FFNN** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | × |
| **CNN** | ✓ | ✓ | ✓ | ✓ | × | ✓* | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | × |
| **RNN** | ✓ | × | ✓ | × | × | × | × | × | × | × | × | × | × | ✓ | × | × |
| **SSNN** | ✓ | × | ✓* | × | × | × | × | × | × | × | × | × | × | ✓ | × | × |
| ***TDNNs*** | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| ***GNNs*** | ✓ | × | × | ✓ | × | × | ✓* | × | × | × | × | × | × | × | × | × |
| ***3D Data*** | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| ***WPP*** | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| **NODEs** | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | ✓ | × | × |
| **NNCS** | ✓ | × | ✓* | ✓ | × | ✓ | × | × | × | × | ✓ | × | × | ✓ | × | ✓ |

Table 8: Comparison of neural network verification tools by supported architectures and applications, with ✓ meaning supported by a tool and × not supported. We use * to indicate partial support. In *purple italics*, the newly supported architecture/applications by NNV3. WPP signifies model perturbations.

variant probes the precision/cost frontier with sub-second runtime at a reduced verification rate. On the CNN benchmarks (Table 6), `approx-star` matches AIVL on Collins RUL and finds one additional counterexample on OVAL21. On MNIST-ResNet-8 (Table 7), both tools verify all 25 instances at every $\varepsilon$. NNV3's runtime scales with $\varepsilon$ while AIVL's remains essentially constant; at smaller perturbations, NNV3 verifies instances faster than the commercial tool. The architectures and threat models that motivate NNV3's extensions to the Star-set family (Table 8) lie outside AIVL's coverage.

## 6 Related Work

Neural network verification has advanced through SMT-based methods (Reluplex [37], Marabou [38,83], NeuralSAT [17]), bound propagation ($\alpha,\beta$-CROWN [79,86]), abstract interpretation (AI2 [21], DeepPoly [60]), and geometric approaches (Star sets [68], zonotopes [59]), among many other tools [10][12]. These methods excel at verifying robustness of FFNNs and CNNs but typically focus on $L_p$-bounded input perturbations. Table 8 summarizes how NNV3's coverage compares to existing tools across architectures and application domains.

[12] See e.g., https://vnn-comp.github.io/#participants

**Weight and Parameter Perturbations.** Most prior work assumes fixed network weights and verifies robustness only against input perturbations. Weight uncertainty from quantization, compression, or hardware faults has been studied via interval neural network (INN) verification [52], interval bound propagation [81], and verification of quantized networks [29]. ModelStar [90, 91] is complementary: INN intervals can be supplied as ModelStar perturbation specifications and propagated jointly with input uncertainty via star-set reachability.

**VolumeStar.** Formal verification of video classifiers has previously been investigated only in [84], using a game-theoretic framework over 2D convolutional and LSTM layers, with perturbations restricted to optical flow rather than video content. VolumeStar [55] extends ImageStar [65] from 2D images to 3D spatio-temporal data, supporting reachability through 3D convolutional and pooling layers under $L_\infty$ perturbations of the video content itself.

**Graph Neural Network Verification.** Early GNN robustness work focused on empirical or probabilistic guarantees without formal soundness [44, 62, 78]. Recent deterministic approaches include encoding message passing as feed-forward networks for Marabou [82], matrix-polynomial-zonotope GCN reachability in CORA [43], and exact methods based on incremental constraint solving [45] or MILP [31]. These target node-classification on a narrow set of architectures; GraphStar [75] generalizes to edge-aware GNNs with joint node/edge uncertainty.

**Probabilistic Verification.** Probabilistic verification offers a middle ground between exact methods (guarantees but limited scalability) and testing (scalability but no guarantees). Prior approaches include statistical sampling [80], importance sampling for volume estimation [6], randomized smoothing for classification [13] and segmentation [20, 25], and conformal inference for classification [22, 87]. NNV3 integrates conformal inference with set-based reachability, extending coverage guarantees to segmentation [27].

**Fairness Verification.** Fairness verification has been approached via probabilistic methods [1], abstract interpretation [76], and SMT [23]. These techniques differ in the guarantee they provide: probabilistic methods yield statistical bounds, abstract interpretation produces conservative over-approximations, and SMT-based methods give satisfiability-style certificates. FairNNV [74] provides exact fairness certificates via reachability over continuous input regions.

## 7 Conclusion

NNV3 advances neural network verification beyond architectural coverage to address new property classes (fairness), data modalities (video, volumetric), and practical deployment concerns (weight perturbations, probabilistic guarantees). While specialized tools may outperform NNV on specific benchmarks, NNV uniquely supports the broadest range of architectures (GNNs, TDNNs, 3D CNNs), data modalities (images, video, time series), and property classes (robustness, fairness, probabilistic guarantees, weight perturbations).

**Limitations and future work.** NNV3 does not yet handle attention or autoregressive architectures, and ModelStar models continuous weight perturbations only, not fixed-point inference arithmetic. Ongoing work targets (i) a Python frontend to broaden adoption beyond the MATLAB ecosystem [57], (ii) extending the Star-set family to transformers, autoencoders, and temporal graph neural networks, and (iii) discrete fixed-point arithmetic for ModelStar so that quantized inference can be verified end-to-end.

## References

1. Albarghouthi, A., D'Antoni, L., Drews, S., Nori, A.V.: Fairsquare: probabilistic verification of program fairness. Proc. ACM Program. Lang. **1**(OOPSLA) (Oct 2017). https://doi.org/10.1145/3133904
2. Althoff, M.: An introduction to CORA 2015. In: Proc. of the 1st and 2nd Workshop on Applied Verification for Continuous and Hybrid Systems. pp. 120–151. EasyChair (2015). https://doi.org/10.29007/zbkv
3. Althoff, M., Grebenyuk, D., Kochdumper, N.: Implementation of taylor models in cora 2018. In: Frehse, G. (ed.) ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems. EPiC Series in Computing, vol. 54, pp. 145–173. EasyChair (2018). https://doi.org/10.29007/zzc7
4. Bak, S.: nnenum: Verification of relu neural networks with optimized abstraction refinement. In: NASA Formal Methods: 13th International Symposium, NFM 2021, Virtual Event, May 24–28, 2021, Proceedings. p. 19–36. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-76384-8_2
5. Bak, S., Liu, C., Johnson, T.: The second international verification of neural networks competition (vnn-comp 2021): Summary and results (2021), https://arxiv.org/abs/2109.00498
6. Baluta, T., Shen, S., Shinde, S., Meel, K.S., Saxena, P.: Quantitative verification of neural networks and its security applications. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. p. 1249–1264. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3319535.3354245
7. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996), https://doi.org/10.24432/C5XW20
8. Bogomolov, S., Forets, M., Frehse, G., Potomkin, K., Schilling, C.: Juliareach: a toolbox for set-based reachability. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. pp. 39–44 (2019)
9. Brix, C., Bak, S., Liu, C., Johnson, T.T.: The fourth international verification of neural networks competition (vnn-comp 2023): Summary and results (2023), https://arxiv.org/abs/2312.16760
10. Brix, C., Müller, M.N., Bak, S., Johnson, T.T., Liu, C.: First three years of the international verification of neural networks competition (VNN-COMP). International Journal on Software Tools for Technology Transfer **25**(3), 329–339 (Jun 2023). https://doi.org/10.1007/s10009-023-00703-4
11. Candemir, S., Jaeger, S., Palaniappan, K., Musco, J.P., Singh, R.K., Xue, Z., Karargyris, A., Antani, S., Thoma, G., McDonald, C.J.: Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Transactions on Medical Imaging **33**(2), 577–590 (2014). https://doi.org/10.1109/TMI.2013.2290491
12. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4733 (2017). https://doi.org/10.1109/CVPR.2017.502

13. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning (ICML). pp. 1310–1320 (2019)
14. Das, S.: SobolBox: Boxed Refinement of Sobol Sequence Samples for Neural Network Verification (Competition Contribution). In: International Symposium on AI Verification. pp. 272–277. Springer (2025)
15. Demarchi, S., Guidotti, D., Pulina, L., Tacchella, A.: NeVer2: learning and verification of neural networks. Soft Computing **28**(19), 11647–11665 (Oct 2024). https://doi.org/10.1007/s00500-024-09907-5
16. Duong, H., Nguyen, T., Dwyer, M.: A dpll(t) framework for verifying deep neural networks (2024), https://arxiv.org/abs/2307.10266
17. Duong, H., Nguyen, T., Dwyer, M.B.: Neuralsat: a high-performance verification tool for deep neural networks. In: International Conference on Computer Aided Verification. pp. 409–423. Springer (2025). https://doi.org/10.1007/978-3-031-98679-6_19
18. Fan, J., Huang, C., Chen, X., Li, W., Zhu, Q.: ReachNN*: A Tool for Reachability Analysis of Neural-Network Controlled Systems. In: Hung, D.V., Sokolsky, O. (eds.) Automated Technology for Verification and Analysis. pp. 537–542. Springer International Publishing, Cham (2020)
19. Ferlez, J., Khedr, H., Shoukry, Y.: Fast batllnn: Fast box analysis of two-level lattice neural networks. In: Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control. HSCC '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3501710.3519533
20. Fischer, M., Baader, M., Vechev, M.: Scalable certified segmentation via randomized smoothing. In: International Conference on Machine Learning (ICML). pp. 3340–3351 (2021)
21. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE (2018)
22. Gendler, A., Weng, T.W., Daniel, L., Romano, Y.: Adversarially robust conformal prediction. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=9L1BsI4wP1H
23. Ghosh, B., Basu, D., Meel, K.S.: Justicia: A Stochastic SAT Approach to Formally Verify Fairness. vol. 35, pp. 7554–7563 (May 2021). https://doi.org/10.1609/aaai.v35i9.16925
24. Guo, Y.: A survey on methods and theories of quantized neural networks (2018), https://arxiv.org/abs/1808.04752
25. Hao, Z., Ying, C., Su, H., Zhu, J., Song, J., Hu, Z.: GSmooth: Certified robustness against semantic transformations via generalized randomized smoothing. In: International Conference on Machine Learning (ICML). pp. 8465–8483 (2022)
26. Hashemi, N., Sasaki, S., Lopez, D.M., Lindemann, L., Oguz, I., Ma, M., Johnson, T.T.: Probabilistic robustness analysis in high dimensional space: Application to semantic segmentation network (2025), https://arxiv.org/abs/2509.11838
27. Hashemi, N., Sasaki, S., Oguz, I., Ma, M., Johnson, T.T.: Scaling data-driven probabilistic robustness analysis for semantic segmentation neural networks. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=liefJOFVfH
28. Henriksen, P., Lomuscio, A.: Efficient neural network verification via adaptive refinement and adversarial search. In: European Conference on Artificial Intelligence (ECAI). pp. 2513–2520 (2020)

29. Henzinger, T.A., Lechner, M., Zikelic, D.: Scalable verification of quantized neural networks. In: AAAI Conference on Artificial Intelligence. pp. 3787–3795 (2021)
30. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994), https://doi.org/10.24432/C5NC77
31. Hojny, C., Zhang, S., Campos, J.S., Misener, R.: Verifying message-passing neural networks via topology-based bounds tightening. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, JMLR.org (2024)
32. Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=HJlWWJSFDH
33. Huang, C., Fan, J., Li, W., Chen, X., Zhu, Q.: Reachnn: Reachability analysis of neural-network controlled systems. ACM Trans. Embed. Comput. Syst. **18**(5s) (Oct 2019). https://doi.org/10.1145/3358228
34. Ivanov, R., Weimer, J., Alur, R., Pappas, G.J., Lee, I.: Verisig: verifying safety properties of hybrid systems with neural network controllers. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. p. 169–178. HSCC '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3302504.3311806
35. Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., Xue, Z., Palaniappan, K., Singh, R.K., Antani, S., Thoma, G., Wang, Y.X., Lu, P.X., McDonald, C.J.: Automatic tuberculosis screening using chest radiographs. IEEE Transactions on Medical Imaging **33**(2), 233–245 (2014). https://doi.org/10.1109/TMI.2013.2284099
36. Johnson, T.T., Lopez, D.M., Tran, H.D.: Tutorial: Safe, secure, and trustworthy artificial intelligence (ai) via formal verification of neural networks and autonomous cyber-physical systems (cps) with nnv. In: 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S). pp. 65–66 (2024). https://doi.org/10.1109/DSN-S60304.2024.00027
37. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: Majumdar, R., Kunčak, V. (eds.) Computer Aided Verification. pp. 97–117. Springer International Publishing, Cham (2017)
38. Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C.: The marabou framework for verification and analysis of deep neural networks. In: Dillig, I., Tasiran, S. (eds.) Computer Aided Verification. pp. 443–452. Springer International Publishing, Cham (2019)
39. Kaulen, K., Ladner, T., Bak, S., Brix, C., Duong, H., Flinkow, T., Johnson, T.T., Koller, L., Manino, E., Nguyen, T.H., Wu, H.: The 6th international verification of neural networks competition (vnn-comp 2025): Summary and results (2025), https://arxiv.org/abs/2512.19007
40. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=SJU4ayYgl
41. Kirov, D., Rollini, S.F.: Benchmark: Remaining Useful Life Predictor for Aircraft Equipment. In: Steffen, B. (ed.) Bridging the Gap Between AI and Reality. pp. 299–304. Springer Nature Switzerland, Cham (2024)
42. Kusner, M., Loftus, J., Russell, C., Silva, R.: Counterfactual fairness. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4069–4079. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)

43. Ladner, T., Eichelbeck, M., Althoff, M.: Formal verification of graph convolutional networks with uncertain node features and uncertain graph structure. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=B6y12Ot0cP
44. Lai, Y., Zhou, J., Zhang, X., Zhou, K.: Toward certified robustness of graph neural networks in adversarial aiot environments. IEEE Internet of Things Journal **10**(15), 13920–13932 (2023). https://doi.org/10.1109/JIOT.2023.3263384
45. Liu, M., Lu, C.H., Kwiatkowska, M.: Exact verification of graph neural networks with incremental constraint solving. In: Sampaio, A., Stoelinga, M. (eds.) Formal Methods. pp. 641–662. Springer Nature Switzerland, Cham (2026)
46. Lopez, D.M., Althoff, M., Benet, L., Coogan, S., Forets, M., Harapanahalli, A., Johnson, T.T., Ladner, T., Schilling, C., Zhang, H., Zhong, X.: Arch-comp25 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants. In: Frehse, G., Althoff, M. (eds.) Proceedings of 12th Int. Workshop on Applied Verification for Continuous and Hybrid Systems. EPiC Series in Computing, vol. 108, pp. 71–121. EasyChair (2025). https://doi.org/10.29007/9vg6, /publications/paper/Gc39
47. Lopez, D.M., Choi, S.W., Tran, H.D., Johnson, T.T.: Nnv 2.0: The neural network verification tool. In: Enea, C., Lal, A. (eds.) Computer Aided Verification. pp. 397–412. Springer Nature Switzerland, Cham (2023)
48. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6) (Jul 2021). https://doi.org/10.1145/3457607
49. Moro, S., Rita, P., Cortez, P.: Bank Marketing. UCI Machine Learning Repository (2014). https://doi.org/https://doi.org/10.24432/C5K306
50. Open Neural Network Exchange (ONNX): https://github.com/onnx/onnx
51. Pal, N., Lopez, D.M., Johnson, T.T.: Robustness verification of deep neural networks using star-based reachability analysis with variable-length time series input. In: Cimatti, A., Titolo, L. (eds.) Formal Methods for Industrial Critical Systems. pp. 170–188. Springer Nature Switzerland, Cham (2023)
52. Prabhakar, P., Afzal, Z.R.: Abstraction based output range analysis for neural networks. Curran Associates Inc., Red Hook, NY, USA (2019)
53. Robinette, P.K., Manzanas Lopez, D., Serbinowska, S., Leach, K., Johnson, T.T.: Case study: Neural network malware detection verification for feature and image datasets. In: Proceedings of the 2024 IEEE/ACM 12th International Conference on Formal Methods in Software Engineering (FormaliSE). p. 127–137. FormaliSE '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3644033.3644372
54. Roy, A., Antony, A., Gimelli, A., Daggitt, M.L.: Vnn-lib 2.0: Rigorous foundations for neural network verification (2026), https://arxiv.org/abs/2605.07451
55. Sasaki, S., Lopez, D.M., Robinette, P.K., Johnson, T.T.: Robustness verification of video classification neural networks. In: 2025 IEEE/ACM 13th International Conference on Formal Methods in Software Engineering (FormaliSE). pp. 22–33 (2025). https://doi.org/10.1109/FormaliSE66629.2025.00009
56. Sasaki, S., Wooding, B., Johnson, T.T., Althoff, M., Benet, L., Coogan, S., Forets, M., Harapanahalli, A., Koller, L., Ladner, T., et al.: Arch-comp26 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants. In: Proceedings of 13th Int. Workshop on Applied Verification for Continuous and Hybrid Systems. vol. 110, pp. 85–130 (2026)

57. Sasaki, S., Wooding, B., Wang, H.D., Tumlin, A.M., Ma, M., Johnson, T.T.: n2v: Neural network verification in python (competition contribution). In: International Symposium on AI Verification. pp. 394–400. Springer (2026)
58. Schilling, C., Forets, M., Guadalupe, S.: Verification of Neural-Network Control Systems by Integrating Taylor Models and Zonotopes. In: AAAI. pp. 8169–8177. AAAI Press (2022). https://doi.org/10.1609/aaai.v36i7.20790
59. Singh, G., Gehr, T., Mirman, M., Püschel, M., Vechev, M.: Fast and effective robustness certification. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 10825–10836. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
60. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang. **3**(POPL) (Jan 2019). https://doi.org/10.1145/3290354
61. Sun, X., Zhang, Z., Ren, X., Luo, R., Li, L.: Exploring the vulnerability of deep neural networks: A study of parameter corruption. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11648–11656 (2021)
62. Tao, S., Shen, H., Cao, Q., Hou, L., Cheng, X.: Adversarial immunization for certifiable robustness on graphs. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. p. 698–706. WSDM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3437963.3441782
63. The MathWorks, Inc.: AI Verification Library. Natick, Massachusetts, United States (2025), `https://www.mathworks.com/products/ai-verification-library.html`
64. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015). https://doi.org/10.1109/ICCV.2015.510
65. Tran, H.D., Bak, S., Xiang, W., Johnson, T.T.: Verification of deep convolutional neural networks using imagestars. In: Lahiri, S.K., Wang, C. (eds.) Computer Aided Verification. pp. 18–42. Springer International Publishing, Cham (2020)
66. Tran, H.D., Choi, S.W., Li, Y., Liu, Q., Okamoto, H., Hoxha, B., Fainekos, G.: Starv: A qualitative and quantitative verification tool for learning-enabled systems. In: Computer Aided Verification: 37th International Conference, CAV 2025, Zagreb, Croatia, July 23-25, 2025, Proceedings, Part II. p. 376–394. Springer-Verlag, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-3-031-98679-6_17
67. Tran, H.D., Lopez, D.M., Yang, X., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., Johnson, T.T.: Demo: The neural network verification (nnv) tool. In: 2020 IEEE Workshop on Design Automation for CPS and IoT (DESTION). pp. 21–22 (2020). https://doi.org/10.1109/DESTION50928.2020.00010
68. Tran, H.D., Manzanas Lopez, D., Musau, P., Yang, X., Nguyen, L.V., Xiang, W., Johnson, T.T.: Star-based reachability analysis of deep neural networks. In: Formal Methods – The Next 30 Years: Third World Congress, FM 2019, Porto, Portugal, October 7–11, 2019, Proceedings. p. 670–686. Springer-Verlag, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-30942-8_39
69. Tran, H.D., Manzanas Lopez, D., Johnson, T.: Tutorial: Neural network and autonomous cyber-physical systems formal verification for trustworthy ai and safe autonomy. In: Proceedings of the International Conference on Embedded Software. p. 1–2. EMSOFT '23, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3607890.3608454

70. Tran, H.D., Musau, P., Lopez, D.M., Yang, X., Nguyen, L.V., Xiang, W., Johnson, T.T.: Star-based reachability analsysis for deep neural networks. In: 23rd International Symposisum on Formal Methods (FM'19). Springer International Publishing (October 2019)
71. Tran, H.D., Pal, N., Musau, P., Lopez, D.M., Hamilton, N., Yang, X., Bak, S., Johnson, T.T.: Robustness verification of semantic segmentation neural networks using relaxed reachability. In: Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20–23, 2021, Proceedings, Part I. p. 263–286. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-81685-8_12
72. Tran, H.D., Yang, X., Lopez, D.M., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., Johnson, T.T.: NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In: 32nd International Conference on Computer-Aided Verification (CAV) (July 2020)
73. Tsai, Y.L., Hsu, C.Y., Yu, C.M., Chen, P.Y.: Formalizing generalization and adversarial robustness of neural networks to weight perturbations. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21, Curran Associates Inc., Red Hook, NY, USA (2021)
74. Tumlin, A.M., Manzanas Lopez, D., Robinette, P., Zhao, Y., Derr, T., Johnson, T.T.: Fairnnv: The neural network verification tool for certifying fairness. p. 36–44. ICAIF '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3677052.3698677
75. Tumlin, A.M., Wooding, B., Shao, Z., Lopez, D.M., Derr, T., Johnson, T.T.: Reachability-based formal verification of graph neural networks with node and edge features. In: International Symposium on AI Verification. pp. 271–298. Springer (2026)
76. Urban, C., Christakis, M., Wüstholz, V., Zhang, F.: Perfectly parallel fairness certification of neural networks. Proc. ACM Program. Lang. **4**(OOPSLA) (Nov 2020). https://doi.org/10.1145/3428253
77. Varbella, A., Amara, K., Gjorgiev, B., El-Assady, M., Sansavini, G.: Powergraph: a power grid benchmark dataset for graph neural networks. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. NIPS '24, Curran Associates Inc., Red Hook, NY, USA (2024)
78. Wang, B., Jia, J., Cao, X., Gong, N.Z.: Certified robustness of graph neural networks against adversarial structural perturbation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 1645–1653. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467295
79. Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.J., Kolter, J.Z.: Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), `https://openreview.net/forum?id=ahYIlRBeCFw`
80. Webb, S., Rainforth, T., Teh, Y.W., Kumar, M.P.: Statistical verification of neural networks. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=S1xcx3C5FX`
81. Weng, T.W., Zhao, P., Liu, S., Chen, P.Y., Lin, X., Daniel, L.: Towards certificated model robustness against weight perturbations. Proceedings of the AAAI Conference on Artificial Intelligence **34**(04), 6356–6363 (Apr 2020). https://doi.org/10.1609/aaai.v34i04.6105

82. Wu, H., Barrett, C., Sharif, M., Narodytska, N., Singh, G.: Scalable verification of gnn-based job schedulers. Proc. ACM Program. Lang. **6**(OOPSLA2) (Oct 2022). https://doi.org/10.1145/3563325
83. Wu, H., Isac, O., Zeljić, A., Tagomori, T., Daggitt, M., Kokke, W., Refaeli, I., Amir, G., Julian, K., Bassan, S., et al.: Marabou 2.0: A Versatile Formal Analyzer of Neural Networks. In: Computer Aided Verification: 36th International Conference, CAV 2024. Springer (2024)
84. Wu, M., Kwiatkowska, M.: Robustness guarantees for deep neural networks on videos. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 308–317 (2020). https://doi.org/10.1109/CVPR42600.2020.00039
85. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386
86. Xu, K., Zhang, H., Wang, S., Wang, Y., Jana, S., Lin, X., Hsieh, C.J.: Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=nVZtXBI6LNn`
87. Yan, G., Romano, Y., Weng, T.W.: Provably robust conformal prediction with improved efficiency. In: The Twelfth International Conference on Learning Representations (2024), `https://openreview.net/forum?id=BWAhEjXjeG`
88. Yan, Z., Hu, X.S., Shi, Y.: Computing-in-memory neural network accelerators for safety-critical systems: Can small device variations be disastrous? ICCAD '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3508352.3549360
89. Yousef, R., Gupta, G., Yousef, N., Khari, M.: A holistic overview of deep learning approach in medical imaging. Multimedia Syst. **28**(3), 881–914 (Jun 2022). https://doi.org/10.1007/s00530-021-00884-5
90. Zubair, M.U., Johnson, T.T., Basu, K., Abbas, W.: Verification of neural network robustness against weight perturbations using star sets. In: 2025 IEEE Conference on Artificial Intelligence (CAI). pp. 637–642 (2025). https://doi.org/10.1109/CAI64502.2025.00117
91. Zubair, M.U., Johnson, T.T., Basu, K., Abbas, W.: Modelstar: Reachability analysis-based safety verification of neural networks against model perturbations. J. Artif. Int. Res. **85** (Apr 2026). https://doi.org/10.1613/jair.1.18922