Benchmark: Formal Verification for Surgical Robots

Jamie Fong[1], Ben Wooding[1], and Taylor T. Johnson[1]

Vanderbilt University, Nashville TN 37235, USA

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

1.1 Motivation

Traditionally, surgical skill assessment has been done with manual feedback from expert surgeons. However, in recent years, this has moved to using machine learning (ML) models to score surgeons [6, 2, 23]. The rise in these ML models raises an important concern about accuracy, especially if these methods will be used in training and credentialing. Networks without formal verification are trusted based on the accuracy from held-out test data alone. This provides no formal guarantees. Input perturbations such as sensor noise, segmentation jitter, and small kinematic variation should not alter a skill judgment [15, 4].

1.2 The Gap

Neural network (NN) verification and benchmarks are usually applied to image classification such as MNIST and CIFAR-10, with ACAS-Xu using sensor input—which more closely aligns with our work and dataset [9, 11, 3, 14]. However, there has not been as much work done on formal verification applied to surgical skill assessment. Additionally, some formal verification tools that do exist, such as n2v (a Python implementation of NNV), do not directly support our models [16, 19, 12]. Because n2v assumes either flat 1D vectors or 2D spatial tensors, it does not support the 3D ($batch \times channels \times time$) shape of our network input. Thus, we needed to create a mathematically equivalent, flat reformulation of the network. This workaround is representative of the fact that typical benchmarks are not always representative of real-world models [10].

1.3 Contributions

1. We built a reusable verification benchmark over the JIGSAWS dataset on three tasks: suturing, knot tying, and needle passing [6]. The fully convolutional network (FCN) was built according to the specifications in [8]. with

VNN-LIB [5] specifications compatible with n2v [16] and α,β–CROWN [22, 20, 21, 24].
2. We implemented a flat reformulation method compatible with n2v. We used a block-diagonal weight matrix to represent the prior per-group weight matrices, which yields a mathematically equivalent network with a max difference of 10^{-7}. This allows Star-set verification without having to retrain a network.
3. We used two verifiers with two different methodologies on identical ONNX models and VNN-LIB specifications. We found that there is full agreement up to $T \leq 60$ between both verifiers, with n2v having a certification ceiling at $T \geq 80$.

2 Background

2.1 Surgical Skill Assessment & JIGSAWS

Our work uses the JIGSAWS dataset [6], the OSATS rubric (to judge surgeon overall performance) [13, 7], and the fully convolutional network (FCN) described in [8]. Benmansour et al. [2] also use OSATS regression to judge surgical skills but with a CNN followed by a BiLSTM architecture, not an FCN. However, n2v—a Star-based verification tool—does not support LSTMs.

2.2 Neural Network Verification

NN verification typically tests for robustness—which means that small perturbations to input should not change the NN's output [1]. This paper tests formal robustness properties of our network. We define robustness over an ϵ-ball on a held-out anchor input. VNN-COMP and VNN-LIB's convention is to assert the negation and return `UNSAT` if the safety property is proven "correct," `SAT` if a counterexample was found, and `unknown` if the safety property cannot be verified and a counterexample cannot be found [3].

The two tools used in this paper are n2v and α,β–CROWN. These two represent two different underlying methods of verifying NNs. n2v is the Python implementation of NNV, which uses Star-set reachability. n2v has two options: exact-star, which is sound and complete, and approx-star, which is sound but not complete. α,β–CROWN uses linear bound propagation with branch-and-bound. α,β–CROWN is, in theory, sound and complete. In practice, however, verification will time out before exploring the whole space.

3 Benchmark Design

3.1 Dataset & Pre-Processing

The JIGSAWS dataset has three surgical tasks, each with a different number of trials: suturing has 39 trials, knot tying has 36, and needle passing has 28.

Table 1. SurgicalFCN architecture. Input shape $(76, T)$.

Stage	Layers	Output Shape	Params
1 — sub-cluster convs	20×Conv1d, kernel 3, pad 1	$(160, T)$	1,984
2 — manipulator convs	4×Conv1d(40→16, k3, pad1)	$(64, T)$	7,744
3 — shared conv	Conv1d(64→32, k3, pad1)	$(32, T)$	6,176
Pool	Temporal mean (ReduceMean)	$(32,)$	0
Output	Linear(32→6)	$(6,)$	198
Total			16,102

Kinematic data from the surgical robot has 19 sensors across 4 manipulators. Thus, there are 76 channels collected from the da Vinci robot.

Keeping with the evaluation protocol described in [8], we perform Leave One Super-Trial Out (LOSO) cross-validation. One subject is held out per fold. There are some trials from the meta file that are missing OSATS labels. These were excluded from training. However, these can still be used as verification anchors as ground-truth labels are not needed. Verification properties are defined over the first `T` consecutive timesteps of each held-out trial.

3.2 Model

The FCN was faithfully reproduced according to [8], without any modification for verifiability. This FCN has three layers. The first layer has one Conv1d per sub-cluster. A sub-cluster separates the different sensor input data of the robot (e.g., the first sub-cluster is Cartesian coordinates, the second is linear velocity). The second layer has one Conv1d per manipulator (4 manipulators). The third layer is a shared convolution across all manipulators. We then apply global average pooling over time. The output layer is a Linear layer that produces the 6 OSATS scores. See Table 1.

Each model is trained per LOSO fold using MSE loss. Afterward, the performance of the models is evaluated using Spearman ρ, with respect to ground-truth OSATS scores. The Spearman ρ on Overall Performance are: Suturing $\rho = 0.603$, Knot Tying $\rho = 0.704$, Needle Passing $\rho = 0.364$.

3.3 Properties

Properties are defined over an ℓ_∞ ϵ-ball around a held-out anchor input. Properties are encoded using VNN-LIB [5]. Four properties were tested. (1) noise robustness: we checked that all inputs within $\epsilon_{physical}$ of the anchor produce an output within δ of the anchor's original predicted score. $\epsilon_{physical} = 0.001$ per the estimated da Vinci kinematic precision [15], and $\delta = 0.5$, which is half an OSATS point. (2) segmentation boundary invariance: we apply $\epsilon_{boundary} = 0.01$ to the first and last two timesteps, $\epsilon_{physical}$ to the other timesteps, and check if output is within δ. (3) output range: we verify that the overall performance score for an expert-class anchor input exceeds a floor of $L = min(\text{expert score}) - 0.25$.

Table 2. SurgicalFCNFlat architecture at $T = 10$. Dense params assume fully populated weights; nonzero weights reflect the block-diagonal transfer.

Layer	Shape	Dense Params	Nonzero Weights
conv1	Conv1d(76→160, k3, pad1)	36,640	(block-diagonal)
conv2	Conv1d(160→64, k3, pad1)	30,784	(block-diagonal)
conv3	Conv1d(64→32, k3, pad1)	6,176	6,176
pool_linear	Linear($32{\cdot}T$→32, no bias)	$1{,}024{\cdot}T$	$32{\cdot}T$
classifier	Linear(32→6)	198	198
Total ($T = 10$)		84,038	16,160

(4) monotonicity: given a novice anchor, we verify that the predicted score is below the floor from (3), L, under perturbation.

4 The Flat-Reformulation Method

4.1 The Problem

One issue with n2v when running the raw verification pipeline is that n2v does not properly perform operations such as `Slice` and `ReduceMean` on our 3D input tensors. Our model is trained by processing channels in separate groups so irrelevant sensor data does not influence each convolution. This is done through channel slicing. n2v takes our 3D input tensor of $(1 \times 76 \times T)$ and strips the batch dimension (always 1). n2v then flattens to a flat 1D star of length $76 \cdot T$. Thus, when we slice for three channels, n2v interprets that as slicing the first three values of the flat 1D vector. In other words, instead of slicing as `V[0:3T, :]`, n2v performs `V[0:3, :]`.

A subsequent problem is that `ReduceMean` will fail as we want to average across the time axis while keeping the channels axis intact. However, because of n2v's flattening, we lose the `(channels × time)` structure. n2v only supports reduction of a vector to a scalar, and thus will be unable to reduce only our time axis.

4.2 The Reformulation

Because n2v does not properly handle `Slice` and `ReduceMean`, we reformulate our model using only `Conv`, `ReLU`, `Flatten`, and `Gemm`. We re-express our 20 sub-cluster convs into one dense `Conv1d`. We place weights block-diagonally. The same block-diagonal technique is applied to the second layer with our grouped manipulator convs. The third layer—which is already dense—remains unchanged. The global average pooling is performed by a bias-free `Linear` layer, with block-diagonal weights set to $\frac{1}{T}$. The classifier remains the same. See Table 2.

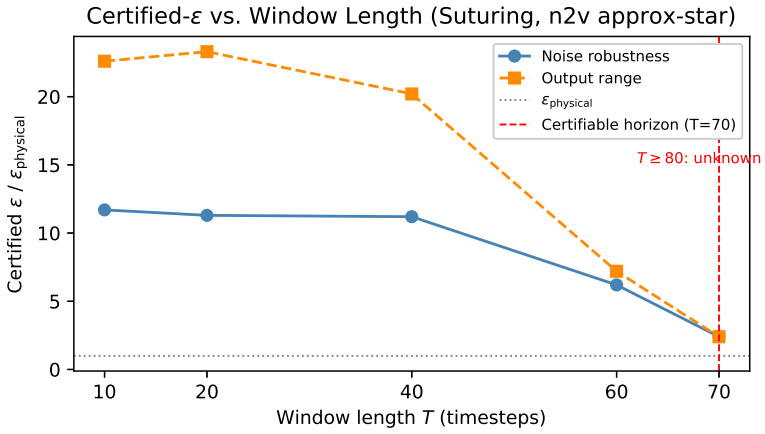

Fig. 1. Certified ϵ over $\epsilon_{physical}$ vs. window length T for Suturing using n2v approx-mode. The decay starting from $T = 40$ reflects n2v's reduced ability to verify larger timesteps due to greater over-approximation. This collapses at $T \geq 80$.

4.3 Equivalence

Using a random input and passing it through both our flattened model and the original model, the maximum difference produced was 10^{-7}. This is consistent with floating-point rounding error. Overall, this flat-reformulation allows us to re-express our network rather than re-train it.

5 Results

5.1 Certified-ϵ

We performed a binary search over an ϵ interval of $[0, 0.1]$ with 12 iterations to find the largest certified-ϵ that still returns `UNSAT`. When running n2v with timestep $T = 10$, the average across five folds yielded 0.011733 for noise robustness and 0.022588 for output range floor (when running in exact mode). This is 11.7× and 22.6× $\epsilon_{physical}$, respectively. However, it is important to note that this value does vary from fold to fold. Fold 3 is only 8× $\epsilon_{physical}$, whereas fold 4 is 45× $\epsilon_{physical}$ (both for output range property).

The certified-ϵ continues to remain above $\epsilon_{physical}$ through $T = 70$, where we get 2.4× for both properties (we run in approx-mode $T > 10$). The certified-ϵ begins to decay from $T = 40$ onward until, at $T = 80$, n2v is no longer able to verify—returning `unknown` when over-approximating. See Fig. 1.

5.2 Cross-Verifier Comparison

We cross-verified our fixed-ϵ verdicts with α,β–CROWN using the same model and specifications. Our results matched n2v exactly up to $T = 60$—n2v began

returning `unknown` at $T = 70$. From our testing, CROWN was able to scale up to $T = 500$ and return `UNSAT` for the noise, segmentation, and range properties. In contrast, n2v's approx-mode begins returning `unknown` verdicts starting at $T = 70$, with $T = 80$ returning all unknowns except for one property on one fold. This is consistent with n2v's approx-star method, where larger input sets cause a larger over-approximation, which causes output bounds to bloat rapidly [18].

Unlike with n2v, we did not perform a binary search to find a certified-ϵ with α,β–CROWN. Because α,β–CROWN uses branch-and-bound to guarantee a correct answer, this can be inefficient if there are a large number of unstable neurons. When looking at the boundary line between robust and non-robust—which is exactly what our binary search will do—this will cause it to constantly split neurons. n2v does not have this problem as it does not branch. It does one pass with a single over-approximate star set.

5.3 Monotonicity

All verdicts in our tests up to $T = 500$ returned `UNSAT` except for monotonicity starting from $T = 100$ for suturing. This implies that our model incorrectly learned the proper ordering of novice and expert surgeons. This reflects the limitation of the JIGSAWS dataset, which only includes two expert surgeon samples. However, it is important to note that this is a nominal failure, not a robustness failure. The predicted overall performance score of the novice was already higher than the performance score of the expert.

6 Discussion

Ultimately, no single tool sufficed for our verification purposes. α,β–CROWN confirmed soundness at a much higher T, pushing far past n2v's horizon. However, α,β–CROWN was unable to efficiently find a certified-ϵ. This benchmark required both tools to properly verify all properties. Additionally, α,β–CROWN was able to verify our original model directly—without having to re-express it through our flat-reformulation method.

Our flat-reformulation model allowed us to re-express the original model without having to re-train it. This was important as it allowed us to retain the original model's learned weights, while having a model that was accurately compatible with n2v. This permitted us to be faithful to the original model described in [8].

The monotonicity result is the most interesting, as it showed how a model can be technically robust but still produce an incorrect result. A network may be robust on some perturbed input—the output being within δ—but still have an incorrect output. Formal verification helped reveal this, whereas regular testing with just Spearman $\rho = 0.6$ might imply that it is fairly monotonic or noticing that we receive the same output despite perturbing input by an ϵ. This is another

case of formal verification discovering something that regular testing may not have discovered [17].

Our benchmark is limited in that the average timestep of a trial is ~ 4000 timesteps. n2v was unable to fully verify at $T = 60$. α,β–CROWN also failed to fully verify a single trial, not because of time constraints but because it ran out of GPU memory.[1] The bound propagation grows quadratically with T. Though theoretically possible to verify, α,β–CROWN also struggles to scale well, though not in the same way as n2v. Future work could include verifying the entire trial length—which is currently beyond any tool we have tested.

7 Conclusion

References

1. Albarghouthi, A.: Introduction to neural network verification (2021), https://arxiv.org/abs/2109.10317
2. Benmansour, M., Malti, A., Jannin, P.: Deep neural network architecture for automated soft surgical skills evaluation using objective structured assessment of technical skills criteria. International Journal of Computer Assisted Radiology and Surgery **18**, 929–937 (2023). https://doi.org/10.1007/s11548-022-02827-5
3. Brix, C., Bak, S., Johnson, T.T., Wu, H.: The fifth international verification of neural networks competition (vnn-comp 2024): Summary and results (2024), https://arxiv.org/abs/2412.19985
4. Cui, Z., Cartucho, J., Giannarou, S., y Baena, F.R.: Caveats on the first-generation da vinci research kit: Latent technical constraints and essential calibrations [survey]. IEEE Robotics & Automation Magazine **32**(2), 113–128 (Jun 2025). https://doi.org/10.1109/mra.2023.3310863, http://dx.doi.org/10.1109/MRA.2023.3310863
5. Demarchi, S., Guidotti, D., Pulina, L., Tacchella, A.: Supporting standardization of neural networks verification with vnnlib and coconet. In: Narodytska, N., Amir, G., Katz, G., Isac, O. (eds.) Proceedings of the 6th Workshop on Formal Methods for ML-Enabled Autonomous Systems. Kalpa Publications in Computing, vol. 16, pp. 47–58. EasyChair (2023). https://doi.org/10.29007/5pdh, /publications/paper/Qgdn
6. Gao, Y., Vedula, S.S., Reiley, C.E., Ahmidi, N., Varadarajan, B., Lin, H.C., Tao, L., Zappella, L., Béjar, B., Yuh, D.D., Chen, C.C.G., Vidal, R., Khudanpur, S., Hager, G.: Jhu-isi gesture and skill assessment working set (jigsaws) : A surgical activity dataset for human motion modeling (2014), https://api.semanticscholar.org/CorpusID:16185857
7. Hatala, R., Cook, D., Brydges, R., Hawkins, R.: Constructing a validity argument for the Objective Structured Assessment of Technical Skills (OSATS): a systematic review of validity evidence. Advances in Health Sciences Education **20**(5), 1149–1175 (2015). https://doi.org/10.1007/s10459-015-9593-1

[1] Experiments were run on an NVIDIA GeForce RTX 4050 Laptop GPU (6 GB dedicated VRAM), 16 GB system RAM, and an Intel Core i7-13700H.

8. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Accurate and interpretable evaluation of surgical skills from kinematic data using fully convolutional neural networks. International Journal of Computer Assisted Radiology and Surgery **14**(9), 1611–1617 (2019). https://doi.org/10.1007/s11548-019-02039-4
9. Katz, G., Barrett, C., Dill, D., Julian, K., Kochenderfer, M.: Reluplex: An efficient smt solver for verifying deep neural networks (2017), https://arxiv.org/abs/1702.01135
10. Kessler, C., Komendantskaya, E., Casadio, M., Viola, I.M., Flinkow, T., Othman, A.A., Malhotra, A., McPherson, R.: Neural network verification for gliding drone control: A case study (2025), https://arxiv.org/abs/2505.00622
11. Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M.J.: Algorithms for verifying deep neural networks (2020), https://arxiv.org/abs/1903.06758
12. Lopez, D.M., Choi, S.W., Tran, H.D., Johnson, T.T.: Nnv 2.0: The neural network verification tool. In: Computer Aided Verification: 35th International Conference, CAV 2023, Paris, France, July 17–22, 2023, Proceedings, Part II. pp. 397–412. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-37703-7_19, https://doi.org/10.1007/978-3-031-37703-7_19
13. Martin, J., Regehr, G., Reznick, R., MacRae, H., Murnaghan, J., Hutchison, C., Brown, M.: Objective structured assessment of technical skill (OSATS) for surgical residents. British Journal of Surgery **84**(2), 273–278 (1997). https://doi.org/10.1046/j.1365-2168.1997.02502.x
14. Pal, N., Lee, S., Johnson, T.T.: Benchmark: Formal verification of semantic segmentation neural networks. In: Bridging the Gap Between AI and Reality: First International Conference, AISoLA 2023, Crete, Greece, October 23–28, 2023, Proceedings. p. 311–330. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-46002-9_20
15. Roberti, A., Piccinelli, N., Meli, D., Muradore, R., Fiorini, P.: Improving rigid 3-d calibration for robotic surgery. IEEE Transactions on Medical Robotics and Bionics **2**(4), 569–573 (Nov 2020). https://doi.org/10.1109/tmrb.2020.3033670, http://dx.doi.org/10.1109/TMRB.2020.3033670
16. Sasaki, S.: n2v (2026), https://github.com/sammsaski/n2v
17. Shafique, M., Naseer, M., Theocharides, T., Kyrkou, C., Mutlu, O., Orosa, L., Choi, J.: Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. IEEE Design & Test **37**(2), 30–57 (Apr 2020). https://doi.org/10.1109/mdat.2020.2971217, http://dx.doi.org/10.1109/MDAT.2020.2971217
18. Tran, H.D., Bak, S., Xiang, W., Johnson, T.T.: Verification of deep convolutional neural networks using imagestars (2020), https://arxiv.org/abs/2004.05511
19. Tran, H.D., Yang, X., Manzanas Lopez, D., 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: Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part I. pp. 3–17. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-53288-8_1, https://doi.org/10.1007/978-3-030-53288-8_1
20. 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 complete and incomplete neural network robustness verification (2021), https://arxiv.org/abs/2103.06624

21. Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.W., Huang, M., Kailkhura, B., Lin, X., Hsieh, C.J.: Automatic perturbation analysis for scalable certified robustness and beyond (2020), https://arxiv.org/abs/2002.12920
22. 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 (2021), https://arxiv.org/abs/2011.13824
23. Yanik, E., Intes, X., Kruger, U., Yan, P., Miller, D., Voorst, B.V., Makled, B., Norfleet, J., De, S.: Deep neural networks for the assessment of surgical skills: A systematic review (2021), https://arxiv.org/abs/2103.05113
24. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions (2018), https://arxiv.org/abs/1811.00866