CENTER FOR TRUSTWORTHY DATA SCIENCE AND SECURITY
UA RUHR RESEARCH ALLIANCE

dortmund university

VeriFlow: Modeling Distributions for Neural Network Verification

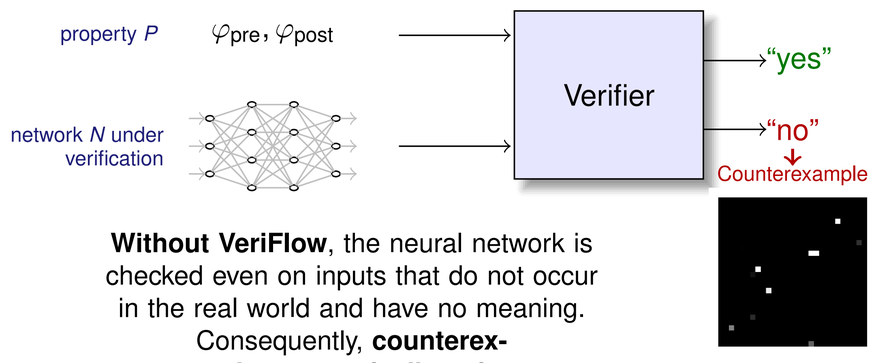
Faried Abu Zaid[1], Daniel Neider[2], Mustafa Yalçıner[2]
1: Independent Researcher, Munich
2: TU Dortmund University, Research Center Trustworthy Data Science and Security, UA Ruhr

Paper

Repository

Verification without VeriFlow

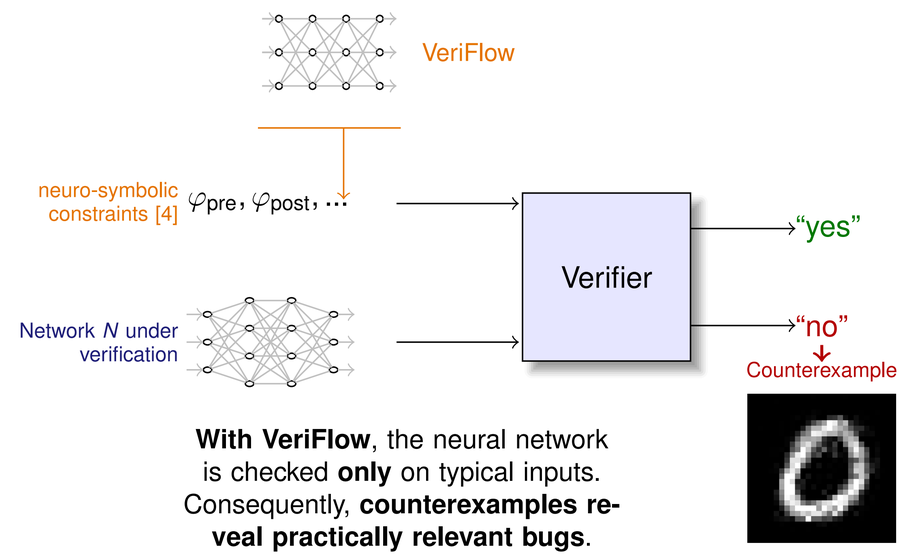
Verifying neural networks involves checking if a network f satisfies a semantic property P, often expressed as $\varphi_{pre}(x) \rightarrow \varphi_{post}(f(x))$, where φ_{pre} and φ_{post} are pre- and postconditions. Here, we verify whether all inputs classified as zero have low confidence.



Without VeriFlow, the neural network is checked even on inputs that do not occur in the real world and have no meaning. Consequently, **counterexamples are typically noise**.

Verification with VeriFlow

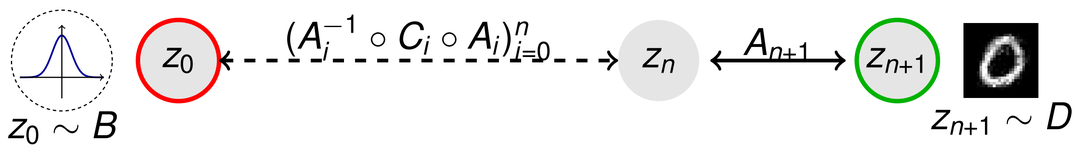
Here, we verify whether all in-distribution inputs classified as zero have low confidence. In-distributionness is specified by VeriFlow.



With VeriFlow, the neural network is checked **only** on typical inputs. Consequently, **counterexamples reveal practically relevant bugs**.

Architecture of VeriFlow

VeriFlow $F_\theta : \mathbb{R}^m \rightarrow \mathbb{R}^m$ maps a simple base distribution B to the target distribution D as follows (Figure inspired by [3]):



- A_i: General bijective affine transform parametrized by LU decomposition [1] (kernel of bijective 1^* convolution for tensors).
- C_i: Masked additive coupling layer $C_i(x) = x + (1 - m)c_i(mx)$ [2].
- All but the last affine transform are applied as *adjoint action* to coupling layers.
- Training maximizes the likelihood of each training sample $x \in \mathcal{D}$ and the learned distribution $p_{F_\theta(B)} = p_B(F_\theta^{-1}(x))|\det J_{F_\theta^{-1}}(x)|$.
- Base Distribution: ℓ_k-radial distributions via *learnable norm-distribution* $p_{|B|}$: $p_B(x) = p_{|B|}(\|x\|_k) / \frac{\partial V_d(r)}{\partial r}(\|x\|_k)$

Features of VeriFlow

Sampling and Density estimation

Both computed efficiently via inference.

F is a Bijection

All points can be sampled and estimated.

F is Piecewise Linear

F is a MILP-encodable ReLU-network.

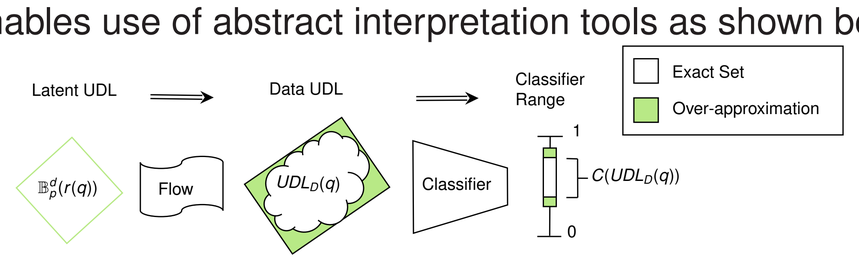
Uniformly Scaling (Constant $|\det J_{F_\theta^{-1}}(x)|$, US)

VeriFlow maps upper density level sets (UDL) sets of the base distribution to UDLs of the data distribution. The figure shows flows mapping data densities (colored) onto a 2D Gaussian.

Non-US Flow (densities unaligned) VeriFlow (densities aligned)

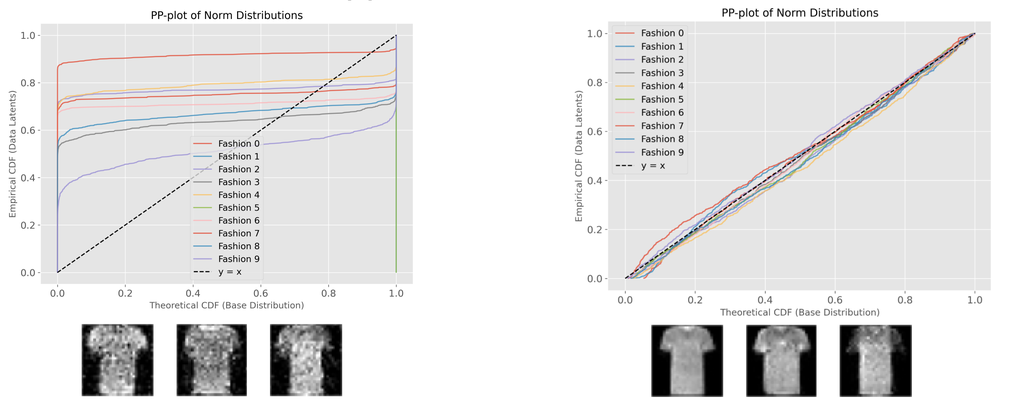
UDL of base distribution is an ℓ_k-norm ball.

- VeriFlow uses a *k-radial* distribution, where the probability density depends solely on the ℓ_k-norm of the input.
- The UDL is then precisely characterized by a simple box (ℓ_∞).
- Enables use of abstract interpretation tools as shown below:



Benchmark

VeriFlow outperforms the US baseline MaCow. Learned distributions tend to be over-approximations, admissible for verification.



MaCow VeriFlow

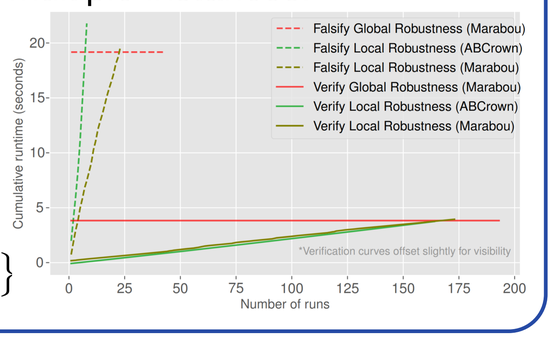
New Notion of Global Robustness

We verify robustness of a classifier c for all inputs x_ℓ drawn from the UDL of VeriFlow F with base distribution B. One global certificate covering a q%-UDL implies local robustness for q% of the dataset, making its overhead worthwhile for multiple instances.

$$\varphi_{pre}: \quad \{x_\ell \in UDL_B(q),\ x' \in \mathbb{R}^n\}$$

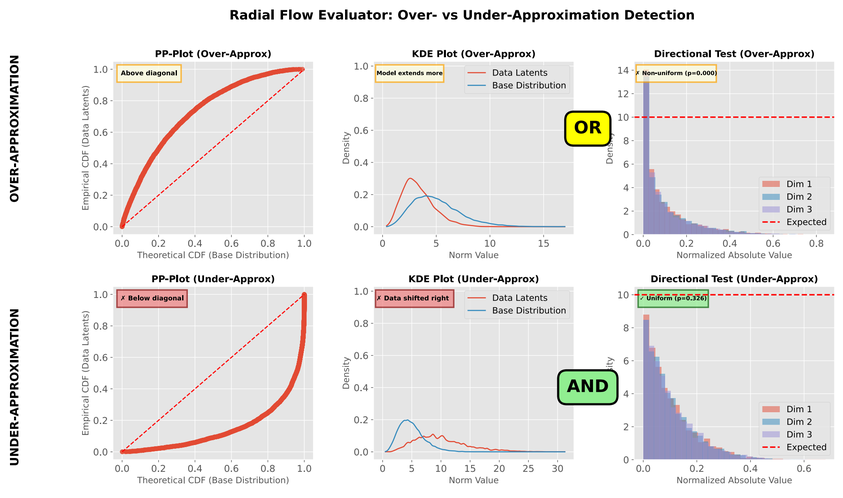
$$x_t \leftarrow F(x_\ell) \quad y \leftarrow c(x_t) \quad y' \leftarrow c(x')$$

$$\varphi_{post}: \quad \{\text{dist}(x_t, x') \leq \epsilon \Rightarrow \arg\max_i y = \arg\max_i y'\}$$



Statistical Evaluation Suite for VeriFlow

Detect discrepancies via latent norms (PP-plots, KDE), and uniformity on the latent simplex (normed absolute values).



References

[1] Robin Chan, Sarina Penquitt, and Hanno Gottschalk. Lu-net: Invertible neural networks based on matrix factorization. *arXiv preprint arXiv:2302.10524*, 2023.

[2] Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard Hovy. MaCow: Masked convolutional generative Flow. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 5893–5902. Curran Associates Inc., 2019.

[3] Lilian Weng. Flow-based deep generative models. *lilianweng.github.io*, 2018.

[4] Xuan Xie, Kristian Kersting, and Daniel Neider. Neuro-symbolic verification of deep neural networks. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3622–3628. ijcai.org, 2022.