Tight Robustness Certification Through the Convex Hull of ℓ_0 Attacks

Yuval Shapira, Dana Drachsler-Cohen

Technion - Israel Institute of Technology
shapirayuval@campus.technion.ac.il, ddana@ee.technion.ac.il

Abstract

Few-pixel attacks mislead a classifier by modifying a few pixels of an image. Their perturbation space is an ℓ_0-ball, which is not convex, unlike ℓ_p-balls for $p \geq 1$. However, existing local robustness verifiers typically scale by relying on linear bound propagation, which captures convex perturbation spaces. We show that the convex hull of an ℓ_0-ball is the intersection of its bounding box and an asymmetrically scaled ℓ_1-like polytope. The volumes of the convex hull and this polytope are nearly equal as the input dimension increases. We then show a linear bound propagation that precisely computes bounds over the convex hull and is significantly tighter than bound propagations over the bounding box or our ℓ_1-like polytope. This bound propagation scales the state-of-the-art ℓ_0 verifier on its most challenging robustness benchmarks by 1.24x-7.07x, with a geometric mean of 3.16.

Code — https://github.com/YuvShap/top-t-CoVerD
Extended version — https://arxiv.org/abs/2511.10576

Introduction

Image network classifiers are a central part of many safety-critical systems in healthcare (Miller 2023), autonomous driving (Hawkins 2025), and automated visual inspection (Marshall 2023). However, neural networks are susceptible to adversarial example attacks (Szegedy et al. 2013). An adversarial attack injects a small perturbation, often confined to the ℓ_p-ball of a correctly classified input, to mislead the classifier. Local robustness (Bastani et al. 2016) is the main safety property for showing the resilience of networks to adversarial attacks.

Many robustness verifiers analyze the local robustness of neural networks to perturbations in ℓ_∞-balls (Tjeng, Xiao, and Tedrake 2019; Zhang et al. 2018; Müller et al. 2021; Gehr et al. 2018; Katz et al. 2017), ℓ_2-balls (Leino, Wang, and Fredrikson 2021; Huang et al. 2021), and ℓ_1-balls (Behl et al. 2021; Wu and Zhang 2021). To scale, most incomplete verifiers rely on linear bound propagation that overapproximates the network's computations with convex polytopes. Since ℓ_p-balls are convex for $p \geq 1$, such analysis does not introduce overapproximation error in the perturbation space. However, ℓ_0-balls are not convex. This raises the question: can linear bound propagation avoid introducing overapproximation error in the non-convex ℓ_0 perturbation space?

We mathematically characterize the convex hull of an ℓ_0-ball around an input x (single- or multi-channel). We show that the convex hull is the intersection of the bounding box of the ℓ_0-ball and an asymmetrically scaled ℓ_1-like polytope. We show that the relative excess volume of this polytope compared to the convex hull converges exponentially to zero, suggesting it may be a good overapproximation for linear bound propagation.

We then present a linear bound propagation that precisely computes the minimum and maximum of a linear function over an ℓ_0-ball, which coincide with those over its convex hull. We show that the minimum and maximum of a linear function f over an ℓ_0-ball of radius t depend on the sum of the t lowest (for the minimum) or highest (for the maximum) input entry contributions to f. This bound propagation generalizes prior bound propagations for few-pixel attacks (Chiang et al. 2020; Xu et al. 2020) to any box input domain and to multi-channel inputs. We also present a bound propagation for our ℓ_1-like polytope and show that the minimum and maximum depend on the product of t and the lowest or highest input entry contribution to f. That is, this polytope is a looser overapproximation than the convex hull, although their volumes are very close.

We integrate our bound propagation in GPUPoly (Müller et al. 2021), which is repeatedly called by CoVerD (Shapira et al. 2024), the state-of-the-art complete (exact) ℓ_0 robustness verifier. We evaluate it over fully-connected and convolutional classifiers, for MNIST, Fashion-MNIST, and CIFAR-10. Our bound propagation boosts CoVerD on its most challenging robustness benchmarks by 1.24x-7.07x, with a geometric mean of 3.16.

In summary, our main contributions are:

- A characterization of the convex hull of ℓ_0 perturbations.
- A linear bound propagation that precisely computes the minimum and maximum of a linear function over an ℓ_0-ball, which is significantly tighter than bound propagations over its bounding box or our ℓ_1-like polytope.
- An integration of our bound propagation in GPUPoly, for boosting ℓ_0 robustness verification.

Preliminaries

In this section, we provide our notation and setting.

An input x is in a bounded box domain $\mathcal{D}_v = \prod_{i=1}^{v}[a_i, b_i]$ if x is single-channel, or $\mathcal{D}_v = \prod_{i=1}^{v}(\prod_{j=1}^{d}[a_i^{(j)}, b_i^{(j)}])$ if x is multi-channel. For example, an RGB image x is in $\mathcal{D}_v = ([0,1]^3)^v$. We denote an entry by x_i for $i \in [v] = \{1, \ldots, v\}$ and a channel by $x_i^{(j)}$ for $i \in [v]$ and $j \in [d]$.

We focus on the local robustness of a network classifier to few-pixel attacks (Croce et al. 2022). A classifier maps an input to a score vector over c labels: $N : \mathbb{R}^v \to \mathbb{R}^c$. The classification of an input is the label with the maximal score: $\text{class}_N(x) = \text{argmax}(N(x))$. An adversarial attack considers an input $\bar{x} \in \mathcal{D}_v$ and computes a small perturbation r such that $\bar{x} + r \in \mathcal{D}_v$ and the perturbed input is classified differently: $\text{class}_N(\bar{x}) \neq \text{class}_N(\bar{x}+r)$. In few-pixel attacks, the attacker is given a bound $t \in [v]$ on the number of entries that can be perturbed, which is often very small, and computes r whose ℓ_0 norm is at most t: $|\{i \in [v] \mid r_i \neq 0\}| \leq t$.

We consider a generalized setting in which the attacker is limited to perturbing a subset of pixels, $\mathcal{K} \subseteq [v]$. Note that $\mathcal{K}$ can be arbitrarily large, and in particular it can be $[v]$, as in our experiments. Since the attacker cannot perturb the pixels in $[v] \setminus \mathcal{K}$, given $\bar{x} \in \mathcal{D}_v$, the effective input space is $\mathcal{D} = \prod_{i \in \mathcal{K}}[a_i, b_i]$ (and similarly for multi-channel inputs), i.e., pixels not in $\mathcal{K}$ are treated as constants. Formally, the extension of $y \in \mathcal{D}$ to $y' \in \mathcal{D}_v$ sets $y'_i = y_i$, for $i \in \mathcal{K}$, and $y'_i = \bar{x}_i$, for $i \in [v] \setminus \mathcal{K}$. We abuse notation: for $\bar{x} \in \mathcal{D}_v$, we also write $\bar{x}$ for its projection onto $\mathcal{K}$, and for $y \in \mathcal{D}$, we also write y for its extension to $\mathcal{D}_v$. The space of all perturbed inputs is the **ℓ_0-ball of $\bar{x}$ with radius t**:

$$\mathcal{B}_0^t(\bar{x}) = \{y \in \mathcal{D} \mid |\{i \in \mathcal{K} \mid y_i \neq \bar{x}_i\}| \leq t\}$$

A network N is locally robust in $\mathcal{B}_0^t(\bar{x})$ if it classifies all inputs the same: $\forall y \in \mathcal{B}_0^t(\bar{x}).\ \text{class}_N(\bar{x}) = \text{class}_N(y)$. Without loss of generality and to simplify notations, we assume $\mathcal{K} = [k]$, for $k \in [t, v]$ (though our formulation, theorems, and bound propagations hold for any $\mathcal{K} \subseteq [v]$). In particular, we write $\mathcal{D} = \prod_{i=1}^{k}[a_i, b_i]$. Figure 1 (left) shows the ℓ_0-balls of three single-channel inputs, for $t = 2$. The figure shows that the ℓ_0-ball is the union of $\binom{3}{2}$ planes. Generally, an ℓ_0-ball is a union of $\binom{k}{t}$ $t \cdot d$-dimensional flats.

Although our focus is on robustness of image classifiers, our formulation, theorems, and bound propagations apply to other classifiers. For example, to text classifiers, which are susceptible to word replacement attacks (Alzantot et al. 2018). Given an input sentence, the attacker replaces up to t words. Technically, if the words are independently embedded (e.g., with Word2Vec (Mikolov et al. 2013)), the problem setting is similar: given an input sentence $\bar{x}$ with v words $\bar{x}_1, \ldots, \bar{x}_v$, the input domain is $\mathcal{D}_v = \prod_{i=1}^{v}(\prod_{j=1}^{d}[a_i^{(j)}, b_i^{(j)}])$, where $\prod_{j=1}^{d}[a_i^{(j)}, b_i^{(j)}]$ is the bounding box of the embedding of $\bar{x}_i$ and its replacements.

The Convex Hull of ℓ_0-Balls

In this section, we characterize the convex hull of an ℓ_0-ball.

The Convex Hull for Single-Channel Inputs

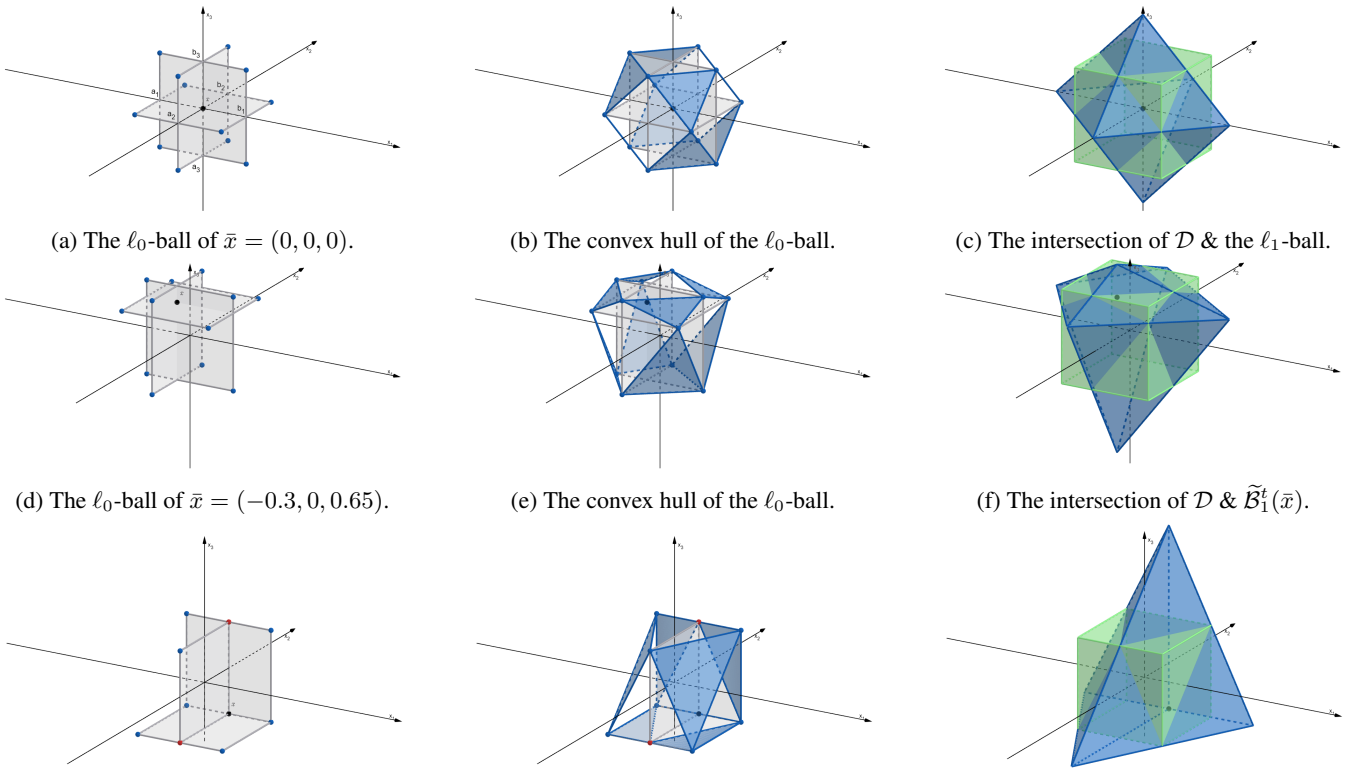
In this section, we focus on a single-channel input $\bar{x}$.

Illustration We illustrate by considering the ℓ_0-ball of $\bar{x} = (0,0,0)$ that lies in the center of $\mathcal{D} = [-1,1]^3$, shown in Figure 1a. The figure illustrates that the ℓ_0-ball is not convex, unlike ℓ_p-balls for $p \geq 1$. The corners $E_{\mathcal{B}_0^t(\bar{x})}$ of the ℓ_0-ball are all inputs $y \in \mathbb{R}^k$ such that $k - t$ of their entries are zeros, and the other entries y_i are one of the bounds -1 or 1 (blue dots in Figure 1a). The convex hull of the ℓ_0-ball is the smallest convex set that contains it. It is obtained by connecting the corners $E_{\mathcal{B}_0^t(\bar{x})}$ (Figure 1b). We shortly provide a characterization of this convex hull. One may wonder whether it is crucial, or whether it suffices to overapproximate the ℓ_0-ball by a containing convex set that can be analyzed by existing robustness verifiers. For example, most verifiers are designed for box neighborhoods. Thus, one may attempt verifying local robustness in an ℓ_0-ball by overapproximating it with its bounding box, whose boundary contains the corners of the ℓ_0-ball, and checking robustness in this box. However, this box equals the input space $\mathcal{D}$. For large k, the network is not locally robust in $\mathcal{D}$ (e.g., for $k = v$, it is not robust since the network does not classify all inputs the same). Thus, this approach usually fails for large k. Instead, one may overapproximate the ℓ_0-ball with the tightest convex ℓ_p-ball, which is the ℓ_1-ball: $\{y \in \mathbb{R}^k \mid \|y\|_1 \leq t\}$ (Zass and Shashua 2006; Baninajjar, Rezine, and Aminifar 2024), and submit it to an ℓ_1-ball robustness verifier (Behl et al. 2021). However, the ℓ_1-ball has sharp corners whose ℓ_∞ distance from $\mathcal{D}$ can be very high (Figure 1c). This overapproximation error can fail the verification. Our first theorem shows that, in the case where $\bar{x}$ is in the center of $\mathcal{D}$, the convex hull of the ℓ_0-ball of $\bar{x}$ is the intersection of $\mathcal{D}$ and the ℓ_1-ball. Figure 1c shows $\mathcal{D}$ (in green) and the ℓ_1-ball with radius $t = 2$ (in blue).

General case We next present our characterization for the general case where $\bar{x}$ may not be in the center of $\mathcal{D}$, e.g., the $\bar{x}$ in Figures 1d and 1g (the black dot). In this case, the corners $E_{\mathcal{B}_0^t(\bar{x})}$ (the blue dots) are inputs $y \in \mathbb{R}^k$ such that $k-t$ of their entries y_i are $\bar{x}_i$, and the other entries y_i are one of the bounds a_i or b_i (which may be equal to $\bar{x}_i$, e.g., Figure 1g). We note that, unlike the simpler case, it can happen that an input y is not a corner even if it has $k - t$ entries $y_i = \bar{x}_i$ and its other entries y_i are a_i or b_i (red dots and $\bar{x}$ itself in Figure 1g). The convex hull is obtained by connecting the corners $E_{\mathcal{B}_0^t(\bar{x})}$ (Figures 1e and 1h). We show that it equals the intersection of $\mathcal{D}$ and an *asymmetrically scaled* ℓ_1-ball around $\bar{x}$. To define this polytope, we first define the *asymmetrically scaled i distance*, for $i \in [k]$. It measures how far y_i is from $\bar{x}_i$: if $y_i = \bar{x}_i$, this distance is zero, otherwise if y_i is one of the bounds a_i or b_i, this distance is one, and otherwise this distance is proportional to the distance between $\bar{x}_i$ and the bound that is closer to y_i (than to $\bar{x}_i$). Formally, the asymmetrically scaled i distance from input y to $\bar{x}$ is the difference of their i^{th} entries normalized by the distance of $\bar{x}_i$ and the bound b_i, if $y_i > \bar{x}_i$, or a_i, if $y_i < \bar{x}_i$:

$$\delta^i_{\bar{x},a_i,b_i}(y) = \frac{y_i - \bar{x}_i}{\mathbf{1}_{\{y_i > \bar{x}_i\}}(b_i - \bar{x}_i) + \mathbf{1}_{\{y_i < \bar{x}_i\}}(a_i - \bar{x}_i)} \quad (1)$$

We omit a_i and b_i to simplify notation. We define $\delta^i_{\bar{x}}(y) = 0$, if $y_i = \bar{x}_i$ (when the numerator and denominator are 0) and



(a) The ℓ_0-ball of $\bar{x} = (0, 0, 0)$. (b) The convex hull of the ℓ_0-ball. (c) The intersection of $\mathcal{D}$ & the ℓ_1-ball.

(d) The ℓ_0-ball of $\bar{x} = (-0.3, 0, 0.65)$. (e) The convex hull of the ℓ_0-ball. (f) The intersection of $\mathcal{D}$ & $\widetilde{\mathcal{B}}_1^t(\bar{x})$.

(g) The ℓ_0-ball of $\bar{x} = (0, 1, -1)$. (h) The convex hull of the ℓ_0-ball. (i) The intersection of $\mathcal{D}$ & $\widetilde{\mathcal{B}}_1^t(\bar{x})$.

Figure 1: Illustration of the perturbation space, for $\mathcal{D} = \prod_{i=1}^{k}[a_i, b_i] = [-1, 1]^3$ and $t = 2$. Left: The ℓ_0-ball of an input $\bar{x}$, $\mathcal{B}_0^t(\bar{x})$. Middle: Its convex hull. Right: The convex hull as the intersection of $\mathcal{D}$ and an ℓ_1-like polytope (Theorem 1), $\widetilde{\mathcal{B}}_1^t(\bar{x}) = \{y \in \mathbb{R}^k \mid \sum_{i=1}^{k} \delta_{\bar{x}}^i(y) \leq t\}$ (see Equation (1)). The plots show $\bar{x}$ as a black dot and the corners of the ℓ_0-ball as blue dots.

$\delta_{\bar{x}}^i(y) = \infty$, if $\bar{x}_i = b_i$ and $y_i > \bar{x}_i$ or if $\bar{x}_i = a_i$ and $y_i < \bar{x}_i$ (when the denominator is 0). Thus, $\delta_{\bar{x}}^i(y) \in [0, \infty]$. If $y \in \mathcal{D}$, then $\delta_{\bar{x}}^i(y) \in [0, 1]$, since for $y_i = \bar{x}_i$, $\delta_{\bar{x}}^i(y) = 0$, and for $y_i \in \{a_i, b_i\} \setminus \{\bar{x}_i\}$, $\delta_{\bar{x}}^i(y) = 1$. As an example, consider $[a_i, b_i] = [-1, 1]$ and $y_i = 0.3$. For $\bar{x}_i = 0$ (the center), $\delta_{\bar{x}}^i(y) = 0.3$. For $\bar{x}_i = 0.7$, $\delta_{\bar{x}}^i(y) = \frac{0.3-0.7}{0+1\cdot(-1-0.7)} = 0.235$.

The *asymmetrically scaled ℓ_1-ball* $\widetilde{\mathcal{B}}_1^t(\bar{x})$ is the set of inputs whose sum over all asymmetrically scaled i distances from $\bar{x}$ (Equation (1)) is at most t:

$$\widetilde{\mathcal{B}}_1^t(\bar{x}) = \{y \in \mathbb{R}^k \mid \sum_{i=1}^{k} \delta_{\bar{x}}^i(y) \leq t\}$$

Note that $\widetilde{\mathcal{B}}_1^t(\bar{x})$ is defined over $y \in \mathbb{R}^k$ (unlike $\mathcal{B}_0^t(\bar{x})$, defined over $y \in \mathcal{D}$), because it generalizes the ℓ_1-ball, defined over $y \in \mathbb{R}^k$. Figures 1f and 1i exemplify $\widetilde{\mathcal{B}}_1^t(\bar{x})$ (in blue).

Theorem We next state and prove our main theorem, which characterizes the convex hull of an ℓ_0-ball. In the following, we denote by $Conv(S)$ the convex hull of a set S.

Theorem 1. $Conv(\mathcal{B}_0^t(\bar{x})) = \mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$.

Proof.

• $Conv(\mathcal{B}_0^t(\bar{x})) \subseteq \mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$: By definition, $\mathcal{B}_0^t(\bar{x}) \subseteq \mathcal{D}$ and $\mathcal{B}_0^t(\bar{x}) \subseteq \widetilde{\mathcal{B}}_1^t(\bar{x})$. Hence, $\mathcal{B}_0^t(\bar{x}) \subseteq \mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$. Since $Conv(\cdot)$ is monotone with respect to set inclusion, $Conv(\mathcal{B}_0^t(\bar{x})) \subseteq Conv(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x}))$. The set $\mathcal{D}$ is convex (as a box) and the set $\widetilde{\mathcal{B}}_1^t(\bar{x})$ is convex (by Lemma 1, appendix, in the extended version) and thus so is $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$. Thus, $Conv(\mathcal{B}_0^t(\bar{x})) \subseteq \mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$.

• $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x}) \subseteq Conv(\mathcal{B}_0^t(\bar{x}))$: The sets $\mathcal{D}$ and $\widetilde{\mathcal{B}}_1^t(\bar{x})$ are convex and compact (as a box or by Lemma 1, appendix), and thus so is $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$. By the Krein-Milman Theorem (Rockafellar 1997), a compact convex set is the convex hull of its extreme points. Thus, $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x}) = Conv(E_\cap)$, where $E_\cap$ is the set of extreme points of $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$. Since $E_\cap \subseteq \mathcal{B}_0^t(\bar{x})$ (Lemma 2, appendix), $Conv(E_\cap) \subseteq Conv(\mathcal{B}_0^t(\bar{x}))$. □

Volumes We next study the volumes of the convex hull, $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$, and $\widetilde{\mathcal{B}}_1^t(\bar{x})$ (the volume of $\mathcal{D}$ is $\prod_{i=1}^{k}(b_i - a_i)$). We show that the relative excess volume of $\widetilde{\mathcal{B}}_1^t(\bar{x})$ compared to $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$ converges exponentially to 0 as k increases. We begin by providing closed-form expressions for these volumes, which depend only on $\mathcal{D}$ and t (not on the input $\bar{x}$). In particular, this implies that the volumes of the convex hulls shown in Figures 1b, 1e and 1h are equal.

Theorem 2. *1.* $vol(\widetilde{\mathcal{B}}_1^t(\bar{x})) = vol(\mathcal{D})\frac{t^k}{k!}$.
2. $vol(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})) = vol(\mathcal{D})\frac{t^k}{k!}\sum_{r=0}^{t-1}(-1)^r\binom{k}{r}(1-\frac{r}{t})^k$.

Proof outline (proof is in the appendix). We partition $\widetilde{\mathcal{B}}_1^t(\bar{x})$ and $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$ into 2^k parts, one for each orthant. Each part is a scaled version of $\Delta_{k,t} = \{z \in [0,\infty)^k \mid \sum_{i=1}^k z_i \leq t\}$ for $\widetilde{\mathcal{B}}_1^t(\bar{x})$ and $[0,1]^k \cap \Delta_{k,t}$ for $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$. It is known that $\text{vol}(\Delta_{k,t}) = \frac{t^k}{k!}$. Thus, summing over the scaled volumes yields $\text{vol}(\widetilde{\mathcal{B}}_1^t(\bar{x})) = \text{vol}(\mathcal{D})\frac{t^k}{k!}$ (proving bullet 1) and $\text{vol}(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})) = \text{vol}(\mathcal{D}) \cdot \text{vol}(\Delta_{k,t} \cap [0,1]^k)$. We show that $\text{vol}(\Delta_{k,t} \cap [0,1]^k)$ is equal to the cumulative distribution function (CDF) of the Irwin–Hall distribution (Irwin 1927; Hall 1927) evaluated at t, which is $\frac{t^k}{k!}\sum_{r=0}^{t-1}(-1)^r\binom{k}{r}(1-\frac{r}{t})^k$ (proving bullet 2). Technically, to compute the volume, we compute the probability that a uniformly sampled input from $[0,1]^k$ lies in $\Delta_{k,t}$, i.e., that the sum of its entries is at most t, and multiply it by $\text{vol}([0,1]^k) = 1$. This probability is obtained by evaluating the Irwin-Hall CDF at t. □

Excess volumes We next compare the volume of the convex hull to $\mathcal{D}$ and to $\widetilde{\mathcal{B}}_1^t(\bar{x})$. To shorten notation, we write $\text{vol}(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})) = \text{vol}(\widetilde{\mathcal{B}}_1^t(\bar{x}))\sum_{r=0}^{t-1} c_{r,k,t}$, where $c_{r,k,t} = (-1)^r\binom{k}{r}(1-\frac{r}{t})^k$, for $r \in \{0,\ldots,t-1\}$. Note that $c_{r,k,t} = 1$, for $r = 0$, and $c_{r,k,t}$ converges exponentially to 0 as k increases, for $r > 0$. The relative excess volume of $\widetilde{\mathcal{B}}_1^t(\bar{x})$:

$$\frac{\text{vol}(\widetilde{\mathcal{B}}_1^t(\bar{x}) \setminus (\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})))}{\text{vol}(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x}))} = \frac{-\sum_{r=1}^{t-1} c_{r,k,t}}{1+\sum_{r=1}^{t-1} c_{r,k,t}}$$

This expression is asymptotically negligible because the numerator approaches 0 and the denominator approaches 1. The relative excess volume of $\mathcal{D}$ is:

$$\frac{\text{vol}(\mathcal{D} \setminus (\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})))}{\text{vol}(\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x}))} = \frac{\frac{k!}{t^k} - \sum_{r=0}^{t-1} c_{r,k,t}}{1+\sum_{r=1}^{t-1} c_{r,k,t}}$$

This expression goes to infinity, since $\frac{k!}{t^k}$ diverges with k. Figure 2 illustrates this behavior for different values of t. It shows that for very small values of k, the relative excess volume of $\mathcal{D}$ is very small, but for $k \geq 20$ it is very large (as context, for MNIST images $k = 784$). In contrast, the relative excess volume of $\widetilde{\mathcal{B}}_1^t(\bar{x})$ converges to zero very quickly.

The Convex Hull for Multi-Channel Inputs

In this section, we focus on a multi-channel input $\bar{x}$. In this setting, the attacker chooses t entries $\bar{x}_i$ to perturb across all their channels (overall, $t \cdot d$ values). To characterize the convex hull, we extend our definitions to multi-channel inputs. The *asymmetrically scaled* (i,j) *distance* from y to $\bar{x}$ is:

$$\delta_{\bar{x}}^{i,j}(y) = \frac{y_i^{(j)} - \bar{x}_i^{(j)}}{\mathbf{1}_{y_i^{(j)} > \bar{x}_i^{(j)}}(b_i^{(j)} - \bar{x}_i^{(j)}) + \mathbf{1}_{y_i^{(j)} < \bar{x}_i^{(j)}}(a_i^{(j)} - \bar{x}_i^{(j)})}$$

This definition is identical to $\delta_{\bar{x}}^i(y)$, but over a channel $y_i^{(j)}$, for $i \in [k], j \in [d]$. We extend the asymmetrically scaled ℓ_1-ball to the set of inputs whose sum, over all entries i,

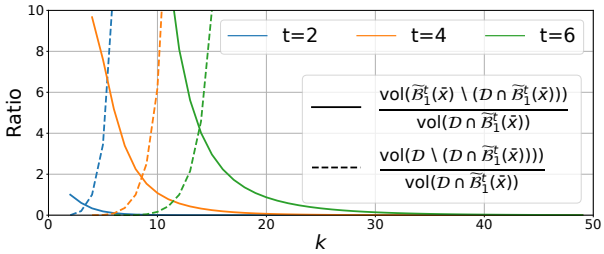


Figure 2: The relative excess volumes.

of the *maximal* asymmetrically scaled (i,j) distances across all channels j is at most t. Since the sum is over maximal values, we denote it $\widetilde{\mathcal{B}}_{1,\infty}^t(\bar{x})$:

$$\widetilde{\mathcal{B}}_{1,\infty}^t(\bar{x}) = \{y \in (\mathbb{R}^d)^k \mid \sum_{i=1}^k \max_{j\in[d]} \delta_{\bar{x}}^{i,j}(y) \leq t\}$$

This definition is identical to $\widetilde{\mathcal{B}}_1^t(\bar{x})$, except that it is over the maximal $\delta_{\bar{x}}^{i,j}(y)$. Like Theorem 1, we show that the convex hull of $\mathcal{B}_0^t(\bar{x})$ is the intersection of $\mathcal{D}$ and $\widetilde{\mathcal{B}}_{1,\infty}^t(\bar{x})$:

Theorem 3. $Conv(\mathcal{B}_0^t(\bar{x})) = \mathcal{D} \cap \widetilde{\mathcal{B}}_{1,\infty}^t(\bar{x})$

The proof is identical to the proof of Theorem 1, except that it relies on Lemmas 3 and 4 (provided in the appendix), which extend Lemmas 1 and 2 to multi-channel inputs. The appendix also shows Theorem 4, which extends Theorem 2 to multi-channel inputs and implies a a similar trend for the relative excess volumes.

Bound Propagations

In this section, we introduce linear bound propagation methods for local robustness verification over an ℓ_0-ball. We begin with background, including bound propagation over a box domain. Then, we present an equivalent formulation for the box domain, which we build on in our subsequent bound propagations. Next, we present bound propagations over an ℓ_0-ball and over an asymmetrically scaled ℓ_1-ball. Lastly, we extend them to multi-channel inputs and compare them.

Background

Bound propagation is a central component in many neural network verifiers (e.g., Zhang et al. (2018); Müller et al. (2021); Singh et al. (2019)). This technique bounds every neuron n within a real-valued interval $[l,u] \in \mathbb{R}^2$. These bounds allow the verifier to efficiently check properties (i.e., constraints over the network's inputs and outputs) as well as to tighten linear relaxations of the network's computations, which reduce the verification time.

To obtain a lower bound l for a neuron n, a verifier computes an affine function that lower bounds the computation of n and computes its minimum. To shorten notations, we assume that the affine function is linear. Formally, for single-channel inputs, a verifier computes a linear function over the network's inputs $y \mapsto \sum_{i=1}^k w_i y_i$, for $w_i \in \mathbb{R}$, lower bounding n in the property's input subspace $\mathcal{I} \subseteq \mathbb{R}^k$:

$\forall y \in \mathcal{I}.\ n(y) \geq \sum_{i=1}^{k} w_i y_i$. It then computes the minimum: $l = \min_{y \in \mathcal{I}} \sum_{i=1}^{k} w_i y_i$, based on the shape of $\mathcal{I}$. Commonly, $\mathcal{I}$ is a box domain (shortly described). Similarly, an upper bound u of n is obtained by computing a linear function that upper bounds the computation of n and computing its maximum. The linear functions are computed by linear relaxation to the nonlinear activation functions (e.g., ReLU). Our bound propagation is orthogonal to the linear relaxation computation. In particular, it is applicable to relaxations computed by a backward pass (e.g., Zhang et al. (2018); Singh et al. (2019)), a forward pass (e.g., Wang et al. (2018a,b)), or a hybrid pass (e.g., Kouvaros et al. (2025)).

Bound propagation for a box domain Consider a neuron n whose computation is lower bounded by $y \mapsto \sum_{i=1}^{k} w_i y_i$ for $y \in \mathcal{I} = \mathcal{D} = \prod_{i=1}^{k} [a_i, b_i]$. In the box domain, it is easy to define an input y obtaining the minimum: every entry y_i is on the boundaries, a_i or b_i, depending on the sign of w_i. Formally, $y_i = a_i$ if $w_i \geq 0$, or $y_i = b_i$ otherwise.

Equivalent Bound Propagation for a Box Domain

We begin with an equivalent formulation of bound propagation over the box domain, to allow a unified formulation for all bound propagations and simplify their extension to multi-channel inputs (shown later). Our formulations are defined over the *minimum and maximum input entry contributions*.

Given $\bar{x} \in \mathcal{D} = \prod_{i=1}^{k} [a_i, b_i]$ and $y \mapsto \sum_{i=1}^{k} w_i y_i$, we write: $\sum_{i=1}^{k} w_i y_i = \sum_{i=1}^{k} w_i \bar{x}_i + \sum_{i=1}^{k} w_i (y_i - \bar{x}_i)$. Note that the minimum and maximum of this linear function over $y \in \mathcal{I}$ depend only on the second sum (the first sum is fixed). We define the *minimum input entry contribution* of $i \in [k]$:

$$d_i^- = \min(w_i(b_i - \bar{x}_i), w_i(a_i - \bar{x}_i)),$$

which lower bounds the term $w_i \cdot (y_i - \bar{x}_i)$, for $y_i \in [a_i, b_i]$. We define the *maximum input entry contribution* of $i \in [k]$:

$$d_i^+ = \max(w_i(b_i - \bar{x}_i), w_i(a_i - \bar{x}_i)),$$

which upper bounds $w_i \cdot (y_i - \bar{x}_i)$, for $y_i \in [a_i, b_i]$. Note that $d_i^- \leq 0$ and $d_i^+ \geq 0$.

Given this notation, if n's function is lower bounded by $y \mapsto \sum_{i=1}^{k} w_i y_i$ for $y \in \mathcal{I} = \mathcal{D}$, the lower bound depends on the sum of all d_i^-:

$$l = \min_{y \in \mathcal{D}} \sum_{i=1}^{k} w_i y_i = \sum_{i=1}^{k} w_i \bar{x}_i + \sum_{i=1}^{k} d_i^-$$

If n's function is upper bounded by $y \mapsto \sum_{i=1}^{k} w_i y_i$ for $y \in \mathcal{D}$, the upper bound depends on the sum of all d_i^+:

$$u = \max_{y \in \mathcal{D}} \sum_{i=1}^{k} w_i y_i = \sum_{i=1}^{k} w_i \bar{x}_i + \sum_{i=1}^{k} d_i^+$$

Example Figure 3 shows a network with inputs y_1, y_2, y_3 and an output o_1. The weights are on the edges and the bias of o_1 is above it. The neurons n_1, n_2, o_1 compute the sum of their bias and the products of their inputs and respective weights. The neurons $\text{ReLU}_1, \text{ReLU}_2$ compute the activation function $\texttt{ReLU}_i(n_i) = \max(0, n_i)$. Our goal is to prove $o_1 \geq 0$ for every input in the ℓ_0-ball of $\bar{x} = (-0.3, 0, 0.65)$, for $\mathcal{D} = [-1, 1]^3$ and $t = 2$ (Figure 1d). The box bound propagation overapproximates the ℓ_0-ball by its bounding box $\mathcal{D}$ (although $\mathcal{B}_t^0(\bar{x})$ is significantly smaller). That is, $y_i \in [-1, 1]$ for $i \in [3]$. Then, a linear relaxation lower and upper bounds every neuron by linear functions. In this example, we use the linear relaxations of ReLU proposed by Zhang et al. (2018); Singh et al. (2019). Then, the box bound propagation computes the minimum and maximum of these linear functions, as described. This results in the intervals marked by $\mathcal{D}$ in Figure 3. For example, the linear functions bounding n_1 are exactly its function (since it is linear): $2 \cdot y_1 - 3 \cdot y_2 + 7 \cdot y_3$. Since $\bar{x} = (-0.3, 0, 0.65) \in [-1, 1]^3$, we get $d_1^- = \min(2 \cdot (1 + 0.3), 2 \cdot (-1 + 0.3)) = -1.4$ and similarly $d_2^- = -3$, $d_3^- = -11.55$. Thus, its lower bound is:

$$\min_{(y_1,y_2,y_3) \in [-1,1]^3} 2 \cdot y_1 - 3 \cdot y_2 + 7 \cdot y_3 =$$

$$\overbrace{(2 \cdot (-0.3) - 3 \cdot 0 + 7 \cdot 0.65)}^{3.95} - 1.4 - 3 - 11.55 = -12$$

Since the box bound propagation computes $o_1 \in [-1, 32]$, it cannot show that the property $o_1 \geq 0$ holds.

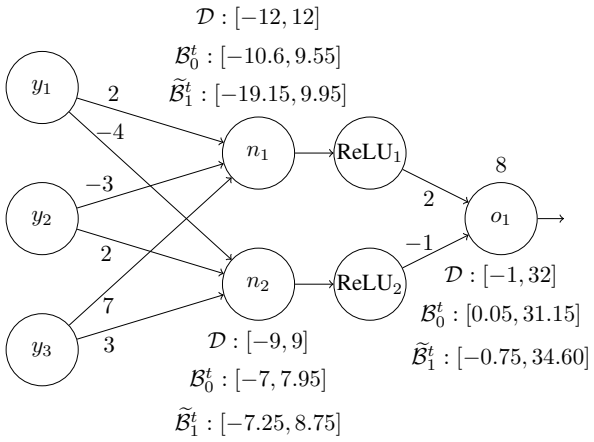


Figure 3: The three approaches for bound propagation over $\mathcal{B}_0^t(\bar{x})$, where $\bar{x} = (-0.3, 0, 0.65)$, $\mathcal{D} = [-1, 1]^3$, and $t = 2$.

Bound Propagation for an ℓ_0-Ball

We next show a bound propagation that precisely captures an ℓ_0-ball input subspace. That is, it computes the minimum and maximum of a linear function over an ℓ_0-ball, which coincide with those over its convex hull. We call it *top-t*, since it relies on the *top-t* input entry contributions. This idea has been proposed by Chiang et al. (2020); Xu et al. (2020), however they assume $\mathcal{D} = [0, 1]^v$ and single-channel inputs.

As before, the goal is to compute the minimum of a neuron n whose function is lower bounded by $y \mapsto \sum_{i=1}^{k} w_i y_i$. Again, we write $\sum_{i=1}^{k} w_i \bar{x}_i + \sum_{i=1}^{k} w_i (y_i - \bar{x}_i)$, however at least $k - t$ of the terms in the second sum are 0 since $y \in \mathcal{B}_0^t(\bar{x})$. Recall that $w_i(y_i - \bar{x}_i) \geq d_i^-$, for $y_i \in [a_i, b_i]$.

Since at most t entries are not 0, the second sum is lower bounded by the sum of the t lowest d_i^-. Thus, we sort them in a non-decreasing order, $d_{l_1}^- \leq d_{l_2}^- \leq \cdots \leq d_{l_k}^-$. The minimum is obtained by y where y_{l_i} is: (1) a_{l_i} if $i \in [t]$ and $w_{l_i} \geq 0$, (2) b_{l_i} if $i \in [t]$ and $w_{l_i} < 0$, and (3) $\bar{x}_{l_i}$ if $i \notin [t]$. The lower bound is:

$$l = \min_{y \in \mathcal{B}_0^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = \left(\sum_{i=1}^{k} w_i \bar{x}_i \right) + (d_{l_1}^- + \cdots + d_{l_t}^-)$$

Similarly, the upper bound depends on the t highest d_i^+:

$$u = \max_{y \in \mathcal{B}_0^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = \left(\sum_{i=1}^{k} w_i \bar{x}_i \right) + (d_{u_1}^+ + \cdots + d_{u_t}^+)$$

for the non-increasing series $d_{u_1}^+ \geq d_{u_2}^+ \geq \cdots \geq d_{u_k}^+$ of d_i^+.

Example The intervals marked by $\mathcal{B}_0^t$ in Figure 3 show the top-t bound propagation. For n_1, we showed $d_1^- = -1.4$, $d_2^- = -3$, $d_3^- = -11.55$. Thus, its $t = 2$ lowest values are d_2^- and d_3^- and its lower bound is: $\min_{y \in \mathcal{B}_0^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = 3.95 - 3 - 11.55 = -10.6$. In this example, $o_1 \in [0.05, 31.15]$, thus top-t successfully proves $o_1 \geq 0$.

Observation 1. *The top-t bound propagation computes the minimum and maximum over $\mathcal{B}_0^t(\bar{x})$ and over its convex hull.*

This follows since the minimum and maximum of a linear function over a compact set (in our setting, $\mathcal{B}_0^t(\bar{x})$) are equal to those over its convex hull. Namely, linear bound propagation over a non-convex perturbation space or over its convex hull are the same. Thus, the top-t bound propagation over $\mathcal{B}_0^t(\bar{x})$ coincides with the same propagation over $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$.

Bound Propagation for $\widetilde{\mathcal{B}}_1^t(\bar{x})$

We next define the *t-times-top* bound propagation over a $\widetilde{\mathcal{B}}_1^t(\bar{x})$ input subspace. The goal is to contrast it with the top-t bound propagation, since Theorem 2 shows that the volumes of $\widetilde{\mathcal{B}}_1^t(\bar{x})$ and $\mathcal{D} \cap \widetilde{\mathcal{B}}_1^t(\bar{x})$ are almost the same, for large k.

As before, we write $\sum_{i=1}^{k} w_i \bar{x}_i + \sum_{i=1}^{k} w_i (y_i - \bar{x}_i)$, for $y \in \widetilde{\mathcal{B}}_1^t(\bar{x})$, and define its minimum. By definition of $\delta_{\bar{x}}^i(y)$:

$$w_i(y_i - \bar{x}_i) = \underbrace{w_i(\mathbf{1}_{\{y_i > \bar{x}_i\}}(b_i - \bar{x}_i) + \mathbf{1}_{\{y_i < \bar{x}_i\}}(a_i - \bar{x}_i))}_{\geq d_i^-} \delta_{\bar{x}}^i(y)$$

Thus, $\sum_{i=1}^{k} w_i(y_i - \bar{x}_i) \geq \sum_{i=1}^{k} d_i^- \delta_{\bar{x}}^i(y)$. Since $\delta_{\bar{x}}^i(y) \geq 0$, $\sum_{i=1}^{k} d_i^- \delta_{\bar{x}}^i(y) \geq d_{l_1}^- \sum_{i=1}^{k} \delta_{\bar{x}}^i(y)$, for $d_{l_1}^- = \min_{i \in [k]} d_i^-$. Since $d_{l_1}^- \leq 0$ and $y \in \widetilde{\mathcal{B}}_1^t(\bar{x})$: $d_{l_1}^- \sum_{i=1}^{k} \delta_{\bar{x}}^i(y) \geq d_{l_1}^- \cdot t$. This minimum is obtained by y where $y_{l_i} = \bar{x}_{l_i}$, for $i \neq 1$, and $y_{l_1} = \bar{x}_{l_1} + t \cdot (c - \bar{x}_{l_1})$, for $c = a_{l_1}$ if $w_{l_1} \geq 0$, and $c = b_{l_1}$ otherwise. Thus, the lower bound depends on the product of t and the lowest d_i^- and is equal to:

$$l = \min_{y \in \widetilde{\mathcal{B}}_1^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = \left(\sum_{i=1}^{k} w_i \bar{x}_i \right) + t \cdot d_{l_1}^-$$

Similarly, the upper bound depends on the product of t and the highest d_i^+:

$$u = \max_{y \in \widetilde{\mathcal{B}}_1^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = \left(\sum_{i=1}^{k} w_i \bar{x}_i \right) + t \cdot d_{u_1}^+$$

Example The intervals marked by $\widetilde{\mathcal{B}}_1^t$ in Figure 3 show the t-times-top bound propagation. For n_1, the minimum of $d_1^- = -1.4$, $d_2^- = -3$, $d_3^- = -11.55$ is d_3^-. Thus, its lower bound is $\min_{y \in \widetilde{\mathcal{B}}_1^t(\bar{x})} \sum_{i=1}^{k} w_i y_i = 3.95 - 2 \cdot 11.55 = -19.15$. This bound propagation cannot prove $o_1 \geq 0$.

Multi-Channel Inputs

Our formulation of the bound propagations as functions of the input entry contributions d_i^- and d_i^+ naturally extends to multi-channel inputs. In this setting, given $i \in [k]$, d_i^- and d_i^+ sum over all channels' contributions. Given $\bar{x} \in \mathcal{D}$ and a linear function $y \mapsto \sum_{i=1}^{k} \sum_{j=1}^{d} w_i^{(j)} y_i^{(j)}$, we define $d_i^- = \sum_{j=1}^{d} \min(w_i^{(j)}(b_i^{(j)} - \bar{x}_i^{(j)}), w_i^{(j)}(a_i^{(j)} - \bar{x}_i^{(j)}))$ and $d_i^+ = \sum_{j=1}^{d} \max(w_i^{(j)}(b_i^{(j)} - \bar{x}_i^{(j)}), w_i^{(j)}(a_i^{(j)} - \bar{x}_i^{(j)}))$. The lower and upper bounds of all bound propagations are the same as for single-channel inputs, except that the first sum is $\sum_{i=1}^{k} \sum_{j=1}^{d} w_i^{(j)} \bar{x}_i^{(j)}$.

Comparison

Our formulation explains why a bound propagation over the ℓ_0-ball is tighter than over our ℓ_1-like polytope or $\mathcal{D}$. All bound propagations have a shared term (i.e., $\sum_{i=1}^{k} w_i \bar{x}_i$, for single-channel inputs) and they differ in the term they add: for the lower bound, top-t adds the sum of the t lowest minimum input entry contributions, while t-times-top adds the product of t and the lowest minimum contribution, and the box bound propagation adds the sum of all minimum contributions (and similarly for the upper bound). The t-times-top and box bound propagations are incomparable for $t > 1$, because t times the top-1 value can be smaller or greater than the sum of all values. For example, in Figure 3, the box bound propagation has a tighter lower bound of n_1 than t-times-top, but a looser upper bound. The theoretical time complexity of our bound propagations is linear in k, for single-channel inputs, or $k \cdot d$, for multi-channel inputs, like the common box bound propagation. In practice, the bound propagations run on a GPU, which affects the complexity (described in the appendix).

Evaluation

In this section, we evaluate the top-t bound propagation. We empirically show that top-t is more precise than the other bound propagations for ℓ_0-balls and that it significantly boosts CoVerD, the state-of-the-art ℓ_0 robustness verifier.

Setup We extended GPUPoly (Müller et al. 2021), which CoVerD calls a large number of times (sometimes 10^6). GPUPoly verifies using box bound propagation with the linear relaxation of Singh et al. (2019). We extended it with

Dataset	Network	# CC	# R	top-t-GP
MNIST	6×200_PGD	99	97	94
	ConvSmall	98	95	36
	ConvSmallPGD	94	93	57
	ConvMedPGD	99	94	43
	ConvBig	97	94	0
Fashion	ConvSmallPGD	83	75	0
	ConvMedPGD	86	83	0
CIFAR-10	ConvSmallPGD	70	61	0

Table 1: The number of verified ℓ_0-balls for $t = 1$.

our bound propagations: top-t (called top-t-GP) and t-times-top (t-times-top-GP). The implementation is described in the appendix, which also shows that the running time of top-t-GP is similar to GPUPoly. We ran experiments on a dual AMD EPYC 7713 server with 2TB RAM and eight A100 GPUs. We consider the same image classifiers as CoVerD, for MNIST (LeCun et al. 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and CIFAR-10 (Krizhevsky 2009). MNIST and Fashion-MNIST consist of single-channel images (grayscale) and CIFAR-10 consists of three-channel images (RGB). The network architectures are provided in ERAN (ETH SRI Lab 2019).

Bound propagation only GPUPoly cannot directly prove robustness for ℓ_0-balls, since it runs box bound propagation and thus requires to overapproximate an ℓ_0-ball with its bounding box. However, the bounding box is $\mathcal{D}_v$ and networks are not robust in $\mathcal{D}_v$ since not all inputs are classified the same. Although top-t-GP improves GPUPoly's precision for ℓ_0-balls, we next show that it is still typically not precise enough to prove robustness in an ℓ_0-ball where all pixels can be perturbed (i.e., $\mathcal{K} = [v]$). We show that even for $t = 1$ (the smallest ℓ_0-ball), it often loses too much precision to prove robustness. In this experiment, for each network, we consider the first 100 images from its test set. For each image, we check whether the network classifies it correctly. If so, we run top-t-GP for its ℓ_0-ball with $t = 1$. Table 1 shows the number of images that are correctly classified (**# CC**), the number of robust ℓ_0-balls (**# R**), and the number of ℓ_0-balls that top-t-GP verifies. We obtain the number of robust ℓ_0-balls from Calzone (Shapira, Avneri, and Drachsler-Cohen 2023), a complete (exact) verifier for ℓ_0-balls. The table shows that except for MNIST classifiers, top-t-GP cannot verify even a single ℓ_0-ball. These results imply that top-t-GP cannot, on its own, prove robustness in ℓ_0-balls with $\mathcal{K} = [v]$. However, we later show that it boosts CoVerD, which does prove robustness in ℓ_0-balls with $\mathcal{K} = [v]$. CoVerD decomposes the verification task into multiple robustness verification tasks over ℓ_0-balls defined over subsets of pixels ($\mathcal{K} \subseteq [v]$) whose size is at most 200 (a more detailed description is provided later). The next experiment explains why top-t-GP boosts CoVerD.

Bound propagation comparison We next show that top-t-GP is more precise than the other bound propagations for verifying robustness in ℓ_0-balls over subsets of pixels ($|\mathcal{K}| < v$). We consider three networks, three respective images, $k \in [200]$, and $t \in [6]$. For each network, image, and k, we randomly sample 500 subsets $S \subseteq [v]$ of size k and let each bound propagation verify robustness in the neighborhood $\mathcal{B}_0^t(\bar{x})$ where $\mathcal{K} = S$, for each t. Figure 4 shows the success rate (the ratio of $\mathcal{B}_0^t(\bar{x})$ proven robust out of all 500 analyzed) as a function of k, for each approach and each t. The results show that top-t-GP's success rate is never lower than GPUPoly (with the box bound propagation) and often improves it, especially for small t or large k. It is also significantly better than t-times-top-GP for $t > 1$, despite the negligible volume difference shown in Theorem 2. In fact, for $t > 4$, even the box bound propagation is better than t-times-top-GP, although the volume of $\widetilde{\mathcal{B}}_1^t(\bar{x})$ is significantly smaller than $\mathcal{D}$ (for large k). This shows that bound propagation highly depends on the shape of the property's input subspace and not only on its volume.

Boosting ℓ_0 certification Lastly, we show that top-t-GP boosts the performance of CoVerD (Shapira et al. 2024), which verifies robustness in ℓ_0-balls where all pixels can be perturbed ($\mathcal{K} = [v]$). CoVerD is a sound and complete local robustness verifier for ℓ_0-balls. It is the state-of-the-art, improving Calzone (Shapira, Avneri, and Drachsler-Cohen 2023), which in turn outperforms a MILP-based verification. A naive verifier would submit $\binom{v}{t}$ box neighborhoods to a verifier (e.g., GPUPoly) and determine robustness if all are robust. However, verifying $\binom{v}{t}$ neighborhoods is infeasible for $t > 2$. Instead, CoVerD defines sets $S_1, \ldots, S_R$ of size k that cover all subsets of t pixels over $[v]$. For each S_r, it submits the box neighborhood $\{y \in \mathcal{D}_v \mid \forall i \notin S_r : y_i = \bar{x}_i\}$ to GPUPoly. If it is robust, CoVerD continues to S_{r+1}; otherwise, it refines S_r to smaller subsets and submits their neighborhoods to GPUPoly. Generally, the larger the sets S_r whose neighborhoods are proven robust, the fewer calls to GPUPoly. With top-t-GP, CoVerD can submit $\mathcal{B}_0^t(\bar{x})$ where $\mathcal{K} = S_r$. Also, since top-t-GP has higher success rate than GPUPoly for large k (Figure 4), CoVerD can verify significantly larger sets S_r, which boosts its performance.

We evaluate on the most challenging benchmarks of CoVerD: for each network, we focus on the largest evaluated t. The appendix includes experiments over less challenging benchmarks of CoVerD. We ran the latest version of CoVerD, with the original hyper-parameters, except that we ran with five MILP instances. We used eight GPUs, as in CoVerD's experiments (the running time decreases roughly linearly with the number of GPUs). The timeout of each verifier is five hours for each ℓ_0-ball. Since CoVerD relies on random choices, we repeat each experiment three times with different seeds to show that our speedup is independent of the choice of seed. Figure 5 reports the average execution time and standard deviation. Each plot corresponds to a network. A pair of bars corresponds to an ℓ_0-ball where each bar shows the execution time in minutes (the y-axis), for CoVerD and ours (CoVerD + top-t-GP). The horizontal line marks the timeout. Bars reaching this line in-

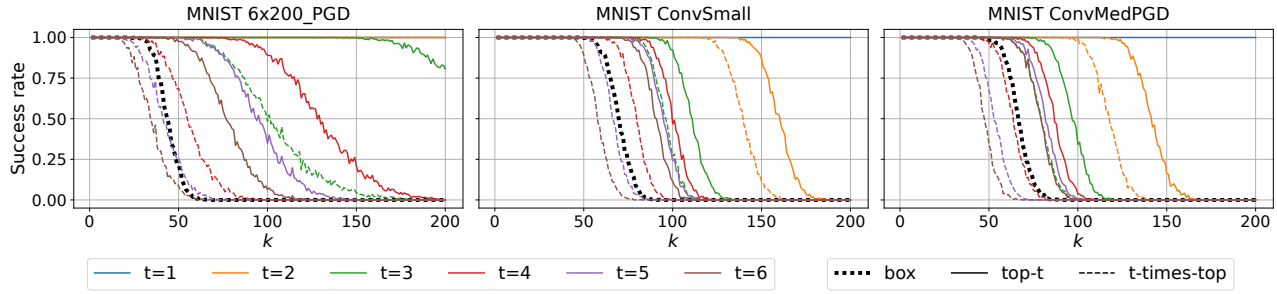


Figure 4: The success rate of the three bound propagations over $\mathcal{B}_0^t(\bar{x})$ for different choices of $\mathcal{K} \subseteq [v]$, as a function of $k = |\mathcal{K}|$.

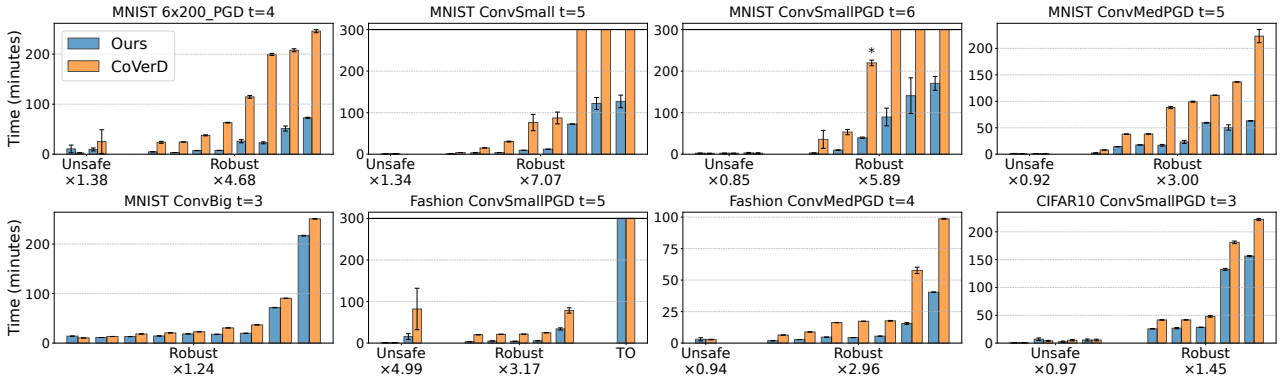


Figure 5: CoVerD + `top-`t`-GP` compared to CoVerD on its most challenging benchmarks.

dicate timeouts, in all three runs. For a single ℓ_0-ball, one run of CoVerD out of three reached the timeout (denoted by *) and is excluded from its average and standard deviation. The x-axis begins with the *unsafe* ℓ_0-balls (non-robust), if they exist, followed by the *robust* ℓ_0-balls. One plot also has a timeout instance (*TO*). Within each group, the ℓ_0-balls are sorted by the execution time of CoVerD to better visualize the time differences. Below each group, we show the ratio between the sums of the average execution times of CoVerD and ours, excluding ℓ_0-balls where CoVerD timed out in all three runs. The results show that `top-`t`-GP` boosts CoVerD's performance on its robustness benchmarks by 1.24x-7.07x, with a geometric mean of 3.16. It is sometimes slower than CoVerD, but only when both verifiers terminate within minutes (as opposed to hours in the more challenging ℓ_0-balls). In the appendix, we demonstrate that CoVerD + `t-times-top-GP` is substantially slower than our approach, consistent with the results in Figure 4.

Related Work

Many verifiers analyze the local robustness of neural networks over convex perturbation spaces, e.g., ℓ_p-balls for $p \geq 1$ (Tjeng, Xiao, and Tedrake 2019; Zhang et al. 2018; Müller et al. 2021; Gehr et al. 2018; Katz et al. 2017; Leino, Wang, and Fredrikson 2021; Huang et al. 2021; Behl et al. 2021; Wu and Zhang 2021), where many rely on bound propagation and convex relaxations to scale. Chiang et al. (2020); Xu et al. (2020) propose bound propagation for ℓ_0-balls, which is similar to ours but is limited to single-channel inputs over $[0, 1]^v$. Our experiments show that this bound propagation often fails to prove robustness even when only a single pixel can be perturbed (i.e., $t = 1$). Behl et al. (2021) propose a tight bound propagation for simplex input spaces. Though it can precisely capture our asymmetrically scaled ℓ_1-ball, it currently does not support the intersection with a box, required to precisely capture the convex hull. Jia et al. (2019) verify robustness to word substitutions using interval bound propagation (IBP), where $t = v$. Huang et al. (2019) propose IBP for $t < v$. They overapproximate the convex hull of the ℓ_0-ball with a simplex and say that manipulating their convex hull is infeasible (whereas we explicitly characterize ours). For a similar setting, Xu et al. (2020) verify using dynamic programming. Chiang et al. (2020); Levine and Feizi (2020) propose certified defenses against patch attacks, where the attacker perturbs a subset of pixels within a contiguous region. In contrast, we focus on few-pixel attacks, where the attacker can perturb any subset of pixels, and the allowed number of perturbed pixels is typically smaller than that in patch attacks.

Conclusion

We show that the convex hull of an ℓ_0-ball is the intersection of its bounding box and an ℓ_1-like polytope. We present a bound propagation over an ℓ_0-ball, tighter than bound propagations over its bounding box or the ℓ_1-like polytope. It boosts the analysis of a neural network robustness verifier for ℓ_0-balls by 1.24x-7.07x, with a geometric mean of 3.16, on its most challenging robustness benchmarks.

References

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.-J.; Srivastava, M.; and Chang, K.-W. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2890–2896. Association for Computational Linguistics.

Baninajjar, A.; Rezine, A.; and Aminifar, A. 2024. VNN: Verification-Friendly Neural Networks with Hard Robustness Guarantees. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2846–2856. PMLR.

Bastani, O.; Ioannou, Y.; Lampropoulos, L.; Vytiniotis, D.; Nori, A. V.; and Criminisi, A. 2016. Measuring Neural Net Robustness with Constraints. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2613–2621.

Behl, H. S.; Kumar, M. P.; Torr, P.; and Dvijotham, K. 2021. Overcoming the Convex Barrier for Simplex Inputs. In *Advances in Neural Information Processing Systems*, volume 34, 4871–4882. Curran Associates, Inc.

Chiang, P.; Ni, R.; Abdelkader, A.; Zhu, C.; Studer, C.; and Goldstein, T. 2020. Certified Defenses for Adversarial Patches. In *International Conference on Learning Representations*.

Croce, F.; Andriushchenko, M.; Singh, N. D.; Flammarion, N.; and Hein, M. 2022. Sparse-RS: A Versatile Framework for Query-Efficient Sparse Black-Box Adversarial Attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6): 6437–6445.

ETH SRI Lab. 2019. ERAN: ETH Robustness Analyzer for Neural Networks. https://github.com/eth-sri/eran.

Gehr, T.; Mirman, M.; Drachsler-Cohen, D.; Tsankov, P.; Chaudhuri, S.; and Vechev, M. T. 2018. AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation. In *IEEE Symposium on Security and Privacy, SP*.

Hall, P. 1927. The Distribution of Means for Samples of Size N Drawn from a Population in which the Variate Takes Values Between 0 and 1, All Such Values Being Equally Probable. *Biometrika*, 19(3/4): 240–245.

Hawkins, A. J. 2025. Waymo is still good at avoiding serious distraction and death after 56.7 million miles. https://www.theverge.com/news/658952/waymo-injury-prevention-human-benchmark-study. Accessed: 2025-06-24.

Huang, P.-S.; Stanforth, R.; Welbl, J.; Dyer, C.; Yogatama, D.; Gowal, S.; Dvijotham, K.; and Kohli, P. 2019. Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4083–4093. Association for Computational Linguistics.

Huang, Y.; Zhang, H.; Shi, Y.; Kolter, J. Z.; and Anandkumar, A. 2021. Training Certifiably Robust Neural Networks with Efficient Local Lipschitz Bounds. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems NeurIPS*.

Irwin, J. O. 1927. On the Frequency Distribution of the Means of Samples from a Population Having any Law of Frequency with Finite Moments, with Special Reference to Pearson's Type II. *Biometrika*, 19(3/4): 225–239.

Jia, R.; Raghunathan, A.; Göksel, K.; and Liang, P. 2019. Certified Robustness to Adversarial Word Substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4129–4142. Association for Computational Linguistics.

Katz, G.; Barrett, C. W.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Computer Aided Verification - 29th International Conference, CAV*.

Kouvaros, P.; Brückner, B.; Henriksen, P.; and Lomuscio, A. 2025. Dynamic Back-Substitution in Bound-Propagation-Based Neural Network Verification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26): 27383–27391.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11): 2278–2324.

Leino, K.; Wang, Z.; and Fredrikson, M. 2021. Globally-Robust Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*. PMLR.

Levine, A.; and Feizi, S. 2020. (De)Randomized Smoothing for Certifiable Defense against Patch Attacks. In *Advances in Neural Information Processing Systems*, volume 33, 6465–6475. Curran Associates, Inc.

Marshall, A. 2023. Amazon's AI-Powered Van Inspections Give It a Powerful New Data Feed. https://www.wired.com/story/amazons-ai-van-inspections-powerful-data-feed. Describes deployment of AI-powered vehicle inspections (UVeye AVI) across Amazon fleet centers.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Miller, K. 2023. Could Self-Supervised Learning Be a Game-Changer for Medical Image Classification? https://hai.stanford.edu/news/could-self-supervised-learning-be-game-changer-medical-image-classification. Accessed: 2025-06-24.

Müller, C.; Serre, F.; Singh, G.; Püschel, M.; and Vechev, M. T. 2021. Scaling Polyhedral Neural Network Verification on GPUs. In *Proceedings of Machine Learning and Systems MLSys*.

Rockafellar, R. T. 1997. *Convex analysis*, volume 28. Princeton university press.

Shapira, Y.; Avneri, E.; and Drachsler-Cohen, D. 2023. Deep Learning Robustness Verification for Few-Pixel Attacks. *Proc. ACM Program. Lang.*, 7(OOPSLA1).

Shapira, Y.; Wiesel, N.; Shabelman, S.; and Drachsler-Cohen, D. 2024. Boosting Few-Pixel Robustness Verification via Covering Verification Designs. In *Computer Aided Verification - 36th International Conference, CAV 2024*, volume 14682 of *Lecture Notes in Computer Science*, 377–400. Springer.

Singh, G.; Gehr, T.; Püschel, M.; and Vechev, M. T. 2019. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL): 41:1–41:30.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tjeng, V.; Xiao, K. Y.; and Tedrake, R. 2019. Evaluating Robustness of Neural Networks with Mixed Integer Programming. *In 7th International Conference on Learning Representations, ICLR*.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018a. Efficient Formal Safety Analysis of Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018b. Formal Security Analysis of Neural Networks using Symbolic Intervals. In *27th USENIX Security Symposium (USENIX Security 18)*, 1599–1614. USENIX Association. ISBN 978-1-939133-04-5.

Wu, Y.; and Zhang, M. 2021. Tightening Robustness Verification of Convolutional Neural Networks with Fine-Grained Linear Approximation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11674–11681.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747.

Xu, K.; Shi, Z.; Zhang, H.; Wang, Y.; Chang, K.-W.; Huang, M.; Kailkhura, B.; Lin, X.; and Hsieh, C.-J. 2020. Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. In *Advances in Neural Information Processing Systems*, volume 33, 1129–1141. Curran Associates, Inc.

Zass, R.; and Shashua, A. 2006. Nonnegative Sparse PCA. In Schölkopf, B.; Platt, J.; and Hoffman, T., eds., *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Zhang, H.; Weng, T.; Chen, P.; Hsieh, C.; and Daniel, L. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.