Appendix A. Proof of Input Refinement

Theorem A.1. *If we can verify that a set S of perturbed versions of an image x are correctly classified for a threat model using one* ***certification cycle*** *(one pass through the algorithm sharing the same linear relaxation values), then we can verify that every perturbed image in the convex hull of S is also correctly classified, where we take the convex hull in the pixel space.*

Proof. When a set S of perturbed inputs and a neural network f_{NN} are passed into a verifier, it produces A_L, b_L, A_U, b_U such that for all $y \in S$

$$A_L \cdot y + b_L \leq \mathrm{f}_{NN}(y)_j \leq A_U \cdot y + b_U \tag{11}$$

Claim A.2. *We claim that if $y, z \in S$, then $x = \frac{y+z}{2}$ satisfies the above inequality.*

Proof. We can prove this by induction on the layers. For the first layer we see that as matrix multiplication and addition are linear transformations, we have that $x_1 = W_1 \cdot x + b_1$ lies between the points $y_1 = W_1 \cdot y + b_1$ and $z_1 = W_1 \cdot z + b_1$. The important property to note here is that every co-ordinate of x_1 lies in the interval between the co-ordinates of y_1 and z_1. Now, we see that the activation layer is linear relaxed such that $A_L^1 \cdot y + B_L^1 \leq Act(y) \leq A_U^1 \cdot y + B_U^1$ for all values of y between the upper and lower bound for a neuron. As we proved before every pixel of x lies within the bounds and hence satisfies the relation.

For the inductive case, we see that given that x satisfies this relation up till layer l, then we have that

$$A_L^l \cdot x + b_L^l \leq \mathrm{f}_{NN}^l(x)_j \leq A_U^l \cdot x + b_U^l \tag{12}$$

where $\mathrm{f}_{NN}^l(x)_j$ gives the output of the j^{th} neuron in layer l post-activation.

Now, we see that as we satisfy the above equation, the certification procedures ensure that the newly computed pre-activation values satisfy the same condition. So, we have

$$A_L^{l+1/2} \cdot x + b_L^{l+1/2} \leq \mathrm{f}_{NN}^{l+1/2}(x)_j \leq A_U^{l+1/2} \cdot x + b_U^{l+1/2}$$

where we use $l + 1/2$ to denote the fact that this is a pre-activation bound. Now, if we can show that our value lies within the range of the output of every neuron, then we prove the inductive case. But then we see that as these $A_L^{l+1/2} \cdot x + b_L^{l+1/2}$ is a linear transform $x_{l+1/2} = A_L^{l+1/2} \cdot x + b_L^{l+1/2}$ lies between the points $y_{l+1/2} = A_L^{l+1/2} \cdot y + b_L^{l+1/2}$, $z_{l+1/2} = A_L^{l+1/2} \cdot z + b_L^{l+1/2}$. So, we see that the values taken by this is lower bounded by the corresponding value taken by at least one of the points in S. Similarly we can prove it for the upper bound. Then, we can use the fact that the linear relaxation gives valid bounds for every values within the upper and lower bound to complete the proof. So, we have that

$$A_L^{l+1} \cdot x + b_L^{l+1} \leq \mathrm{f}_{NN}^{l+1}(x)_j \leq A_U^{l+1} \cdot x + b_U^{l+1} \tag{13}$$

□

Then we see that the verifier only certifies the set S to be correctly classified if for all $y \in S$

$$(A_j^U \cdot y + b_j^U) \leq (A_c^L \cdot y + b_c^L)$$

Now, we see that from the equation above that if $z \in conv(S)$, then we have that $z = \sum_{i=1}^n a_i x_i$, where $x_i \in S$ and $\sum_{i=1}^n a_i = 1, a_i \geq 0$. Then using the above claim we see that

$$\begin{aligned}
(\mathrm{f}_{NN}(z))_j &\leq (A_j^U \cdot z + b_j^U) \\
&= (A_j^U \cdot \sum_{i=1}^n (a_i x_i) + b_j^U) \\
&= \sum_{i=1}^n a_i (A_j^U \cdot x_i + b_j^U) \\
&\leq \sum_{i=1}^n a_i (A_c^L \cdot x_i + b_c^L) \\
&= (A_c^L \cdot \sum_{i=1}^n (a_i x_i) + b_c^L) \\
&= (A_c^L \cdot z + b_c^L) \\
&\leq (\mathrm{f}_{NN}(z))_c
\end{aligned}$$

□

Remark A.3. *For some non-convex attack spaces embedded in high-dimensional pixel spaces, the convex hull of the attack space associated with an image can contain images belonging to a different class (an example of rotation is illustrated in Figure 3). Thus, one cannot certify large intervals of perturbations using a single certification cycle of linear relaxation based verifiers.*

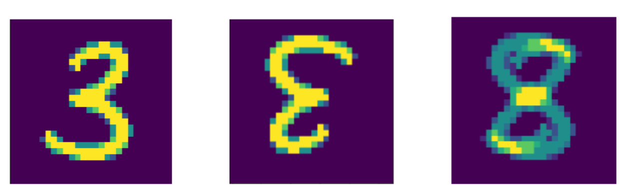


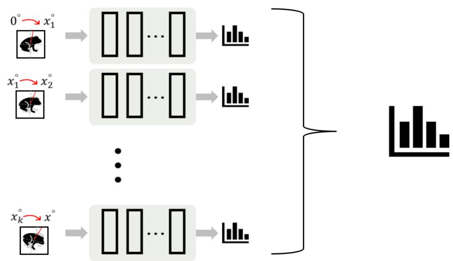
Figure 3: Convex Hull in the Pixel Space

Proof for Figure 3. Consider the images given in Figure 3, denote them as x_1, x_2, x_3 and $x_3 = \frac{x_1+x_2}{2}$ by construction.

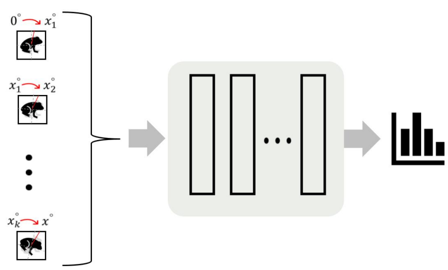
We can observe that for an ideal neural network f, we expect that f classifies x_1, x_2 as 3 and classifies x_3 as 8. Now, we claim that for this network f, it is not possible for a linear-relaxation based verifier to verify that both x_1, x_2 are classified as 3 using just one certification cycle. If it could, then we have by Theorem A.1 that we would be able to verify it for the point $x_3 = \frac{x_1+x_2}{2}$. However, we see that this is not possible as f classifies x_3 as 8. Therefore, we need the verification for x_1 and for x_2 to belong to different certification cycles making input-splitting necessary.

$\square$

Appendix B. Input Space Splitting



(a) Explicit Splitting



(b) Implicit Splitting

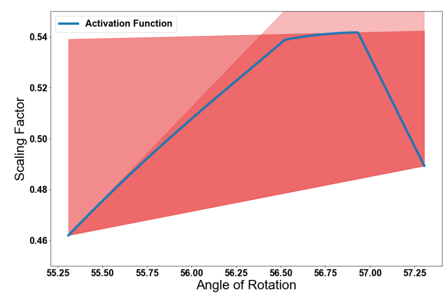
Figure 4: Illustration of refinement techniques.

Figure 4 illustrates the difference between explicit and implicit input space splitting. In Figure 5a, we give the form of the activation function for rotation. Even in a small range of roation angle θ (2°), we see that the function is quite non-linear resulting in very loose linear bounds. Splitting the images explicitly into 5 parts and running them separately (i.e. explicit splitting as shown in Figure 5b) gives us a much tighter approximation. However, explicit splitting results in a high computation time as the time scales linearly with the number of splits. In order to efficiently approximate this function we can instead make the splits to get explicit bounds on each sub-interval and then run them through certification simultaneously (i.e. implicit splitting as shown in Figure 5c). As we observe in Figure 5c, splitting into 20 implicit parts gives a very good approximation with very little overhead (number of certification cycles used are still the same).

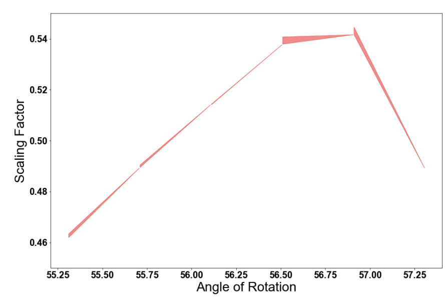
Table 5 gives a more detailed overview of the effect of implicit splitting. For a large explicit split interval size, we see that using a lot of implicit splits allows us to certify larger radius. However, we also see a pattern that beyond a point adding more implicit splits does not give better bounds. Using implicit splits still results in a single certification cycle. By theorem A.1 we see certifying this relaxation is a harder problem than certifying all the rotated images. This could explain the reason we are unable to certify big explicit interval even after using a large number of implicit splits.

Table 5: Evaluation of averaged certified bounds for rotation space perturbation on MNIST MLP 3×1024 and 10 images. The results demonstrate the effectiveness of implicit splits.

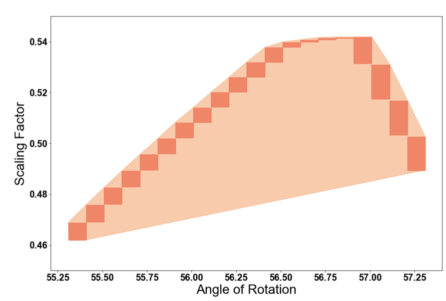
Explicit Split Interval Size	Number of Implicit Splits					
	1	5	8	10	15	20
Experiment (II): Rotations						
0.3	0.27	50.0	50.09	50.12	50.18	50.24
0.5	0.0	40.0	50.0	50.0	50.1	50.2
0.8	0.0	40.0	40.0	40.1	50.0	50.0
1.0	0.0	30.2	40.0	40.0	50.0	50.2
1.2	0.0	10.6	40.0	40.0	40.0	50.0
1.5	0.0	0.0	30.15	40.0	40.0	40.0
2.0	0.0	0.0	0.4	30.0	40.0	40.0
3.0	0.0	0.0	0.0	0.0	0.9	30.0



(a) Without splitting the input range



(b) Explicitly splitting the input (5 divisions)



(c) Implicitly splitting the input (20 divisions)

Figure 5: Bounds for activation function of SP layer in rotation

Appendix C. Additional Experimental Results

Table 6: Additional results of Table 2

Network	Certified Bounds				Ours Improvement (vs Weighted)		Attack
	Naive	Weighted	**SPL**	**SPL + Refine**	w/o refine	w/ refine	Grid
Experiment (I)-A: Hue							
CIFAR, MLP 5×2048	0.00489	0.041	0.370	1.119	8.02x	26.29x	1.449
Experiment (I)-B: Saturation							
CIFAR, MLP 5×2048	0.00286	0.007	0.119	0.325	16.00x	45.42x	0.346
Experiment (I)-C: Lightness							
CIFAR, MLP 5×2048	0.00076	0.001	0.059	0.261	58.00x	260.00x	0.276

Table 7: Additional result of Table 3

Network	Certified Bounds (degrees)				Attack (degrees)
	Number of Implicit Splits			**SPL + Refine**	Grid Attack
	1 implicit No explicit	5 implicit No explicit	10 implicit No explicit	100 implicit + explicit intervals of $0.5°$	
Experiment (II): Rotations					
MNIST, MLP $4\times$ 1024	0.256	0.644	1.129	46.63	48.75
MNIST, MLP $3\times$ 1024	0.486	1.177	1.974	48.47	49.76
MNIST, CNN 4×5	0.437	0.952	1.447	49.20	54.61