A. Full Certified Accuracy Curves

Figure 6 reports the full certified accuracy curves that compare 3DeformRS against a previous certification approach (3DCertify [28]). Note that 3DeformRS achieves comparable-to-better certificates, while also enjoying full certified accuracy curves instead of individual points in these plots.

B. Certified Accuracy Curves for Individual σ values

Every envelope curve we presented is the result of certifying with various σ values and computing the maximum certified accuracy achieved at each certified radius. Here, we elaborate on the results shown in the main paper, and report the curves corresponding to all the σ values we considered. Figures 10–17 report these curves for ModelNet40 and ScanObjectNN. Each curve, corresponding to a particular σ, is reported as a dashed line underneath the envelope.

C. Detailed Analysis per Deformation

Next, we provide a comprehensive and detailed analysis of all the results reported in Section 4.2. The envelope certified accuracy curves are reported in Figures 3 and 4, and Table 2 in the main paper. Figures 10–17 in this supplementary material also detail the certified accuracy curve per deformation, with individual σ values. Here we report analyses of the main observations we draw for each deformation.

Rotations. *Rot Z*. In ModelNet40, PointNet is vastly superior to all other point cloud DNNs against rotations: its ACR is over 80% more than the runner-up (PointNet++). Furthermore, its robust accuracy is, for the most part, maintained across the entire regime we consider (from $-\pi$ to $+\pi$ radians, *i.e.* the entire range of possible rotations). That is, PointNet correctly classifies most objects independently of their rotation around the z axis. In stark contrast, PointNet does *not* show this superiority in ScanObjectNN. As mentioned in the main paper, we attribute this phenomenon almost exclusively to the training augmentations that this version of PointNet enjoyed. Furthermore, we also find that other point cloud DNNs also show some robustness against rotations in the z axis (although to a lesser extent). That is, there is a non-negligible amount of objects that point cloud DNNs can accurately recognize regardless of z rotations. *Rot XZ*. Combining z rotations with x rotations dramatically changes the phenomena we observed with z rotations alone. In particular, PointNet's superiority in ModelNet40 entirely disappears: while its certified accuracy was larger than that of other point cloud DNNs for most z rotations, this is no longer the case in Rot XZ, where PointNet now ranks last with an ACR of 0.31. In turn, CurveNet displays the largest robustness with an ACR of 0.39. PoinNet's drop in ranking, from first to last, remarks the generalization problem of training augmentations. Specifically, we interpret this observation as follows: despite the robustness boost that training augmentations can provide against a target transformation, they may catastrophically fail to generalize to different (yet semantically similar) transformations. Notably, we also observe that CurveNet enjoys a larger performance margin (compared to the other DNNs) in ScanObjectNN than in ModelNet40, suggesting CurveNet's architecture generalizes well to the more realistic objects from ScanObjectNN. *Rot XYZ*. Moreover, introducing another rotation preserves the patterns from Rot XZ: CurveNet shows better performance than other models, while PointNet shows the worse performance of all. Notably, for the most general XYZ rotation, CurveNet's performance is about 30% larger than that of the runner-up (DGCNN). Furthermore, we also note that, for most of the regime we consider, DGCNN is superior to PointNet++ in ModelNet40. However, this order is reversed when experimenting in ScanObjectNN.

Translation. For each dataset, the ranking of models is preserved across the entire range of values we consider: for ModelNet40 the ranking is CurveNet, PointNet++, DGCNN, and PointNet; on the other hand, for ScanObjectNN the ranking is PointNet++, CurveNet, DGCNN and PointNet. That is, we observe that *(i)* PointNet is the worse performer in both cases, and *(ii)* CurveNet is superior to PointNet++ in ModelNet40 (although this order is reversed in ScanObjectNN).

Affine and No-Translation Affine. In ModelNet40, we notice that the affine and affine (NT) transformations do not share exactly the same DNN ranking: while CurveNet and PointNet are the best and worst performers, respectively, the affine (NT) curve suggests DGCNN performs better than PointNet++, while the affine curve suggests the opposite. Additionally, we note that the affine transformation (which *includes* translation), suggests exactly the same DNN ranking as the translation transformation. This observation suggests that the translation transformation is a consequential component in the affine transformation. On the other hand, for ScanObjectNN, we also see that the DNN ranking suggested by the affine and affine (NT) transformations is not exactly the same: both suggest DGCNN and PointNet are the two worst performers (in that order), but affine (NT) suggests CurveNet performs better than PointNet++, while affine suggests the opposite. Finally, analogous to our observation of these transformations on ModelNet40, we notice that the translation and affine transformations suggest the *same* DNN ranking.

Twisting. The twisting transformation we considered is around the z axis. In ModelNet40, PointNet shows remarkable performance superiority (compared to the other DNNs) against twisting. As mentioned in the main paper, we at-

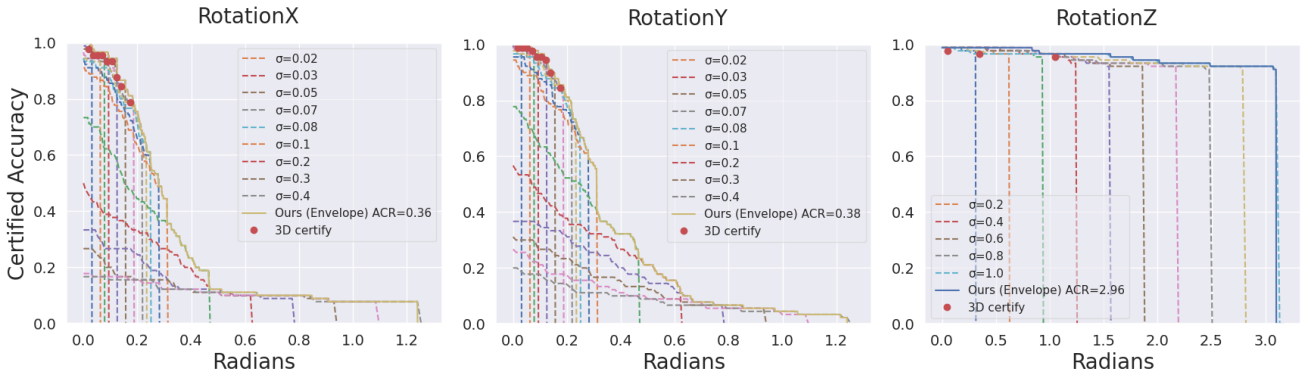

Figure 6. **Certified accuracy against $x-$, $y-$ and $z-$ rotations for PointNet's 64-point version.** The $z-$augmented PointNet maintains a certified accuracy of over 90% for up to 360° rotations.

Name	Deformation Flow	ϕ	Name	Deformation Flow	ϕ
$x-$ Rotation	$\tilde{x} = 0$ $\tilde{y} = (c_\alpha - 1)y - s_\alpha z$ $\tilde{z} = s_\alpha y + (c_\alpha - 1)z$	$[\alpha]$	$xz-$ Rotation	$\tilde{x} = (c_\gamma - 1)x - s_\gamma c_\alpha y + s_\gamma s_\alpha z$ $\tilde{y} = (s_\gamma)x + (c_\gamma c_\alpha - 1)y - c_\gamma s_\alpha z$ $\tilde{z} = s_\alpha y + (c_\alpha - 1)z$	$\begin{bmatrix}\alpha\\ \gamma\end{bmatrix}$
$y-$ Rotation	$\tilde{x} = (c_\beta - 1)x + s_\beta z$ $\tilde{y} = 0$ $\tilde{z} = -s_\beta x + (c_\beta - 1)z$	$[\beta]$	$xyz-$ Rotation	$\tilde{x} = (c_\gamma c_\beta - 1)x + (c_\gamma s_\beta s_\alpha - s_\gamma c_\alpha)y + (c_\gamma s_\beta c_\alpha + s_\gamma s_\alpha)z$ $\tilde{y} = (s_\gamma c_\beta)x + (s_\gamma s_\beta s_\alpha + c_\gamma c_\alpha - 1)y + (s_\gamma s_\beta c_\alpha - c_\gamma s_\alpha)z$ $\tilde{z} = -s_\beta x + c_\beta s_\alpha y + (c_\beta c_\alpha - 1)z$	$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}$

Table 7. **Point flows for rotations along $x-$, $y-$, $xz-$, $xyz-$axes**.

tribute this phenomenon to the z rotation augmentations with which PointNet was trained. Regarding the rest of the DNNs, we observe that CurveNet (analogous to the results on z rotation) also displays low performance for most rotations. This last observation may suggest that ModelNet40 has some idiosyncratic properties that induce noise into the assessment, since, in ScanObjectNN, CurveNet is remarkably superior to other methods for almost all rotation magnitudes.

Tapering. In both ModelNet40 and ScanObjectNN, CurveNet and PointNet are, respectively, the best and worst performers. Remarkably, the slope of the curves, for both datasets (but specially for ScanObjectNN), are rather smooth, showing a slow but steady decrease in performance as the transformation grows stronger. This observation suggests that point cloud DNNs possess somewhat of an inherent robustness against tapering.

Shearing. For the entire regime we consider, PointNet is the worst performer in both datasets. In ModelNet40, the other DNNs, *i.e*. PointNet++, DGCNN and CurveNet, have similar performances for the entire regime. In ScanObjectNN, on the other hand, the performances display larger variation across DNNs.

Gaussian Noise. In ModelNet40, performances drop rapidly as the perturbation's magnitude increases. These drops are, however, consistent across most of the regime we consider. In particular, we notice that DGCNN performs slightly better than PointNet; in turn, PointNet performs better than CurveNet which performs better than PointNet++. The performances of all DNNs change dramatically when testing on ScanObjectNN. Specifically, the performance drops in ScanObjectNN are not as rapid as in ModelNet40. Remarkably, PointNet, which consistently scored the worst in most of the transformations, scores the *best* against Gaussian noise, displaying superior performance against the other DNNs for almost the entirety of the deformation regime. This observation may invite for a thorough study into why, apparently, there is little (or even inverse) correlation between robustness against spatial transformations and per-point perturbations with Gaussian noise.

D. Formulation for other Rotations

Table 7 reports the detailed formulation for the more complex types of rotations we considered in our work, *i.e*. xz-Rotation & xyz-Rotation.

E. Homogeneous Coordinates

Table 8 reports details into how the deformations we considered in our work are modeled in the standard homogeneous-coordinates form.

$x-$Rotation [α]	$z-$Rotation [γ]	$xyz-$Rotation [α,β,γ]	Translation [t_x,t_y,t_z]	Affine [$a,\ldots,l$]
$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} c_\gamma c_\beta & c_\gamma s_\beta s_\alpha - s_\gamma c_\alpha & c_\gamma s_\beta c_\alpha + s_\gamma s_\alpha & 0 \\ s_\gamma c_\beta & s_\gamma s_\beta s_\alpha + c_\gamma c_\alpha & s_\gamma s_\beta c_\alpha - c_\gamma s_\alpha & 0 \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b & c & d \\ e & f+1 & g & h \\ i & j & k+1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$y-$Rotation (β)	$z-$Twisting (γ)	$z-$Tapering (a,b)	$z-$Shearing (a,b)	Affine (NT) [$a,\ldots,k$]
$\begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} c_{\gamma z} & -s_{\gamma z} & 0 & 0 \\ s_{\gamma z} & c_{\gamma z} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2}a^2z+bz+1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}a^2z+bz+1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b & c & 0 \\ d & e+1 & f & 0 \\ g & h & i+1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 8. **Transformation matrices $\mathbf{T} \in \mathbb{R}^{4\times4}$ for common perturbations under homogeneous coordinates.** A point cloud p_i in homogeneous coordinates is transformed into $\tilde{p}_i = \mathbf{T}(\phi)\, p_i$. c_α and s_α correspond to $\sin(\alpha)$ and $\cos(\alpha)$, respectively. "$xyz-$Rotation" corresponds to the extrinsic Euler angles in the specific order: "$x-$Rotation", "$y-$Rotation" then "$z-$Rotation".

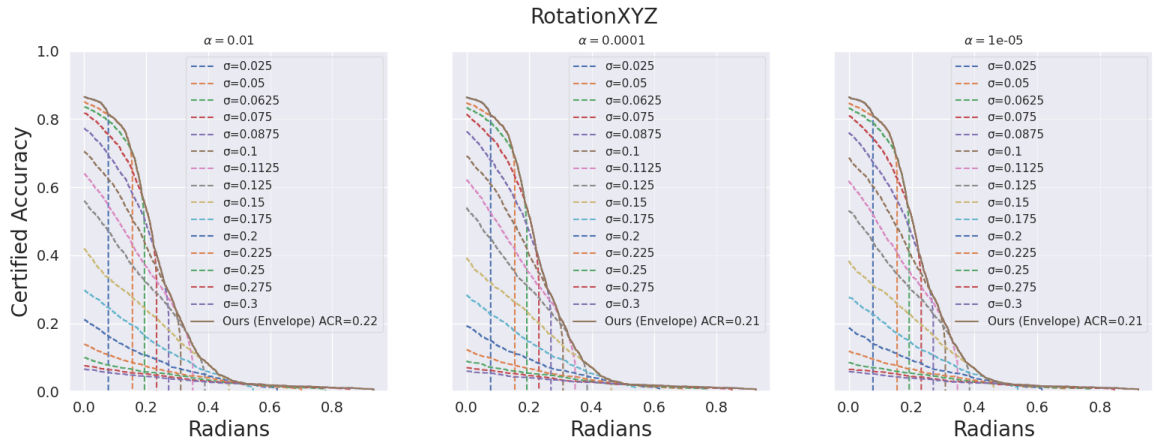

Figure 7. **Certified accuracy curves with failure ratio $\alpha \in [10^{-2}, 10^{-4}, 10^{-5}]$ under xyz-Rotation deformations using 3DeformRS.** Our certification method appears to be insensitive to the failure ratio α.

F. Curves for Failure Ratios α

Figure 7 reports the certified accuracy curves for the failure ratios α we considered in Section 4.5.

G. 3DCertify comparison

G.1. Locally-run experiments

We compared against 3DCertify for Rotations around single axes. Whilst the important z-Rotation results were reported in 3DCertify, we ran local experiments for x-Rotation and y-Rotation using their public implementation (available at `https://github.com/eth-sri/3dcertify`).

Table 9 reports the certified accuracy achieved by 3DCertify. We used 3DCertify's original experimental setup (Taylor relaxation with 2° splits).

Apart from the x-Rotation and y-Rotation experiments, we also computed a single value for z-Rotation (3°) to verify consistency with the larger values reported in 3DCertify (20° and 60°). These results are reported in Figure 6.

In Table 10 we report the runtimes for all experiments. It is worth to notice the proportionality between the runtimes and the deformation parameter: certifying large deformation parameters would take significantly longer. These val-

Degrees	RotX	RotY	RotZ
1	0.98	0.99	–
2	0.96	0.99	–
3	0.96	0.99	0.99
4	0.96	0.98	–
5	0.93	0.96	–
6	0.93	0.96	–
7	0.88	0.94	–
8	0.84	0.90	–
10	0.79	0.84	–
15	0.54	0.67	–
20	–	–	0.97
60	–	–	0.96

Table 9. **Certified Accuracy from 3DCertify.** Taylor relaxation and improved MaxPool layer with DeepPoly for single-axis rotations.

Degrees	RotX	RotY	RotZ
1	13346	14095	–
2	26614	26785	–
3	38960	39247	32832
4	49931	49803	–
5	58679	58699	–
6	66781	66460	–
7	73440	73238	–
8	79223	78798	–
10	89327	88744	–
15	89974	93572	–
20	–	–	–
60	–	–	–

Table 10. **Running Time (seconds) for 3DCertify.** Taylor relaxation and improved MaxPool layer with DeepPoly for single-axis rotations.

ues oscillate in the tenths of thousands of seconds and, beyond 15°, runtimes above a hundred thousand seconds are expected.

G.2. Point Sampling

As seen on Table 5, 3DeformRS outperforms previous certification approaches based on linear relaxations for 3° z-Rotations. Furthermore, Figure 18 shows how 3DeformRS consistently certifies PointNet for all z-Rotations regardless of point cloud cardinality. Note that we used the weights for PointNet provided by the authors, pre-trained with the corresponding point cloud size.

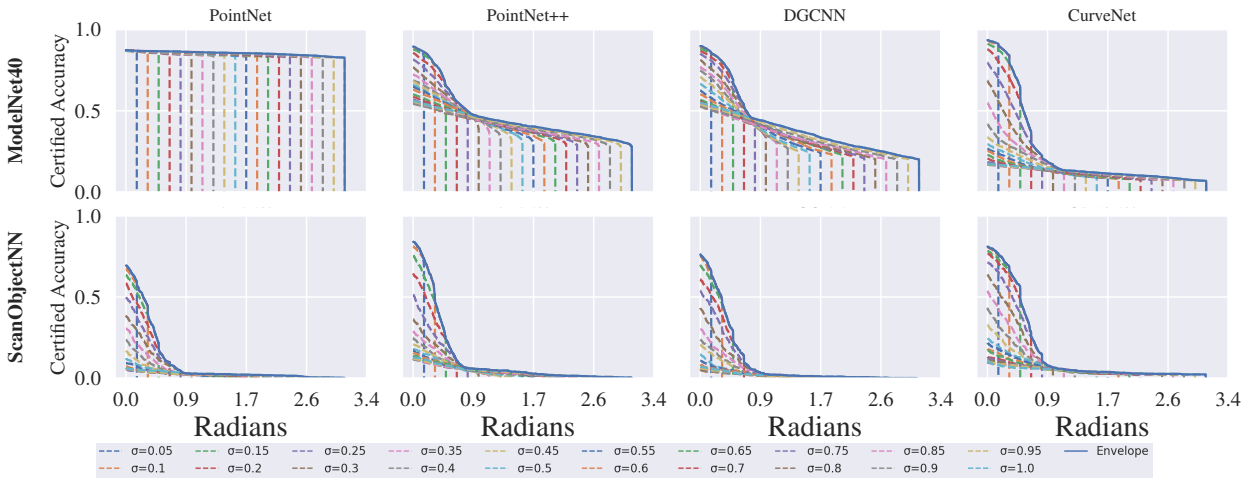

Figure 8. **RotationZ**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

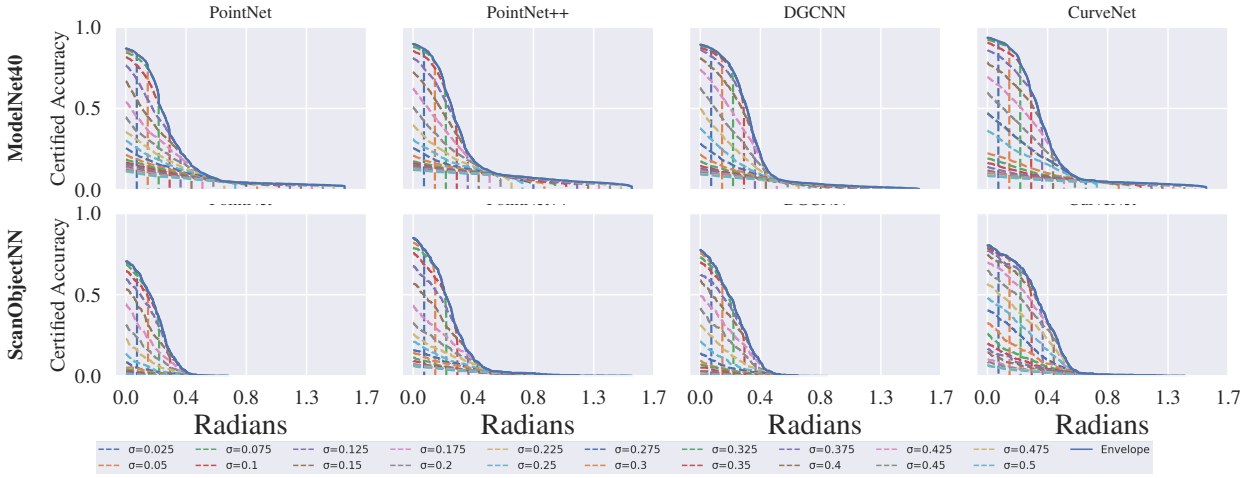

Figure 9. **RotationXZ**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

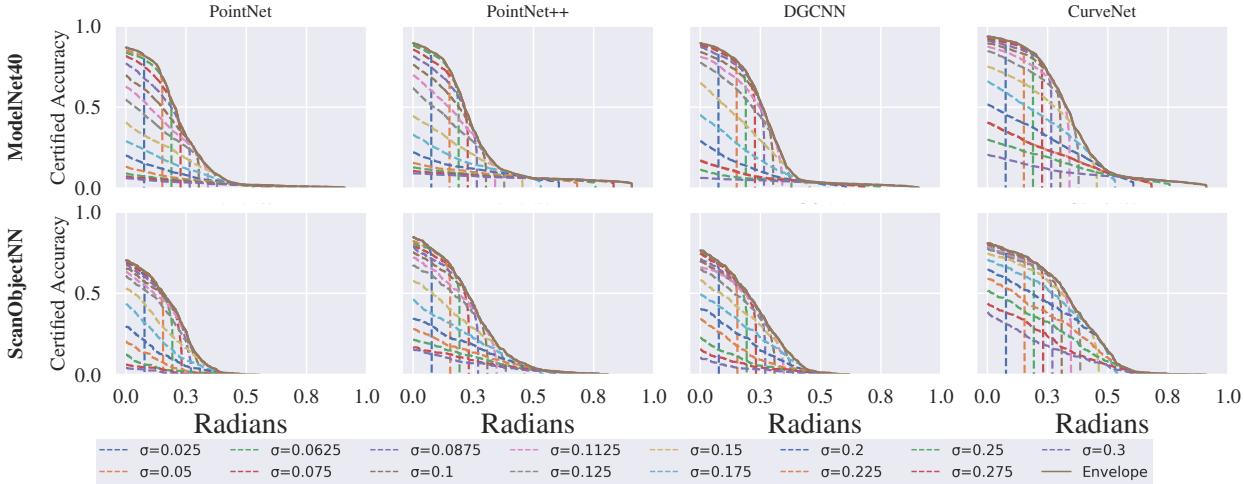

Figure 10. **RotationXYZ**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

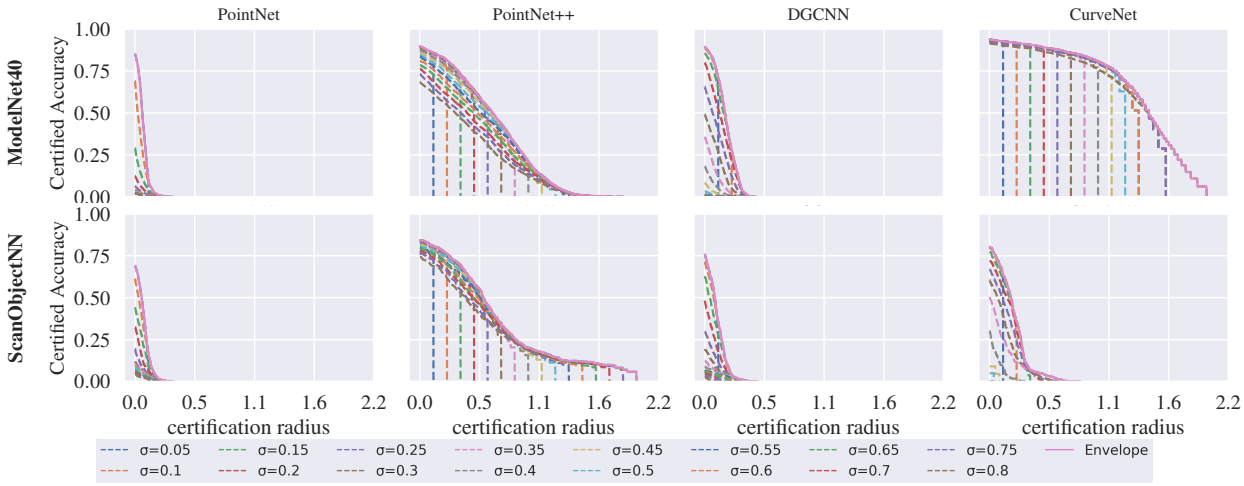

Figure 11. **Translation**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

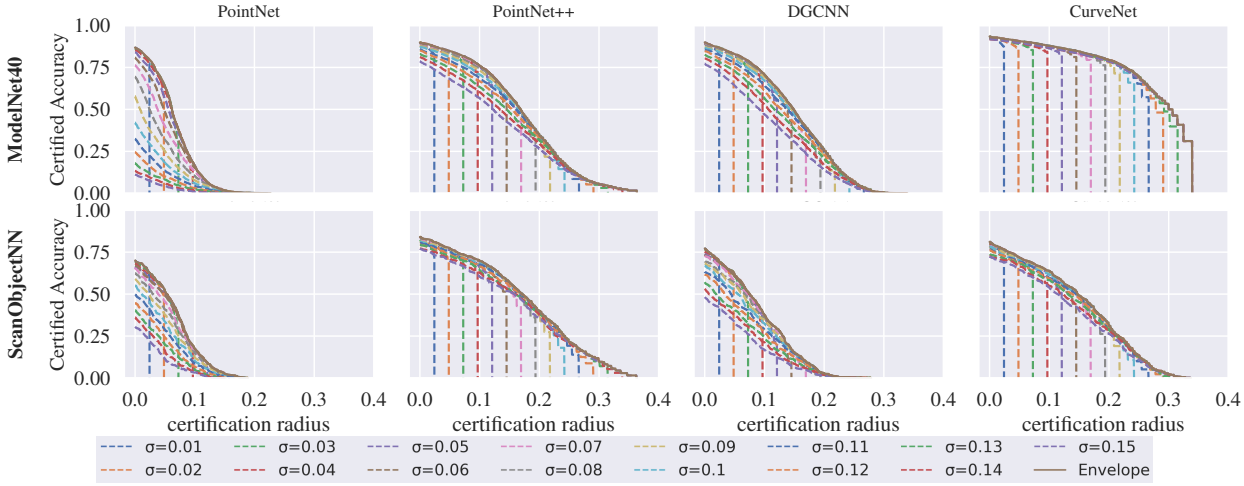

Figure 12. **Affine**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

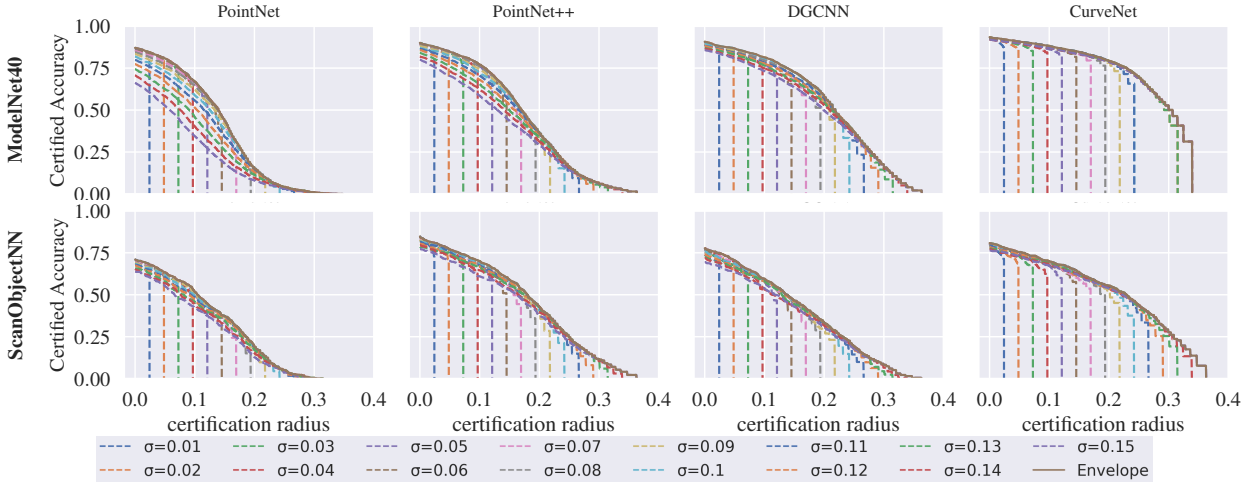

Figure 13. **Affine (NT)**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

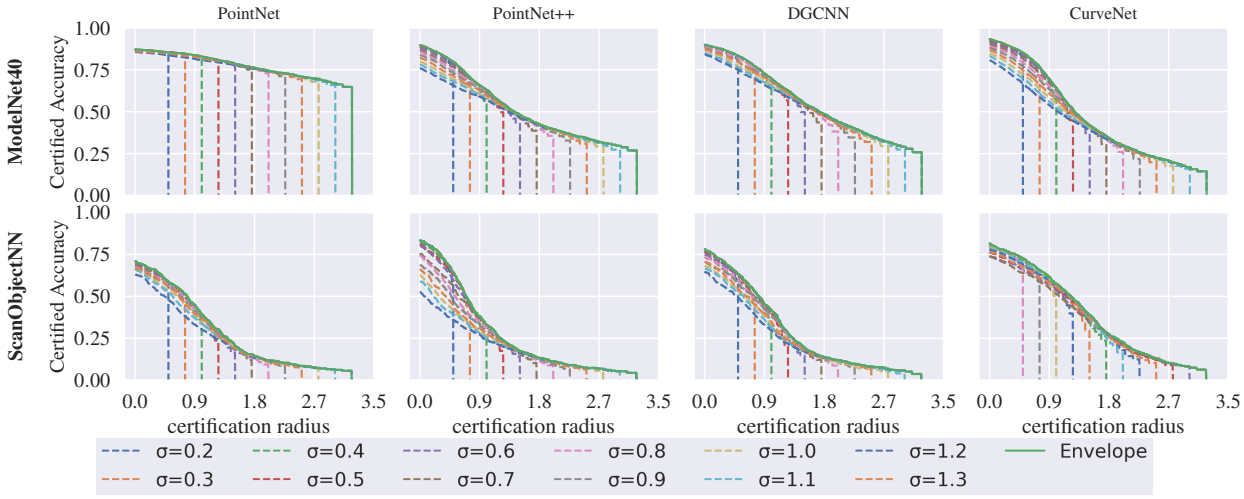

Figure 14. **Twisting**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

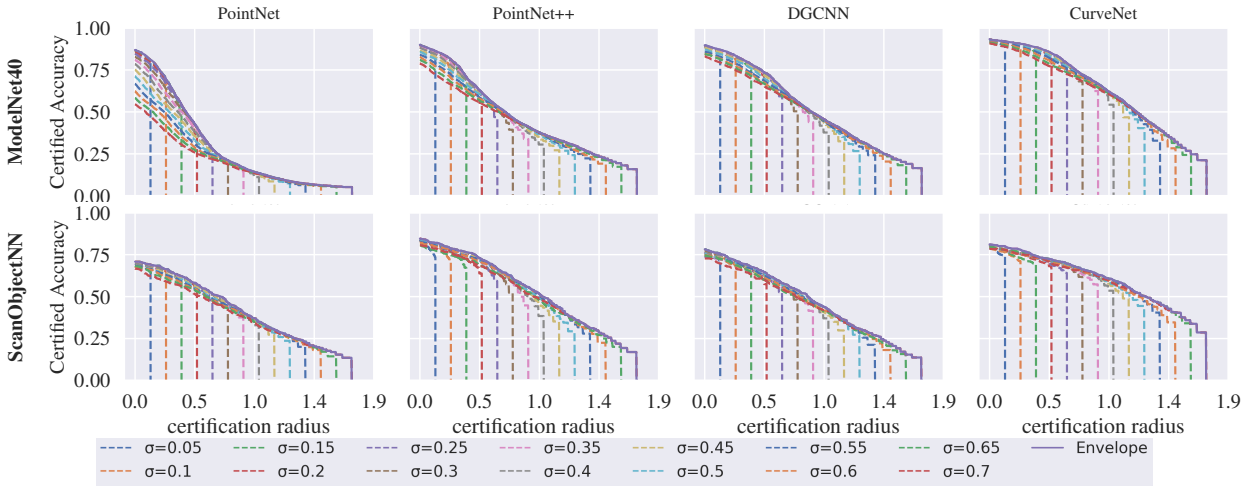

Figure 15. **Tapering**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

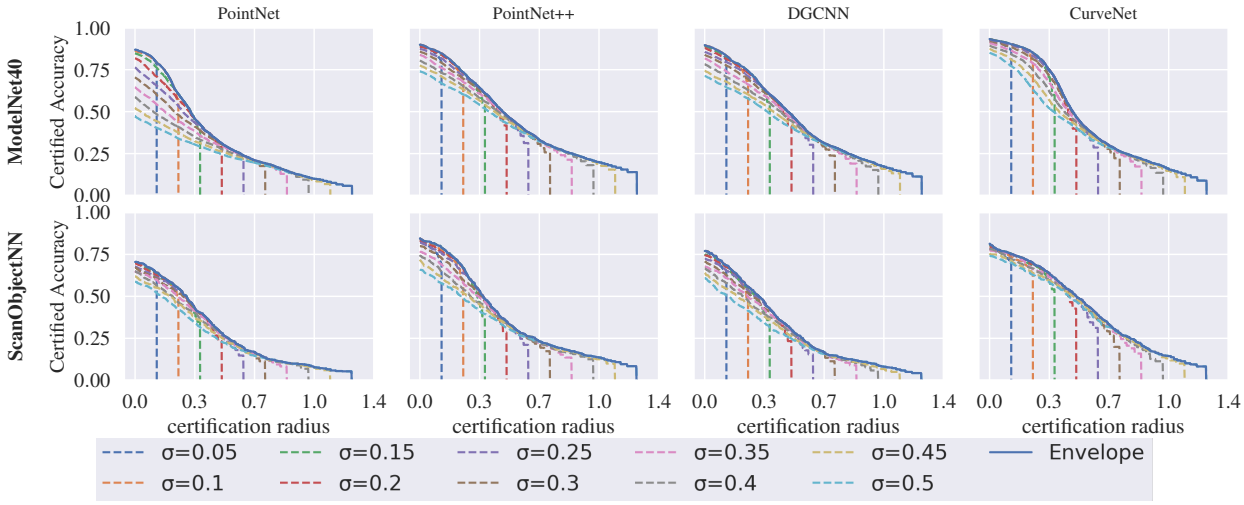

Figure 16. **Shearing**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

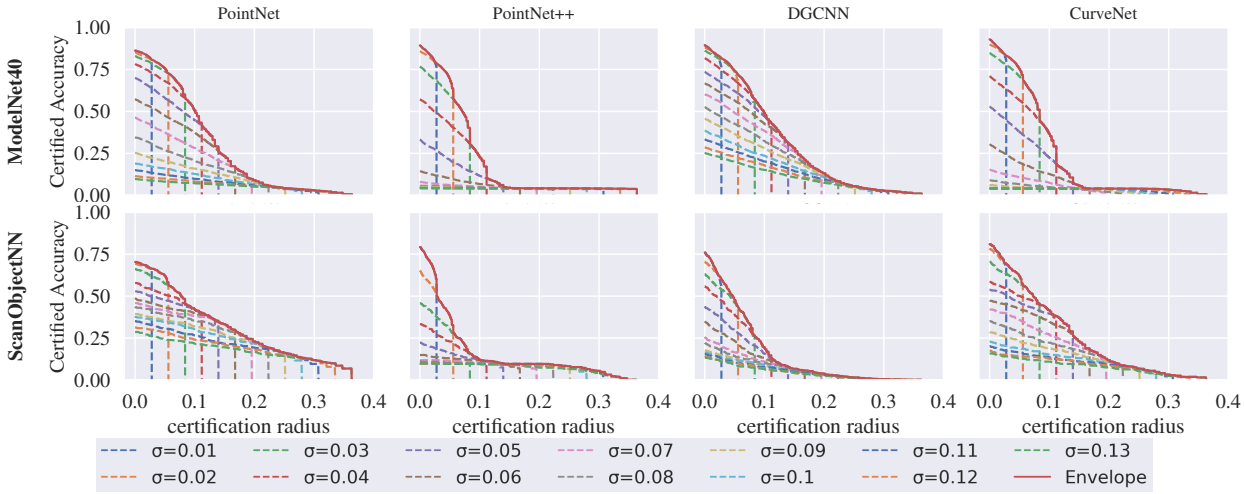

Figure 17. **Gaussian Noise**. Certified accuracy for PointNet, PointNet++, DGCNN and CurveNet on ModelNet40 and ScanObjectNN.

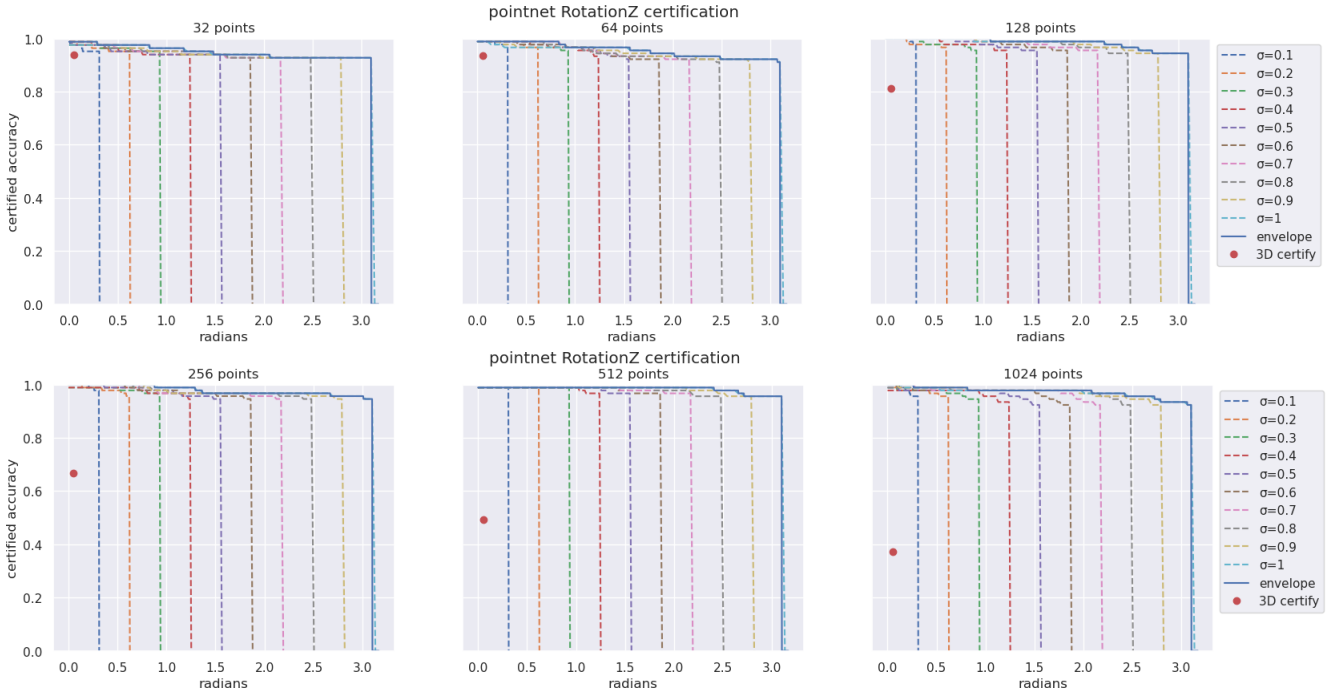

Figure 18. **Certification under different Point Cloud Cardinality.** Larger point sampling leads to severe loss in certification ability for 3DCertify. In contrast, the variation of 3DeformRS performance is minimal.