A. Example Backdoor Training Samples

In Sec. 5.2, we created 36 attacks for 9 (source, target) class pairs (P1, ..., P9), and 4 different types of local geometry (GS, RS, RP, and HS) for the embedded backdoor points. In Fig. 8 we show an example backdoor training sample for each attack.

B. ASR and ACC for 36 Attacks (Completed Results)

In Sec. 5.4 we have shown the statistics of ASR and ACC for the 36 attacks we created. In Tab. 4, we show complete results, i.e. the ASR and ACC for all 36 attacks.

C. BA against PC AD (Completed Results)

In Sec. 5.6, we have shown the ASR for the BAs we created against the state-of-the-art point AD proposed in [66], but only for the victim classifier architecture PointNet due to space limitations. Here, we show the results for DNN architectures PointNet++ and DGCNN in Tab. 5 and Tab. 6, respectively.

D. BA against Other Defenses

In Sec. 5.6, we mentioned that the state-of-the-art BA defenses (that are designed for image BAs) are not applicable to our PC BA. Here, we provide a more detailed discussion about our BA facing other BA defenses.

Existing BA defenses can be divided into several categories. One major category is reverse-engineering defenses[44, 14, 52, 45, 54]. These defenses aim to detect whether a classifier (pre-trained by a possibly malicious third party) is backdoor attacked, without access to the classifier's training set. Using a small, clean dataset independently collected by the defender, these defenses first estimate a backdoor pattern for each putative target class. Then, a detection statistic related to the estimated pattern is obtained for each class. The classifier is deemed to be attacked if there exists a detection statistic that is atypical to the others. However, the reverse-engineering process of these defenses rely on the backdoor embedding mechanism. Since our PC BA adopts an entirely different backdoor embedding mechanism with the images BAs considered in those defenses, they are clearly not applicable to our PC BA.

Another category of BA defenses assumes that the defender has access to the training set. This category includes many early defenses that hypothesize that the backdoor training samples inserted into the training set are separable from clean target class samples in terms of internal layer representation [42, 5, 50]. Although these defenses are not constraint to a particular domain, they heavily rely on the architecture of the classifier. Here, we shown an example using attack P1-GS and PointNet we created in Sec. 5.2. Following the protocol of [5], we obtained the penultimate layer representations for the backdoor training samples and clean target class samples, and found that they are actually not separable. A visualization (projected on 2D) of such non-separability is shown in Fig. 9.

Other BA defenses include a meta-learning approach which simulates classifiers being attacked [58]. However, this method requires a large number of clean samples to train a sufficient number of classifiers. Also, it is more difficult to build an attack distribution in the entire 3D space than in pixel space. A fine-pruning approach prunes neurons that are dormant for clean input samples [23]. They hypothesize these neurons are corresponding to the "backdoor mapping". However, this approach also depends on the compactness of the DNN architecture. [8] and [9] are test-time BA detectors, but their extension to PC domain is not trivial. A detector that reverse-engineers backdoor patterns on the possibly poisoned training set is proposed in [55]. However, this defense relies on the backdoor embedding mechanism and also access to the training set.

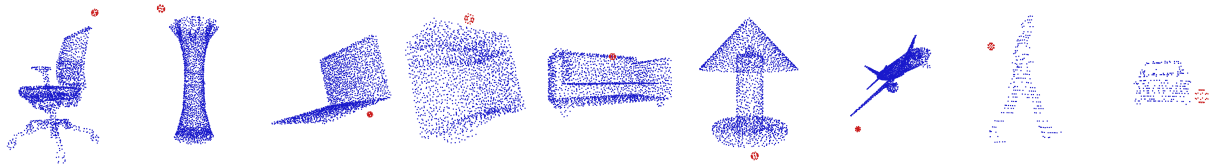
In summary, how to detect our PC BA is still an open problem.

E. Visualization of Learned Backdoor Mapping

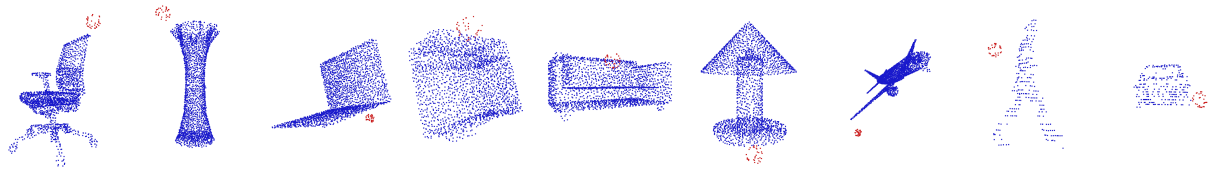
According to [4], PointNet selects a subset of points from a PC, using max pooling, as the "critical points" that visually represent the skeleton of the object from which the PC is obtained (see Fig. 7 of [4]). To validate that the backdoor mapping has been learned by the victim classifier trained on the poisoned training set, we feed each PC (with backdoor points embedded) in Fig. 8 into the victim classifier for its associated attack. In Fig. 10, we show the critical points selected for each PC by the victim classifier with PointNet architecture – for all 36 PCs, a subset of backdoor points are selected as the critical points; such selection is decisive for misclassification to the attacker's target class.

F. Influence of the Number of Backdoor Points

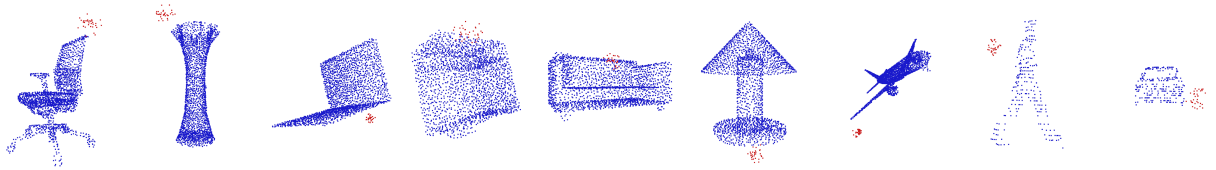
In Sec. 4.1, we have indicated that the number of inserted backdoor points $n' = |\mathbf{V}|$ should be sufficiently large, such that point sub-sampling cannot effectively remove all backdoor points. In our experiments in the main paper, we set $n' = 32$ uniformly. Here, we study the influence of the number of inserted backdoor points on the effectiveness of our BA. Again, we consider the three attacks with local geometry RP and associated with class pairs P1, P2, and P3, respectively (as examples). We use exactly the same configurations for attack implementation, training, and performance evaluation as in the main paper for these three attacks. Especially, we use the same optimized spatial location for these three attacks as in the main paper. How-



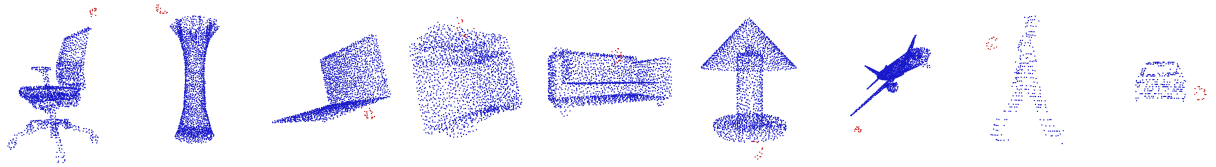
(a) Example backdoor training samples for local geometry GS, for class pairs P1-P9.



(b) Example backdoor training samples for local geometry RS, for class pairs P1-P9.



(c) Example backdoor training samples for local geometry RP, for class pairs P1-P9.



(d) Example backdoor training samples for local geometry HS, for class pairs P1-P9.

Figure 8: Example backdoor training samples for the 36 attacks.

		ModeNet40														KITTI			
		P1		P2		P3		P4		P5		P6		P7		P8		P9	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
PointNet [4]	GS	94.0	88.5	93.0	88.8	95.0	88.9	91.9	88.6	95.0	88.2	95.0	88.9	94.0	89.0	96.4	99.4	89.1	99.2
	RS	93.0	88.7	98.0	88.9	100	88.8	95.0	88.2	95.0	88.8	95.0	88.9	96.0	88.7	99.4	99.4	87.3	99.4
	RP	94.0	88.7	96.0	88.9	100	88.5	96.5	88.4	90.0	87.8	90.0	89.2	98.0	88.9	97.0	99.7	90.9	99.1
	HS	93.0	88.9	97.0	88.4	100	88.7	95.3	88.2	93.0	88.9	100	88.4	94.0	88.5	91.2	99.5	91.2	99.5
PointNet++ [37]	GS	97.0	91.9	94.0	91.0	94.0	91.3	89.5	91.0	91.0	91.5	100	91.3	97.0	91.6	99.1	99.5	92.7	99.5
	RS	96.0	91.0	98.0	91.1	100	91.5	92.0	91.5	94.0	90.2	100	91.2	98.0	90.7	98.5	99.4	87.6	99.4
	RP	95.0	90.2	95.0	91.2	95.0	91.0	96.5	90.9	98.0	91.2	100	91.4	99.0	91.2	97.3	99.8	89.7	99.5
	HS	95.0	91.3	88.0	91.4	100	91.5	90.7	91.3	96.0	91.1	90.0	91.4	96.0	91.6	87.6	99.5	89.5	99.5
DGCNN [46]	GS	91.0	90.8	94.0	91.5	92.0	90.9	94.2	91.7	95.0	91.7	90.0	91.0	96.0	91.2	97.9	99.5	95.5	99.5
	RS	87.0	91.0	98.0	91.3	100	91.5	94.0	91.1	92.0	91.0	90.0	90.8	96.0	90.7	98.5	99.7	91.5	99.8
	RP	96.0	91.4	96.0	90.9	100	90.8	97.7	91.0	97.0	90.6	90.0	91.3	96.0	91.0	99.7	99.4	93.1	99.7
	HS	92.0	91.2	92.0	91.0	100	91.0	93.0	91.0	87.0	90.8	95.0	91.0	97.0	90.9	94.9	99.4	90.6	99.5

Table 4: ASR and ACC (in %) for the 36 attacks (for 9 class pairs P1, P2, ..., P9, and four types of local geometry GS, RS, RP, HS), for victim classifier architectures PointNet, PointNet++, and DGCNN.

ever, different from the main paper where $n' = 32$, we vary the number of inserted backdoor points – we consider $n' \in \{5, 10, \cdots, 40\}$. In Fig. 11, we show the ASR versus n' for these three attacks (without a point anomaly detector). All three attacks achieve ASR $> 80\%$. Note that the ASRs are evaluated with point sub-sampling, which keeps only 1024 points for classification. In other words, when we insert only 5 points into a PC with 2048 points, there is roughly $0.5^5 = 0.03125$ probability that all the inserted points will be removed. Such "3-percent" degradation in ASR is also reflected in Fig. 11 (see the degradation in ASR for the three attacks with 5 inserted points).

	GS	RS	RP	HS
P1	50.0 (97.0)	94.0 (96.0)	90.0 (95.0)	92.0 (95.0)
P2	9.0 (94.0)	89.0 (98.0)	93.0 (95.0)	80.0 (88.0)
P3	52.0 (94.0)	95.0 (100)	90.0 (95.0)	90.0 (100)
P4	8.1 (89.5)	85.0 (92.0)	93.0 (96.5)	82.6 (90.7)
P5	1.0 (91.0)	86.0 (94.0)	97.0 (98.0)	93.0 (96.0)
P6	40.0 (100)	100 (100)	100 (100)	85.0 (90.0)
P7	65.0 (97.0)	95.0 (98.0)	97.0 (99.0)	89.0 (96.0)
P8	97.3 (99.1)	96.9 (98.5)	94.0 (97.3)	89.1 (87.6)
P9	85.2 (92.7)	87.6 (87.6)	86.7 (89.7)	88.2 (89.5)

Table 5: Attack success rate (ASR) (in %) for the 36 attacks for victim classifier architecture PointNet++, when the PC AD in [66] is deployed during testing. ASRs (in %) without the AD deployed are shown in parentheses for reference.

	GS	RS	RP	HS
P1	49.0 (91.0)	81.0 (87.0)	88.0 (96.0)	84.0 (92.0)
P2	9.0 (94.0)	87.0 (98.0)	84.0 (96.0)	85.0 (92.0)
P3	48.0 (92.0)	95.0 (100)	100 (100)	90.0 (100)
P4	4.7 (94.2)	89.0 (94.0)	91.9 (97.7)	86.0 (93.0)
P5	2.0 (95.0)	85.0 (92.0)	91.0 (97.0)	80.0 (87.0)
P6	30.0 (90.0)	90.0 (90.0)	90.0 (90.0)	90.0 (95.0)
P7	63.0 (96.0)	94.0 (96.0)	96.0 (96.0)	90.0 (97.0)
P8	92.1 (97.9)	99.4 (98.5)	98.8 (99.7)	96.1 (94.9)
P9	90.3 (95.5)	93.4 (91.5)	95.2 (93.1)	91.5 (90.6)

Table 6: Attack success rate (ASR) (in %) for the 36 attacks for victim classifier architecture DGCNN, when the PC AD in [66] is deployed during testing. ASRs (in %) without the AD deployed are shown in parentheses for reference.

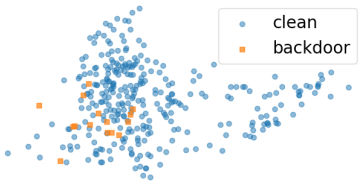


Figure 9: Example of the non-separability between backdoor training samples and clean target class samples for the penultimate layer features obtained using the method in [5].

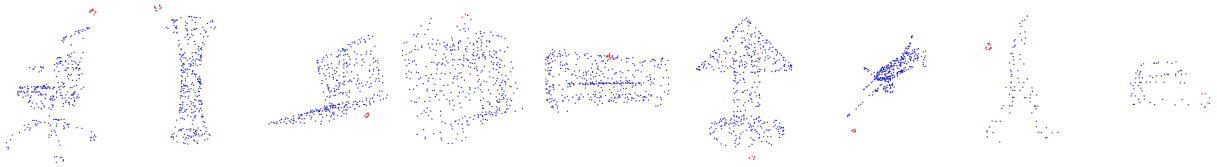
Moreover, in Fig. 12, we show the ASR versus n' for the three attacks when the PC anomaly detector (AD) in [66] is deployed during testing. All three attacks achieve ASR $> 80\%$ when no less than 15 points are inserted into PCs with 2048 points. Note that the PC AD in [66] removes outlier points based on kNN distance with $k = 2$. Thus, a cluster with less than 3 points will be removed with high probability. For an inserted cluster of 5 points, due to the presence of the point sub-sampling, there is roughly $0.5^5 \times (1+5+10) = 0.5$ probability that there will be less than 3 inserted points left. Such "50-percent" degradation is also reflected in Fig. 12 (see the degradation in ASR for the three attacks with 5 inserted points).

G. Choice of ϵ

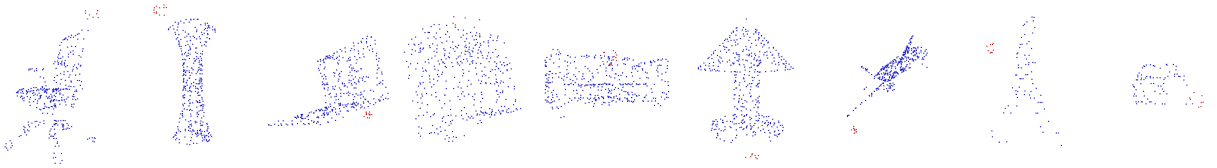
In our experiments in the main paper, we set $\epsilon = 0.02$ when solving (9) for spatial location optimization for all attacks. Here, we study the influence of the choice of ϵ on our BA. As an example, we consider the attack for class pair P1 and with RP backdoor point local geometry. All the configurations for attack implementation, training, and performance evaluation are the same as in the main paper unless specified otherwise.

For each $\epsilon \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$, we perform spatial location optimization by solving (9) using Alg. 1, for 500 random initializations[4] of $\mathbf{c}$. In Fig. 13, we show the optimal objective value of (9) (which represents the average distance from the spatial location to all PCs from $\mathcal{D}_s$) versus the ϵ choices. In Fig. 15, we show example PCs with backdoor points embedded at the optimal spatial location obtained for each ϵ. From both Fig. 13 and Fig. 15, we observe that as ϵ increases, the optimal spatial location for backdoor point embedding will be further apart from the object of interest. When a PC for classification is extracted from a scene using a bounding box, an overly large ϵ may cause the embedded backdoor points to fall outside the bounding box; thus the backdoor points will not be included in the PC for classification. Finally, for each choice of ϵ, we create an attack using its associated optimal spatial location and the same configurations as described in the main paper. In Fig. 14, we show the ASR for the attacks with the above ϵ choices. In general, ASR increases with the growth of ϵ. This is consistent with our intuition for spatial location optimization in Sec. 4.2 – the closer we push the backdoor training samples towards the target class, the easier the backdoor mapping will be learned. However, as we have discussed above, it is infeasible in practice to have an overly large ϵ such that the backdoor points may not be included in the PC for classification. Fortunately, even with $\epsilon = 0.005$, our BA achieves 89% success rate, as shown in Fig. 14. With $\epsilon = 0.005$, the backdoor points are very close to the points for the object of interest (see Fig. 15a).

[4] In our experiments in the main paper, for each attack we solve for the optimal spatial location with only 10 random initializations, which yields a relatively good optimal solution with low time consumption. Here, to compare different choices of ϵ for their best-case scenario, we perform a more thorough search over the entire space in order to find the best optimal spatial location for each ϵ.



(a) Critical points selected by the victim PointNet classifier for the example backdoor training samples in Fig. 8a.



(b) Critical points selected by the victim PointNet classifier for the example backdoor training samples in Fig. 8b.

(c) Critical points selected by the victim PointNet classifier for the example backdoor training samples in Fig. 8c.

(d) Critical points selected by the victim PointNet classifier for the example backdoor training samples in Fig. 8d.

Figure 10: Critical points selected by the associated victim PointNet classifiers for the example training samples for the 36 attacks. For all PC examples, there is a subset of backdoor points selected as the critical points.

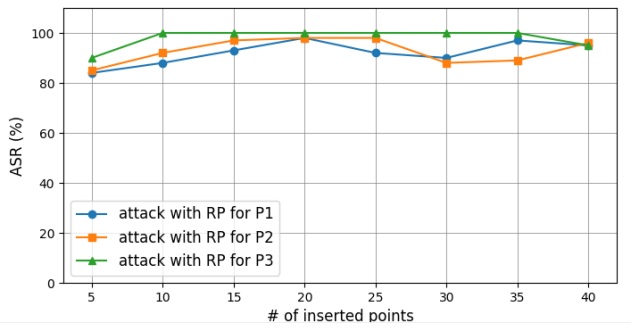


Figure 11: ASR versus number of inserted backdoor points for attacks with local geometry RP associated with class pairs P1, P2, and P3 (as examples), when there is no PC anomaly detector. All attacks achieve ASR $> 80\%$.

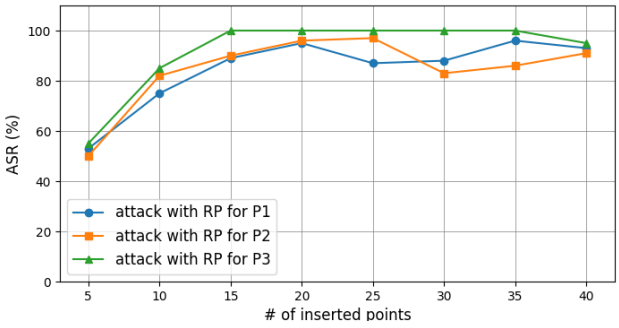


Figure 12: ASR versus number of inserted backdoor points for attacks with local geometry RP associated with class pairs P1, P2, and P3 (as examples), when the PC anomaly detector in [66] is deployed during testing. Attacks with no less than 15 inserted points (into PCs with 2048 points) all achieve ASR $> 80\%$.

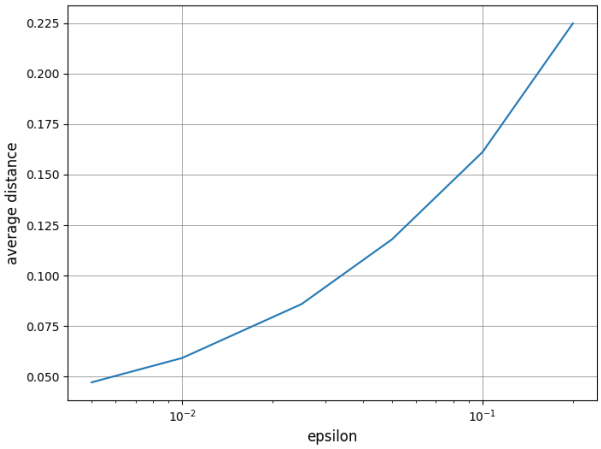


Figure 13: Optimal objective value (i.e. average distance from the optimal spatial location to all PCs from $\mathcal{D}_s$) versus ϵ for solving (9) for spatial location optimization.

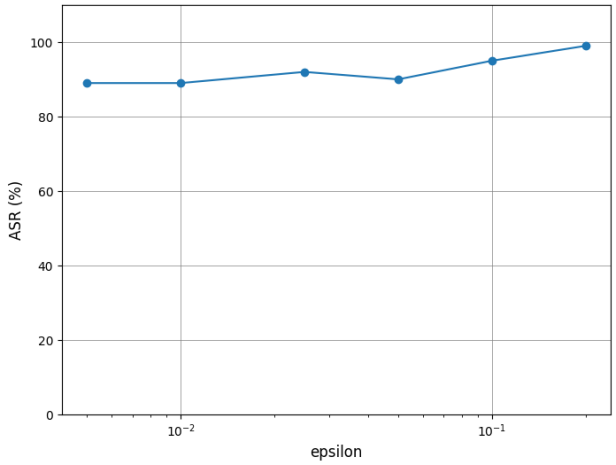
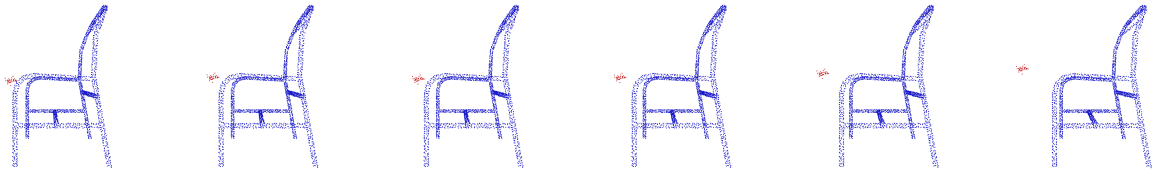


Figure 14: ASR for attacks with $\epsilon \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$. All attacks are successful with ASR $\geq 89\%$.



(a) $\epsilon = 0.005$ (b) $\epsilon = 0.01$ (c) $\epsilon = 0.025$ (d) $\epsilon = 0.05$ (e) $\epsilon = 0.1$ (f) $\epsilon = 0.2$

Figure 15: Example PCs with backdoor points embedded at the optimal spatial location obtained for $\epsilon \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$. The larger the ϵ, the further the backdoor points (in red) are apart from the object of the PC, i.e. the chair (in blue).