# *Supplementary Material*: R3D-AD: Reconstruction via Diffusion for 3D Anomaly Detection

Zheyuan Zhou[1*], Le Wang[1*], Naiyu Fang[1,2],
Zili Wang[1], Lemiao Qiu[1(✉)], and Shuyou Zhang[1]

[1] State Key Laboratory of Fluid Power & Mechatronic System, Zhejiang University, Hangzhou, China
[2] S-Lab, Nanyang Technological University, Singapore, Singapore

https://zhouzheyuan.github.io/r3d-ad

## A Appendix

### A.1 Additional implement details

**Patch-Gen pseudocode** We formulate the process of the proposed 3D anomaly simulation strategy Patch-Gen in Algorithm 1. The procedure begins by taking an initial point cloud $\mathcal{P}$ as input and aims to produce an augmented point cloud $\mathcal{P}_a$ that reflects the addition of anomaly. The rotation matrix $\mathcal{R}$ is obtained by applying arbitrary rotation angles to all the rotation axes. The translation matrix $\mathcal{T}$ is sampled from a Gaussian distribution, and after normalization and scaling, it dictates the displacement of the nearest points towards the viewpoint, while the rest of the point cloud remains unchanged.

**Algorithm 1** Patch-Gen

**Input:** $\mathcal{P}$: input point cloud
  $N$: number of points to select
  $S$: scaling factor for transformation
**Output:** $P_a$: augmented point cloud
  $\mathcal{R} \leftarrow$ random rotation matrix ▷ $\mathbb{R}^{3\times3}$
  $\mathcal{P}_a = \mathcal{P} \cdot \mathcal{R}$ ▷ apply rotation
  $\mathcal{P}_v \leftarrow$ random viewpoint ▷ $\mathbb{R}^{1\times3}$
  $\mathcal{P}_n = NN(\mathcal{P}_a, \mathcal{P}_v, N)$ ▷ select $N$ nearest neighbor points to the viewpoint
  $\mathcal{T} \leftarrow$ random translation matrix ▷ $\mathbb{R}^{N\times3}$
  $\mathcal{P}_n = \mathcal{P}_n + S \cdot \mathit{normalize}(\mathcal{P}_n - \mathcal{P}_v) \odot \mathcal{T}$ ▷ update selected points only

[*] Equal contribution.

**R3D-AD pseudocode** To further clarify the overall architecture of the proposed network R3D-AD, we provide the training and testing iteration procedures more compactly in Algorithm 2 and Algorithm 3, respectively.

During **training**, anomalies are simulated by Patch-Gen, and noise is artificially added following a Gaussian distribution. The model predicts this noise and calculates a displacement to correct for it. The reconstruction loss is measured by comparing the original and corrected point clouds.

**Algorithm 2** R3D-AD training iteration

**Input:** $\mathcal{P}$: input point cloud
**Output:** $\mathcal{L}$: reconstruction loss
$\mathcal{P}^{'} \sim \text{Uniform}(normalize(\mathcal{P}))$ ▷ normalize and downsample the input point cloud
$\mathcal{P}_a^{(0)} = Patch\text{-}Gen(\mathcal{P}^{'})$ ▷ 3D anomaly simulation strategy (Algorithm 1)
$c = PointNet(\mathcal{P}_a^{(0)})$ ▷ feature extraction
$\mathcal{P}_a^{(0)} \sim q(\mathcal{P}_a^{(0)})$ ▷ point distribution
$t \sim \text{Uniform}(\{1, \dots, T\})$ ▷ step distribution
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ noise distribution
$\boldsymbol{\mu} = \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathcal{P}_a^{(0)} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{c}, t)$ ▷ noise prediction
$\Delta = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\sqrt{\bar{\alpha}_t}\mathcal{P}_a^{(0)} + \sqrt{1-\bar{\alpha}_t}(\boldsymbol{\epsilon} - \boldsymbol{\mu})\right)$ ▷ displacement prediction
$\mathcal{L} = \left\|\mathcal{P}^{'} - (\mathcal{P}_a^{(0)} + \Delta)\right\|^2$ ▷ relative reconstruction loss

During **testing**, noise is progressively removed from a simulated noisy version of the cloud, aiming to reconstruct its anomaly-free outfits. The anomaly score is assessed by comparing the clusters after KNN of the original and reconstructed point clouds.

**Algorithm 3** R3D-AD testing iteration

**Input:** $\mathcal{P}$: input point cloud
**Output:** $\mathcal{A}$: anomaly score
$\mathcal{P}^{'} \sim \text{Uniform}(normalize(\mathcal{P}))$ ▷ normalize and downsample the input point cloud
$c = PointNet(\mathcal{P}^{'})$ ▷ feature extraction
$\Delta^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
**for** $t = T, \dots, 1$ **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
    $\Delta^{(t-1)} = \frac{1}{\sqrt{\alpha_t}}(\Delta^{(t)} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon_\theta}(\Delta^{(t)}, \boldsymbol{c}, t)) + \sigma_t\mathbf{z}$
**end for**
$\widehat{\mathcal{P}} = \mathcal{P}^{'} + \Delta^{(0)}$ ▷ reconstructed point cloud
$\widehat{cluster} = KNN(\widehat{\mathcal{P}}, k)$ ▷ reconstructed point-cluster
$cluster = KNN(\mathcal{P}^{'}, k)$ ▷ input point-cluster
$\mathcal{A} = \left\|cluster - \widehat{cluster}\right\|^2$ ▷ euclidean distance for point-cluster

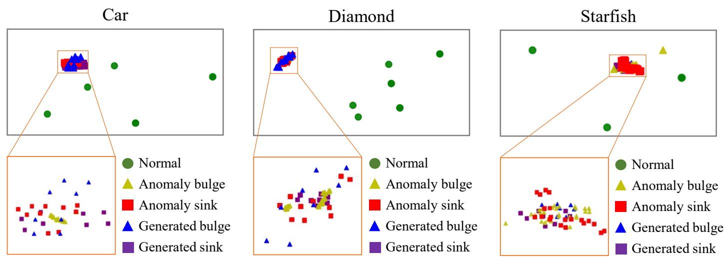



**Fig. 1:** t-SNE visualization on Real3D-AD.

| Anomaly type | Bulge | Sink | Oracle |
|---|---|---|---|
| Airplane | 1.31 | 1.35 | 1.58 |
| Candybar | 2.43 | 2.30 | 2.54 |
| Car | 1.15 | 1.23 | 1.37 |
| Chicken | 3.50 | 2.92 | 4.02 |
| Diamond | 0.84 | 0.83 | 0.97 |
| Duck | 1.53 | 1.29 | 1.67 |
| Fish | 1.42 | 1.45 | 1.57 |
| Gemstone | 2.58 | 5.23 | 5.26 |
| Seahorse | 2.37 | 2.35 | 2.45 |
| Shell | 1.30 | 1.29 | 1.40 |
| Starfish | 2.47 | 2.46 | 2.64 |
| Toffees | 1.73 | 1.71 | 1.79 |

**Table 1:** PSNR of generated anomalous with Patch-Gen on Real3D-AD.

### A.2 Additional experiments

**Quality of the generated anomalies** The proposed 3D anomaly simulation strategy Patch-Gen is designed to address the problem of the lack of 3D anomalous samples in the training phase.

T-distributed Stochastic Neighbor Embedding (t-SNE) [7] is particularly effective at visualizing high-dimensional samples by giving each data point a corresponding location in a low-dimensional map, allowing complex data to be understood at a glance. We follow [4] and use the t-SNE to validate the quality and effectiveness of our generated anomaly samples. As shown in Fig 1, the generated anomalies are clearly distinguished from normal samples and overlap with real anomalous samples, which strengthens our model to reconstruct well on unseen anomalies.

Peak Signal-to-Noise Ratio (PSNR) is an engineering term that quantifies the quality of the reconstruction of a signal. PSNR is typically measured in decibels (dB) and calculated based on the mean squared error between the origin and the reconstruction. The higher the PSNR value, the better the quality of the reconstruction. In Table 1, the PSNR is computed by comparing the generated

| Training Dataset | Training Category | Testing Dataset | Testing Category | I-AUROC | CD | Oracle |
|---|---|---|---|---|---|---|
| Real3D-AD | Airplane | ShapeNetCore.v2 | Airplane | - | 0.032 | 0.001 |
| Real3D-AD | Car | ShapeNetCore.v2 | Car | - | 0.077 | 0.004 |
| ShapeNetCore.v2 | Airplane | Real3D-AD | Airplane | 0.614 | - | 0.772 |
| ShapeNetCore.v2 | Car | Real3D-AD | Car | 0.601 | - | 0.713 |
| Anomaly-ShapeNet | {bowl0..3} | Anomaly-ShapeNet | bowl4 | 0.715 | - | 0.744 |

**Table 2:** Generalization capability of R3D-AD for unseen data.

samples with real anomalies. We randomly select two normal samples to calculate their PSNR, and we average the PSNR obtained from multiple times of randomization to obtain the upper bound of the PSNR limit for each category. The Oracle PSNR servers are a reference to the generation quality.

**Generalization on unseen data** To assess the robustness and generalization capabilities of our proposed model, we conduct a series of experiments on different categories from diverse datasets, as outlined in Table 2. The oracle result represents the performance ceiling of our model, which is obtained by training on the category that is identical to the testing.

For **known categories**, we focus on the well-regarded ShapeNetCore.v2 dataset [2], which includes categories such as Airplanes and Cars, also featured in the Real3D-AD dataset [6]. It's pertinent to note that ShapeNetCore.v2 is not an anomaly detection dataset; it does not encompass anomalous samples. Therefore, for the first and second rows in Table 2, the AUROC metric cannot be utilized in this context. Instead, we resort to evaluating the generalization performance of models trained on Real3D-AD of the same category on ShapeNet-Core.v2 using the Chamfer Distance (CD) metric. The marked decline in performance observed upon transitioning from ShapeNetCore.v2 to Real3D-AD, and vice versa, illuminates the hurdles presented by inconsistencies between datasets. This highlights the importance of our reconstruction approach, which effectively learns inductive biases, allowing for better generalization across different data distributions.

For **unknown categories**, we utilize the Anomaly-ShapeNet dataset [5], as shown in the last row of Table. 2. The model was trained on a subset of Bowl and tested a category it had never encountered during training. Remarkably, despite this lack of prior exposure, our model achieves an impressive score of 0.715 image-level AUROC. This performance surpassed all other methods trained and tested exclusively on "bowl4", thus demonstrating the superior generalization capability of our method.

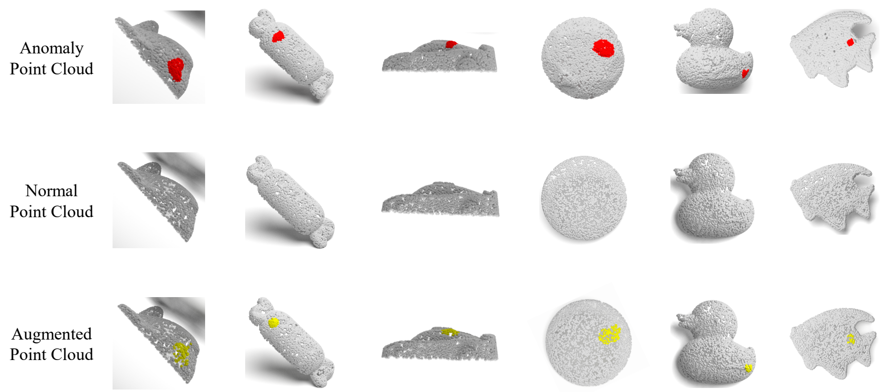
These results not only validate the effectiveness of our approach in handling both known and unknown categories but also underscore its potential for real-world applications where data diversity and unseen scenarios are commonplace.

| **Method** | BTF [3] | | M3DM [9] | PatchCore [8] | | CPMF [1] | Reg3D-AD [6] | IMRNet [5] | **Ours** |
|---|---|---|---|---|---|---|---|---|---|
| **Feat.** | *Raw* | *FPFH* | *PointMAE* | *FPFH* | *PointMAE* | *ResNet* | *PointMAE* | *PointMAE* | *Raw* |
| ashtray0 | 0.578 | 0.420 | 0.577 | 0.587 | 0.591 | 0.353 | 0.597 | 0.671 | **0.833** |
| bag0 | 0.410 | 0.546 | 0.537 | 0.571 | 0.601 | 0.643 | 0.706 | 0.660 | **0.720** |
| bottle0 | 0.597 | 0.344 | 0.574 | 0.604 | 0.513 | 0.520 | 0.486 | 0.552 | **0.733** |
| bottle1 | 0.510 | 0.546 | 0.637 | 0.667 | 0.601 | 0.482 | 0.695 | 0.700 | **0.737** |
| bottle3 | 0.568 | 0.322 | 0.541 | 0.572 | 0.650 | 0.405 | 0.525 | 0.640 | **0.781** |
| bowl0 | 0.564 | 0.509 | 0.634 | 0.504 | 0.523 | 0.783 | 0.671 | 0.681 | **0.819** |
| bowl1 | 0.264 | 0.668 | 0.663 | 0.639 | 0.629 | 0.639 | 0.525 | 0.702 | **0.778** |
| bowl2 | 0.525 | 0.510 | 0.684 | 0.615 | 0.458 | 0.625 | 0.490 | 0.685 | **0.741** |
| bowl3 | 0.385 | 0.490 | 0.617 | 0.537 | 0.579 | 0.658 | 0.348 | 0.599 | **0.767** |
| bowl4 | 0.664 | 0.609 | 0.464 | 0.494 | 0.501 | 0.683 | 0.663 | 0.676 | **0.744** |
| bowl5 | 0.417 | 0.699 | 0.409 | 0.558 | 0.593 | 0.685 | 0.593 | **0.710** | 0.656 |
| bucket0 | 0.617 | 0.401 | 0.309 | 0.469 | 0.593 | 0.482 | 0.610 | 0.580 | **0.683** |
| bucket1 | 0.321 | 0.633 | 0.501 | 0.551 | 0.561 | 0.601 | 0.752 | **0.771** | 0.756 |
| cap0 | 0.668 | 0.618 | 0.557 | 0.580 | 0.589 | 0.601 | 0.693 | 0.737 | **0.822** |
| cap3 | 0.527 | 0.522 | 0.423 | 0.453 | 0.476 | 0.551 | 0.725 | **0.775** | 0.730 |
| cap4 | 0.468 | 0.520 | **0.777** | 0.757 | 0.727 | 0.553 | 0.643 | 0.652 | 0.681 |
| cap5 | 0.373 | 0.586 | 0.639 | **0.790** | 0.538 | 0.697 | 0.467 | 0.652 | 0.670 |
| cup0 | 0.403 | 0.586 | 0.539 | 0.600 | 0.610 | 0.497 | 0.510 | 0.643 | **0.776** |
| cup1 | 0.521 | 0.610 | 0.556 | 0.586 | 0.556 | 0.499 | 0.538 | **0.757** | **0.757** |
| eraser0 | 0.525 | 0.719 | 0.627 | 0.657 | 0.677 | 0.689 | 0.343 | 0.548 | **0.890** |
| headset0 | 0.378 | 0.520 | 0.577 | 0.583 | 0.591 | 0.643 | 0.537 | 0.720 | **0.738** |
| headset1 | 0.515 | 0.490 | 0.617 | 0.637 | 0.627 | 0.458 | 0.610 | 0.676 | **0.795** |
| helmet0 | 0.553 | 0.571 | 0.526 | 0.546 | 0.556 | 0.555 | 0.600 | 0.597 | **0.757** |
| helmet2 | 0.602 | 0.542 | 0.623 | 0.425 | 0.447 | 0.462 | 0.614 | **0.641** | 0.633 |
| helmet3 | 0.526 | 0.444 | 0.374 | 0.404 | 0.424 | 0.520 | 0.367 | 0.573 | **0.707** |
| helmet4 | 0.349 | 0.719 | 0.427 | 0.484 | 0.552 | 0.589 | 0.381 | 0.600 | **0.720** |
| jar0 | 0.420 | 0.424 | 0.441 | 0.472 | 0.483 | 0.610 | 0.592 | 0.780 | **0.838** |
| microphone0 | 0.563 | 0.671 | 0.357 | 0.388 | 0.488 | 0.509 | 0.414 | 0.755 | **0.762** |
| shelf0 | 0.164 | 0.609 | 0.564 | 0.494 | 0.523 | 0.685 | 0.688 | 0.603 | **0.696** |
| tap0 | 0.525 | 0.560 | 0.754 | 0.753 | 0.458 | 0.359 | 0.676 | 0.676 | **0.736** |
| tap1 | 0.573 | 0.546 | 0.739 | 0.766 | 0.538 | 0.697 | 0.641 | 0.696 | **0.900** |
| vase0 | 0.531 | 0.342 | 0.423 | 0.455 | 0.447 | 0.451 | 0.533 | 0.533 | **0.788** |
| vase1 | 0.549 | 0.219 | 0.427 | 0.423 | 0.552 | 0.345 | 0.702 | **0.757** | 0.729 |
| vase2 | 0.410 | 0.546 | 0.737 | 0.721 | 0.741 | 0.582 | 0.605 | 0.614 | **0.752** |
| vase3 | 0.717 | 0.699 | 0.439 | 0.449 | 0.460 | 0.582 | 0.650 | 0.700 | **0.742** |
| vase4 | 0.425 | 0.510 | 0.476 | 0.506 | 0.516 | 0.514 | 0.500 | 0.524 | **0.630** |
| vase5 | 0.585 | 0.409 | 0.317 | 0.417 | 0.579 | 0.618 | 0.520 | 0.676 | **0.757** |
| vase7 | 0.448 | 0.518 | 0.657 | 0.693 | 0.650 | 0.397 | 0.462 | 0.635 | **0.771** |
| vase8 | 0.424 | 0.668 | 0.663 | 0.662 | 0.663 | 0.529 | 0.620 | 0.630 | **0.721** |
| vase9 | 0.564 | 0.268 | 0.663 | 0.660 | 0.629 | 0.609 | 0.594 | 0.594 | **0.718** |
| **Average** | 0.493 | 0.528 | 0.552 | 0.568 | 0.562 | 0.559 | 0.572 | 0.659 | **0.749** |

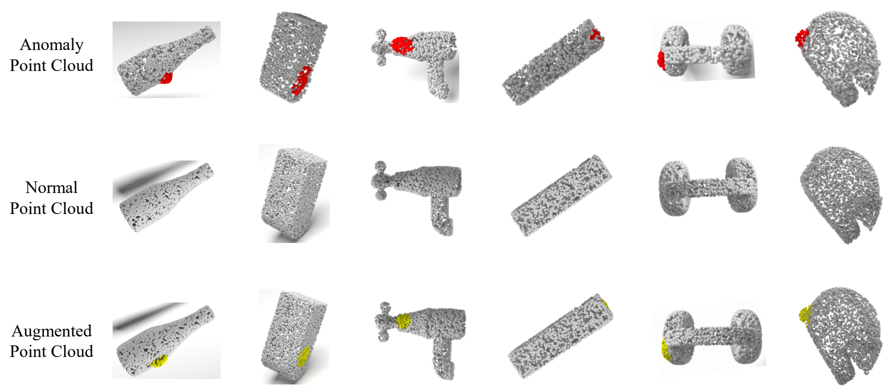
**Table 3:** Complete image-level anomaly detection AUROC on Anomaly-ShapeNet dataset. We highlight the best result in **bold**.

### A.3 Additional main results

Anomaly-ShapeNet [5] contains a total of 40 categories. In the main text, due to the space limitation, we consider objects that belong to the same kind but with differing appearances to be in the same category (e.g., bottle0, bottle1, bottle3 are categorized as Bottle). Here, we provide the specific image-level AUROC as in Table 3.




**(a)** Visualization on Real3D-AD dataset.




**(b)** Visualization on Anomaly-ShapeNet dataset.

**Fig. 2:** Visualization on Real3D-AD dataset and Anomaly-ShapeNet dataset. The red region indicates the real abnormal area of the anomaly point cloud in the testing set, while the yellow region indicates the simulated abnormal area generated by Patch-Gen based on the normal point cloud in the training set.

### A.4 Additional qualitative results

To further demonstrate and compare the effect of our proposed 3D anomaly simulation strategy Patch-Gen, we conduct additional qualitative analysis on the Real3D-AD dataset and the Anomaly-ShapeNet dataset.

The first row shows the anomaly samples in the testing split, where the second row shows the normal samples in the training split, and the third row shows the anomaly samples simulated by Patch-Gen. It can be seen from Fig. 2 that our method fully simulates the defects that vary in different classes, proving that our method can well compensate for the domain gap caused by using only positive samples for training in 3D anomaly detection.

## References

1. Cao, Y., Xu, X., Shen, W.: Complementary pseudo multimodal feature for point cloud anomaly detection. arXiv preprint (2023)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint (2015)
3. Horwitz, E., Hoshen, Y.: Back to the feature: Classical 3d features are (almost) all you need for 3d anomaly detection. In: CVPRW (2023)
4. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: CVPR (2021)
5. Li, W., Xu, X., Gu, Y., Zheng, B., Gao, S., Wu, Y.: Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network. arXiv preprint (2023)
6. Liu, J., Xie, G., Li, X., Wang, J., Liu, Y., Wang, C., Zheng, F., et al.: Real3d-ad: A dataset of point cloud anomaly detection. In: NeurIPS (2023)
7. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research (2008)
8. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: CVPR (2022)
9. Wang, Y., Peng, J., Zhang, J., Yi, R., Wang, Y., Wang, C.: Multimodal industrial anomaly detection via hybrid fusion. In: CVPR (2023)