Removing Geometric Bias in One-Class Anomaly Detection with Adaptive Feature Perturbation
–Supplementary Material–

Romain Hermary Vincent Gaudillière Abd El Rahman Shabayek Djamila Aouada

University of Luxembourg, Esch-sur-Alzette, Luxembourg

{romain.hermary, vincent.gaudilliere, abdelrahman.shabayek, djamila.aouada}@uni.lu

A. Experimental Environment

Experiments were conducted with Python 3.10 [25], PyTorch 1.12.1 [16] (compiled for hardware compatible with CUDA 11.3 [15]), and PyTorch Lightning 2.0.0 [5]. We provide the code and additional configuration details along with this document. This includes a requirement file with all versions of the packages used and a Docker [13] image specification file.

B. Detailed Architectures

Table 1 depicts the detailed architecture and configuration of the networks used as perturbator and classifier in PLUME. Total numbers of learnable parameters are also reported; note that only the classifier is used for inference. For the perturbator, layers 3.1 and 3.2 are both using the output of layer 2; their respective outputs are μ and σ, used to draw a new vector from a Gaussian distribution (then given to the 4^{th} layer), as in a classic VAE. Regarding the classifier, embeddings used to calculate the contrastive loss are taken after layer 6.

	No.	Type	Input Size	Output Size	Affine	Bias	# Param.
Perturbator	1	Linear	3072	3072	-	Yes	9.4M
	2	LeakyReLU	3072	3072	-	-	-
	3.1	Linear	3072	3072	-	Yes	9.4M
	3.2	Linear	3072	3072	-	Yes	9.4M
	4	Linear	3072	3072	-	Yes	9.4M
	5	LeakyReLU	3072	3072	-	-	-
	6	Linear	3072	6144	-	Yes	18.8M
						Total	56.6M
Classifier	1	Linear	3072	1024	-	No	3.1M
	2	BatchNorm	1024	1024	No	-	0
	3	LeakyReLU	1024	1024	-	-	-
	4	Linear	1024	512	-	No	524K
	5	BatchNorm	512	512	No	-	0
	6	LeakyReLU	512	512	-	-	-
	7	Linear	512	1	-	No	512
						Total	3.7M

Table 1. Detailed architecture of the perturbator and classifier used in PLUME. The layers are numbered for each network following their order of appearance in the layer sequence. When a configuration parameter (Affine, Bias...) does not apply to a layer type, a dash is used.

C. Additional Results

To study the stability of PLUME with respect to a varying hyperparameter λ, we report per class results on CIFAR-10 with values 5, 10 and 20 (Table 3). Compared to PLAD [2], PLUME shows SotA performance regardless of the value of λ and greater stability. In the end, we prove that PLUME does not require any fine-tuning of the optimiser, learning rate, or other hyperparameter to achieve SotA performance. Table 2 highlights the maximum AUC values reached for each class and λ.

Table 4 depicts detailed results of PLUME and PLAD which we also ran on CIFAR-100 (results were not provided in the original paper). PLUME outperforms PLAD on each meta-class and averages at 80.3% AUC, which is higher than PLAD by 16.5pp, without any increase of learnable parameter, change in the architecture or training procedure.

Table 5 shows a detailed report of the results of all recent methods on CIFAR-10, including the ones based on rotations.

	λ	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
Max.	5	88.0	87.8	77.1	79.4	89.7	79.6	**88.4**	86.1	**92.6**	92.9	86.16
	10	**91.0**	**90.1**	**80.2**	**80.8**	89.6	**83.7**	84.9	85.8	89.4	93.8	86.92
	20	89.4	86.5	75.6	79.0	**90.6**	79.8	**88.4**	**89.5**	88.3	**94.0**	86.13
	Bests	91.0	90.1	80.2	80.8	90.6	83.7	88.4	89.5	92.6	94.0	88.09

Table 2. Detailed results of the maximum AUC reached for each CIFAR-10 class, over three different λ values (5, 10, 20). The mean of those maximums are reported in the last column, and the best AUC for each class in the last row.

D. Vector Space Analysis

This section details the 3D plots in Figures 1 to 4. t-SNE [24] was used to reduce the different vectors to 3D vectors (with a perplexity of 30), and each plot shows multiple view angles. Models from the ablation study and features extracted from CIFAR-10 with ResNet50 are used.

Method	λ	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
PLAD [2]	$\sim$	82.5$_{\pm 0.4}$	80.8$_{\pm 0.9}$	68.8$_{\pm 1.2}$	65.2$_{\pm 1.2}$	71.6$_{\pm 1.1}$	71.2$_{\pm 1.6}$	76.4$_{\pm 1.9}$	73.5$_{\pm 1.0}$	80.6$_{\pm 1.8}$	80.5$_{\pm 0.3}$	75.1
PLAD†	5	79.9$_{\pm 1.1}$	78.8$_{\pm 1.4}$	64.4$_{\pm 2.9}$	62.2$_{\pm 2.6}$	59.2$_{\pm 3.8}$	67.1$_{\pm 4.3}$	61.4$_{\pm 5.7}$	71.6$_{\pm 2.9}$	78.3$_{\pm 2.4}$	78.7$_{\pm 1.8}$	70.2
PLUME	5	86.8$_{\pm 1.3}$	**85.7**$_{\pm 1.7}$	**74.5**$_{\pm 2.3}$	75.0$_{\pm 6.8}$	87.4$_{\pm 1.7}$	77.0$_{\pm 2.7}$	86.1$_{\pm 1.7}$	84.0$_{\pm 2.6}$	**87.8**$_{\pm 3.1}$	89.6$_{\pm 3.2}$	83.4
	10	**89.5**$_{\pm 1.0}$	85.6$_{\pm 2.7}$	71.7$_{\pm 6.0}$	**78.3**$_{\pm 3.6}$	**87.7**$_{\pm 1.3}$	**79.5**$_{\pm 3.7}$	82.9$_{\pm 3.4}$	84.3$_{\pm 1.5}$	87.1$_{\pm 1.4}$	89.9$_{\pm 2.7}$	83.7
	20	82.5$_{\pm 7.6}$	81.7$_{\pm 3.6}$	71.1$_{\pm 2.9}$	73.2$_{\pm 3.5}$	86.5$_{\pm 3.0}$	78.0$_{\pm 1.5}$	**87.5**$_{\pm 0.9}$	**84.6**$_{\pm 3.7}$	85.4$_{\pm 3.7}$	**90.0**$_{\pm 2.7}$	82.0
	$\sim$	**89.5**$_{\pm 1.0}$	**85.7**$_{\pm 1.7}$	**74.5**$_{\pm 2.3}$	**78.3**$_{\pm 3.6}$	**87.7**$_{\pm 1.3}$	**79.5**$_{\pm 3.7}$	**87.5**$_{\pm 0.9}$	**84.6**$_{\pm 3.7}$	**87.8**$_{\pm 3.1}$	**90.0**$_{\pm 2.7}$	**84.5**

Table 3. Study on the impact of λ (CIFAR-10 dataset). Official results of the baseline PLAD [2] are achieved with different λ values per class ($\sim$). We reproduce the experiments (PLAD†) fixing λ to the most frequent value ($\lambda = 5$). PLUME performance is also reported with λ values fixed over classes (5,10,20), and considering the best value per class ($\sim$).

D.1. Pseudo-Anomalies

In Figures 1 and 2, normal data (green, Bird class) and the generated pseudo-anomalies (red) are displayed; features are taken before the classifier. We compare the pseudo-anomalies generated after training with the introduced LinearMap (PLUME, Figure 2) and with the baseline mapping (Figure 1, AddMult). We observe that the linear mapping seems to allow for better separated features compared to the baseline noising procedure, with which normal and pseudo-anomalous features are entangled. An interesting observation is that two "blobs" are generated with our linear mapping (Figure 2). In fact, we found that this seems to be an amplification of the dissimilarities already present in the normal data. Indeed, by measuring the cosine similarity between the normal vectors generating one blob and the other, we observed that data of each group had greater similarities within the group than with any data from the other one.

D.2. Embedded Vectors

In Figures 3 and 4, embedded vectors from normal data (blue, Truck class), the generated pseudo-anomalies (orange) and real anomalies (other classes, pink) are displayed; vectors are taken after the first part of the classifier f_1, where the contrastive loss is applied. We compare the vectors extracted with the AddMult perturbation and without contrastive guidance (Figure 3) and PLUME (Figure 4). With PLUME we see a more important aggregation of the normal data points, and the separation with the anomalies seems clearer.

E. Supplements on the Ablation Study

E.1. Mean Contrastive Loss

In the ablation study, we report results with a simplified version of the fully contrastive loss which does not compute the similarity with respect to all vectors for each vector, but instead with respect to the mean embeddings of the normal data ($\hat{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i$) and the pseudo-anomalies ($\hat{\tilde{\mathbf{z}}} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{z}}_i$):

$$\hat{\mathfrak{L}}_c = \frac{1}{N}\sum_{i=1}^{N} -\log\left[\frac{\exp(\mathfrak{c}(\mathbf{z}_i, \hat{\mathbf{z}}))}{\exp(\mathfrak{c}(\mathbf{z}_i, \hat{\tilde{\mathbf{z}}})) + \exp(\mathfrak{c}(\mathbf{z}_i, \hat{\mathbf{z}}))}\right] \quad (1)$$

Meta-Class	AUC	
	PLUME	PLAD†
0	**75.8**$\pm$2.7	70.2$\pm$2.5
1	**74.9**$\pm$1.8	65.9$\pm$1.5
2	**88.8**$\pm$1.2	70.3$\pm$5.4
3	**83.2**$\pm$1.6	62.2$\pm$1.8
4	**84.3**$\pm$1.4	72.2$\pm$3.1
5	**83.5**$\pm$2.7	61.2$\pm$2.4
6	**88.2**$\pm$3.8	69.7$\pm$1.9
7	**69.8**$\pm$2.7	57.7$\pm$2
8	**80.1**$\pm$1.8	60.3$\pm$2.6
9	**78.2**$\pm$0.9	68.8$\pm$3.8
10	**94.7**$\pm$0.4	70.6$\pm$4
11	**77**$\pm$1.7	64.5$\pm$2.6
12	**74.1**$\pm$2.4	57.8$\pm$3.6
13	**73.6**$\pm$2.4	51.1$\pm$1.7
14	**86.5**$\pm$2.2	70.3$\pm$3.4
15	**73.1**$\pm$2.5	51.2$\pm$1.7
16	**65.1**$\pm$2.2	64.7$\pm$3.1
17	**90.1**$\pm$1.2	66$\pm$2.6
18	**86**$\pm$2.5	64.2$\pm$2
19	**78.2**$\pm$2.8	57.4$\pm$1.5
Average	**80.3**	63.8

Table 4. Detailed results of average AUC achieved by PLUME and PLAD† (reproduced results), over 5 runs, for each meta-class of the **CIFAR-100** dataset.

E.2. Other Perturbations

To supplement our ablation study we provide in Table 6 results with only the additive noise (Add) and only the multiplicative noise (Mult). Both are learned with the same

	Method	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
Rotation Based	Geom [7]	74.7	95.7	78.1	72.4	87.8	87.8	83.4	95.5	93.3	91.3	86.0
	Rot* [9]	78.3±0.2	94.3±0.3	86.2±0.4	80.8±0.6	89.4±0.5	89.0±0.4	88.9±0.4	95.1±0.2	92.3±0.3	89.7±0.3	88.4
	Rot+Trans* [9]	80.4±0.3	96.4±0.2	85.9±0.3	81.1±0.5	91.3±0.3	89.6±0.3	89.9±0.3	95.9±0.1	95.0±0.1	92.6±0.2	89.8
	GOAD* [1]	75.5±0.3	94.1±0.3	81.8±0.5	72.0±0.3	83.7±0.9	84.4±0.3	82.9±0.8	93.9±0.3	92.9±0.3	89.5±0.2	85.1
	DROC [22]	90.9±0.5	98.9±0.1	88.1±0.1	83.1±0.8	89.9±1.3	90.3±1.0	93.5±0.6	98.2±0.1	96.5±0.3	95.2±1.3	92.5
	CSI [23]	89.9±0.1	99.1±0.0	93.1±0.2	86.4±0.2	93.9±0.1	93.2±0.2	95.1±0.1	98.7±0.0	97.9±0.0	95.5±0.1	94.3
	iDECODe [11]	86.5±0.0	98.1±0.0	86.0±0.5	82.6±0.1	90.9±0.1	89.2±0.1	88.2±0.4	97.8±0.1	97.2±0.0	95.5±0.1	91.2
	SSD [21]	82.7	98.5	84.2	84.5	84.8	90.9	91.7	95.2	92.9	94.4	90.0
	NDA [3]	98.5	76.5	79.6	79.1	92.4	71.7	97.5	69.1	98.5	75.2	84.3
	UniCon-HA [27]	91.7±0.1	99.2±0	93.9±0.1	89.5±0.2	95.1±0.1	94.1±0.2	96.6±0.1	98.9±0.0	98.1±0.0	96.6±0.1	95.4
	UNODE [14]	97.0	98.8	96.0	92.4	96.5	94.7	98.5	98.6	98.6	97.8	96.9
Geometrically Unbiased	AnoGAN [19]	67.1	54.7	52.9	54.5	65.1	60.3	58.5	62.5	75.8	66.5	61.8
	OCSVM [20]	61.6	63.8	50.0	55.9	66.0	62.4	74.7	62.6	74.9	75.9	64.7
	IF [12]	66.1	43.7	64.3	50.5	74.3	52.3	70.7	53.0	69.1	53.2	59.7
	DAE [26]	41.1	47.8	61.6	56.2	72.8	51.3	68.8	49.7	48.7	37.8	53.6
	DAGMM [30]	41.4	57.1	53.8	51.2	52.2	49.3	64.9	55.3	51.9	54.2	53.1
	ADGAN [4]	63.2	52.9	58.0	60.6	60.7	65.9	61.1	63.0	74.4	64.2	62.4
	DSVDD [18]	61.7	65.9	50.8	59.1	60.9	65.7	67.7	67.3	75.9	73.1	64.8
	OCGAN [17]	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	82.0	55.4	65.6
	TQM [28]	40.7	53.1	41.7	58.2	39.2	62.6	55.1	63.1	48.6	58.7	52.1
	DROCC [8]	79.2	74.9	68.3	62.3	70.3	66.1	68.1	71.3	62.3	76.6	69.9
	HRN-L2 [10]	80.6	48.2	64.9	57.4	73.3	61.0	74.1	55.5	79.9	71.6	66.7
	HRN [10]	77.3	69.9	60.6	64.4	71.5	67.4	77.4	64.9	82.5	77.3	71.3
	DPAD [6]	78.0±0.3	75.0±0.2	68.1±0.5	66.7±0.4	77.9±0.8	68.6±0.3	81.2±0.4	74.8±0.2	79.1±1.0	76.1±0.2	74.6
	DO2HSC [29]	81.3±0.2	82.7±0.3	71.3±0.4	71.2±1.3	72.9±2.1	72.8±0.2	83.0±0.6	75.5±0.4	84.4±0.5	82.0±0.9	77.7
	PLAD [2]	82.5±0.4	80.8±0.9	68.8±1.2	65.2±1.2	71.6±1.1	71.2±1.6	76.4±1.9	73.5±1.0	80.6±1.8	80.5±0.3	75.1
	PLUME (ours)	89.5±1.0	85.7±1.7	74.5±2.3	78.3±3.6	87.7±1.3	79.5±3.7	87.5±0.9	84.6±3.7	87.8±3.1	90.0±2.7	84.5

Table 5. Detailed results (AUC) on CIFAR-10 of all methods depicted in Figure 4 of the main article. * denotes values taken from CSI [23].

VAE in PLUME with a reduced output dimension to generate only α_i or β_i. The results stay on par with the AddMult perturbation, though only the multiplicative part of the noise gives on average a better performance.

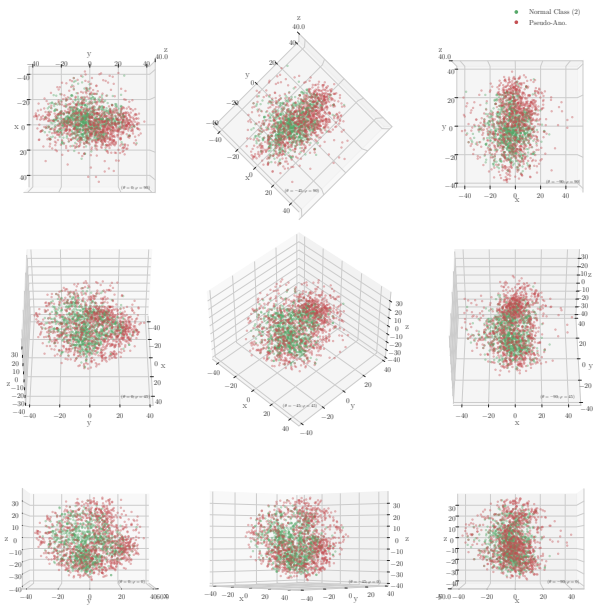

Figure 1. Study done on CIFAR-10 Bird class, with features extracted with ResNet50. Illustration of normal data (green) and generated pseudo-anomalies (red). Pseudo-anomalies were produced by the PLUME with the AddMult perturbation method.

Perturbation	Guidance	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
Gaussian	-	83.3±2.6	75.7±10.0	69.8±3.2	72.8±9.1	80.6±3.4	72.0±3.3	84.5±1.7	79.6±5.7	79.9±3.6	76.9±9.3	77.5
Gaussian	✓	75.1±3.2	79.6±3.4	68.4±1.3	70.4±1.6	74.5±3.7	69.1±2.6	78.5±2.1	69.1±3.2	75.2±5.0	77.6±5.3	73.7
AddMult	-	66.8±3.5	70.5±9.5	67.4±1.5	64.9±3.6	67.5±5.4	67.0±3.5	70.6±7.8	68.1±7.7	72.5±4.6	68.7±5.2	68.4
AddMult	✓	59.2±2.6	62.6±2.6	61.2±1.5	58.7±2.0	63.0±1.9	62.6±1.0	68.6±3.0	59.5±4.9	61.1±1.2	57.5±4.6	61.4
Add	✓	59.8±2.2	62.7±4.4	59.7±2.3	60.1±3.3	59.0±2.5	56.6±4.3	61.5±7.7	59.7±3.2	59.7±4.1	62.0±3.5	60.1
Mult	✓	62.0±3.2	65.0±3.7	62.6±3.4	58.5±3.0	61.2±2.7	59.4±6.1	66.0±5.2	59.2±2.9	66.8±3.4	64.5±1.0	62.5
LinearMap	-	69.5±9.6	75.7±10.1	76.2±1.9	58.8±8.7	74.5±6.7	78.4±2.0	85.1±1.0	78.0±5.4	85.6±3.4	86.5±2.5	76.8
LinearMap	✓ (Mean)	81.4±6.0	72.4±7.2	74.8±3.5	60.3±7.9	74.5±5.5	77.9±1.8	86.4±1.8	82.6±2.7	84.0±3.1	74.3±11.9	76.9
LinearMap	✓	86.8±1.3	85.7±1.7	74.5±2.3	75.0±6.8	87.4±1.7	77.0±2.7	86.1±1.7	84.0±2.6	87.8±3.1	89.6±3.2	83.4

Table 6. Ablation study of the main article supplemented by results with only additive (Add) and only multiplicative (Mult) noises.

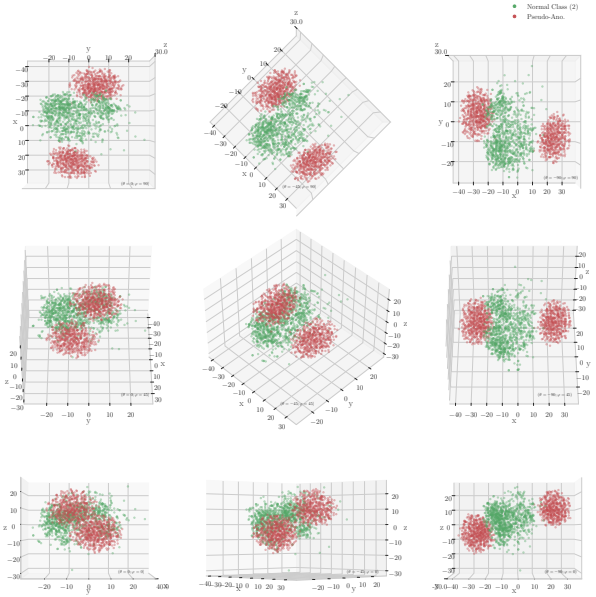

Figure 2. Study done on CIFAR-10 Bird class, with features extracted with ResNet50. Illustration of normal data (green) and generated pseudo-anomalies (red). Pseudo-anomalies were produced by a trained PLUME configuration.

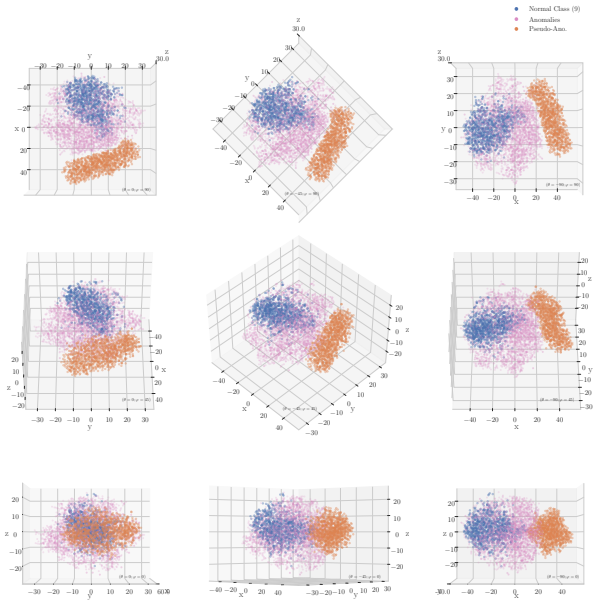

Figure 3. Study done on CIFAR-10 Truck class, with features extracted with ResNet50. Illustration of embedded vectors from normal data (blue), generated pseudo-anomalies (orange) and real anomalies (pink). Pseudo-anomalies were produced by training PLUME with the AddMult perturbation and without contrastive loss.

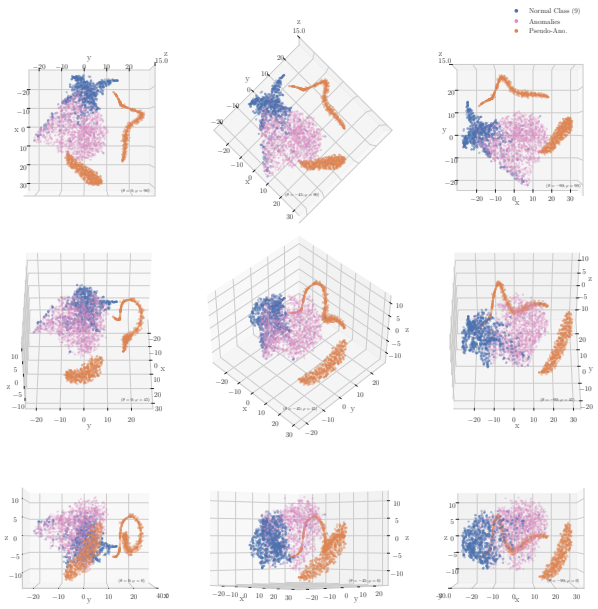

Figure 4. Study done on CIFAR-10 Truck class, with features extracted with ResNet50. Illustration of embedded vectors from normal data (blue), generated pseudo-anomalies (orange) and real anomalies (pink). Pseudo-anomalies were produced by a trained PLUME configuration.

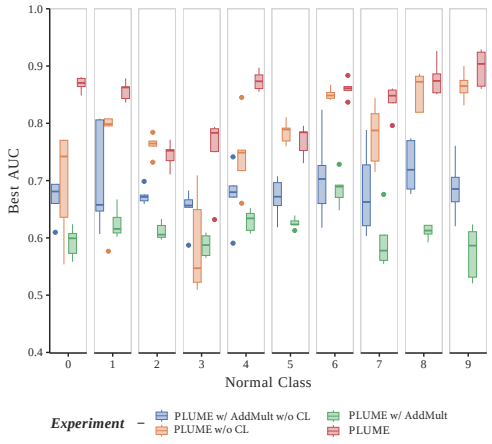

Figure 5. Visual interpretation of the ablation study conducted in the main paper, in the form of box plots. Results are displayed by class, and coloured by experiment.

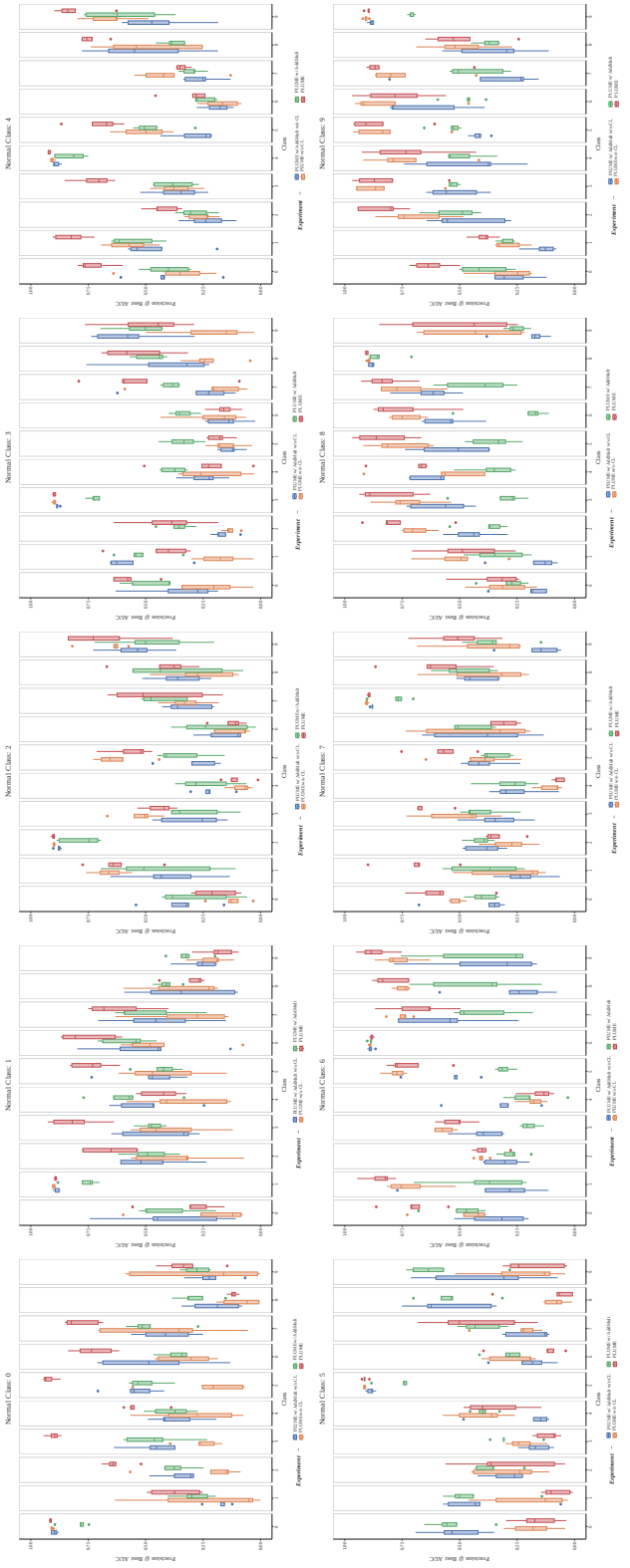

Figure 6. In-depth illustration of the ablation study. Each subfigure shows the results of the trained models on one normal class. In each one, precision over the different classes are reported.

References

[1] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 3

[2] Jinyu Cai and Jicong Fan. Perturbation learning based anomaly detection. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 1, 2, 3

[3] Chengwei Chen, Yuan Xie, Shaohui Lin, Ruizhi Qiao, Jian Zhou, Xin Tan, Yi Zhang, and Lizhuang Ma. Novelty detection via contrastive learning with negative data augmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 606–614. ijcai.org, 2021. 3

[4] Lucas Deecke, Robert A. Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*, volume 11051 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2018. 3

[5] William Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019. 1

[6] Dazhi Fu, Zhao Zhang, and Jicong Fan. Dense projection for anomaly detection. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 8398–8408. AAAI Press, 2024. 3

[7] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9781–9791, 2018. 3

[8] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. DROCC: deep robust one-class classification. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3711–3721. PMLR, 2020. 3

[9] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15637–15648, 2019. 3

[10] Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. HRN: A holistic approach to one class learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[11] Ramneet Kaur, Susmit Jha, Anirban Roy, Sangdon Park, Edgar Dobriban, Oleg Sokolsky, and Insup Lee. idecode: In-distribution equivariance for conformal out-of-distribution detection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7104–7114. AAAI Press, 2022. 3

[12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 413–422. IEEE Computer Society, 2008. 3

[13] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014. 1

[14] Hossein Mirzaei, Mojtaba Nafez, Mohammad Jafari, Mohammad Bagher Soltani, Mohammad Azizmalayeri, Jafar Habibi, Mohammad Sabokrou, and Mohammad Hossein Rohban. Universal novelty detection through adaptive contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22914–22923, June 2024. 3

[15] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with CUDA. *ACM Queue*, 6(2):40–53, 2008. 1

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 1

[17] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: one-class novelty detection using gans with constrained latent representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2898–2906. Computer Vision Foundation / IEEE, 2019. 3

[18] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4390–4399. PMLR, 2018. 3

[19] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging - 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, volume 10265 of *Lecture Notes in Computer Science*, pages 146–157. Springer, 2017. 3

[20] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001. 3

[21] Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[22] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[23] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[24] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 1

[25] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. 1

[26] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1096–1103. ACM, 2008. 3

[27] Guodong Wang, Yunhong Wang, Jie Qin, Dongming Zhang, Xiuguo Bao, and Di Huang. Unilaterally aggregated contrastive learning with hierarchical augmentation for anomaly detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 6865–6874. IEEE, 2023. 3

[28] Jingjing Wang, Sun Sun, and Yaoliang Yu. Multivariate triangular quantile maps for novelty detection. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5061–5072, 2019. 3

[29] Yunhe Zhang, Yan Sun, Jinyu Cai, and Jicong Fan. Deep orthogonal hypersphere compression for anomaly detection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 3

[30] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3