Supplementary Material for A Unified, Resilient, and Explainable Adversarial Patch Detector

Vishesh Kumar, Akshay Agarwal
Trustworthy BiometraVision Lab, IISER Bhopal
{vishesh22, akagarwal}@iiserb.ac.in

This supplementary presents the details which could not be included in the main paper due to space constraints:

1. AdvPatchXAI on the Standard Datasets in Prototype Literature

Since our proposed model is based on prototypes, building upon the prototype-based approach described in the main paper, we evaluate our model's performance on CUB-200-2011 [8], and Stanford Cars [3], standard benchmarks in prototype learning. We compare our results with several current state-of-the-art prototype-based models. The following metrics are used for evaluation.

- **Top-1 Acc:**

$$\text{Top-1 Acc} = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions}}$$

- **Global Size:** Global size presents the number of prototypes in the model with at least one non-zero weight.
- **Local size:** The local size counts all present relevant prototypes for any class. For a local explanation, we count all relevant prototypes with a similarity > 0.1 for a local explanation.

As shown in Table 1, our proposed AdvPatchXAI model utilizes significantly fewer prototypes compared to PIPNet with a ConvNeXt backbone. Specifically, the global size of AdvPatchXAI (AdvPatchXAI-C) employs 31 and 32 fewer prototypes than PIPNet-C on the CUB and CARS datasets, respectively, while still maintaining competitive classification accuracy.

2. AdaptiveAttacks

It is to note that the proposed algorithm has been found robust and generalized in handling unseen physical adversarial attack patches and patches perturbed with natural corruption. However, to further strengthen the effectiveness of the proposed defense algorithm, we have evaluated its resiliency against adaptive attacks in various forms: (i) ***gray-box:*** where the attacker has complete access to the defense model except the color channel information used as input and (ii) ***white-box:*** where the attacker has full access to the model including color channel. Following the standard protocols, we have applied the BPDA & Auto attacks [1] following benchmark attack parameter settings to fool the proposed defense algorithm. The results reported in Table 2 demonstrate the resiliency of the proposed attack in handling adaptive adversaries. We want to highlight that the proposed defense has been trained on the unseen patches and even found to be generalized when we combined the adaptive adversary with corruption.

[1]There is a significant discrepancy of 5.2% between our proposed AdvPatchXAI-R and PIP-Net R. Furthermore, upon reproducing the results of PIP-Net R on the CUB dataset using the code implementation of authors, we achieved a performance of 76.1%, which is 1.1% lower than that of our proposed model.

Table 1. Performance comparison in object recognition of different methods on CUB and CARS datasets.

Dataset	Method	TOP-1 Acc ↑	Global Size ↓	Local Size ↓
CUB	AdvPatchXAI-C	**84.5**	464	10 (4)
	AdvPatchXAI-R	77.2	673	13 (4)
	PIP-Net C [4]	84.3	495	10 (4)
	PIP-Net R [4]	82.0[1]	731	12 (5)
	ProtoPNet [2]	79.2	2000	2000
	ProtoTree [5]	82.2	202	8.3
	ProtoPShare [7]	74.7	400	400
CARS	AdvPatchXAI-C	**88.3**	483	9 (3)
	AdvPatchXAI-R	85.7	570	8 (3)
	PIP-Net C [4]	88.2	515	9 (4)
	PIP-Net R [4]	86.5	669	11 (4)
	ProtoPNet [2]	86.1	1960	1960
	ProtoTree [5]	86.6	195	8.5
	ProtoPShare [7]	86.4	480	480

Table 2. Adaptive attack robustness of the proposed algorithm AdvPatchXAI-R (training patch 9). For completeness, we want to mention that the performance on other patches is at least 88%.

	Patch 0	Patch 1	Patch 2	Patch 4	Patch 6	Patch 7	Patch 9
Gray-box	95.12	95.25	94.87	95.25	93.17	94.68	95.25
White-box	93.75	93.81	93.31	93.68	91.25	92.93	93.81

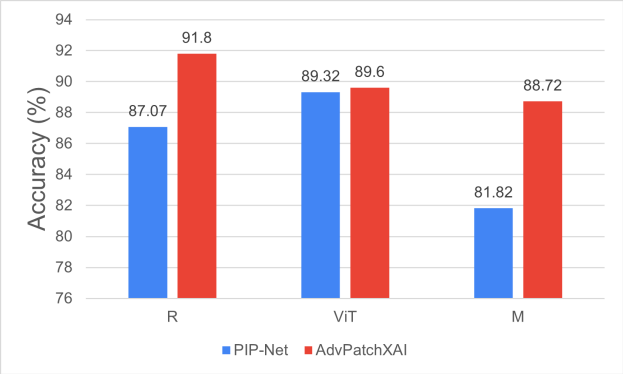

Figure 1. Baseline comparison of AdvPatchXAI with recent prototype-based model PIP-Net with several backbones R, ViT, and M representing ResNet50, Vision Transformer, and MobileNet, respectively. Here, the proposed model is trained on the COCO patched train dataset and evaluated on the COCO test patched dataset in silent (unseen patch) settings and reported mean accuracy.

3. Adversarial Patch Detection

In this section, we extend the presentation of our results by including both mean accuracy and standard deviation (SD). While the main paper emphasizes mean accuracy to summarize the central tendency of our data, we also provide the SD to quantify variability. This measure is crucial for illustrating the dispersion of data points around the mean, offering insights into the consistency and reliability of our findings across different patches.

Table 3 and Table 4 present the results of AdvPatchXAI on COCO subsets under silent and noisy evaluation settings, respectively. Similarly, Table 5 and Table 6 show the results on the ImageNet subset under the same conditions. Our analysis reveals that while AdvPatchXAI with backbones such as ConvNeXt (C), ResNet (R), and MobileNet (M) exhibits high standard deviations in many cases, AdvPatchXAI with the Vision Transformer (ViT) shows relatively small SD values, except for Patch-5 and Patch-6. This indicates the robustness of our proposed algorithm with the ViT backbone. Additionally, it is noteworthy that in 10 out of 16 evaluations with Patch-4, low SD values further support the effectiveness of Patch-4 in conjunction with our proposed algorithm. These findings demonstrate the reliability and robustness of AdvPatchXAI, particularly with the ViT backbone, across different patches and evaluation settings.

3.1. Comparison & Discussion on Defense

The global size in the context of the AdvPatchXAI model refers to the total number of prototypes utilized within the model for classification tasks. This number is critical as it impacts both the model's interpretability and efficiency. In our research, the global size is defined as the number of prototypes in the model with at least one non-zero weight,

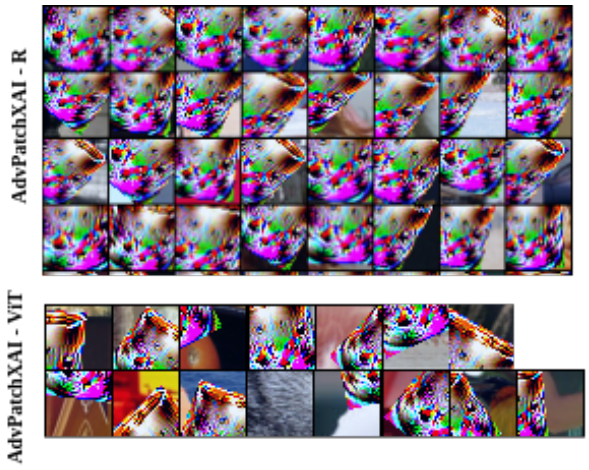

Figure 2. Samples of the prototypes available in AdvPatchXAI-R and AdvPatchXAI-ViT after fully training on the most effective patch **Patch-4** for adversarial patch detection.

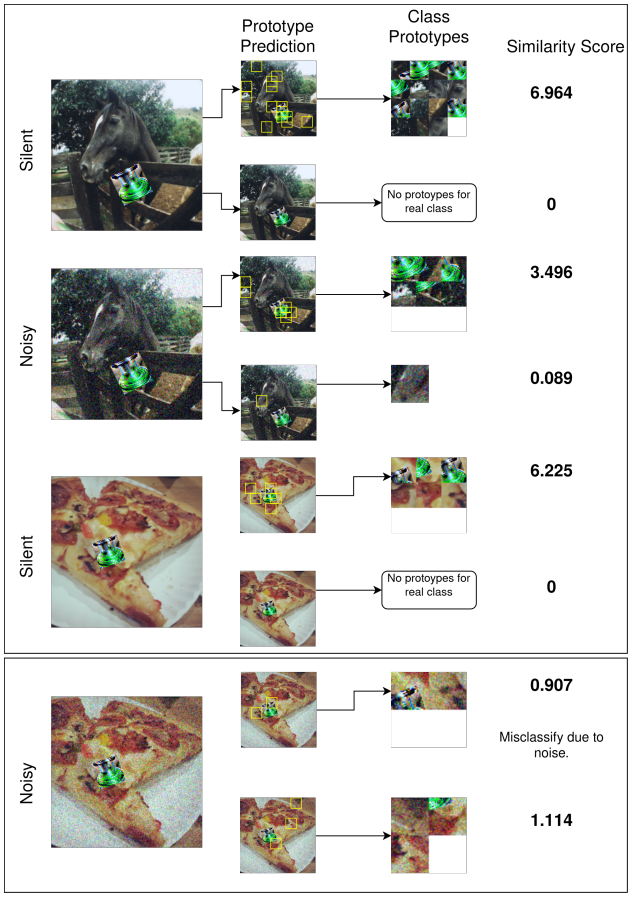

Figure 3. Explanation of proposed AdvPatchXAI with only a few prototypes for the correct class. Here AdvPatchXAI-ViT trained on Patch-4 and evaluated on Patch-0. AdvPatchXAI learns part-prototypes visuaalized as patches from the training data, and localizes similar image patches in an unseen input image.

Table 3. Adversarial patch detection accuracy of the proposed AdvPatchXAI with different backbone on **COCO** (seen dataset) subset in **silent** (unseen patch without any noise) setting. The results are reported as mean along with their standard deviation (SD). **Patch-{0-9}\{1}** indicates models are trained on Patch-1 and tested on **all other patches except Patch-1**. Here, R, M, ViT, and C represent ResNet50, MobileNet, Vision Transformer, and ConvNeXt backbones, respectively. The best mean and SD values are highlighted and underlined across each network.

Method	Train → Test → Mean ↓	Patch-0 Patch-{1-9}	Patch-1 Patch-{0-9}\{1}	Patch-2 Patch-{0-9}\{2}	Patch-3 Patch-{0-9}\{3}	Patch-4 Patch-{0-9}\{4}	Patch-5 Patch-{0-9}\{5}	Patch-6 Patch-{0-9}\{6}	Patch-7 Patch-{0-9}\{7}	Patch-8 Patch-{0-9}\{8}	Patch-9 Patch-{0-8}
AdvPatchXAI-R	Mean	94.09	95.57	84.69	83.30	97.99	79.65	85.53	96.91	94.76	**99.33**
	SD	05.20	09.31	14.75	14.64	02.92	16.53	13.37	04.84	07.27	00.63
AdvPatchXAI-M	Mean	94.17	97.45	77.33	64.88	98.47	72.28	93.90	**98.61**	92.90	97.24
	SD	08.51	05.57	18.91	15.14	00.63	13.89	08.20	02.09	10.79	04.06
AdvPatchXAI-ViT	Mean	95.84	95.19	93.20	81.74	**96.42**	67.27	89.08	93.90	89.00	94.32
	SD	01.54	02.87	05.15	11.12	00.10	14.46	11.28	02.38	08.77	01.22
AdvPatchXAI-C	Mean	90.08	85.29	86.08	75.94	**96.27**	82.42	78.63	87.18	92.73	95.28
	SD	10.58	14.96	15.87	19.87	04.93	15.12	13.60	13.84	07.50	06.92

Table 4. Adversarial patch detection accuracy and standard deviation (SD) on COCO subsets in an **noisy** (unseen patch and unseen noise) evaluation setting. This evaluates the 'dual' resiliency of the detectors when unseen patches perturbed with natural noises are classified. The best mean and SD values are highlighted and underlined across each network. **Patch-{0-9}\{1}** indicates models are trained on Patch-1 and tested on **all other patches except Patch-1**.

Method	Train → Test → Mean ↓	Patch-0 Patch-{1-9}	Patch-1 Patch-{0-9}\{1}	Patch-2 Patch-{0-9}\{2}	Patch-3 Patch-{0-9}\{3}	Patch-4 Patch-{0-9}\{4}	Patch-5 Patch-{0-9}\{5}	Patch-6 Patch-{0-9}\{6}	Patch-7 Patch-{0-9}\{7}	Patch-8 Patch-{0-9}\{8}	Patch-9 Patch-{0-8}
AdvPatchXAI-R	Mean	85.81	88.86	78.07	72.01	**96.09**	69.05	78.95	94.59	83.79	95.92
	SD	12.19	16.11	19.10	18.95	04.18	16.54	18.21	08.13	06.47	02.79
AdvPatchXAI-M	Mean	76.09	70.82	56.91	50.24	**89.63**	51.34	70.45	83.65	71.13	68.11
	SD	19.51	13.06	08.31	00.60	08.49	07.40	14.36	15.04	13.76	14.38
AdvPatchXAI-ViT	Mean	92.31	92.61	88.91	78.91	92.95	64.57	85.74	90.52	80.56	**93.13**
	SD	02.01	05.11	06.51	11.92	00.72	14.34	13.47	03.75	08.83	02.98
AdvPatchXAI-C	Mean	79.16	79.88	74.62	64.69	**86.57**	59.72	72.74	71.09	76.41	84.06
	SD	16.38	15.94	20.14	14.66	11.34	14.54	16.82	17.88	18.32	14.79

Table 5. Adversarial patch detection accuracy of the proposed AdvPatchXAI with different backbone on **ImageNet** (unseen dataset) subset in **Silent** (unseen patch without any noise), **Noisy** (unseen patch+noise). The results are reported as mean along with standard deviation (SD). **Patch-{0-9}\{1}** indicates models are trained on Patch-1 and tested on **all other patches except Patch-1**. Here, R, M, ViT, and C represent ResNet50, MobileNet, Vision Transformer, and ConvNeXt backbones, respectively. The best mean values are highlighted across each network.

Method	Train → Test → Mean ↓	Patch-0 Patch-{1-9}	Patch-1 Patch-{0-9}\{1}	Patch-2 Patch-{0-9}\{2}	Patch-3 Patch-{0-9}\{3}	Patch-4 Patch-{0-9}\{4}	Patch-5 Patch-{0-9}\{5}	Patch-6 Patch-{0-9}\{6}	Patch-7 Patch-{0-9}\{7}	Patch-8 Patch-{0-9}\{8}	Patch-9 Patch-{0-8}
AdvPatchXAI-R	Mean	92.96	95.12	84.49	82.56	98.17	79.40	84.68	97.18	95.60	**98.94**
	SD	06.02	09.80	14.25	14.88	02.16	16.67	14.01	04.47	06.18	00.67
AdvPatchXAI-M	Mean	93.68	95.28	77.30	64.13	95.71	70.72	91.93	**96.81**	91.90	95.92
	SD	08.25	06.49	18.47	14.44	00.67	14.41	09.38	02.10	11.70	04.59
AdvPatchXAI-ViT	Mean	**94.31**	92.67	92.03	81.37	92.94	66.41	86.71	91.77	87.37	90.94
	SD	01.61	03.11	05.53	10.63	00.26	13.90	11.79	02.45	07.99	01.91
AdvPatchXAI-C	Mean	89.41	85.26	85.12	75.91	**96.23**	81.21	77.30	86.67	91.53	94.65
	SD	10.99	14.61	15.37	19.75	05.12	15.15	14.38	14.21	08.20	07.49

Table 6. Adversarial patch detection accuracy of the proposed AdvPatchXAI with different backbone on ImageNet subset (unseen dataset) under **noisy** (unseen patch and unseen noise) setting. The results are reported as mean and standard deviation (SD). **Patch-{0-9}\{1}** indicates models are trained on Patch-1 and tested on **all other patches except Patch-1**. The best mean and SD values are highlighted and underlined across each network.

Method	Train → Test → Mean ↓	Patch-0 Patch-{1-9}	Patch-1 Patch-{0-9}\{1}	Patch-2 Patch-{0-9}\{2}	Patch-3 Patch-{0-9}\{3}	Patch-4 Patch-{0-9}\{4}	Patch-5 Patch-{0-9}\{5}	Patch-6 Patch-{0-9}\{6}	Patch-7 Patch-{0-9}\{7}	Patch-8 Patch-{0-9}\{8}	Patch-9 Patch-{0-8}
AdvPatchXAI-R	Mean	84.39	88.51	77.81	71.42	**95.67**	68.34	78.30	93.72	78.88	94.76
	SD	12.57	15.89	18.75	19.28	03.85	16.42	18.37	07.66	05.41	02.82
AdvPatchXAI-M	Mean	75.10	70.16	56.88	50.22	**87.74**	51.15	69.73	82.00	68.55	67.21
	SD	19.30	13.25	08.17	00.51	08.56	07.16	14.53	14.66	12.97	13.88
AdvPatchXAI-ViT	Mean	**90.18**	90.14	87.98	78.88	89.19	64.22	83.73	87.19	78.16	90.04
	SD	02.31	05.32	06.49	11.16	01.12	14.16	13.35	04.02	08.31	03.59
AdvPatchXAI-C	Mean	78.14	79.55	73.29	63.66	**85.20**	59.13	71.54	70.16	75.21	83.41
	SD	16.79	15.82	20.36	14.27	11.85	14.43	17.08	17.70	17.78	15.27

Table 7. Global size presents the total number of prototypes of different methods (AdvPatchXAI and PIP-Net) with various backbones R, M, ViT, C represents ReNet50, MobileNet, Vision Transformer, and ConvNeXT respectively. All the models are trained on the COCO subset across patches.

Method	Backbone	Patch-0	Patch-1	Patch-2	Patch-3	Patch-4	Patch-5	Patch-6	Patch-7	Patch-8	Patch-9
AdvPatchXAI	R	263	220	182	209	232	213	241	233	250	210
	ViT	187	168	203	166	203	144	183	194	192	190
	M	82	55	79	87	65	85	52	88	83	65
	C	295	291	229	269	293	320	273	316	269	299
PIP-Net	R	370	284	251	252	347	265	315	309	279	269
	ViT	195	151	161	152	161	153	161	170	176	156
	M	67	61	100	91	101	81	68	85	99	113
	C	397	459	348	301	353	363	367	373	274	364

Table 8. Mean IOU for patch detection using Faster R-CNN on the proposed COCO patched dataset.

Metric	Test → Train ↓	Patch-0	Patch-1	Patch-2	Patch-3	Patch-4
Mean IOU	Patch-0	97.51	95.01	93.86	95.38	95.14
Mean IOU	Patch-1	92.92	97.77	95.03	95.63	96.03

which contributes to the decision-making process. Table 7 shows the total number of prototypes with at least one non-zero weight present in the model after full training. Initially, before training, the total number of prototypes (D) in ResNet50, MobileNet, ConvNeXt, and ViT were 2048, 1280, 768, and 768, respectively. The significant drop in global size showcases the better interpretability and efficiency of our proposed AdvPatchXAI.

However, as Table 7 demonstrates that our proposed AdvPatchXAI with ResNet (R), MobileNet (M), and ConvNeXt (C) backbones contains fewer prototypes while providing better performance as shown in Figure 1 compared to PIP-Net. For instance, the proposed AdvPatchXAI with backbone R, ViT, and M outperforms PIP-Net with the same backbone by 4.73%, 0.28%, and 6.9%, respectively, showcasing the robustness of our model. Some sample of the prototypes available in the AdvPatchXAI-ViT and AdvPatchXAI-R corresponding to Patch-4 are shown in Figure 2. More explanation and impact of noise on AdvPatchXAI can be followed by Figure 3. We are not limited to only one metric; we have evaluated the robustness by using AUC-ROC to verify the robustness of our proposed AdvPatchXAI. The average AUC across all unseen patches on COCO patched dataset of AdvPatchXAI-C is 95.91%, compared to 89.76% for PIP-Net-C, indicating a **6.15%** improvement. The AUC of AdvPatchXAI-ViT is 66.95%, compared to 57.92% for PIP-Net-ViT, reflecting a **9.03%** improvement.

3.2. Patch Detection

Apart from classifying an image into a clean or adversarial image, we have performed a preliminary study to demonstrate whether the added patches can be detected in seen and unseen settings. For that, we have trained an object detection model namely Faster R-CNN [6] on COCO patched train images for Patch-0 and Patch-1 and tested on COCO patched test images for the patches 0-5. We have not used clean images for training and testing in this case, i.e., only 1200 images are used for training, and 800 images are used for testing for a particular patch. The result shown in Table 8 achieves more than 90% mean IOU for each patch. In terms of, mean average precession (mAP); in each patch, the model yields **99%** mAP score showcases the potential integration of object detector in our model as part of our future work.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018. 1

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[3] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 1

[4] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. 2023. 1

[5] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021. 1

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region

proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 4

[7] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 1

[8] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1