Residual Pattern Learning for Pixel-wise Out-of-Distribution Detection in Semantic Segmentation (Supplementary Material)

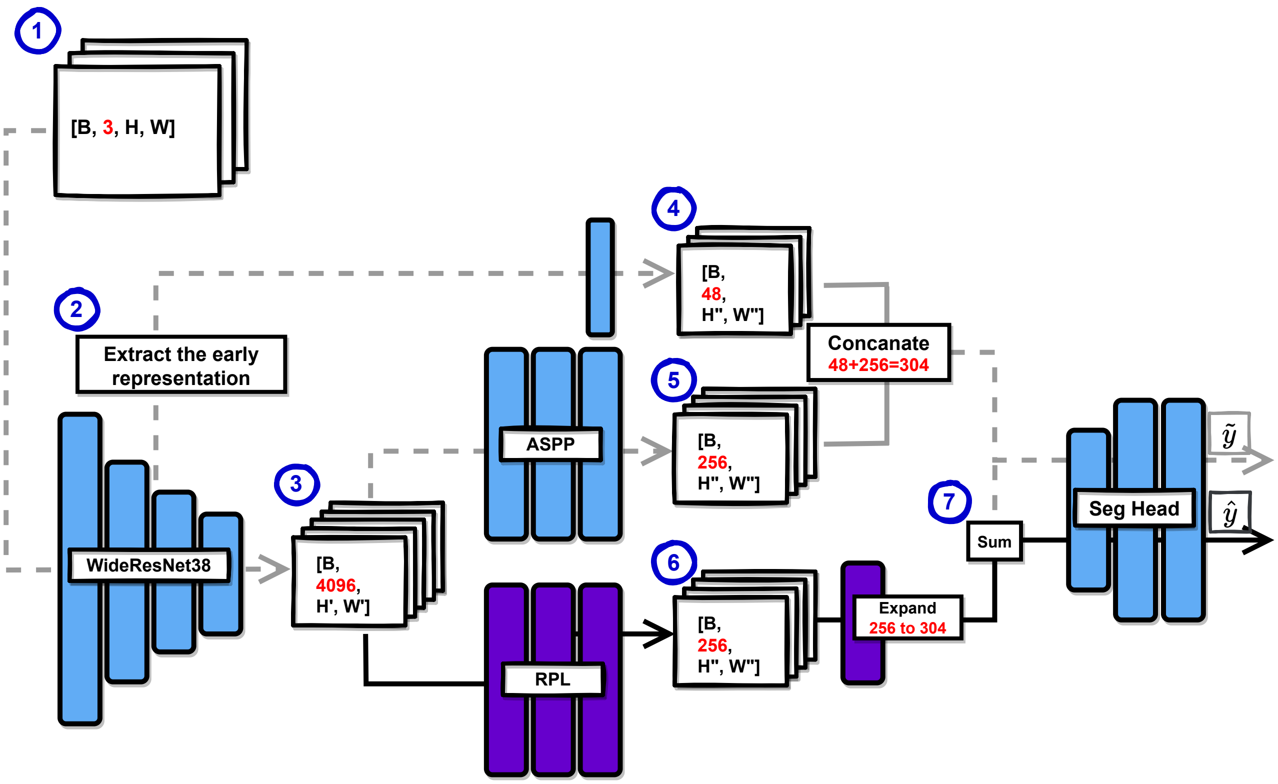


Figure 1: **The detailed workflow of the RPL module.** The blue blocks denote the convolutional layers in the original closed-set segmentation models, the violet blocks represent our RPL module, and the dashed line ("- - -") means the blocks that are processed without requiring training. We produce the pseudo label ($\tilde{\mathbf{y}}$) via the path of ① → ② → ③ → ⑤ ∪ ④ → $\tilde{\mathbf{y}}$, and we produce prediction ($\hat{\mathbf{y}}$) via ① → ③ → ⑥ → ⑦ + (⑤ ∪ ④) → $\hat{\mathbf{y}}$

.

A. The Architecture of Residual Pattern Learning (RPL) and DeepLabV3+

The RPL module is externally attached to the closed-set segmentation network that assists in deciding the potential anomalies, where we utilise DeeplabV3+ [2] as our architecture. As shown in Fig. 1, the batch (**B**) of RGB-based images under height (**H**) and width (**W**) in ① will be fed to the FCN encoder network (e.g., WiderResNet38) first to produce the feature map with **4096** channels in ③. This feature map will then go through the Atrous Spatial Pyramid Pooling (ASPP) layers and RPL module to produce the outputs under the same resolution (i.e., **256** channels) in ⑤ and ⑥, respectively. After that, the representation extracted from shallow layers (in ②) will go through a convolutional layer to produce the feature maps in ④ that are concatenated with ⑤. Then the combined feature map will be fed into the final classifier (Seg. head) to produce $\tilde{\mathbf{y}}$. The feature map in ⑥ will be processed by the following convolutional layer to expand the channels to **304**, which are added to the intermediate feature map from the original segmentation model in ⑦. Finally, such feature map with a potential anomaly will be classified to produce $\hat{\mathbf{y}}$. During training, we utilise $H = 700$, $W = 700$ and $B = 8$ in stage ①, to produce $H' = W' = 88$ in stage ③ and $H" = W" = 350$ in ④ ⑤ ⑥. Finally, the Seg. head will produce $\tilde{\mathbf{y}}$, $\hat{\mathbf{y}}$ with shape $8 \times 19 \times 700 \times 700$ based on the bilinear upsampling, where 19 is the closed-set (i.e., Cityscapes [3]) categories.

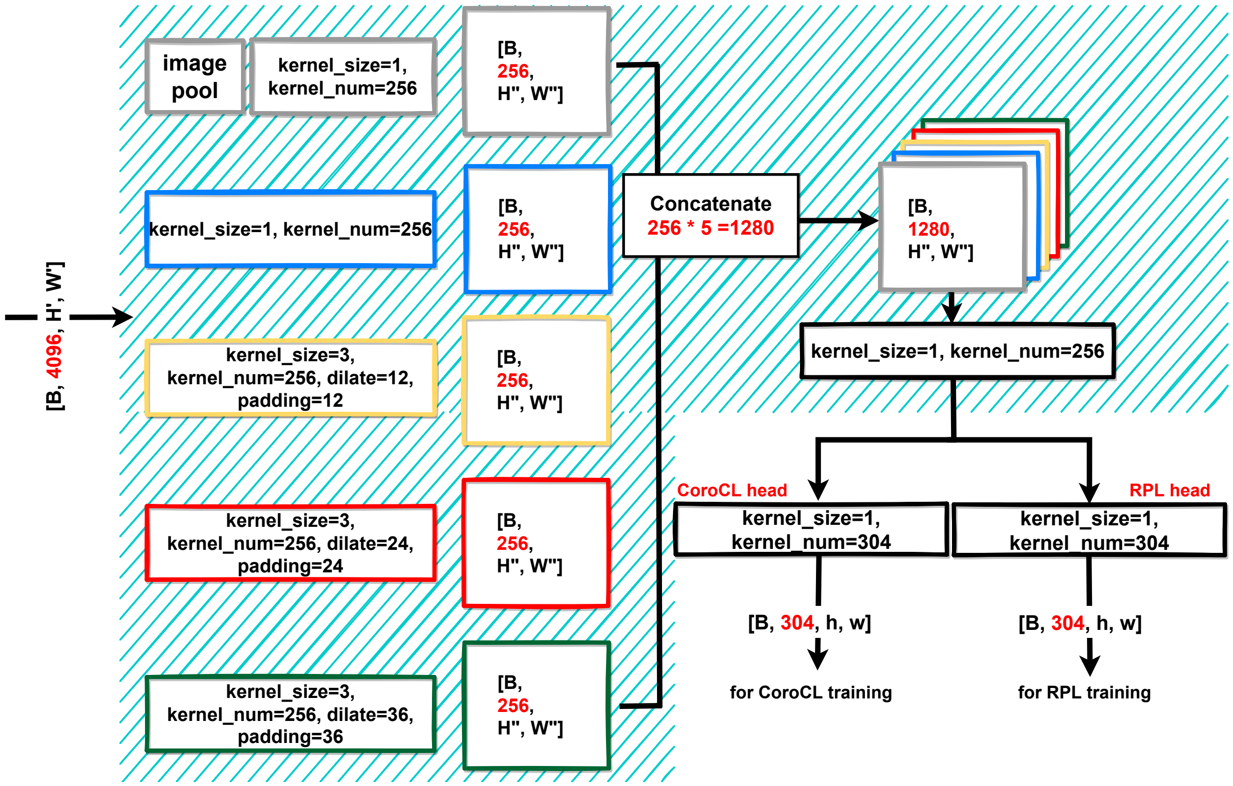

[B, 4096, H', W']
image pool
kernel_size=1, kernel_num=256
[B, 256, H", W"]
kernel_size=1, kernel_num=256
[B, 256, H", W"]
kernel_size=3, kernel_num=256, dilate=12, padding=12
[B, 256, H", W"]
kernel_size=3, kernel_num=256, dilate=24, padding=24
[B, 256, H", W"]
kernel_size=3, kernel_num=256, dilate=36, padding=36
[B, 256, H", W"]
Concatenate
256 * 5 =1280
[B, 1280, H", W"]
kernel_size=1, kernel_num=256
CoroCL head
RPL head
kernel_size=1, kernel_num=304
kernel_size=1, kernel_num=304
[B, 304, h, w]
[B, 304, h, w]
for CoroCL training
for RPL training

Figure 2: **The detailed architecture of RPL and CoroCL.** Our proposed RPL firstly encodes the incoming features from the segmentation network into a set features extracted from different dilated rates and concatenate them together. After being processed by the following convolutional layer, RPL will output the results for CoroCL optimisation (main paper Eq. (7)) and segmentation head (main paper Eq. (3)) based on two separate heads. Note: the region inside the cyan region is motivated from ASPP [1, 2].

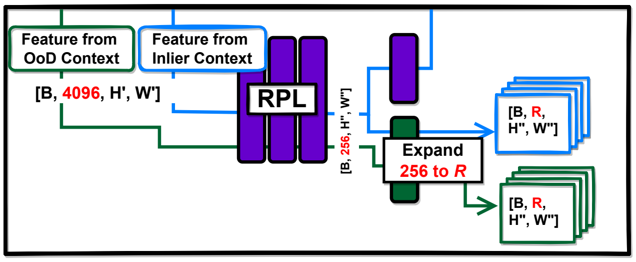


Figure 3: **The detailed workflow of the CoroCL.** The blue lines denote the forward pass with the inlier context features (i.e., based on the input of $\mathbf{x}^{oe}$), while the green lines represent the OoD context features (i.e., based on $\mathbf{x}^{out}$).

A.1. The Architecture of RPL with Context-Robust Contrastive Learning (CoroCL)

On top of the RPL module, we propose CoroCL to generalise the OoD detector for various open-world contexts, as demonstrated in Fig. 3. CoroCL pulls the embedding features that belong to the same class (i.e., both are OoD or inliers) closer and pushes apart those embeddings from different classes (i.e., one is inlier and the other is the outlier, or vice versa). We extract those embeddings based on an extra convolutional layer (also known as the ”projector”) via the intermediate features from inlier and OoD contexts, where the projector expands the features from 256 channels to the R depth of the embedding features. In our experiments, $R = 304$ shows the best performance which is demonstrated in Fig. 6 of the main paper.

A.2. The detailed architecture of RPL and CoroCL

As shown in Fig. 2, we design our proposed RPL module based on the Atrous Spatial Pyramid Pooling (ASPP) [1, 2] block in [15], followed by one convolutional head for CoroCL and one for RPL. During training, the incoming feature (with **4096** channels) will go through a set of convolutional layers that have different dilation rates which produce a set of features that are concatenated to form the feature map with depth **1280**. There is one more convolutional layer to extract the information from such concatenated feature map and reduce the channels to **256**. Finally, the heads of CoroCL and RPL will produce the outputs with **304** depth for training.

Table 1: **Comparing with SOTAs on Fihsyscapes and SMIYC test benchmarks**[1,2] with extra datasets [10, 14]. Our results are in **bold**, and the *gray* row shows the method [4] that utilises a post-processing to narrow the anomaly detection area.

Methods	Anomaly Detection Area	Fishyscapes (test)				SMIYC (test)			
		Static		Lost&Found		AnomalyTrack		ObstacleTrack	
		FPR	AuPRC	FPR	AuPRC	FPR	AuPRC	FPR	AuPRC
NFlowJS [4]	whole image	15.41	52.12	8.98	39.36	34.71	56.92	0.41	85.55
DenseHybrid [5]	whole image	-	-	-	-	9.81	77.96	0.24	87.08
Ours	whole image	**0.53**	**95.80**	**2.24**	**59.43**	**6.22**	**90.78**	**0.40**	**88.61**
NFlowJS (w/ GF) [4]	road/sidewalks pixels	100	50.11	1.96	69.43	-	-	-	-

Table 2: **Improvements for different backbones** on Fihsyscapes, SMIYC and RoadAnomaly validation sets. "Before" represents the pixel-wise anomaly detection performance based on the closed-set segmentation model, while "After" denotes the results after the training of RPL with CoroCL. We use red to represent a decrease and green to represent an increase in the "Improve" row and the results reported in the main paper are in **boldface**.

Backbone		Fishyscapes						SMIYC						RoadAnomaly		
		Static			L&F			Anomaly			Obstacle					
		FPR↓	AP↑	ROC↑	FPR↓	AP↑	ROC↑	FPR↓	AP↑	ROC↑	FPR↓	AP↑	ROC↑	FPR↓	AP↑	ROC↑
MobileNet	Before	47.94	17.45	88.84	42.87	6.32	90.56	46.19	51.92	87.17	6.86	52.18	98.32	67.81	20.27	73.76
	After	18.64	74.42	96.89	20.77	48.54	96.59	26.74	63.52	91.43	3.26	80.07	99.23	38.11	62.49	91.72
	Improve	29.30	57.27	8.05	22.10	42.22	6.03	19.45	11.60	4.26	3.60	27.89	0.91	29.70	42.22	17.96
ResNet50	Before	46.66	28.64	89.01	42.04	10.15	91.24	65.75	46.46	81.07	6.55	49.12	91.33	67.61	22.08	72.78
	After	5.69	87.27	99.07	16.78	49.92	97.78	22.51	72.18	94.08	2.62	74.40	99.42	26.18	63.96	93.24
	Improve	40.97	58.63	10.06	25.26	39.77	6.54	43.24	25.72	13.01	3.93	25.28	8.09	41.43	41.88	20.46
ResNet101	Before	42.85	30.15	90.16	38.07	24.57	92.36	44.92	53.91	86.50	23.75	13.30	93.78	66.21	24.05	77.25
	After	1.61	89.88	99.14	8.82	60.08	98.84	15.13	74.83	95.14	2.46	78.78	99.66	24.54	65.42	94.24
	Improve	41.24	59.73	8.98	29.25	35.51	6.48	29.79	20.92	9.64	21.29	65.48	5.88	41.67	41.37	16.99
WiderResNet38	Before	17.78	41.68	95.90	41.78	16.05	93.72	67.75	44.54	80.26	4.50	34.44	99.67	69.99	19.95	73.61
	After	**0.85**	**92.46**	**99.73**	**2.52**	**70.61**	**99.39**	**7.18**	**88.55**	**98.06**	**0.09**	**96.91**	**99.97**	**17.74**	**71.61**	**95.72**
	Improve	16.93	50.78	3.83	39.26	54.56	5.67	60.57	44.01	9.80	4.41	62.47	1.30	52.25	51.66	22.11

B. Experiments with Extra Training Set

Dataset descriptions. The NFlowJS [4] and Densehybrid [5] have additional experimental setups that fine-tune the OoD detector to extra training sets, including Vistas [10] and Wilddash2 [14]. Vistas [10] contain $20,000$ images from real-world driving scenes with high resolution (2592×1944 pixels) and 66 categories of finely-annotated pixel-wise labels. Similarly, Wilddash2 [14] is another driving scene dataset containing $4,255$ images with 80 categories in total, where each image has 1920×1080 pixels. Given that the experimental setup presented in our submitted main paper only utilises Cityscapes [3] (i.e., $29,75$) images, fine-tuning the OoD detector with those extra training sets enables better robustness to hard inliers.

Results from Fishyscape[1] and SMIYC[2]. To enable a fair comparison, we follow [4, 5] to fine-tune our RPL module with 10 epochs for both Vistas [10] and Wilddash2 [14]. Tab. 1 shows that our method outperforms other approaches under the same setup. For example, our AuPRC results are 12.82% and 1.53% higher than Densehybrid [5] on SMIYC-Anomaly and SMIYC-Obstacle subsets, respectively.

The *Ground-Focus (GF) post-process* in the last row of Tab. 1 merges all road and sidewalk pixels to a common "ground" class by creating a convex hull that encapsulates all such pixels. During inference, all the pixels outside this hull will be ignored, producing significant improvements for Fishyscapes-Lost&Found dataset. However, real-world anomalies (e.g., birds) might not be located on the "ground", reducing its practicability. For example, Fishyscapes-Static has anomalies outside the road that are never detected, leading to unsatisfactory performance. In addition, the inaccurate prediction of the road/sidewalks categories also results in the misdetection of the anomalies.

C. More Implementation Details

We provide more implementation details in this section. **In the training of RPL**, we partially load the parameters from the pre-trained ASPP block in DeepLabV3+ [1, 2] to the main RPL module, as they share the same architecture. We initialise the convolutional head for RPL based on [6] and we apply 10 times the learning rate (with $7.5e^{-4}$) to the head compared with other convolutional layers that are trained. The images from Cityscape [3]

[1] https://fishyscapes.com/results
[2] https://segmentmeifyoucan.com/leaderboard

Table 3: **Ablations for CoroCL on Fishyscapes, SMIYC and RoadAnomaly validation sets.** We define the OoD, inlier with {△,🟢} in COCO context and {▲, 🔵} in city context. The best performance are in bold.

Anchor	Contrastive	Fishyscapes						SMIYC						RoadAnomaly		
		Static			L&F			Anomaly			Obstacle					
		FPR	AP	ROC	FPR	AP	ROC	FPR	AP	ROC	FPR	AP	ROC	FPR	AP	ROC
🔵▲	🔵▲	1.43	91.23	99.47	4.26	67.37	98.10	21.91	77.76	94.91	6.37	92.13	99.81	25.84	62.61	93.70
🟢△	🟢△	1.70	88.72	99.57	6.97	56.84	98.70	9.41	86.62	97.57	0.11	95.92	99.94	21.0	69.89	95.70
🔵▲	🟢🔵▲	1.79	89.90	99.52	3.74	68.17	98.95	11.31	86.37	97.27	0.29	94.31	99.93	26.78	66.91	93.87
🔵▲	🔵▲🟢△	**0.85**	**92.46**	**99.73**	**2.52**	**70.61**	**99.39**	**7.18**	**88.55**	**98.06**	**0.09**	**96.91**	**99.97**	**17.74**	**71.61**	**95.72**
🔵▲🟢△	🔵▲🟢△	1.57	90.24	99.58	4.90	60.72	98.53	15.29	82.94	97.18	0.10	96.58	99.96	19.62	68.97	95.44

Table 4: **The impact of the projector architecture** on the SMIYC-Anomaly and RoadAnomaly datasets.

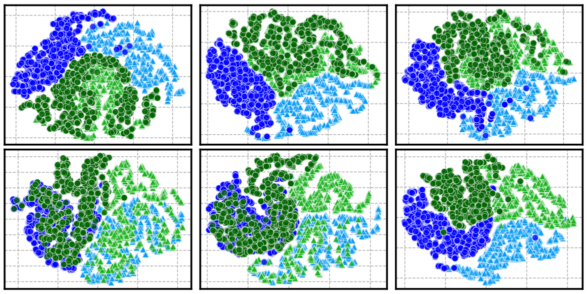
Architecture	SMIYC-Anomaly			RoadAnomaly		
	FPR	AuPRC	AuROC	FPR	AuPRC	AuROC
2 layers (w/o BN)	14.56	82.08	95.51	21.94	62.59	94.47
2 layers (w/ BN)	13.72	83.24	95.95	21.11	63.81	94.53
single-layer	**7.18**	**88.55**	**98.06**	**17.74**	**71.61**	**95.72**

are randomly cropped with 700 × 700 resolution, while the COCO [8] images are randomly scaled with ratio in $\{.1, .125, .25, .5, .75\}$ and then mixed with the city images based on outlier exposure (OE) [11]. Meanwhile, we copy the vanilla COCO images based on padding or centre cropping to the same resolution of 700 × 700 as city images for contrastive learning. **In the training of CoroCL**, we concatenate the context images from COCO and Cityscapes and extract the embeddings of both contexts via single forward propagation. We randomly choose **512** pixel-wise samples from both inlier and OoD in city and COCO contexts to perform CoroCL based on the Eq. (7) (from the main paper), where $\tau = 0.10$ for all the experiments.

We train the RPL module with one Tesla V100 16GB and RPL+CoroCL with one RTX A6000, as the contrastive learning needs more GPU memory. Following previous works [7, 12], we discard the projector head after training and directly utilise RPL outputs to induce the closed-segmentation to produce high-uncertainty in potential anomalous regions.

D. Results from Different Backbones

Tab. 2 displays the results of our approach with different backbones, while we measure them based on the area under the receiver operating characteristics (AuROC), average precision (AP), and false positive rate at a true positive rate of 95% (FPR). We report the closed-set segmentation performance in "Before" and our performance in "After", while the improvements in all backbones demonstrate the generalisation of our approach. For example, our approach improves the performance by 29.70%, 41.43% and 52.25% FPR in the RoadAnomaly validation set for MobileNet, ResNet50 and WiderResNet38, respectively.



case (a) case (b) case (c)

Fig †: T-SNE visualisation of RPL outputs **w/o CoroCL (first row)** and **with CoroCL (second row)**. Each column uses the same images, where city and non-city contexts inliers are 🔵 and 🟢, while city and non-city contexts outliers are ▲ and △. Better viewed in zoomed-in mode.

E. More Details of CoroCL

Impact of Projector Architecture. Differently from previous contrastive learning methods [12, 13], we find that a projector with a single-layer performs better than with a multi-layer, as shown in Tab. 4, which may be due to a better robustness to overfitting given the smaller number of layers. **Construction of Anchor and Contrastive Sets** We implement the ablation of the anchor and contrastive sets based on AuROC, AP and false positive rate at a true positive rate of 95% (FPR) in Tab. 3. The pixel-wise embedding samples in our training have OoD (△) and inlier (🟢) in the COCO context and OoD (▲) and inlier (🔵) in the city context, as shown in Fig. 2 in the main paper. The choice of the samples that build the anchor and contrastive sets will heavily impact the final performance. For example, using city context samples {🔵,▲} for both anchor and contrastive sets (in the first row) yields great performance in Fishyscapes (e.g., 91.23% AP in Static and 67.37% AP in L&F) but the poor performance in SMIYC. On the contrary, using COCO context samples {🟢,△} for both anchor and contrastive sets (in the second row) improves the results by 12.5% and 6.26% FPR in both Anomaly and Obstacle of SMIYC but demonstrates worse performance in Fishyscapes. The best performance is observed when we use {🔵,▲} to construct anchor set and {🔵, ▲, 🟢, △} to be the contrastive set, which

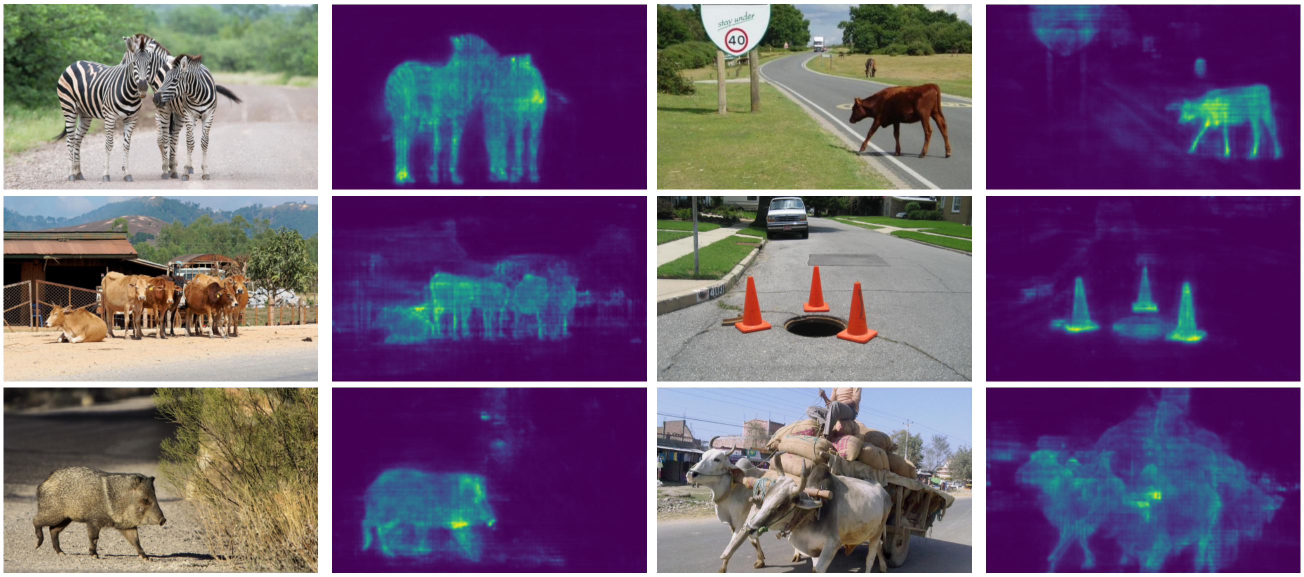


Figure 5: The self-attention results of the learned residual pattern feature from RoadAnomaly [9].

Table 5: Results on StreetHazards w/ LDN-121 net based on the closed-set checkpoint and evaluation code on https://github.com/matejgrcic/DenseHybrid. The Closed/Open sets are measured by mean IoU and we use *energy* to compute anomaly score. $*$ denotes the results from pretrained inlier model for both our approach and [5].

Method	Anomaly Detection			Closed-set	Open-set	
	FPR	AP	AuC		(t5)	(t6)
LDN-121*	15.6	16.7	95.1	**65.0**	39.3	44.5
DenseH [5]	13.0	30.2	95.6	63.0	46.1	45.3
RPL	**8.22**	**31.15**	**97.19**	**65.0**	**58.14**	**54.38**

achieves the reported performance in the main paper. Compared with our results (in the fourth row of Tab. 3.), the last row additionally enforces ●→ ← ●and ●← → ▲(based on Eq. 7 in the main paper). Due to the training and validation datasets based on the driving scenes, we suspect that the optimisation applied to the daily natural images (i.e., COCO contexts) damages the convergence of our approach, which yields unsatisfactory performance.

T-SNE visulaisation. As shown in Fig. †, we apply T-SNE on the outputs of RPL block for both city and other context images. Using the same images (each column), we randomly sample **4000** pixel-wise RPL embeddings. We observe the RPL results **w/o CoroCL (first row)** can only separate anomalies in city contexts (● and ▲), but fail in non-city contexts (● and ▲). **CoroCL (second row)** clusters the inliers from various scenes while pushing the outliers away, independently of city/non-city contexts.

F. Generalization and results in StreetHazard

RPL can be easily adopted by other FCN-based architectures by attaching the RPL module before the pixel-wise classifier head. For example, we easily attach the RPL module to the LDN-121 segmentation model, with results in Tab Tab. 5. Based on same pre-trained checkpoint, the RPL's results outperform the previous SOTA Dense-hybrid [5] with 4.8% improvement in FPR and over 10% mIoU in Open-set evaluation.

G. Visualisations of Residual Patterns

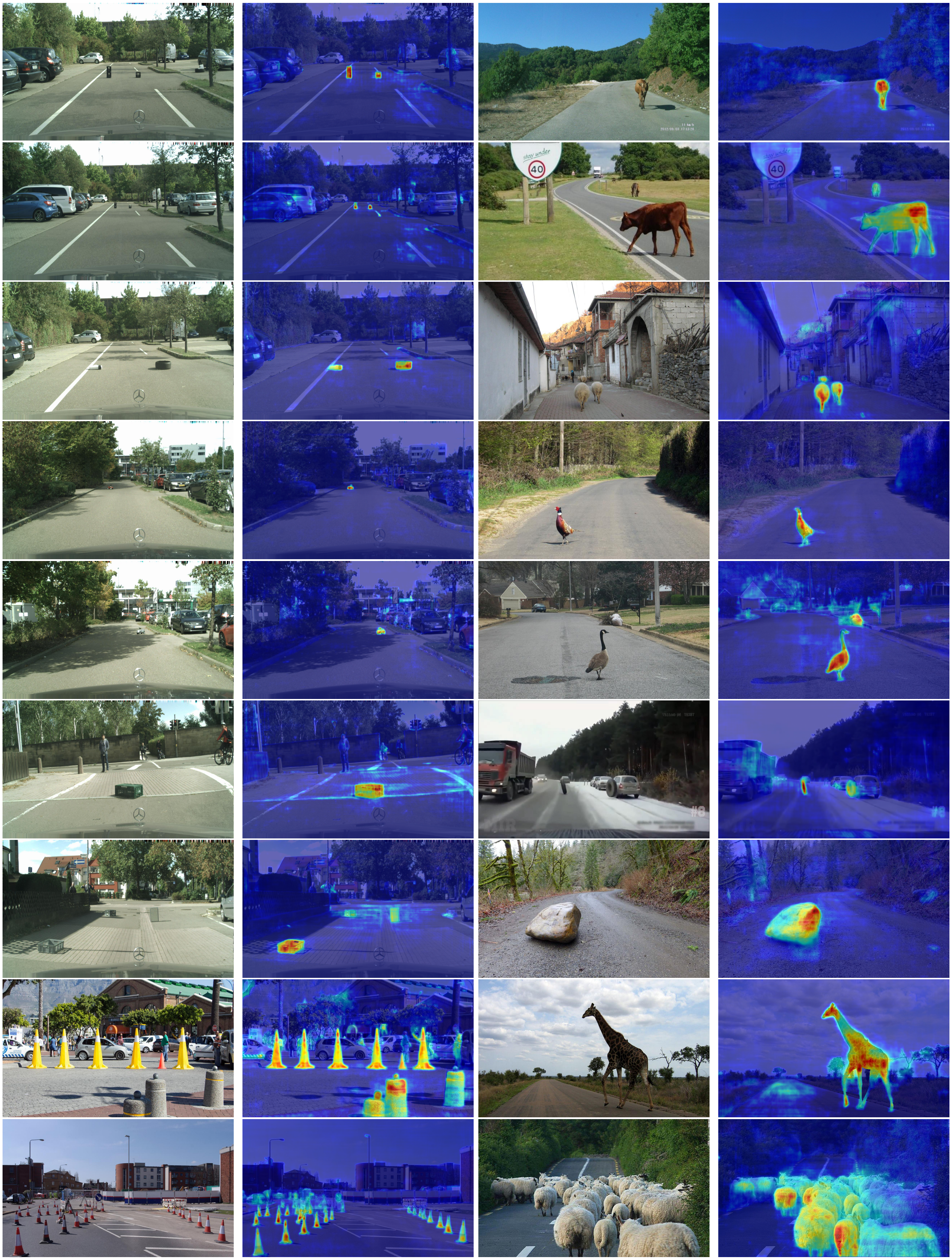
The learned residual patterns from the RPL feature map $\mathbf{r} \in \mathbb{R}^{304\times H\times W}$ can be visualised via self-attention and the pytorch-like code is in below:

```
torch.einsum('abc,bca->bc', r, r.permute(1,2,0)).
```

The visualisations of such learned residual patterns are shown in Fig. 5, where the anomaly objects are highlighted.

H. More Visualisations of OoD Maps

Fig. 6 shows the anomaly segmentation visualisation results of our method. The results indicate that our method successfully detects and segments anomaly objects in different scenarios, including various hard anomalies. Rows 1-5 of the city scenes demonstrate that our method can detect small and distant anomalies well, while rows 8 and 9 show the robustness of our method to many anomalous objects, which are successfully detected and segmented. The country context scene results show that our method can accurately segment hard objects. Rows 1-5 and 8 also show that our method accurately segments OoD animals. Row 9 of the country scene is a hard challenge for anomaly segmentation because most of the pixels in the image are anomalies, and it is difficult for previous methods to identify such large-scale anomalies. Nevertheless, our method can still detect the anomalies on the road in this hard case.



(a) City Context Scenes

(b) Country Context Scenes

Figure 6: **More visualisations** for our method in different contexts.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Dense anomaly detection by robust learning on synthetic negative data. *arXiv preprint arXiv:2112.12833*, 2021.

[5] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. *arXiv preprint arXiv:2207.02606*, 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[9] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2152–2161, 2019.

[10] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[11] Yu Tian, Yuyuan Liu, Guansong Pang, Fengbei Liu, Yuanhong Chen, and Gustavo Carneiro. Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. *arXiv preprint arXiv:2111.12264*, 2021.

[12] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021.

[13] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[14] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–416, 2018.

[15] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.