Supplementary Material for "Random Walk on Pixel Manifolds for Anomaly Segmentation of Complex Driving Scenes"

Zelong Zeng[1] and Kaname Tomite[1]

SenseTime Japan, Tokyo, Japan
{zengzelong,tomite}@sensetime.jp

Abstract. The following items are included in the supplementary material.

- The Architecture of Pixel-based and Mask-based Networks.
- Anomaly Scoring Function Details.
- Additional Results and Implementation Details.
- Different Methods for Constructing Locally Constrained Graph.
- Analysis for RWPM.
- Some Reviewers' Comments and Our Explanations.

S.1 The Architecture of Pixel-based and Mask-based Networks

In this section, we introduce the structures of pixel-based and mask-based segmentation models, respectively. Additionally, we also illustrate the position of the pixel embedding map used by RWPM in different network architectures. In the main paper, we utilize the Deeplabv3+ [2] with WideResNet38 as the pixel-based segmentation model, and the Mask2Former [3] with Swin-L as the mask-based segmentation model.

Fig 1 illustrates their architecture. Specifically, Fig 1(a) shows the architecture of the Deeplabv3+ with WideResNet38, which consists of three parts: a WideResNet38 backbone, a Atrous Spatial Pyramid Pooling (ASPP) layers and a Segmentation Head (Seg Head). During inference, the input image will be fed into the backbone first to produce a feature map. Then, this feature map will go through the ASPP layers to output a feature map with multiscale information. Finally, such feature map will be classified by the Seg Head to produce the dense predictions. The last layer of Seg Head is a classifier. In the main paper, we use the output from the penultimate layer of the Seg Head as the pixel embedding map for RWPM.

Fig 1(b) depicts the architecture of the Mask2Former with Swin-L, consisting of a Pixel Encoder-Decoder, a Transformer Decoder, a Linear Classifier and a Multilayer Perceptron (MLP). During inference, the input image is first fed into the Pixel Encoder-Decoder to generate a feature map. Simultaneously, the intermediate upsampled features from the Pixel Encoder-Decoder are fed to the

Transformer Decoder. Next, N learnable queries go through the Transformer Decoder with the intermediate upsampled features to produce N refined queries. These N refined queries are then fed into the Linear Classifier and MLP. The Linear Classifier is the classifier of queries, generating the class predictions of each refined query. Meanwhile, the MLP adjusts the dimension of each refined queries to match the size of the feature map from the Pixel Encoder-Decoder. After going through the MLP, the N refined queries can be seen as N binary classifiers, which are multiplied with the feature map from the Pixel Encoder-Decoder to produce the mask predictions. Finally, by combining the mask predictions with the corresponding class predictions, the model outputs the classification predictions for each pixel of the input image. In summary, each refined query generates a mask on the image, where the prediction scores on the mask represent the binary classification results of corresponding pixels, describing the confidence level of the refined query for that pixel. Then, the Linear Classifier categorizes each refined query, obtaining classification predictions for the corresponding refined queries. The classification prediction for each pixel is obtained by the weighted sum of the classification predictions of all refined queries. The weight is the mask prediction score generated by each refined query for that pixel. In other words, the higher the confidence level of a refined query for a pixel, the more similar the classification result of that pixel is to the classification result of the refined query. In the main paper, we use the output from the Pixel Encoder-Decoder as the pixel embeddings map for RWPM.

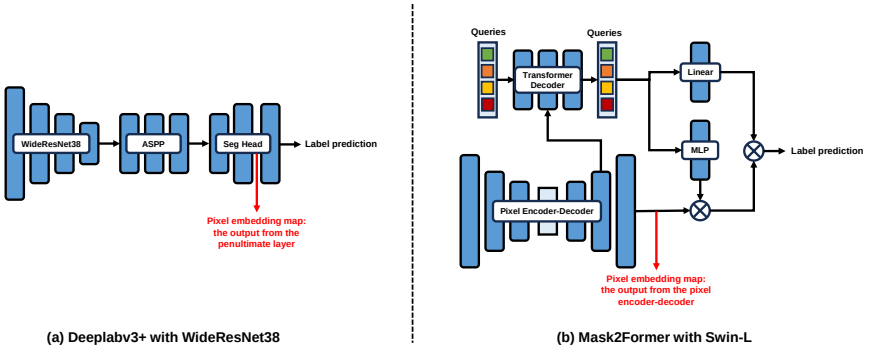


Fig. 1: The architecture of pixel-based and mask-based networks. (a) depicts the architecture of the pixel-based network used in the main paper, namely Deeplabv3+ with WideResNet38. (b) illustrates the architecture of the mask-based network used in the main paper, which is Mask2Former with Swin-L. The red arrows indicate the position of the pixel embeddings map used by RWPM in different network architectures.

S.2 Anomaly Scoring Function Details

In our main paper, we integrate our RWPM with four recent representative anomaly segmentation methods, namely PEBAL [8], Balanced Energy [4], RbA [6] and Mask2Anomaly [7]. In this section, we provide a detailed introduction to the anomaly scoring functions of these methods.

S.2.1 PEBAL and Balanced Energy

PEBAL and Balanced Energy utilize Deeplabv3+ (see Fig 1(a)), a pixel-based network, as their backbones. Both methods utilize the EBM function [5] as their anomaly scoring function. Let's define $\boldsymbol{l}_{h,w}(k)$ as the logit of a pixel $\boldsymbol{x}_{h,w}$ for class k. $\boldsymbol{l}_{h,w}(k)$ is calculated from the inner product of the pixel embedding and the vector of network weights for class k of the classifier. The anomaly scoring function at the pixel $\boldsymbol{x}_{h,w}$ is defined as:

$$E(\boldsymbol{x})_{h,w} = -\log \sum_{k \in \{1 \cdots K\}} \exp\left(\boldsymbol{l}_{h,w}(k)\right), \tag{S.1}$$

where K is the number of inlier class. Obviously, when the anomaly score $E(\boldsymbol{x})_{h,w}$ is large, it indicates that the logits of the pixel across all inlier classes are small, *i.e.*, the pixel $\boldsymbol{x}_{h,w}$ is dissimilar from all inlier classes.

S.2.2 RbA

RbA uses Mask2Former (see Fig 1(b)), a mask-based network, as its backbone. Let's define $\boldsymbol{l}_{h,w}(k)$ is the class logit of a pixel $\boldsymbol{x}_{h,w}$ for class k. In Mask2Former model, the $\boldsymbol{l}_{h,w}(k)$ is described as:

$$\boldsymbol{l}_{h,w}(k) = \sum_{i=1}^{N} P_n(\boldsymbol{x_{h,w}}; k) M_n(\boldsymbol{x_{h,w}}), \tag{S.2}$$

where N is the total number of refined query in the network, $M_n(\boldsymbol{x_{h,w}})$ denotes the mask prediction score, generated by the n-th query, on the pixel $\boldsymbol{x_{h,w}}$, and $P_n(\boldsymbol{x_{h,w}}; k)$ indicates the k-th class predicted probability of the n-th query. In other words, the class logit $\boldsymbol{l}_{h,w}(k)$ is the weighted sum of the classification predictions of each query, and the weights are the mask prediction scores generated by each query for the pixel $\boldsymbol{x_{h,w}}$. Based on the Eq S.2, the anomaly scoring function RbA is defined as:

$$\text{RbA}(\boldsymbol{x_{h,w}}) = -\sum_{k=1}^{K} \sigma(\boldsymbol{l}_{h,w}(k)), \tag{S.3}$$

where K is the total number of inlier class. When the anomaly score has a large value, it indicates that all class logits, $\boldsymbol{l}_{h,w}(k)$ where $k \in \{1, \cdots, K\}$, are small. This means that the mask scores generated by each refined query for this pixel are all small, indicating that the pixel is not similar to any inlier class.

S.2.3 Mask2Anomaly

Similar to RbA, Mask2Anomaly also uses Mask2Former as its backbone. Therefore, its definition of the class logit $\boldsymbol{l}_{h,w}(k)$ is the same as Eq S.2. Mask2Anomaly compute the anomaly score $f(\boldsymbol{x_{h,w}})$ as:

$$f(\boldsymbol{x_{h,w}}) = 1 - \max_{k=1}^{K} \sigma(\boldsymbol{l}_{h,w}(k)), \tag{S.4}$$

When the value of $f(\boldsymbol{x_{h,w}})$ is large, it implies that the value of $\max_{k=1}^{K} \sigma(\boldsymbol{l}_{h,w}(k))$ is small, meaning that all class logits, $\boldsymbol{l}_{h,w}(k)$ where $k \in \{1, \cdots, K\}$, are small. This indicates that the mask scores generated by each refined query for this pixel are small, implying that the pixel is not similar to any inlier class.

S.3 Additional Results and Implementation Details

In our main paper, we validate that the proposed RWPM can improve the performance of existing anomaly segmentation methods by integrating our approach with four recent representative methods (see Section 4.1 of the main paper). However, due to page limitations, we only present results on the Fishyscapes Lost & Found dataset and the Road Anomaly dataset. Therefore, in Table S.1, we provide additional results, including on the Anomaly track and Obstacle track of the SMYIC dataset. From Table S.1, it demonstrates that our proposed RWPM consistently and significantly enhances the performance of existing anomaly segmentation methods across different datasets.

Furthermore, in Table S.2, we present the implementation details of the RWPM parameters α (the transition probability) and T (the number of iteration) in each experiments in Table S.1.

Table S.1: Comparison with strong and representative anomaly segmentation baselines across different backbone Architectures (Arch). **Bold** denotes the better results between using RWPM and not using RWPM with the same anomaly segmentation baselines. † indicates that use the calibration in sub-map concatenation as described in Section 3.4 of the main paper.

Benchmark→		Fishyscapes Lost&Found			Road Anomaly			Anomaly Track		Obstacle Track	
Method ↓	Arch	AuROC↑	AP↑	FPR95↓	AuROC↑	AP↑	FPR95↓	AP↑	FPR95↓	AP↑	FPR95↓
PEBAL [8]	Pixel-based	98.96	58.81	4.77	87.63	45.10	44.58	49.13	40.87	4.96	12.71
PEBAL + RWPM† (Ours)	Pixel-based	**99.20**	**66.85**	**3.68**	**89.48**	**50.29**	**36.81**	**51.49**	**34.04**	**14.47**	**11.23**
Balanced Energy [4]	Pixel-based	99.03	67.07	2.93	88.31	41.48	41.46	47.70	52.00	9.07	18.78
Balanced Energy + RWPM† (Ours)	Pixel-based	**99.29**	**72.95**	**2.38**	**90.51**	**48.26**	**32.10**	**50.44**	**51.13**	**20.00**	**16.65**
Mask2Anomaly[1] [7]	Mask-based	95.47	65.27	7.79	96.54	80.04	13.95	88.62	**14.57**	93.10	0.20
Mask2Anomaly + RWPM (Ours)	Mask-based	**95.48**	**65.38**	**7.33**	**97.44**	**80.09**	**7.45**	**88.98**	15.88	**93.82**	**0.18**
RbA [6]	Mask-based	98.62	70.81	6.30	97.99	85.42	6.92	90.86	11.59	91.85	0.46
RbA + RWPM (Ours)	Mask-based	**98.82**	**71.16**	**6.12**	**98.04**	**87.34**	**5.27**	**92.00**	**10.15**	**93.30**	**0.28**

[1] We carefully reproduce the experiments by using the official code.

Table S.2: The implementation detail of RWMP parameters α (the transition probability) and T (the number of iteration) for each experiment in in Table S.1. † indicates that use the calibration in sub-map concatenation as described in Section 3.4 of the main paper.

Benchmark→	Fishyscapes L&F		Road Anomaly		Anomaly Track		Obstacle Track	
Method ↓	α	T	α	T	α	T	α	T
PEBAL + RWPM† (Ours)	0.99	5	0.99	20	0.99	5	0.99	5
Balanced Energy + RWPM† (Ours)	0.90	5	0.99	20	0.99	5	0.99	5
Mask2Anomaly + RWPM (Ours)	0.10	5	0.99	20	0.90	5	0.70	5
RbA + RWPM (Ours)	0.50	5	0.99	20	0.99	5	0.90	5

S.4 Different Methods for Constructing Locally Constrained Graph

In our main paper (see Section 3.2), we utilize a Softmax function to obtain locally constrained graph. In this section, we first present a comparison between using the Softmax function and employing k-NN search algorithm in our RWPM. When using the k-NN algorithm, for each pixel, we only retain its connections to the k most similar pixels (neighborhoods) in the graph $\mathbf{S}$. Table S.3 shows the results. We observe that when using the k-NN algorithm, as the value of k decreases (from $k = 500$ to $k = 50$), the performance improves due to the reduction of noise effect. This demonstrates the effectiveness of the locally constrained graph. However, when the value of k becomes too small (from $k = 50$ to $k = 10$), it leads to the loss of relationship information between pixels, resulting in a decrease in performance. Nevertheless, although the k-NN algorithm can achieve good performance, it significantly reduces operational efficiency (with only 0.93 FPS). In contrast, our proposed Softmax method can achieve better performance and higher operational efficiency (with 2.04 FPS). This demonstrates the superiority of our Softmax method.

Table S.3: The comparison between the performance of Softmax functions and k-NN search algorithms using different k values. We set $T = 20$ and $n = 2$ for all experiments. **Bold** and underline denote the best and the second best results, respectively.

Benchmark→	Road Anomaly		
	AP↑	**FPR95↓**	**FPS↑**
k-NN ($k = 10$)	86.38	6.57	0.93
k-NN ($k = 20$)	86.65	6.47	0.93
k-NN ($k = 50$)	<u>86.71</u>	<u>5.96</u>	0.93
k-NN ($k = 100$)	86.24	6.00	0.93
k-NN ($k = 200$)	84.61	6.72	0.93
k-NN ($k = 500$)	81.00	19.08	0.93
Softmax ($\tau = 0.01$)	**87.34**	**5.27**	**2.04**

Additionally, we show the effect of the temperature parameter τ in the Softmax function. A smaller τ results in fewer neighborhoods being connected to each pixel in the graph $\mathbf{S}$, which resembles the reduction of the k value in k-NN. Table S.4 presents the results. We can observe a similar phenomenon to that in Table S.3, where the locally constrained graph can improve the performance, but when too few pixels are connected to each pixel, it may actually decrease the performance. In our experiments, the best results were achieved when $\tau = 0.01$.

Table S.4: The effect of temperature parameter τ. We set $n = 2$ and $T = 20$ for all experiments. **Bold** denotes the best results.

Benchmark→	Road Anomaly	
	AP↑	**FPR95↓**
RbA w/o RWPM	85.42	6.92
$\tau = 1.0$	21.60	79.25
$\tau = 0.1$	31.45	68.65
$\tau = 0.01$	**87.34**	**5.27**
$\tau = 0.001$	85.97	6.70
$\tau = 0.0001$	85.57	6.93

S.5 Analysis for RWPM

In this section, we analyze why RWPM can optimize pixel embeddings. In the main paper, the process of RWPM can be described as:

$$\mathbf{m}^{t+1} = \alpha \mathbf{S}\mathbf{m}^{t} + (1-\alpha)\,\mathbf{m}^{0},\ \alpha \in (0,1), \tag{S.5}$$

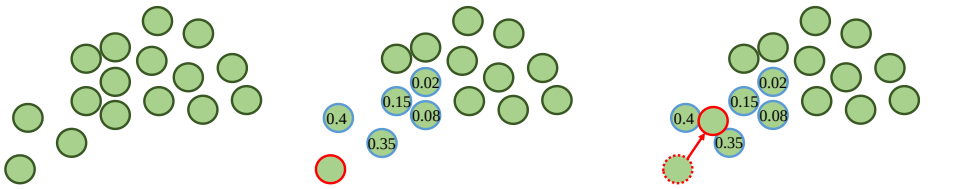
where $\mathbf{S}$ denotes the graph representing the manifolds of pixel embeddings, $\mathbf{m}^0$ denotes the original pixel embeddings, $\mathbf{m}^t$ denotes the updated pixel embeddings in the t-th iteration of the random walks and α denotes the continuing probability. Eq S.5 shows that each updated pixel embedding is obtained by taking a weighted ensemble of other pixel embeddings. The weights are determined by the similarities between the pixels depicted on graph $\mathbf{S}$. As shown in Fig 2, after updating, each pixel will become more similar with its neighborhoods. In other words, all pixel embeddings in the same manifolds will become more similar and form a more compact cluster.

Additionally, similar to the derivation in [1], the closed-form solution of Eq S.5 can be formulated as the following optimization problem:

$$\min_{\mathbf{m}^{\infty}} \frac{1}{2} \sum_{i,j=1}^{HW} \mathbf{S}_{ij} \left(\mathbf{m}_i^{\infty} - \mathbf{m}_j^{\infty}\right) + \frac{1-\alpha}{\alpha} \sum_{i=1}^{HW} \left(\mathbf{m}_i^{\infty} - \mathbf{m}_i^{o}\right)^2, \tag{S.6}$$

where $\mathbf{m}^{\infty}$ is the finial updated embeddings, $\mathbf{m}_i$ denotes the embedding of the i-th pixel and $\mathbf{S}_{ij}$ depicts the similarity between the i-th and j-th pixels on the

manifolds. We can observe that Eq S.6 consists of two terms: The first term indicates that if two pixels are more similar on manifolds, their final embeddings will also be more similar. In other words, the first term considers higher-order information among pixels. For example, if the i-th pixel is similar to the j-th pixel, and the j-th pixel is similar to the k-th pixel, then in the final state, the embedding of the i-th pixel will also be similar to that of the k-th pixel. These encourage pixels on the same manifold to form more compact clusters, consistent with the description and experimental results in the main paper. The second term suggests that the initial embeddings should be retain to a certain degree. In our limited iteration strategy, $\mathbf{m}^t$ is regarded as an approximation of $\mathbf{m}^\infty$.



(a) Pixel Distribution (b) Pixel and Its Neighborhoods (c) Pixel Embedding Updating

Fig. 2: The process of updating pixel embeddings. (a) shows the distribution of pixel embeddings, all data points belong to the same category. (b) shows a pixel embedding (red circle) that deviates from the clustering and its 5 neighborhoods (blue circle). The number inside each neighborhood (blue circle) represents its similarity to the pixel embedding (red circle). (c) shows that, after the computation by the weighted ensemble, the updated pixel embedding moves closer to the clustering.

S.6 Some Reviewers' Comments and Our Explanations

Comment#1 [Why are random walks particularly effective for anomaly segmentation?]:

Existing anomaly score functions rely on the logits output by segmentation models to infer anomaly scores. However, diverse real-world driving scenarios often distort manifolds of pixel features (but most pixels of same class still lie on the same manifold), leading to inaccurate anomaly score inference (see Fig. 1 of the main paper). Thus, we employ random walks on manifolds to propagate and update pixel features. Random walks can measure the similarity of pixels on manifolds (*i.e.*, distance on manifolds rather than Euclidean distance), resulting in high similarity for pixels on the same manifold and low similarity for pixels on different manifolds. This forms more compact clusters on each manifold, mitigating manifold distortion. This is why random walks are effective for anomaly segmentation.

Comment#2 [The proposed partitioning strategy sounds reasonable, but what effect does it have when anomaly parts are split among different sub-maps? Did the authors observe this as an issue?]:
From the analysis of numerous samples, we observe that when the number of sub-maps n is small, the high similarity between sub-maps contents ensures that dividing the anomaly target into separate parts does not affect the results. However, when n is larger, our proposed calibration method can alleviate the issue (see Section 3.4 of the main paper). Naturally, finding more elegant solutions to this problem will be part of our future work.
Comment#3 [It would be interesting to know RWPM's limitations and which solutions could be adopted to alleviate/solve such limitations.]:
One limitation of RWPM is running efficiency. Although we have proposed Partial Random Walk and verified its effectiveness in improving efficiency, there are still some methods that can further enhance operational efficiency in practical applications. For instance, in practical applications, we can use image segmentation results to remove background elements (*e.g.*, sky) or other irrelevant regions. This significantly reduces the pixel count and improves computational efficiency, helping to meet real-time requirements. Besides, as we mentioned in the conclusions section, another limitation is that RWPM's effectiveness depends on the model's quality. Training models with RWPM to enhance generalization is a promising solution.

References

1. Bai, S., Bai, X., Tian, Q., Latecki, L.J.: Regularized diffusion process for visual retrieval. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017). https://doi.org/10.1609/aaai.v31i1.11216
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018). https://doi.org/10.1007/978-3-030-01234-2_49
3. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022). https://doi.org/10.1109/CVPR52688.2022.00135
4. Choi, H., Jeong, H., Choi, J.Y.: Balanced energy regularization loss for out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15691–15700 (2023)
5. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020)
6. Nayal, N., Yavuz, M., Henriques, J.F., Güney, F.: Rba: Segmenting unknown regions rejected by all. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 711–722 (2023)
7. Rai, S.N., Cermelli, F., Fontanel, D., Masone, C., Caputo, B.: Unmasking anomalies in road-scene segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4037–4046 (2023)

8. Tian, Y., Liu, Y., Pang, G., Liu, F., Chen, Y., Carneiro, G.: Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. In: European Conference on Computer Vision. pp. 246–263. Springer (2022). https://doi.org/10.1007/978-3-031-19842-7_15