Appendix of "On the Robustness of Open-World Test-Time Training: Self-Training with Dynamic Prototype Expansion"

Yushu Li [1] Xun Xu [2†] Yongyi Su [1] Kui Jia [1†]
[1] South China University of Technology
[2] Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR)
{eeyushuli, eesuyongyi}@mail.scut.edu.cn, {alex.xun.xu, kuijia}@gmail.com

In this supplementary material, we first declare the limitations and failure cases for the proposed method. Then We perform more extensive experiments, including compatibility with the transformer-based model, ViT[1], robustness to different data streaming orders, and the evaluation results on 3D point cloud data. In addition, we further conduct some visualization experiments to provide a more intuitive demonstration of how our approach works.

1. Limitation and Failure Case

We explicitly discuss the limitation of the proposed method and analyze the failure case when the assumptions are violated. Our method is built to resolve a common scenario during open-world test-time training, i.e. target domain consists of strong OOD samples. Nevertheless, we do not rule out the scenario where the target domain is not contaminated with strong OOD samples. As our method relies on detecting strong OOD samples to improve robustness, it may mistakenly treat some weak OOD samples as strong OOD ones. As such, *test-time training performance may be compromised when the target domain is not contaminated with strong OOD samples, and this is not a known priori*. To verify this limitation, we first evaluate our method, as well as other competing methods, on the "clean" CIFAR10-C test set, i.e. the target domain only contains the CIFAR10-C test set. To enable OOD detection under a potentially single-modal OOD score distribution, we restrict the strong OOD detection threshold to between 0.4 and 1.0. We compare the results with the target domain contaminated with random noise as strong OOD samples. As shown in Tab. 1, we make the following observations from the results. First, when it is known a priori that the target domain only contains weak OOD samples, our method without using an OOD detector performs comparably to the state-of-the-art TTT methods, e.g. TTAC. However, all competing methods with strong OOD detectors would suffer when this prior knowledge is not available,

Table 1: The performance under CIFAR10-C with (w/ Str. OOD) and without (w/o Str. OOD) strong OOD samples.

Method	OOD Det.	w/o Str. OOD Acc_S	w/ Str. OOD Acc_H
TEST	×	70.59	N/A
BN	×	79.73	N/A
TENT	×	80.91	N/A
SHOT	×	82.58	N/A
TTT++	×	80.74	N/A
TTAC	×	**86.36**	N/A
Ours	×	<u>85.25</u>	N/A
TEST	✓	54.19	81.36
BN	✓	74.22	<u>85.11</u>
TENT	✓	76.70	32.77
SHOT	✓	74.64	67.23
TTT++	✓	76.86	47.86
TTAC	✓	<u>78.45</u>	70.35
Ours	✓	**78.63**	**91.56**

e.g. Ours drops from 85.25% to 78.63% on clean accuracy. Nevertheless, our proposed method still outperforms all competing methods with a large margin on the harmonic mean Acc_H and there is a good balance between clean accuracy and accuracy under strong OOD samples when OOD detection is included. In contrast, without strong OOD detection, all methods fail to identify strong OOD samples. As direct calculating Acc_H under this circumstance yields $Acc_H = 0$, we use N/A to indicate this situation. Overall, it still remains an open question of how to trade off a balance between being robust to strong OOD samples and maintaining good performance when the target domain only contains clean OOD samples.

2. Compatibility with Transformer Backbone

In this section, we perform additional experiments with ViT backbone [1]. The ViT model pre-trained in the clean CIFAR-10 dataset is utilized as the source domain model. Then we test it on the Cifar10-C test set under the strongest corruption level. All experiments were conducted under our OWTTT protocol, where random noise, MNIST, SVHN,

†Corresponding authors

CIFAR100-C, and Tiny-ImageNet are respectively selected as strong OOD data. The results presented in Tab. 2 demonstrated that our method is compatible with a more advanced backbone network.

3. Data Streaming Order

In this section, we explore the impact of the testing data streaming order on our approach. We randomly shuffled the test data four times and performed test-time Training separately. The experimental results are shown in Tab. 3. Since we use a moving average queue N_m to select the optimal threshold τ^*, which is less affected by the data streaming order, our method demonstrates strong stability regardless of the order of data streaming.

4. Imapct of Thresholding Ratio

To further reduce the effect of incorrect pseudo labeling, we only use 50% of samples with od_i far from τ^* to perform prototype clustering for each batch. We explored the proportions of testing samples for clustering by setting the proportions to 25% 50% 75% and 100%, and using CIFAR10-C as the weak OOD. The results are presented in Tab. 4. It is evident that our method is not sensitive to the proportion of used pseudo labels.

5. Additional Details

Source Domain Prototypes: We obtain the source domain prototypes by first running inference on all source domain training samples. More specifically, the prototypes are obtained via the following equation.

$$p_k = \frac{1}{\sum\limits_{y_i \in \mathcal{D}_s} \mathbb{1}(y_i = k)} \sum_{x_i, y_i \in \mathcal{D}_s} \mathbb{1}(y_i = k) \cdot f(x_i) \quad (1)$$

6. Evaluation on 3D Point Cloud Data

To demonstrate the applicability of our proposed method across diverse tasks, we extended the OWTTT protocol to the 3D point cloud classification task.

Method: We maintain consistency with the methods in the manuscript, making only minor adaptations specifically for 3D point cloud data. Due to the inherently discrete nature of point cloud data compared to image data, we employ strong OOD prototypes $\mathcal{P}_u$ and weak OOD prototypes $\mathcal{P}_s$ together to enhance the discriminative power of the Strong OOD Score OS'_i, defined as Eq. 2. To enable the adaptation of $\mathcal{P}_u$ with respect to changes in the test data, we employ a momentum-based updating approach. Specifically, for a given sample x_s that is predicted as a strong OOD instance, we update the most similar strong OOD prototype p_i in $\mathcal{P}_u$, as shown in Eq. 3.

$$\begin{aligned} OS'_i &= (1 - s_s) \cdot \frac{s_s}{s_s + s_u} + s_u \cdot \frac{s_u}{s_s + s_u} \\ s_s &= \max_{p_k \in \mathcal{P}_s} < f(x_i), p_k > \\ s_u &= \frac{1}{10} \sum_{j=1}^{10} d_{ij} \\ & s.t. \quad \{d_{ij}\}_{j=0}^{|\mathcal{P}_u|-1} = sort(\{< f(x_i), p_k >\}_{p_k \in \mathcal{P}_u}) \\ & and \quad d_{i0} = \max_{p_k \in \mathcal{P}_u} < f(x_i), p_k > \end{aligned} \quad (2)$$

$$p_i = (1 - \delta) p_i + \delta f(x_s) \quad (3)$$

Datasets: We choose ModelNet40-C [3] as weak OOD, which consists of 15 common corruptions of point cloud data, with 9,843 training samples and 2,468 test samples. We select random noise and the 3D representation of MNIST [4] as strong OOD.

Training Details: We follow [2] and use the DGCNN [5], with learning rate α=1e-4, batch size N_B = 64, λ = 1.

Results: We observe from the results in Tab. 5 that our method outperforms all competing methods on the point cloud datasets. The results demonstrate that our proposed method also exhibits a strong fit for 3D point cloud data, showcasing its potential for broader application in various fields.

7. Adaptive Threshold VS Fixed Threshold

We visualize testing samples on CIFAR10-C with SVHN as strong OOD via t-SNE in Fig. 1 to compare the fixed threshold and our adaptive threshold. Green, black and red dots indicate correctly classified weak OOD samples, correctly classified strong ODD samples and misclassified samples respectively. We clearly observe fewer misclassified samples with adaptive thresholds, suggesting the advantage.

(a) Adaptive Threshold (b) Fixed Threshold

Figure 1: T-SNE visualizations of adaptive threshold and fixed threshold on CIFAR10-C with SVHN as strong OOD.

8. Dynamic Representations

We further present a t-SNE visualization at four different stages (indicated by the percentage TTT progress) of TTT in

Table 2: Open-world test-time training under ViT backbone.

Method	Noise			MNIST			SVHN			Tiny-ImageNet			CIFAR100-C		
	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H
TEST	86.22	**100.00**	92.60	82.24	**96.52**	88.81	78.82	92.71	85.20	82.29	71.41	76.46	80.48	76.50	78.44
TENT	88.56	99.93	93.90	**87.49**	91.65	**89.52**	75.98	49.51	59.95	84.16	61.63	71.16	78.91	56.50	65.85
SHOT	89.37	85.88	87.59	85.68	76.18	80.65	78.17	50.93	61.67	89.22	63.13	73.94	**86.44**	62.96	72.85
TTAC	90.14	**100.00**	94.81	77.28	54.01	63.58	85.03	92.51	88.61	85.55	68.09	75.82	85.30	74.06	79.29
OURS	**92.47**	**100.00**	**96.09**	73.67	65.22	69.19	**89.53**	**98.50**	**93.80**	**90.30**	**78.75**	**84.13**	83.12	**82.97**	**83.05**

Table 3: The performance of our method under different random seeds.

Seed	Noise			MNIST			SVHN			Tiny-ImageNet			CIFAR100-C		
	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H
#1	85.46	98.60	91.56	83.89	97.83	90.32	84.99	87.94	86.44	71.77	84.71	77.70	74.08	84.64	79.01
#2	85.00	98.40	91.21	84.40	99.12	91.17	85.19	88.38	86.76	72.63	83.25	77.58	75.69	85.09	80.11
#3	85.57	98.79	91.71	84.48	99.01	91.17	85.26	87.94	86.58	72.46	82.37	77.10	73.89	84.09	78.66
#4	85.35	98.37	91.40	84.04	98.14	90.54	85.29	89.62	87.40	71.82	84.09	77.47	75.00	85.74	80.01

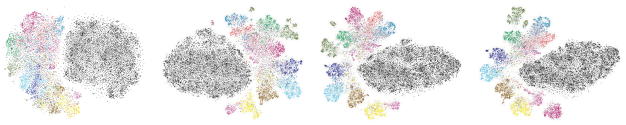
Table 4: The performance of our method under different thresholding rates.

Rate	Noise			MNIST			SVHN			Tiny-ImageNet			CIFAR100-C		
	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H
25%	84.25	97.64	90.45	83.68	97.65	90.13	84.88	84.88	84.88	72.83	80.04	76.26	75.0	79.83	77.34
50%	85.39	98.69	91.56	83.89	97.71	90.27	85.00	88.03	86.49	71.77	84.71	77.70	74.22	84.39	78.98
75%	85.87	98.82	91.89	84.22	97.95	90.57	85.00	90.99	87.89	69.44	84.94	76.41	72.69	86.70	79.08
100%	85.17	97.12	90.76	84.07	97.63	90.35	84.56	92.84	88.51	67.25	85.37	75.23	69.12	87.50	77.23

Table 5: Open-world test-time training on point cloud data.

Method	Noise			3DMNIST		
	Acc_S	Acc_N	Acc_H	Acc_S	Acc_N	Acc_H
TEST	44.77	88.77	59.52	42.11	79.94	55.16
BN	58.06	85.20	69.06	47.81	69.72	56.73
TENT	19.50	60.37	29.47	17.33	57.22	26.60
SHOT	62.75	79.79	70.25	**61.58**	78.74	69.11
TTAC	49.99	87.20	63.55	43.97	77.02	55.98
OURS	**69.15**	**93.39**	**79.46**	59.24	**88.95**	**71.12**

Fig. 2. It is obvious that different semantic classes (colorful dots) become better separated as TTT progresses and the strong OOD samples (black dots) are always well separated from weak OOD ones.



(a) 0% (b) 33% (c) 67% (d) 100%

Figure 2: T-SNE visualizations on CIFAR10-C with SVHN as strong OOD samples as TTT progresses.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[2] Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. In *Advances in Neural Information Processing Systems*, 2022.

[3] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, Z Morley Mao, Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, et al. Benchmarking robustness of 3d point cloud recognition against common corruptions. In *International Conference on Machine Learning*, 2021.

[4] Thunguyenphuoc. 3dmnist. https://github.com/thunguyenphuoc/3DMNIST, 2016.

[5] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.