3D Test-time Adaptation via Graph Spectral Driven Point Shift

Xin Wei Qin Yang Yijie Fang Mingrui Zhu Nannan Wang✉
State Key Laboratory of Integrated Services Networks,
School of Telecommunications Engineering, Xidian University
{weixin, mrzhu, nnwang}@xidian.edu.cn {qinyang, fangyijie}@stu.xidian.edu.cn

1. Experiments with Additional Backbone

In Sect. 4, we evaluated the efficacy of the proposed GSDTTA on the ModelNet40-C [1] and ScanObjectNN-C [2] benchmarks using DGCNN [3], CurveNet [4], and PointNeXt [5] as the point cloud classification backbone f_θ across all comparable methods. To further assess the robustness and adaptability of GSDTTA across diverse architectures, we extend our experiments to RSCNN [6], with results shown in Table 1.

As illustrated in Table 1, GSDTTA achieves the highest mean accuracy across all methods on both datasets with RSCNN [6]. Specifically, GSDTTA attains a mean accuracy of 78.03% on ModelNet40-C, surpassing the second-best TTA method SHOT [7] by 0.48%. For the experiments on ScanObjectNN-C, which represents real-world scenarios, GSDTTA maintains its strong performance. Our GSDTTA achieves a mean accuracy of 58.44%, outperforming the previous state-of-the-art 3DTTA methods Cloudfixer [8] by 1.18%. The consistent improvements across datasets and backbone networks demonstrate GSDTTA could generalize well across different architectures and corruption types.

2. Comparble Methods in Experiments

In Sect. 4, we evaluate GSDTTA against various TTA methods on the ModelNet40-C and ScanObjectNN-C datasets. The comparison includes 2D TTA methods such as BN [9], PL [10], DUA [11], TENT [12], and SHOT [7], as well as 3D-specific approaches like BFTT3D [13] and Cloudfixer [8]. Below, we provide a detailed overview of these methods along with their experimental configurations and hyperparameters.

BN. Batch Normalization (BN) [9] enhances model robustness by replacing the activation statistics of BatchNorm layers with estimated statistics (i.e., μ and σ) derived from the incoming batch of testing point clouds. This approach requires no additional test-time hyperparameters. The results reported are obtained using the publicly available implementation provided at https://github.com/jiachens/ModelNet40-C.

PL. Pseudo-Labeling (PL) [10] assigns pseudo-labels to unlabeled data by selecting the model's highest-confidence predictions and updates the model at test time using cross-entropy loss. Two essential hyperparameters for this method, the learning rate and the number of iteration steps, are configured as 0.01 and 1, respectively. We get the results utilizing the official code repository available at https://github.com/shimazing/CloudFixer.

DUA. Dynamic Unsupervised Adaptation (DUA) [11] recalibrates BatchNorm layer statistics by directly utilizing each incoming test batch, applying moving averages to update these statistics without requiring backpropagation. For this method, the number of iteration steps and decay factor, which regulate the rate of the moving average update, are set to 5 and 0.9, respectively. We reproduce the results using the implementation accessible at https://github.com/shimazing/CloudFixer.

TENT. Test-time Entropy Minimization (TENT) [12] optimizes the scale and shift parameters of BatchNorm layers by minimizing entropy, while disregarding batch normalization statistics from the source data. The critical test-time hyperparameters for this approach, the learning rate and the number of iteration steps, are set to 0.001 and 10, respectively. The official repository used to reproduce the results is avalible at https://github.com/jiachens/ModelNet40-C.

SHOT. Source HypOthesis Transfer (SHOT) [7] aligns target domain representations with the source hypothesis by leveraging information maximization and self-supervised pseudo-labeling. To further enhance pseudo-label accuracy, SHOT employs K-means clustering. The method relies on three test-time hyperparameters: the learning rate, the number of iteration steps, and the pseudo-label loss weight, which are set to 0.0001, 5, and 0.2, respectively. The

Dataset	Method	uniform	gaussian	background	impulse	upsampling	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	occlusion	lidar	Mean
ModelNet40-C [1]	Source-only	78.93	71.27	28.36	35.82	83.71	74.03	76.99	82.01	76.62	74.55	42.83	77.55	74.39	42.50	33.51	63.54
	BN [9]	85.29	83.10	67.58	70.58	87.15	78.76	80.87	85.21	83.42	79.70	60.78	83.67	78.56	50.20	45.75	74.71
	PL [10]	86.35	83.63	69.41	74.23	83.75	79.57	81.97	84.44	82.29	78.73	62.07	82.82	79.90	47.33	45.87	74.82
	DUA [11]	85.86	83.31	69.73	72.45	86.71	79.05	81.65	85.17	83.79	79.50	62.31	84.20	78.77	50.85	47.24	75.37
	TENT [12]	86.87	84.40	70.95	74.51	86.79	80.06	82.09	85.41	84.48	80.14	63.61	**84.84**	79.25	50.57	48.10	76.14
	SHOT [7]	86.55	85.53	75.61	78.12	84.80	**80.83**	**83.14**	**86.26**	**84.84**	**81.60**	**66.17**	84.16	**80.71**	**52.96**	**51.90**	77.55
	BFTT3D [13]	79.34	72.16	30.40	41.07	84.05	73.50	76.66	83.08	78.21	75.20	44.93	79.59	74.51	43.55	33.89	64.68
	Cloudfixer [8]	**89.16**	**89.45**	77.19	**88.68**	**90.22**	80.07	81.21	78.04	73.90	75.08	66.07	78.77	76.34	38.92	36.81	74.66
	GSDTTA (ours)	86.14	83.95	**88.45**	82.41	87.40	80.11	82.13	86.22	84.72	80.71	64.22	83.79	79.66	51.66	48.82	**78.03**
ScanObjectNN-C [2]	Source-only	43.20	33.91	32.87	34.42	49.39	66.09	67.47	70.91	65.06	67.81	55.59	64.89	68.85	11.02	13.43	49.66
	BN [9]	53.36	46.47	38.04	44.23	58.18	**68.50**	68.50	71.77	71.43	69.19	60.58	69.02	68.85	11.53	12.05	54.11
	PL [10]	56.45	51.81	34.77	45.96	55.59	65.75	66.44	71.08	69.71	68.16	60.41	65.92	66.61	12.22	10.84	53.45
	DUA [11]	57.14	51.81	37.18	47.16	59.38	67.47	68.85	73.49	71.43	69.19	**62.31**	69.53	68.16	11.19	10.84	55.01
	TENT [12]	56.28	52.66	37.01	47.85	58.69	66.44	68.33	72.63	70.40	69.02	**62.31**	68.33	67.30	11.02	11.70	54.66
	SHOT [7]	57.83	54.91	36.83	52.32	56.80	65.92	65.92	70.57	69.02	67.30	59.04	66.27	66.09	12.05	9.81	54.05
	BFTT3D [13]	45.31	39.41	33.33	41.15	53.30	66.15	68.92	68.23	60.76	65.62	55.73	61.46	**69.62**	**12.85**	**13.54**	50.36
	Cloudfixer [8]	**66.15**	**64.24**	49.48	**72.05**	**76.04**	61.98	67.36	63.89	63.89	65.28	57.64	63.72	67.53	9.03	10.59	57.26
	GSDTTA (ours)	56.63	53.53	**64.54**	65.06	61.10	66.61	**69.54**	**74.35**	**71.77**	**70.05**	61.79	**70.22**	68.67	11.70	11.02	**58.44**

Table 1. Classification accuracy (%) across various distributional shifts in the ScanObjectNN-C dataset[2] with DGCNN [3] as backbone network [6]. Mean accuracy scores are reported with the highest values highlighted in bold and the second highest underlined.

GSDPS	GSGMA	EGSS	uniform	gaussian	background	impulse	upsampling	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	occlusion	lidar	Mean
✗	✗	-	48.70	44.57	40.61	67.46	56.79	70.39	72.28	67.46	73.66	73.49	62.65	69.53	73.14	10.67	10.32	56.11
✗	✓	✓	61.44	60.06	18.24	70.05	65.74	70.74	73.32	68.84	72.28	72.97	63.51	67.64	72.12	10.33	11.88	57.28
✓	✗	✓	51.98	45.09	63.85	72.29	56.97	72.63	72.29	66.95	74.01	73.66	61.61	68.16	73.32	11.01	11.36	58.35
✓	✓	✗	62.30	59.03	69.01	72.81	64.54	71.42	73.32	69.36	74.01	72.46	63.51	69.02	74.01	11.87	11.35	61.20
✓	✓	✓	63.17	58.52	69.54	73.67	66.09	71.26	74.01	70.74	75.04	74.87	66.61	69.02	73.67	10.67	10.51	**61.83**

Table 2. Accuracy (in %) of variants of GSDTTA for point cloud classification on ScanObjectNN-C with different architectures.

experimental results are obtained from the code hosted at https://github.com/shimazing/CloudFixer.

BFTT3D. The Backpropagation-Free Test-Time 3D (BFTT3D) [13] employs a backpropagation-free adaptation module to generate target-specific logits, which are then fused with logits from the source model to produce the final predictions. Four hyperparameters are utilized in this method: k, α, and β, which are used to construct the non-parametric network, and γ, a scaling factor for calculating output logits. These parameters are set to 120, 1000, 100, and 205, respectively. The results are reproduced using the authors' publicly available code at https://github.com/abie-e/BFTT3D.

Cloudfixer. Cloudfixer [8] is an input adaptation method for 3D point clouds that employs a pre-trained diffusion model to directly transform test instances into the source domain. In addition to input adaptation, Cloudfixer also adjusts the model using data from each test batch. Several test-time hyperparameters are involved in this method, including the timestep schedule $(t_{\min}, t_{\max})$ and the number of iterations S for the diffusion model; the nearest neighbor parameter k, the input learning rate η_{input} and regularization scheduling $\lambda(\cdot)$ for input adaptation; and the number of votes K and the model learning rate η_{model} for model adaptation. These hyperparameters are set to 0.02, 0.12, 30, 5, 0.0001, 10, 3, and 0.00001, respectively. The results are derived by directly running their published code https://github.com/shimazing/CloudFixer.

3. Per corruption results of ablation study.

In Sect. 4.4, we present the results of ablation studies to evaluate the effectiveness of the components of GSDTTA. Here, we provide a detailed breakdown of the performance across individual corruptions to analyze the contribution of each component. This analysis highlights how Graph Spectral Driven Point Shift (GSDPS) module for input adaptation, Graph Spectral Guided Model Adaptation (GSGMA) module for model adaptation, and eigenmap-guided self-training strategy (EGSS) improve robustness across various corruption types.

Table 2 presents the performance analysis of GSDTTA variants on ScanObjectNN-C across 15 corruption types. The baseline variant, excluding all components, achieves the lowest mean accuracy of 56.11%, with significant drops under uniform noise (48.70%) and impulse corruption (67.46%). Introducing GSGMA alone raises the mean accuracy by 1.17%, demonstrating moderate gains through model adaptation, with notable improvements under uniform noise (61.44%*v.s.*48.70%) and impulse corruption (70.05%*v.s.*67.46%). Adding GSDPS increases the mean accuracy by 2.24%, underscoring the importance of input adaptation in the graph spectral domain, particularly under background noise (23.24%) and impulse corruption (4.83%). The full GSDTTA, combining all components, achieves the highest mean accuracy of 61.83%, excelling under upsampling (66.09%), background corruption (69.54%), and shear (74.87%). Replacing EGSS with a deep-feature-guided self-training approach, where pseudo-labels are generated solely from the global deep descriptor, slightly decreases accuracy to 61.20%. These results confirm the effectiveness of GSDTTA integrating GSDPS, GSGMA, and EGSS in addressing diverse distribution shifts.

References

[1] 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 *ICML*, 2021. 1, 2

[2] M Jehanzeb Mirza, Inkyu Shin, Wei Lin, Andreas Schriebl, Kunyang Sun, Jaesung Choe, Mateusz Kozinski, Horst Possegger, In So Kweon, Kuk-Jin Yoon, et al. Mate: Masked autoencoders are online 3d test-time learners. In *ICCV*, pages 16709–16718, 2023. 1, 2

[3] 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 TOG*, 38(5):1–12, 2019. 1, 2

[4] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *ICCV*, pages 915–924, 2021. 1

[5] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *NeurIPS*, volume 35, pages 23192–23204, 2022. 1

[6] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, pages 8895–8904, 2019. 1, 2

[7] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039, 2020. 1, 2

[8] Hajin Shim, Changhun Kim, and Eunho Yang. Cloudfixer: Test-time adaptation for 3d point clouds via diffusion-guided geometric transformation. In *ECCV*, 2024. 1, 2

[9] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, volume 33, pages 11539–11551, 2020. 1, 2

[10] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, volume 3, page 896, 2013. 1, 2

[11] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, pages 14765–14775, 2022. 1, 2

[12] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2

[13] Yanshuo Wang, Ali Cheraghian, Zeeshan Hayder, Jie Hong, Sameera Ramasinghe, Shafin Rahman, David Ahmedt-Aristizabal, Xuesong Li, Lars Petersson, and Mehrtash Harandi. Backpropagation-free network for 3d test-time adaptation. In *CVPR*, pages 23231–23241, 2024. 1, 2