Backpropagation-free Network for 3D Test-time Adaptation

Yanshuo Wang[1,2], Ali Cheraghian[2], Zeeshan Hayder[2], Jie Hong[1,2]*, Sameera Ramasinghe[5], Shafin Rahman[3], David Ahmedt-Aristizabal[2], Xuesong Li[1,2], Lars Petersson[2], Mehrtash Harandi[2,4]
[1]Australian National University, [2]Data61-CSIRO, Australia
[3]North South University, Bangladesh, [4]Monash University, Australia
[5]Amazon, Australia
{yanshuo.wang, jie.hong, xuesong.li}@anu.edu.au,
{ali.cheraghian, zeeshan.hayder, david.ahmedtaristizabal, lars.petersson}@data61.csiro.au,
shafin.rahman@northsouth.edu, mehrtash.harandi@monash.edu, ramasisa@amazon.com

Abstract

Real-world systems often encounter new data over time, which leads to experiencing target domain shifts. Existing Test-Time Adaptation (TTA) methods tend to apply computationally heavy and memory-intensive backpropagation-based approaches to handle this. Here, we propose a novel method that uses a backpropagation-free approach for TTA for the specific case of 3D data. Our model uses a two-stream architecture to maintain knowledge about the source domain as well as complementary target-domain-specific information. The backpropagation-free property of our model helps address the well-known forgetting problem and mitigates the error accumulation issue. The proposed method also eliminates the need for the usually noisy process of pseudo-labeling and reliance on costly self-supervised training. Moreover, our method leverages subspace learning, effectively reducing the distribution variance between the two domains. Furthermore, the source-domain-specific and the target-domain-specific streams are aligned using a novel entropy-based adaptive fusion strategy. Extensive experiments on popular benchmarks demonstrate the effectiveness of our method. The code will be available at https://github.com/abie-e/BFTT3D.

1. Introduction

In recent years, 3D point cloud processing has experienced significant growth [32, 33, 47, 51], driven largely by advances in deep learning techniques. While researchers have made commendable contributions in these areas, their focus has predominantly been on controlled environments. A notable real-world scenario is Test-Time Adaptation (TTA),

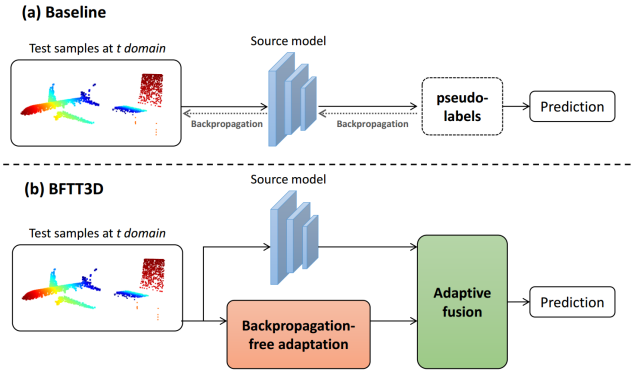

Figure 1. (a) Baseline. When faced with new point cloud samples at test time t, most existing methods generate the pseudo-labels and train the source model in a self-supervised manner. (b) Backpropagation-free test-time 3D model (BFTT3D). BFTT3D adopts a backpropagation-free adaptation module to output the target-specific logit, which fuses with the logit from the source model for prediction. Compared to the baseline, BFTT3D does not require any pseudo-labeling process and backpropagation.

which has recently gained substantial attention from researchers [9, 10, 18, 23, 24, 28, 37, 43, 54], owing to its critical applications in practical settings. In TTA, the model must rapidly adapt to new test samples during testing to provide accurate final predictions. For example, consider a self-driving car equipped with a lidar object detector designed to locate humans and cars on the street. Although this detector may have been trained exclusively under ideal weather conditions, it must swiftly adapt to new environments during test time, such as snowy or rainy weather conditions. This highlights the pressing need to adopt TTA to ensure the car can adapt swiftly to these dynamic environments without extensive retraining. In this paper, we endeavor to propose an innovative and efficient method for

*Corresponding author

TTA on 3D data [11, 27], an area that has received comparatively less attention in the literature compared to 2D images [4, 20, 21, 45]. Nonetheless, from a problem-solving standpoint, it holds equal significance for the computer vision community.

In a TTA scenario, the model must swiftly adapt to a new target domain on the fly, using unlabeled test samples. It accomplishes this task without direct access to the original training data, instead relying on a pre-trained model derived from it. Traditionally, two prominent techniques are employed to tackle the TTA challenge. In the first category, the model utilizes a teacher-student architecture [6, 31, 46] to generate pseudo-labels for given test samples, enabling it to adjust to a new domain. The model is then updated through back-propagation using these pseudo-labels. While this strategy leverages pseudo-labels effectively, relying solely on them can introduce noisy supervision, leading to the accumulation of errors from previous test samples and significantly hindering performance on subsequent new test samples. Moreover, it may result in losing valuable knowledge from the source domain during adaptation to a new one due to the forgetting issue. In the second category, self-supervised methods [27] are employed to adapt the model to new domains. Although they perform well, they tend to be slower, making them less suitable for real-time applications where rapid adaptation is crucial. In this paper, we introduce a novel approach for 3D test-time learning that overcomes forgetting and error accumulation issues and allows for fast adaptation. This makes it a promising choice for addressing real-world application problems, which is pivotal in test-time learning.

In this paper, as shown in Figure 1, we introduce a novel approach, backpropagation-free test-time 3D model (BFTT3D), for TTA that circumvents the reliance on pseudo-labels and avoids needing a complex and sluggish self-supervised learning technique. It operates without necessitating updates to the model parameters during adaptation, thus sidestepping the issue of error accumulation. To delve deeper into our proposed method, we leverage an existing 3D point cloud model [32, 33, 47, 51] as the source model, meticulously trained on the source data. The source model crafts a source-domain-specific (or source-specific) feature description when presented with a test sample from a target domain. The source model is kept frozen during upcoming test samples, which helps us to keep useful knowledge of the source domain. Concurrently, we generate a target-domain-specific (or target-specific) description tailored to the given test samples from the target domain. Within our designed backpropagation-free adaptation module, our focus lies in narrowing the domain gap between the emerging target domain and the established source domain within a defined subspace, effectively minimizing the distribution divergence between the two. Notably, our dynamic domain adaptation methods enable seamless adjustment to forthcoming test samples from the target domain, rendering them highly versatile in TTA scenarios. In the final stage, we integrate the information from both source-specific and target-specific information, leveraging the entropy data from both models during adaptation. Our proposed solution is highly adaptive as it can grow with the data, accommodating new information without requiring updates to predefined parameters (avoiding the computational cost of backpropagation), and well-suited for scenarios where data is constantly changing or expanding.

In summary, the main contributions of our paper are as follows:

- We present a novel and efficient approach, BFTT3D, for 3D TTA, eliminating the necessity for extensive backpropagation. This approach is less susceptible to the impact of noisy supervision derived from pseudo-labels and does not entail parameter fine-tuning during adaptation, consequently sidestepping error accumulation and forgetting issues.
- The backpropagation-free adaptation is proposed along with the source model to enhance the model's adaptability to specific domains. It is introduced without adding the extra model parameters that need backpropagation.
- Using entropy information, we fuse the source-specific feature from the source model and the target-specific feature from the backpropagation-free adaptation. Like the backpropagation-free adaptation module, the fusion module does not introduce extra model parameters that need backpropagation.
- The experimental outcomes demonstrate the superiority of the proposed BFTT3D on popular benchmarks, including including ModelNet-40C [42] and ScanObjectNN-C [27]. The algorithm is validated to be effective and efficient in 3D TTA.

2. Related work

Point cloud domain adaptation: In recent years, 3D sensors have become integral components of perception systems. The field of 3D point cloud processing has experienced remarkable growth, primarily fueled by advancements in deep learning techniques [16, 19, 32, 33, 47, 51]. Unsupervised domain adaptation (UDA) is increasingly prominent in the 3D vision for mitigating domain gaps between source and target datasets without target label information. UDA methods can be categorized into two groups. The first group requires labeled source data and source-trained models for adaptation to the target domain, while the second group employs source-free domain adaptation approaches. Wang *et al.* [48] introduce a statistic normalization approach to handle the gaps. Many of the following works leverage self-training to resolve the domain adaptation problem. These methods make improve-

ments mainly on obtaining pseudo labels of higher quality *e.g.* with a memory bank (ST3D) [53] or by harnessing scene flow information (FAST3D) [8]. Some works suggested multi-level self-supervised learning at global and local scales [7]. Recently, Cardace *et al.* [3] introduced a point cloud-specific self-training strategy with pseudo label refinement that exploits self-distillation to learn effective representations.

In addition to the self-training approaches, other approaches concentrate on establishing alignment between the source and target domains. Domain alignments are established through various strategies, including a multi-level alignment of local and global features (PointDAN) [34], a scale-aware and range-aware alignment strategy (SR-DAN) [57], a student-teacher network along with pseudo-labeling (MLC-Net) [26], and a contrastive co-training scheme (3D-CoCo) [55]. Shen *et al.* [38] introduced geometry-aware implicit in point clouds to mitigate domain biases and adopted a pseudo-labeling approach as part of their two-step method. CoSMix [35, 36] addresses domain shift in point cloud data through a compositional semantic mixup strategy with a teacher-student learning scheme. Traditional UDA face limitations in scenarios where data privacy and data portability are critical. To address these constraints, source-free approaches facilitate domain adaptation exclusively using a source-trained model, eliminating the need for source domain labels [17]. Hegde and Patel [13] utilize a transformer module to calculate an attentive class prototype, pinpointing accurate regions of interest for self-training. UAMT [14] introduces a mean teacher approach with Monte-Carlo dropout uncertainty for supervising the student model using iteratively generated pseudo labels. In their subsequent work [15], the same authors propose an uncertainty-aware mean teacher framework that enhances conventional pseudo-label based on self-training methods by reducing the impact of label noise and assigning lower weights to samples when the teacher model is uncertain. While UDA and SFDA approaches address critical challenges, they assume prior knowledge about test data distributions.

Test-time domain adaptation: Test-time adaptation (TTA) is in contrast to traditional unsupervised domain adaptation, as it adapts a source-trained model for a new target domain during inference without using source data. A common method to minimize the domain difference when source data is unavailable is fine-tuning the source model using an unsupervised loss function derived from the target distribution. TTT [43] updates model parameters in an online manner by applying a self-supervised proxy task on the test data. TENT [45] updates trainable batch normalization parameters at test time by minimizing the entropy of model prediction. SHOT [21] also minimizes the expected prediction entropy derived from the output softmax distribution. SHOT++ [22] employs target-specific clustering for denoising pseudo labels. A gradient-free TTT approach by Boudiaf *et al.* [2] emphasizes output prediction consistency while incorporating Laplacian regularization. TTT++ [24] introduces an extra self-supervised branch using contrastive learning in the source model to facilitate adaptation in the target domain. TTT-MAE [9], an extension of TTT that utilizes a transformer backbone and replaces the self-supervision with masked autoencoders. AdaContrast [4] employs contrastive learning into TTA and utilizes both pseudo-label loss and diversity loss for optimization. Finally, TTTFlow [29] employs unsupervised normalizing flows as an alternative to self-supervision for the auxiliary task.

Test-time point cloud domain adaptation: While the TTA concept, originally designed for 2D image data, may encounter performance challenges when extended to 3D data, specialized designs are often required for 3D scenarios. However, a significant research gap remains in developing networks that can adapt a pre-trained source model during testing without requiring access to either source data or target labels. MM-TTA [39] introduces a selective fusion strategy to ensemble predictions from multiple modalities, enhancing self-learning signals in 3D semantic segmentation. It includes intra- and inter-modules for generating and selecting pseudo-labels.

MATE [27] uses masked autoencoding as a robust self-supervised auxiliary objective to improve the network's resilience to distribution shifts in 3D point clouds. DSS [49] is a method that addresses noisy pseudo-labels in TDA by proposing dynamic thresholding, positive learning, and negative learning processes. It demonstrates versatility for 3D point cloud classification by monitoring prediction confidence online and selecting low- and high-quality samples for training. Hatem *et al.* [12] proposes a test-time adaptation approach for point cloud upsampling using meta-learning to enhance model generalization at inference time. Point-TTA [11] proposes a test-time adaptation approach for point cloud registration. The approach adapts the model parameters in an instance-specific manner during inference and obtains a different set of network parameters for each instance.

3. Method

3.1. Problem Formulation

In the context of Test-time Adaptation (TTA), we define two domains: the source domain denoted as $\mathcal{Q}_s$ and the target domain denoted as $\mathcal{Q}_t$. We assume the existence of a model, $h_{\theta_s}(.)$, representing the source model, where θ_s represents the pre-trained parameters on the source data of $\mathcal{Q}_s$. In the context of the 3D TTA task, the primary objective is for the source model $h_{\theta_s}(.)$ to accurately predict the point cloud

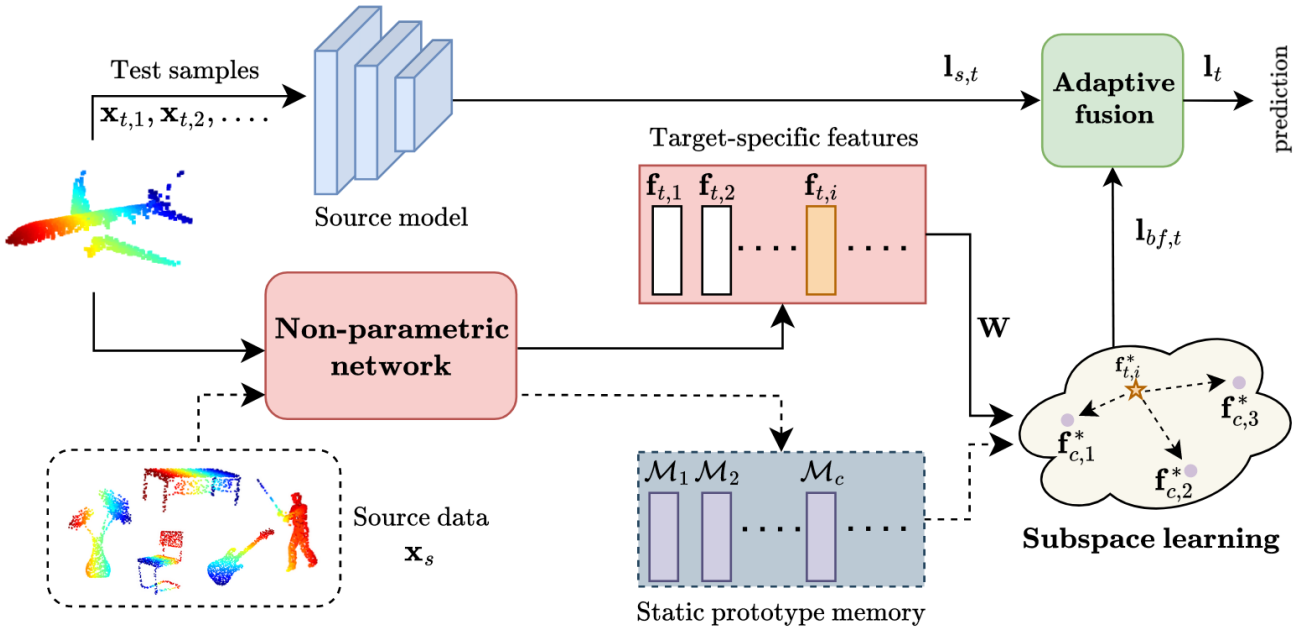

Figure 2. The framework of backpropagation-free test-time 3D model (BFTT3D). In the preparation stage, we first extract general features for source point cloud data $\mathbf{x}_s$ using a non-parametric network and then select a subset of all general features as static prototype memory $\mathcal{M}$. At test time, BFTT3D again adopts the non-parametric network to extract the general feature representation $\mathbf{f}_t$ from the given test point cloud sample $\mathbf{x}_t$ of t domain. The feature $\mathbf{f}_t$ is then compared with static prototype feature $\mathbf{f}_c \in \mathcal{M}_c$ on a shared subspace to compute the target-specific logit $\mathbf{l}_{bf}$. Finally, the logit $\mathbf{l}_{bf}$ supplements the logit produced by the source model, $\mathbf{l}_s$, via an adaptive fusion module based on prediction entropy to output the final logit $\mathbf{l}_t$ for prediction. Notably, each module of BFTT3D, including the non-parametric network, subspace learning, and adaptive fusion, does not introduce any parameters that need backpropagation during adaptation.

sample $\mathbf{x}_t$ from the target domain $\mathcal{Q}_t$ during the test phase. To achieve this, the model undergoes adaptation to samples $\mathbf{x}_t$ from $\mathcal{Q}_t$ at test time. After this adaptation process, the updated model predicts the class label of $\mathbf{x}_t$. It is important to emphasize that, for reasons such as privacy or memory limitations, we do not have access to the source dataset $\mathcal{Q}_s$ during test time. In this paper, following [5], we utilize a limited set of prototype feature representations from the source domain.

3.2. Model Overview

We provide a concise overview of our proposed method, BFTT3D, illustrated in Figure 2. Given test point cloud samples, $\{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, ..., \mathbf{x}_{t,n_t}\}$ from domain $\mathcal{Q}_t$, our goal is to predict class labels for these samples. The designed BFTT3D first extracts information using a non-parametric network that typically remains unaffected by domain bias, and thus it produces powerful target-specific features for the test 3D samples, $\{\mathbf{f}_{t,1}, \mathbf{f}_{t,2}, ..., \mathbf{f}_{t,n_t}\}$. These features will be compared with static prototypes in a shared subspace, and then the target-specific logit, $\mathbf{l}_{bf,t}$, will be computed. At last, $\mathbf{l}_{bf,t}$ complements the logit generated from the source model, $\mathbf{l}_{s,t}$, via an adaptive fusion module to output the final logit, $\mathbf{l}_t$, for prediction. It should be noted that modules we design along with the source model, including non-parametric network, subspace learning, and adaptive fusion, do not require any parameters that need backpropagation.

3.3. Backpropagation-free Adaptation

To better adapt the model to diverse domain distributions, we complement the source model logit $\mathbf{l}_{s,t}$ during test time of $\mathcal{Q}_t$. To achieve this, we generate the logit $\mathbf{l}_{bf,t}$ from the backpropagation-free adaptation module. In particular, the non-parametric network transforms the incoming test 3D data $\mathbf{x}_t$ into the target-specific feature $\mathbf{f}_t$. Then, the target-specific logit $\mathbf{l}_{bf,t}$ is calculated based on $\mathbf{f}_t$ and the static prototype feature $\mathbf{f}_c$ in a common subspace.

3.3.1 Non-parametric Network

Non-parametric point cloud network [56] does not require training and is constructed solely from non-learnable components, including farthest point sampling (FPS), k-nearest neighbors (k-NN), and pooling operations, supplemented by trigonometric functions. It has been demonstrated that the non-parametric network has a strong generalization ability from seen to unseen 3D data due to its training-free manner [56]. Given an input point cloud, $\mathbf{x}_{t,i}$ from $\mathcal{Q}_t$, trigonometric functions first encode the sample into the channel

embedding with size d for X, Y, and Z coordinates. For each channel s $\in \{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$, the channel embedding is computed as follows:

$$g(\mathbf{x}_{t,i})^{(s)} = \begin{cases} \sin(\alpha \mathbf{x}_{t,i}^{(s)} / \beta^{\frac{6i}{d}}), & i = 2k \\ \cos(\alpha \mathbf{x}_{t,i}^{(s)} / \beta^{\frac{6i}{d}}), & i = 2k+1 \end{cases} \in \mathbb{R}^{d/3} \quad (1)$$

where α and β represent the wavelength and scale hyperparameters. The raw point embedding $\mathbf{f}_{raw,t,i}$ is then calculated by the concatenated channel-wise embeddings from XYZ coordinates as follows:

$$\mathbf{f}_{raw,t,i} = [g(\mathbf{x}_{t,i})^{(\mathrm{X})}, g(\mathbf{x}_{t,i})^{(\mathrm{Y})}, g(\mathbf{x}_{t,i})^{(\mathrm{Z})}] \in \mathbb{R}^{d} \quad (2)$$

Given each point $\mathbf{x}$, we can obtain the embedding $\mathbf{f}_{raw}$. FPS is used to find the down-sampled local center points $\mathbf{x}_{cen}$ from all $\mathbf{x}$, and k-NN is then applied to group $\mathcal{N}_{ct}$ neighbor points $\mathbf{x}_{nb}$ for each center point $\mathbf{x}_{cen}$. To incorporate both center and neighbor information, we first calculate the expanded center feature $\bar{\mathbf{f}}_{cen}$ for $\mathbf{x}_{cen}$ by simply concatenating with neighboring point features:

$$\bar{\mathbf{f}}_{cen} = \mathrm{Concat}(\mathbf{f}_{cen}, \mathbf{f}_{nb,j}), j \in \mathcal{N}_{cen} \quad (3)$$

where $\mathbf{f}_{cen}$ is the feature of point $\mathbf{x}_{cen}$ (See Eq. (2)), and $\mathbf{f}_{nb}$ denotes the feature of the neighbor point $\mathbf{x}_{nb}$. To show the spatial distribution of neighboring points $\mathbf{x}_{nb}$, the feature $\bar{\mathbf{f}}_{cen}$ is further reweighted by its relative position encoding $g(\Delta_{nb})$ to the center point:

$$\bar{\mathbf{f}}_{cen}^{w} = (\bar{\mathbf{f}}_{cen} + g(\Delta_{nb})) \odot g(\Delta_{nb}) \quad (4)$$

where $\odot$ represents element-wise multiplication and Δ_{nb} is the relative position of $\mathbf{x}_{nb}$ to its center point $\mathbf{x}_{cen}$. Finally, the maximum pooling and average pooling are used to condense the information of $\bar{\mathbf{f}}_{cen}^{w}$:

$$\bar{\mathbf{f}}_{cen}^{p} = \mathrm{MaxPool}(\bar{\mathbf{f}}_{cen}^{w}) + \mathrm{AvgPool}(\bar{\mathbf{f}}_{cen}^{w}) \quad (5)$$

The process from Eq. (3) to (5) will be repeated four times. After the repetition is completed, a global pooling operation is applied to get the sample feature $\mathbf{f}_{t,i} \in \mathbb{R}^{d}$ for $\mathbf{x}_{t,i}$, which is shown in Figure 2.

3.3.2 Subspace Learning

Prototype memory. Before test time, we use the non-parametric network to extract features from the source data $\mathbf{x}_s$ and save them in the static memory. To predict the logit, the non-parametric network uses similarity matching instead of the typical classification head. Hence, static memory should be employed to gather necessary category information from the source domain for achieving similarity matching [56].

To be resource-efficient regarding memory usage, we employ the herding algorithm introduced by [40] combined with a nearest-mean-selection technique to select a subset from all features, ensuring that we only conserve limited memory space. We denote the whole feature set without selection for each class c as $\mathcal{M}_{c,all}$, and the selected exemplars set as $\mathcal{M}_c$. During the selection, for $\forall \mathbf{f} \in \mathcal{M}_c$,

$$\mathrm{dist}\left(\mathbf{f}, \bar{\mathbf{f}}_c\right) \leq \min_{\mathbf{f}' \in \mathcal{M}_{c,all} \setminus \mathcal{M}_c} \mathrm{dist}\left(\bar{\mathbf{f}}_c, \mathbf{f}'\right) \quad (6)$$

where $\bar{\mathbf{f}}_c$ denotes mean averaged feature in $\mathcal{M}_{c,all}$. $\mathrm{dist}(.,.)$ is the distance function between two features. The selected features $\mathbf{f}_c \in \mathcal{M}_c$ are then stored in a fixed prototype memory. The feature $\mathbf{f}_t$ from the non-parametric network and $\mathbf{f}_c$ from the prototype memory $\mathcal{M}_c$ share the same dimension. To make a prediction, we construct the similarity matrix $\mathcal{J}$ between the feature vector $\mathbf{F}_t = [\mathbf{f}_{t,1}, \mathbf{f}_{t,2}, ..., \mathbf{f}_{t,n_t}]$ and the source prototype memory $\mathcal{M} = [\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_c]$ via cosine similarity:

$$\mathcal{J} = \tilde{\mathbf{F}}_t \cdot \tilde{\mathcal{M}}^T \quad (7)$$

where $\tilde{\mathbf{F}}_t$ and $\tilde{\mathcal{M}}$ denote the corresponding normalized vectors of $\mathbf{F}_t$ and static memory $\mathcal{M}$. Given the source memory label $\mathbf{L}_m$, we compute the output logit $\mathbf{l}_{bf,t}$ via:

$$\mathbf{l}_{bf,t} = \varphi(\mathcal{J}\mathbf{L}_m) \quad (8)$$

where $\varphi(x) = \exp(-\gamma(1-x))$ serves as an activation for prediction and γ is a scaling hyperparameter. In this manner, the source prototypes contribute to the target-specific logit $\mathbf{l}_{bf,t}$.

Similarity matrix. Due to the distribution gap between the source and target domain, making direct predictions by similarity matrix $\mathcal{J}$ across two domains could be problematic. As such, before Eq. (7), we would like to have a projection function ψ that first maps both static prototypes and test sample features, $\mathbf{f} \in \mathcal{M}$ and $\mathbf{f}_t$, into a shared subspace before calculating the similarity:

$$\begin{aligned} \mathbf{f} \leftarrow \mathbf{f}^* &= \psi(\mathbf{f}) \\ \mathbf{f}_t \leftarrow \mathbf{f}_t^* &= \psi(\mathbf{f}_t) \end{aligned} \quad (9)$$

Using the projection function $\psi(.)$, we aim to minimize the domain divergence between the prototype and test features within the shared subspace after mapping. Here, we use the Maximum Mean Discrepancy (MMD) distance [1] to measure the statistical difference between the prototype and test domain. This kernel-based measure is distribution-free and can be applied without imposing strict requirements on the distribution.

$$\mathrm{Dist}\left(\mathcal{Q}_s, \mathcal{Q}_t\right) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{f}_i^* - \frac{1}{n_t} \sum_{i=1}^{n_t} \mathbf{f}_{t,i}^* \right\|_{\mathcal{H}}^2 \quad (10)$$

where n_s is the number of features $\mathbf{f}$ in $\mathcal{M}$, and $\|\cdot\|_{\mathcal{H}}^2$ is the l_2 norm computed in a reproducing kernel Hilbert space.

To approximate the mapping function $\psi(.)$ in practice, which can mitigate the difference between these static source prototypes and test sample features in the common space, we employ a classical transfer learning technique named Transfer Component Analysis (TCA) [30]. In general, TCA finds a transformation matrix $\mathbf{W} \in \mathbb{R}^{(n_s+n_t)\times m}$, as the empirical measurement of mapping function ψ induced by a universal kernel, to project both $\mathbf{f}_s$ and $\mathbf{f}_t$ onto a commonly shared subspace of dimension m. The objective is to minimize the MMD distance on the mapped subspace formulated in a kernel learning problem [1]. The source-target kernel matrix $\mathbf{K}$ is formulated as follows, reflecting data similarities in the source, target, and across domains.

$$\mathbf{K} = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle_{i,j=1}^{n_s+n_t} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)} \tag{11}$$

$$\mathbf{L}_{i,j} = \begin{cases} \frac{1}{n_t^2} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathcal{Q}_t \\ \frac{1}{n_t^s} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathcal{Q}_s \\ \frac{-1}{n_s n_t} & \text{otherwise} \end{cases}, \tag{12}$$

where $\langle \cdot \rangle$ denotes the dot product in $\mathcal{H}$, and $\mathbf{L}$ is used to scale the kernel matrix $\mathbf{K}$. The overall kernel learning objective for MMD distance can be rewritten as follows, according to Eq. (10):

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{(n_s+n_t)\times m}} & \operatorname{tr}\left(\mathbf{W}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W}\right) + \mu \operatorname{tr}\left(\mathbf{W}^T\mathbf{W}\right), \\ \text{s.t. } & \mathbf{W}^T\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I} \end{aligned} \tag{13}$$

where μ represents the regularization penalty to control the complexity of $\mathbf{W}$ and $\mathbf{H} = \mathbf{I} - \mathbf{1}\mathbf{1}^T/(n_s+n_t) \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ is the centering matrix. By solving this constrained trace optimization problem with regularization, the solution of $\mathbf{W}$ is explicitly given by the m largest eigenvectors of

$$(\mathbf{K}\mathbf{L}\mathbf{K} + \mu\mathbf{I})^{-1}\mathbf{K}\mathbf{H}\mathbf{K}. \tag{14}$$

Once $\mathbf{W}$ is obtained, the transformed source prototype and target-specific feature vectors, $\mathbf{f}^*$ and $\mathbf{f}_t^*$ in Eq. (9), are the corresponding indexed columns of $\mathbf{W}^T\mathbf{K} \in \mathbb{R}^{m\times(n_s+n_t)}$.

3.4. Adaptive Fusion Module

After obtaining the target-specific logit $\mathbf{l}_{bf,t}$ from the backpropagation-free adaptation, as shown in Figure 2, we then fuse $\mathbf{l}_{bf,t}$ and the source-specific logit $\mathbf{l}_{s,t}$ from the source model, into a final logit $\mathbf{l}_t$. The distribution distance between the source and target domain varies since the target domain is diverse. As such, to tackle the varying distribution gaps, we adaptively fuse $\mathbf{l}_{s,t}$ and $\mathbf{l}_{bf,t}$ for a more accurate prediction. When the incoming test point cloud samples are drawn from a target domain whose distribution is similar to the source data, we hope to emphasize a larger proportion of the logit from the source model branch since it is reliable in predicting in-distribution test samples. Conversely, when the domain sample is drawn from a significantly different distribution than the source domain, we prioritize using a larger proportion of the logit from the backpropagation-free adaptation to provide more target-specific information. To achieve this, we first adopt a weight p to control the proportion of the logit used in the final prediction.

$$\mathbf{l}_t = p \cdot \mathbf{l}_{bf,t} + (1-p) \cdot \mathbf{l}_{s,t} \tag{15}$$

To dynamically calculate the value of p in the test time, we employ the softmax entropy of logits as the measurement to give p since entropy is related to the errors and shifts [45]. Intuitively, entropy is related to the model's certainty about the test point cloud samples, as low-entropy predictions are most likely correct with high confidence. Thus, we calculate the entropy ratio between two logits as follows to reflect the degree of the difference between the current target and source domain:

$$p = 1 - \frac{\mathrm{E}(\mathbf{l}_{bf,t})}{\mathrm{E}(\mathbf{l}_{s,t}) + \mathrm{E}(\mathbf{l}_{bf,t})} \tag{16}$$

where $\mathrm{E}(.)$ denote the entropy. In this way, we achieve the adaptive fusion of the logits from the source model and backpropagation-free adaptation module.

4. Experiments

In this section, we provide extensive experiments to demonstrate the effectiveness of the proposed BFTT3D model in TTA for 3D data. We evaluate our method on multiple 3D point cloud datasets, including ModelNet-40C [42] and ScanObjectNN-C [27].

4.1. Implementation

For the non-parametric network, we follow the implementations in [56]. Regarding the static prototype memory, we construct the feature memory only once before adaptation and keep it fixed throughout the adaptation stage. We selectively choose 25% of the samples as the prototype memory by herding algorithm, and the transformed feature dimension m for subspace learning is fixed to 150. Note that the severity of corruption applied is fixed to the highest 5 for all test experiments. In addition, we also utilized various backbones for the source model, each trained independently on its corresponding clean set, to validate the robustness of BFTT3D. The number of points for the input point cloud is set to 1024. We run all experiments on a single NVIDIA RTX 4080.

4.2. Datasets

ModelNet-40C. ModelNet-40C [42] is a benchmark with 15 common types of corruptions induced on the original test set of ModelNet-40 [50]. The goal of building the

Method	Backbone	uniform	gaussian	background	impulse	upsampling	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean ↓
Source-only		14.63	18.68	95.30	33.39	15.03	29.46	27.63	12.93	10.49	42.71	72.81	14.95	34.85	56.28	59.00	35.88
TENT [45]		14.71	18.35	90.36	27.03	15.36	28.28	26.66	13.86	12.36	40.68	65.68	15.76	33.87	56.56	59.85	34.62
BN [20]		14.83	18.84	89.63	27.47	15.15	28.69	26.66	14.59	12.40	40.44	65.88	16.25	34.68	57.58	60.21	34.89
SHOT [21]	**PointNet** [32]	14.91	17.22	90.44	25.61	14.51	27.07	25.85	13.78	12.20	39.95	65.80	16.21	32.98	56.93	60.29	34.25
LAME [2]		76.64	84.67	95.95	95.48	67.83	93.37	91.72	59.92	34.81	96.45	97.50	68.52	92.53	95.99	95.95	83.16
DSS [49]		14.20	18.71	94.13	33.12	14.33	29.31	27.55	13.31	10.92	41.42	66.31	15.45	34.10	56.31	59.33	35.23
BFTT3D (ours)		14.55	18.27	80.75	31.93	14.83	28.97	27.19	12.76	10.49	39.71	68.56	14.67	34.04	54.13	55.88	**33.78**
Source-only		23.46	28.20	57.41	37.93	33.23	22.73	20.91	27.59	16.57	15.96	41.33	23.78	19.94	65.52	85.25	34.65
TENT [45]		18.92	19.94	61.91	22.85	24.64	21.80	20.87	25.00	18.15	18.76	31.97	23.46	21.72	62.56	72.29	30.99
BN [20]	**DGCNN** [47]	19.81	20.42	63.01	23.22	25.41	22.41	21.52	26.09	17.54	18.80	32.09	23.10	21.76	63.13	72.12	31.36
SHOT [21]		18.88	19.73	57.13	21.27	22.77	22.29	19.73	24.31	17.59	18.48	31.28	22.12	21.72	62.28	72.97	30.17
BFTT3D (ours)		19.45	20.02	58.51	22.33	24.59	21.96	20.71	24.03	16.49	18.80	31.48	21.64	21.39	56.28	62.24	**29.33**
Source-only		15.60	17.42	81.24	35.49	12.80	14.22	12.88	20.87	10.90	12.32	29.58	20.38	12.93	62.12	68.35	28.47
TENT [45]		13.94	13.01	56.20	20.06	12.76	13.45	12.72	15.15	10.90	12.32	19.57	16.65	13.70	53.57	55.06	22.60
BN [20]	**Curvenet** [52]	14.95	14.59	60.66	21.35	13.05	15.07	13.61	17.06	11.43	13.65	21.27	17.75	14.10	55.15	56.77	24.03
SHOT [21]		11.75	11.35	43.01	16.17	10.58	11.30	10.94	11.75	9.44	10.01	16.03	12.28	12.75	55.19	54.28	19.79
BFTT3D (ours)		11.87	12.16	40.68	15.36	10.74	11.51	11.02	11.99	10.01	10.72	16.82	13.74	11.75	50.32	49.88	**19.23**

Table 1. Experimental results on ModelNet-40C [42]. The classification errors in % are provided.

Method	Backbone	uniform	gaussian	background	impulse	upsampling	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean ↓
Source-only		63.17	40.62	87.61	70.57	62.31	60.24	58.52	23.24	22.89	65.23	69.54	26.51	60.59	89.33	89.85	59.35
TENT [45]		61.10	38.55	84.34	69.36	60.24	58.69	57.14	23.24	22.72	63.51	68.16	25.82	59.55	89.50	90.71	58.18
BN [20]		60.76	37.01	83.65	68.16	59.21	58.86	57.14	24.27	23.58	63.34	67.99	25.82	58.18	89.5	90.53	57.87
SHOT [21]	**PointNet** [32]	62.20	55.77	88.64	86.57	63.86	70.91	67.13	27.54	25.47	74.01	75.39	28.23	68.33	93.29	95.18	65.63
LAME [2]		60.24	33.31	91.91	70.22	59.71	59.90	54.77	14.53	15.98	55.52	62.90	16.44	58.04	92.73	88.92	55.67
DSS [49]		59.83	34.74	84.81	69.26	58.21	57.13	55.42	21.69	20.11	61.52	65.53	22.45	58.21	88.47	90.53	56.53
BFTT3D (ours)		57.14	39.93	70.05	65.23	55.94	54.56	53.01	21.34	21.51	58.69	68.85	24.10	55.94	85.03	85.54	**54.46**
Source-only		57.83	49.91	55.94	43.37	53.53	37.01	34.08	24.10	19.62	33.39	38.73	25.13	34.60	92.08	92.08	46.09
TENT [45]		54.73	47.16	55.25	43.89	52.84	36.83	32.53	24.78	20.48	31.50	38.55	23.92	32.53	91.39	91.22	45.17
BN [20]	**DGCNN** [47]	55.25	47.16	55.42	42.86	52.84	36.32	32.19	24.78	19.45	32.01	38.38	24.61	32.36	91.39	91.57	45.11
SHOT [21]		86.92	67.81	81.93	65.75	86.75	43.20	37.01	30.46	27.19	36.32	46.13	32.53	40.45	91.22	89.33	57.53
BFTT3D (ours)		57.31	41.82	52.32	43.37	53.36	36.83	34.07	22.55	17.73	33.05	38.55	22.20	34.58	85.54	85.37	**43.91**
Source-only		69.54	52.32	81.93	80.21	67.99	67.13	66.27	35.46	31.15	64.03	69.36	36.32	67.13	90.53	92.25	64.77
TENT [45]		70.22	53.87	81.93	80.21	70.05	68.16	67.81	36.49	32.53	66.09	70.57	36.83	67.99	91.05	92.77	65.77
BN [20]	**Curvenet** [52]	69.36	50.95	80.21	79.52	68.16	66.95	66.09	34.77	31.33	64.37	67.99	35.63	66.09	90.36	92.08	64.26
SHOT [21]		87.26	77.11	91.91	93.98	87.09	87.78	87.44	72.12	73.84	87.26	87.44	74.18	87.09	95.01	90.71	85.35
BFTT3D (ours)		61.96	48.19	68.85	71.43	60.93	58.00	57.83	30.29	28.23	57.49	62.65	31.50	58.86	88.64	85.54	**57.80**

Table 2. Experimental results on ScanObjectNN-C [27]. The classification errors in % are provided.

ModelNet-40C dataset is to mimic distribution shifts that occur in the real world.

ScanObjectNN-C. ScanObjectNN [44] is a point cloud classification dataset collected from the real world. It contains 15 classes, with 2309 samples in the train set and 581 in the test set. To build ScanObjectNN-C, we employ the setting proposed by [27] to generate 15 corruptions in the test set of ScanObjectNN with the object only.

4.3. Main Results

ModelNet-40C. We first examine the adaptation performance on ModelNet-40C, and the experimental results are given in Table 1. As shown in the table, all baseline methods, including TENT [45], BN [20], and SHOT [21] have improvements over the source model. However, our method, BFTT3D, has the lowest error rate on average. Interestingly, our approach consistently demonstrates significant improvements in domains like *occlusion* and *lidar*. For example, for the target domain of *occlusion*, BFTT3D achieves improvements of 5.02%, 9.24%, and 11.82% over the source model. This observation suggests that our method is particularly effective when the target and source domains have a large gap.

ScanObjectNN-C. The experimental result of 3D TTA performance on ScanObjectNN-C [27], a more difficult case, are reported in Table 2. Notably, the source model exhibits a high error rate for each backbone, indicating substantial domain divergence on average and increased difficulty in adaptation. Baseline methods show limited improvement in adapting the source model to test the domain. However, our BFTT3D model can still overcome the large domain gap. The table shows that our method has 4.89%, 2.18%, and 6.59% less men error compared to the source model, using PointNet, DGCNN, and Curvenet as the backbone, respectively. This demonstrates that our approach provides useful target-specific information to the source model, even when the test target domain distribution differs largely from the source domain. Another fact is that most baseline methods face a limitation of insufficient test data for adaptation, particularly because ScanObjectNN-C has a very limited number of test samples for each target domain. However, this constraint does not affect BFTT3D as our method does not require samples for training the source model. It only involves concatenating the logits from the frozen source model and the backpropagation-free adaptation, offering greater flexibility.

4.4. Ablation Study

Number of prototypes. Here, we explore the influence of the number of exemplars utilized for constructing the sim-

Method	Ratio	Mean ↓
Source	None	46.09
BFTT3D	100%	**43.75**
BFTT3D	75%	43.80
BFTT3D	50%	43.91
BFTT3D	25%	43.91

Table 3. Ablation study: number of prototype. The experiments run on ScanObjectNN-C [27]. The mean classification errors in % are provided. The ratio reflects the proportion of the prototype memory relative to the total static memory, indicating the size of the prototype memory.

Subspace method	Mean ↓
None	56.84
JDA [25]	54.87
CORAL [41]	56.80
TCA [30]	**54.46**

Table 4. Ablation study: subspace learning method. The experiments run on ScanObjectNN-C [27]. Using our BFTT3D with PointNet backbone, different subspace learning methods are tested, including JDA, CORAL, and TCA. The mean classification errors in % are provided.

ilarity matrix. The results in Table 3 indicate that solely storing 100% of source features for similarity construction yields limited benefits compared to using only 25% of the data. Our setting of 25% not only conserves less memory space but also reduces the size of the similarity matrix to improve processing speed.

Subspace learning method. As shown in Table 4, we present the performance of alternative subspace learning methods for creating shared subspace of the similarity matrix. The results show that certain subspace learning approaches for domain alignment, such as TCA [30] and JDA [25], have relatively low errors compared to other setups. Others like CORAL [41] do not demonstrate effective improvements in performance. Overall, we note that the highest error occurs when no subspace learning is applied. This indicates that transforming onto a shared subspace before computing the similarity matrix in non-parametric adaptation is necessary.

Adaptive ratio. In this section, we first evaluate the effectiveness of the entropy-based ratio from the adaptive fusion module in comparison with fixed thresholds. As seen from Table 5, simply using $p = 0.5$ to mix logits from the target-specific branch and source model helps to reduce the error compared with the source-only setting. By leveraging the adaptive ratio, we get the lowest mean error across the board compared with other setups. This may indicate that the adaptive ratio reaches a good balance between the two

Method	Ratio	Mean ↓
Source-only	None	46.09
BFTT3D	$p = 0.5$	44.13
BFTT3D	adaptive p	**43.91**

Table 5. Ablation study: adaptive ratio. Using the backbone CurveNet, the experiments run on ScanObjectNN-C [27]. The mean classification errors in % are provided.

branches of logits. In addition, we further conducted experiments on ModelNet40-C [42] to compare the adaptive ratio with a best p found by exhaustive search to show the reliability of our adaptive ratio. Note that from Figure 3, the error rate of the adaptive threshold is very similar to the error with the optimal p, staying within a certain small range. Moreover, it is unlikely for us to manually set an optimal and reliable threshold without validation. In contrast, our adaptive ratio provides a good estimated mixing threshold for test data, and we can calculate this without any validation in test time.

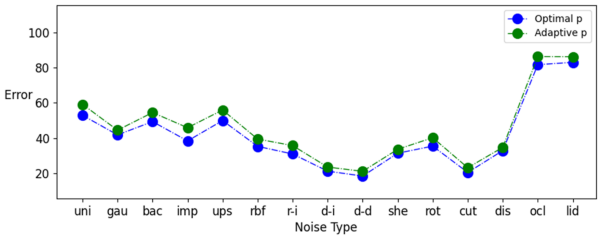

Figure 3. Comparisons of mean error between adaptive and optimal ratio. Using the backbone PointNet [32] and testing on ModelNet40-C [42]. The blue line represents the optimal p from an exhaustive search, and the green line represents our adaptive ratio p.

5. Conclusion

In this work, we propose the model BFTT3D for the task of 3D TTA. In particular, BFTT3D first generates the complementary target-domain-specific logit via similarity matching on shared subspace. Then, it is combined with the source-domain-specific model logit via an entropy-based adaptive fusion strategy to output final refined predictions for test samples from diverse distributions. The entire framework of BFTT3D does not introduce any parameters that need backpropagation, thus avoiding the complex pseudo-labeling process. For future work, meta-training in the pre-training stage, for the simplicity of our method, could be a promising direction.

Acknowledgment. Mehrtash Harandi expresses gratitude for the support provided by the Australian Research Council via the Discovery Program (DP230101176).

References

[1] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14): e49–e57, 2006. 5, 6

[2] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022. 3, 7

[3] Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Self-distillation for unsupervised 3d domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4166–4177, 2023. 3

[4] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[5] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *European Conference on Computer Vision*, pages 440–458. Springer, 2022. 4

[6] Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. *arXiv preprint arXiv:2211.13081*, 2022. 2

[7] Hehe Fan, Xiaojun Chang, Wanyue Zhang, Yi Cheng, Ying Sun, and Mohan Kankanhalli. Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6377–6386, 2022. 3

[8] Christian Fruhwirth-Reisinger, Michael Opitz, Horst Possegger, and Horst Bischof. Fast3d: Flow-aware self-training for 3d object detectors. *arXiv preprint arXiv:2110.09355*, 2021. 3

[9] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. *arXiv preprint arXiv:2209.07522*, 2022. 1, 3

[10] Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven test-time adaptation. *arXiv preprint arXiv:2207.03442*, 2022. 1

[11] Ahmed Hatem, Yiming Qian, and Yang Wang. Point-tta: Test-time adaptation for point cloud registration using multitask meta-auxiliary learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16494–16504, 2023. 2, 3

[12] Ahmed Hatem, Yiming Qian, and Yang Wang. Test-time adaptation for point cloud upsampling using meta-learning. *arXiv preprint arXiv:2308.16484*, 2023. 3

[13] Deepti Hegde and Vishal M Patel. Attentive prototypes for source-free unsupervised domain adaptive 3d object detection. *arXiv preprint arXiv:2111.15656*, 2021. 3

[14] Deepti Hegde, Vishwanath Sindagi, Velat Kilic, A Brinton Cooper, Mark Foster, and Vishal Patel. Uncertainty-aware mean teacher for source-free unsupervised domain adaptive 3d object detection. *arXiv preprint arXiv:2109.14651*, 2021. 3

[15] Deepti Hegde, Velat Kilic, Vishwanath Sindagi, A Brinton Cooper, Mark Foster, and Vishal M Patel. Source-free unsupervised domain adaptation for 3d object detection in adverse weather. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6973–6980. IEEE, 2023. 3

[16] Jie Hong, Shi Qiu, Weihao Li, Saeed Anwar, Mehrtash Harandi, Nick Barnes, and Lars Petersson. Pointcam: Cut-and-mix for open-set point cloud learning. *arXiv preprint arXiv:2212.02011*, 2023. 2

[17] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Advances in Neural Information Processing Systems*, 34:3635–3649, 2021. 3

[18] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021. 1

[19] Xuesong Li, Jose Guivant, Ngaiming Kwok, Yongzhi Xu, Ruowei Li, and Hongkun Wu. Three-dimensional backbone network for 3d object detection in traffic scenes. *arXiv preprint arXiv:1901.08373*, 2019. 2

[20] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016. 2, 7

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020. 2, 3, 7

[22] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021. 3

[23] Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch nor-

malization in test-time adaptation. *arXiv preprint arXiv:2302.05155*, 2023. 1

[24] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021. 1, 3

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013. 8

[26] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8866–8875, 2021. 3

[27] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16709–16718, 2023. 2, 3, 6, 7, 8

[28] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *Internetional Conference on Learning Representations*, 2023. 1

[29] David Osowiechi, Gustavo A Vargas Hakim, Mehrdad Noori, Milad Cheraghalikhani, Ismail Ben Ayed, and Christian Desrosiers. Tttflow: Unsupervised test-time training with normalizing flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2126–2134, 2023. 3

[30] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010. 6, 8

[31] Zicheng Pan, Xiaohan Yu, Miaohua Zhang, and Yongsheng Gao. Ssfe-net: Self-supervised feature enhancement for ultra-fine-grained few-shot class incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6275–6284, 2023. 2

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 7, 8

[33] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 2

[34] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[35] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 586–602. Springer, 2022. 3

[36] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Fabio Poiesi, and Elisa Ricci. Compositional semantic mix for domain adaptation in point cloud segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[37] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020. 1

[38] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7223–7232, 2022. 3

[39] Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16928–16937, 2022. 3

[40] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021. 5

[41] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, 2016. 8

[42] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*, 2022. 2, 6, 7, 8

[43] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248, 2020. 1, 3

[44] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *International Conference on Computer Vision (ICCV)*, 2019. 7

[45] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 2, 3, 6, 7

[46] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 2

[47] 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)*, 2019. 1, 2, 7

[48] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 2

[49] Yanshuo Wang, Jie Hong, Ali Cheraghian, Shafin Rahman, David Ahmedt-Aristizabal, Lars Petersson, and Mehrtash Harandi. Continual test-time domain adaptation via dynamic sample selection. 2023. 3, 7

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6

[51] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 915–924, 2021. 1, 2

[52] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021. 7

[53] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10368–10378, 2021. 3

[54] Teresa Yeo, Oğuzhan Fatih Kar, Zahra Sodagar, and Amir Zamir. Rapid network adaptation: Learning to adapt neural networks using test-time feedback. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4674–4687, 2023. 1

[55] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021. 3

[56] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2023. 4, 5, 6

[57] Weichen Zhang, Wen Li, and Dong Xu. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6769–6779, 2021. 3