MODE-TTA: MULTI-MODE DENSITY MODELING FOR TEST-TIME ADAPTATION OF 3D VISION-LANGUAGE FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

3D vision–language foundation models (VLFMs) for point clouds exhibit strong zero-shot generalization capabilities, yet their performance often deteriorates under real-world distribution shifts. Test-time adaptation (TTA) seeks to address this challenge by adapting models during inference using only unlabeled target data. Recent cache-based TTA methods address distribution shifts in online settings by maintaining a memory of representative test samples and refining predictions based on feature affinities between the current sample and the cached instances. However, their effectiveness is largely dependent on the quality of pseudo-labels, which may deteriorate under substantial domain shifts. Moreover, they do not capture the underlying structure and dynamics of the incoming test distribution. To overcome these limitations, we propose MODE-TTA, a streaming density modeling framework that explicitly models class-conditional feature distributions. Each class is represented by a Gaussian Mixture Model (GMM) to capture the multi-cluster structure of 3D features, with mixture parameters updated incrementally via a hierarchical expectation–maximization procedure guided by soft zero-shot assignments. To further improve cross-modal alignment, we introduce a lightweight residual adaptation for text embeddings. Extensive experiments across diverse distribution shifts demonstrate that MODE-TTA consistently outperforms competitive baselines.

1 INTRODUCTION

Large-scale 3D vision-language foundation models (VLFMs) Zhou et al. (2024); Xue et al. (2024); Liu et al. (2023) have recently demonstrated remarkable zero-shot generalization ability. Models such as Uni3D Zhou et al. (2024) achieve competitive performance on downstream tasks without task-specific training, indicating semantic alignment between 3D geometry and language supervision. However, this zero-shot capability does not guarantee robustness under distribution shifts.

In real-world 3D perception scenarios, particularly those involving LiDAR sensors, distribution shifts are inevitable. Variations in sensor hardware, beam configuration, environmental conditions, weather, object sparsity, occlusions, and viewpoint changes introduce substantial discrepancies between training distributions and deployment environments. As a result, the performance of 3D VLFMs degrades significantly when deployed in unseen domains Sun et al. (2025); Tamjidi et al. (2025); Mirza et al. (2023).

Test-time adaptation (TTA) Wang et al. (2020); Shu et al. (2022); Karmanov et al. (2024); Iwasawa & Matsuo (2021) aims to improve robustness under such shifts without accessing labeled target data. In online TTA settings, samples arrive sequentially in a stream, and the model must adapt progressively as data arrive Karmanov et al. (2024); Zhang et al. (2024b); Han et al. (2024). Existing TTA approaches for VLFMs generally fall into two major paradigms. (1) Methods with parameter updates: these approaches update model parameters at test time using unsupervised objectives Xiao et al. (2025); Liu et al. (2024); Osowiechi et al. (2024). While effective in some cases, they suffer from high computational cost due to backpropagation during inference, which contradicts the requirement for lightweight deployment in real-time 3D perception systems. (2) Methods without parameter updates: to avoid modifying model parameters, recent approaches adopt cache-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

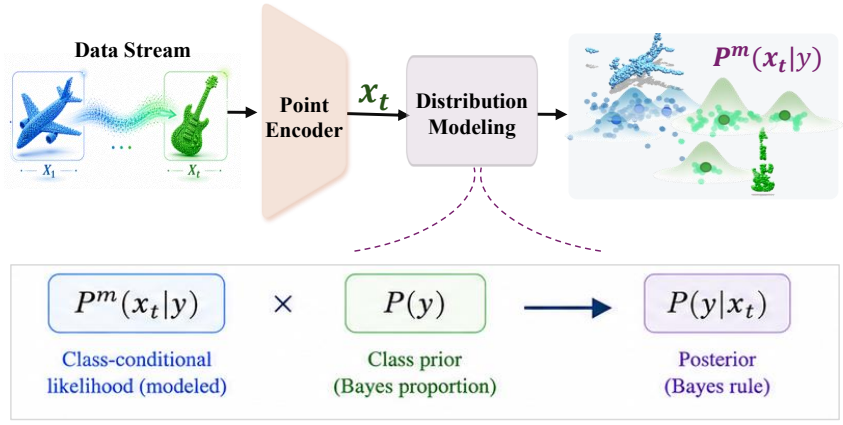

Figure 1: Point clouds arrive sequentially, with only the sample $\mathbf{X}_t$ available at time t. The extracted feature $\mathbf{x}_t$ updates the distribution $P^m(\mathbf{x} \mid y)$. The t-SNE plot illustrates multimodal feature clusters, motivating a mixture-based model.

based strategies Karmanov et al. (2024); Zhang et al. (2024b;a;c); Tamjidi et al. (2025). They store high-confidence test samples as prototypes and refine predictions using cosine similarity between incoming features and cached representations. Although computationally efficient, this strategy introduces a structural bias: high-confidence samples cluster locally far from decision boundaries, limiting prototypes from capturing the full target distribution, especially under intra-class diversity Tamjidi et al. (2025).

Recent empirical analysis suggests that 3D VLFMs exhibit inherently multi-peaked feature distributions within each semantic class Tamjidi et al. (2025), as illustrated by the t-SNE visualization in Fig. 1 (right) for three example classes. Cache-based methods struggle to capture such multi-cluster structures, as they rely on a small set of high-confidence samples representing limited regions of the feature space. UniAdapter Tamjidi et al. (2025) mitigates this issue through online clustering that maintains multiple prototypes per class. However, these clusters are still treated as deterministic representatives and do not explicitly model distributional uncertainty or the geometry of the class-conditional feature space. Distribution-modeling TTA methods, such as DOTA Han et al. (2024) and FreeTTA Dai & Yang (2025), model class-conditional feature distributions using Gaussian assumptions. While more principled, these methods typically assume a single Gaussian per class, enforcing a unimodal distribution and limiting their ability to capture multi-modal feature structures.

In this work, we argue that effective online TTA requires modeling the class-conditional feature distribution with an expressive parametric model capable of capturing multi-cluster structures. The key challenge, as illustrated in Fig. 1, is that data arrives in a streaming manner, where at each time step only the current sample is accessible. Consequently, the model must incrementally learn the class-conditional density from an evolving stream of feature representations. To address this challenge, we introduce MODE-TTA (Multi-MOde DEnsity modeling for Test-Time Adaptation), a streaming distribution modeling framework that represents each class using a Gaussian Mixture Model (GMM). MODE-TTA incrementally updates mixture means, covariances, mixture weights, and class priors through a hierarchical expectation–maximization (EM) procedure, where soft assignments are derived from zero-shot predictions. This enables the model to progressively refine the class-conditional feature distribution as new samples arrive. In addition, we employ residual learning to adjust text embeddings to better align with the evolving feature distribution, improving class separability in the shared feature–text embedding space.

Our main contributions are summarized as follows: (1) We identify the limitations of prototype-based test-time adaptation methods in capturing multi-cluster feature distributions in 3D VLFMs. (2) We propose MODE-TTA, a streaming distribution modeling framework that represents each class with a Gaussian Mixture Model and incrementally updates mixture parameters via a hierarchical EM procedure with soft assignments from zero-shot predictions, leveraging both original and augmented features to stabilize online distribution estimation. (3) We introduce a residual text-

embedding adaptation strategy that refines class embeddings to better align with the learned feature distribution, improving class separability in the embedding space.

2 Related work

3D Vision-Language Foundation Models (3D VLFMs) have emerged as a powerful paradigm for enhancing point cloud comprehension by transferring rich semantic knowledge learned from large-scale image-text corpora into the 3D domain Zhu et al. (2023); Chen et al. (2023); Xue et al. (2024). Notable approaches such as Uni3D Zhou et al. (2024), ULIP Xue et al. (2023), ULIP-2 Xue et al. (2024), and OpenShape Liu et al. (2023) leverage contrastive objectives over large collections of aligned image, text, and point cloud triplets to learn well-structured cross-modal representations. The resulting pre-trained 3D VLFMs exhibit impressive zero-shot generalization and fine-grained geometric-semantic reasoning across a variety of downstream tasks. Nevertheless, distribution shifts between pre-training and deployment settings often degrade their effectiveness, posing significant challenges for reliable operation in real-world environments.

Test-Time Adaptation (TTA) seeks to dynamically adjust model predictions to unseen target domains without requiring target annotations or source data access Niu et al. (2023); Boudiaf et al. (2022). Early methods such as TENT Wang et al. (2020), SHOT Liang et al. (2020), and MEMO Zhang et al. (2022) minimize softmax entropy to boost confidence and generalization. With the rise of VLFMs, TPT Shu et al. (2022), DiffTPT Feng et al. (2023), and SCAP Zhang et al. (2025a) combine entropy minimization with learnable prompt tuning, though at the cost of expensive test-time backpropagation. Training-free methods such as TDA Karmanov et al. (2024), BoostAdapter Zhang et al. (2024b), and PointCache Sun et al. (2025) refine predictions using cached prototypes. However, these approaches do not explicitly model the underlying feature density of the test distribution. Recent methods address this limitation through probabilistic modeling, including Bayesian adaptation Zhou et al. (2025); Zhang et al. (2025b) and Gaussian-based density estimation Han et al. (2024); Dai & Yang (2025). While effective, they typically assume a unimodal class distribution, limiting their ability to capture the complex multimodal structure often present in real-world data.

Test-Time Point Cloud Adaptation has recently attracted increasing attention for improving the generalization of 3D point cloud models across various tasks, including recognition Sun et al. (2025); Wang et al. (2024); Shim et al. (2024), segmentation Zhao et al. (2025); Zou et al. (2024), registration Hatem et al. (2023), object detection Lin et al. (2024); Chen et al. (2024); Yuan et al. (2024), and scene completion Jang et al. (2025). Early approaches focused on adapting model parameters during inference. For example, MATE Mirza et al. (2023) updates encoder parameters through self-training, while Bahri et al. (2025) adapt normalization layers following TENT Wang et al. (2020). To avoid costly parameter updates, subsequent methods perform adaptation through alternative mechanisms. BFTT3D Wang et al. (2024) leverages source representations via a non-parametric adapter, while CloudFixer Shim et al. (2024) and 3DD-TTA Dastmalchi et al. (2025) adapt the input point cloud through diffusion-guided geometric corrections. More recently, adaptation methods have been extended to 3D VLFMs Zhou et al. (2024); Xue et al. (2024); Liu et al. (2023). PointCache Sun et al. (2025) constructs global and local caches from high-confidence predictions and summarizes local patch features using k-means clustering. Uni-Adapter Tamjidi et al. (2025) maintains class-specific cluster centers as dynamic prototypes in a cache, updating them online and enforcing consistency through graph-based label smoothing.

3 Method

3.1 Overview and Setting

We study online test-time adaptation for 3D VLFMs Zhou et al. (2024); Xue et al. (2024); Liu et al. (2023) under distribution shift. In the online setting, unlabeled target samples arrive sequentially as a stream, and the model must adapt without access to the full target set. At each time step t, only the current sample is available, with no access to future samples or ground-truth labels.

We assume a pretrained and frozen 3D VLFM composed of a text encoder $\mathcal{E}_T(\cdot)$ and a point cloud encoder $\mathcal{E}_P(\cdot)$ as depicted in Fig. 2. Given an incoming point cloud $\mathbf{X}_t \in \mathbb{R}^{N\times 3}$ at time t, the

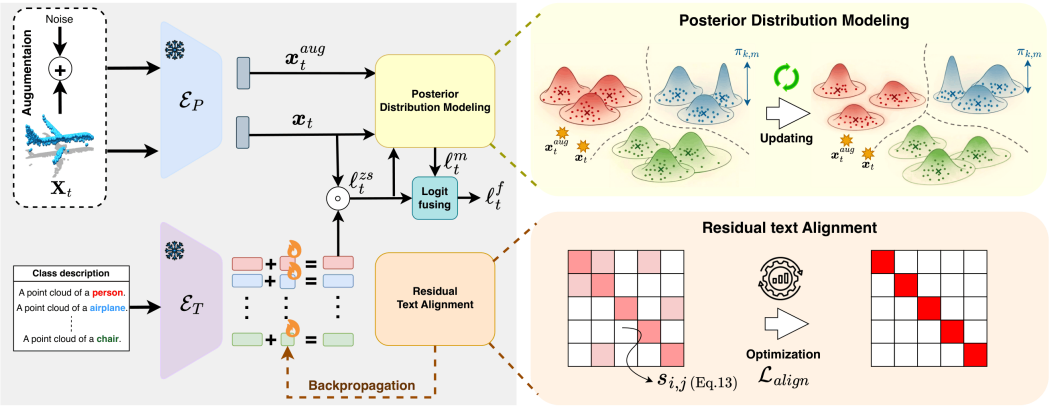

Figure 2: **Overview of the proposed method.** Given a test sample $\mathbf{X}_t$ and its noise-augmented version, both are encoded by the point encoder $\mathcal{E}_P$ to obtain features $\mathbf{x}_t$ and $\mathbf{x}_t^{aug}$. The zero-shot score ℓ_t^{zs} is computed via cosine similarity with class text embeddings from the text encoder $\mathcal{E}_T$. Both features are fed into the Posterior Distribution Modeling module to produce the modeling score ℓ_t^m. In parallel, the Residual Text Alignment module refines class text embeddings to improve separation. Finally, ℓ_t^{zs} and ℓ_t^m are fused via an entropy-based strategy to obtain the final logit ℓ_t^f.

point cloud encoder produces a D-dimensional feature vector $\mathbf{x}_t = \mathcal{E}_P(\mathbf{X}_t) \in \mathbb{R}^D$, while the frozen text encoder provides K class embeddings $\{\mathbf{w}_k\}_{k=1}^K$, where $\mathbf{w}_k = \mathcal{E}_T(\text{prompt}_k) \in \mathbb{R}^D$. The zero-shot classifier computes cosine similarities between $\mathbf{x}_t$ and the class text embeddings, followed by softmax normalization:

$$\gamma_{t,k}^{zs} = P^{zs}(y = k \mid \mathbf{x}_t) = \frac{\exp(\tau \text{sim}(\mathbf{x}_t, \mathbf{w}_k))}{\sum_{j=1}^K \exp(\tau \text{sim}(\mathbf{x}_t, \mathbf{w}_j))} \tag{1}$$

where y denotes the class label, $\text{sim}(\cdot, \cdot)$ is the cosine similarity, τ is the temperature parameter, and $P^{zs}(y = k \mid \mathbf{x}_t)$ is the zero-shot posterior. We model the target feature space with a streaming GMM updated online via EM, using both original and noise-augmented features for robustness. Its density posterior is fused with the zero-shot posterior through confidence-aware weighting. We further introduce *Residual Text Alignment*, which learns class-wise residual embeddings $\{\mathbf{\Delta}k\}_{k=1}^K$ by maximizing likelihood under the learned densities while keeping the VLFM backbone frozen.

3.2 Posterior Distribution Modeling

Point cloud features often exhibit multi-modal distributions due to geometric variability, viewpoint changes, and intra-class shape diversity Tamjidi et al. (2025). Consequently, samples from the same class may form multiple distinct clusters rather than a single compact region, as illustrated in Fig. 1. To capture this structure, we model each class-conditional feature distribution with a Gaussian Mixture Model, which efficiently represents multiple modes while supporting online updates. The posterior induced by the learned density model is computed via Bayes' rule:

$$P^m(y = k \mid \mathbf{x}_t; \theta) \;\propto\; P^m(\mathbf{x}_t \mid y = k; \theta_k)\, P^m(y = k), \tag{2}$$

where $\boldsymbol{\theta}_k$ denotes the parameters of the class-conditional distribution for class k. Thus, density-based classification requires estimating the class-conditional density $P^m(\mathbf{x}_t \mid y = k; \theta_k)$ and the class prior $P^m(y = k)$.

To capture the multi-modal structure of point cloud features, we model the class-conditional density of each class k with an M-component GMM:

$$P^m(\mathbf{x} \mid y = k; \theta_k) = \sum_{m=1}^{M} \pi_{k,m}\, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{k,m}, \mathbf{\Sigma}_{k,m}) \tag{3}$$

where $\pi_{k,m} \geq 0$ and $\sum_{m=1}^M \pi_{k,m} = 1$. The parameter set is $\theta_k = \{\pi_{k,m}, \boldsymbol{\mu}_{k,m}, \mathbf{\Sigma}_{k,m}\}_{m=1}^M$, where $\boldsymbol{\mu}_{k,m} \in \mathbb{R}^D$ and $\mathbf{\Sigma}_{k,m} \in \mathbb{R}^{D \times D}$ denote the mean and covariance of the m-th component, respectively.

Since high-dimensional VLFM embeddings (e.g., $D = 1024$) make full covariance estimation computationally expensive and statistically unstable, we adopt diagonal covariance matrices,

$$\mathbf{\Sigma}_{k,m} = \mathrm{diag}\big(\sigma^2_{k,m,1}, \ldots, \sigma^2_{k,m,D}\big)\,, \tag{4}$$

which capture per-dimension uncertainty while enabling efficient, stable online updates. We develop an online EM algorithm that incrementally updates the GMM parameters $\pi_{k,m}, \boldsymbol{\mu}_{k,m}, \mathbf{\Sigma}_{k,m}$ as samples arrive. Since target samples are unlabeled, MODE-TTA uses zero-shot predictions as soft class assignments to guide online estimation of the class-conditional density model.

E-Step (Responsibility Estimation). For each incoming sample $\mathbf{x}_t$, we estimate its responsibility with respect to both class labels and mixture components. At the class level, we do not have access to ground-truth labels. Instead, we use the zero-shot posterior (Eq. 1) as a soft proxy for class responsibility, $\gamma^{zs}_{t,k} = P^{zs}(y = k \mid \mathbf{x}_t)$, which provides a label-free estimate of how likely $\mathbf{x}_t$ belongs to class k. For class k, the likelihood of $\mathbf{x}_t$ under each mixture component is given by:

$$\ell_{t,k,m} = \mathcal{N}\Big(\mathbf{x}_t \mid \boldsymbol{\mu}^{(t-1)}_{k,m}, \mathbf{\Sigma}^{(t-1)}_{k,m}\Big)\,. \tag{5}$$

Within each class, we normalize component likelihoods to obtain mixture responsibilities:

$$r_{t,k,m} = \frac{\pi^{(t-1)}_{k,m}\,\ell_{t,k,m}}{\sum_{j=1}^{M}\pi^{(t-1)}_{k,j}\,\ell_{t,k,j}}. \tag{6}$$

Finally, we combine the class-level and within-class responsibilities to obtain the joint responsibility:

$$\gamma_{t,k,m} = \gamma^{zs}_{t,k}\,r_{t,k,m} \tag{7}$$

This hierarchical assignment enables each sample softly update all classes and mixture components, while emphasizing plausible classes and well-matched modes.

M-step (Streaming Parameter Updates). Given the responsibilities $\gamma_{t,k,m}$ from the E-step, the GMM parameters are updated online using streaming sufficient statistics. These updates are obtained by maximizing the expected complete-data log-likelihood and admit efficient recursive forms suitable for online adaptation (see Appendix A for the derivation).

Let $c^{(t)}_{k,m}$ denote the cumulative soft count associated with component (k, m):

$$c^{(t)}_{k,m} = c^{(t-1)}_{k,m} + \gamma_{t,k,m}. \tag{8}$$

The recursive updates for the component means and variances are given by

$$\begin{aligned}\boldsymbol{\mu}^{(t)}_{k,m} &= \frac{c^{(t-1)}_{k,m}\,\boldsymbol{\mu}^{(t-1)}_{k,m} + \gamma_{t,k,m}\,\mathbf{x}_t}{c^{(t)}_{k,m}},\\ \boldsymbol{\sigma}^{2,(t)}_{k,m} &= \frac{c^{(t-1)}_{k,m}\,\boldsymbol{\sigma}^{2,(t-1)}_{k,m} + \gamma_{t,k,m}(\mathbf{x}_t - \boldsymbol{\mu}^{(t)}_{k,m}) \odot (\mathbf{x}_t - \boldsymbol{\mu}^{(t)}_{k,m})}{c^{(t)}_{k,m}}.\end{aligned} \tag{9}$$

where $\odot$ denotes element-wise multiplication. The mixture weights and class priors are then estimated from the normalized soft counts:

$$\pi^{(t)}_{k,m} = \frac{c^{(t)}_{k,m}}{n^{(t)}_k}, \qquad P^m(y = k) = \frac{n^{(t)}_k}{\sum_{j=1}^{K} n^{(t)}_j}, \tag{10}$$

where $n^{(t)}_k = \sum_{m=1}^{M} c^{(t)}_{k,m}$ denotes the effective soft count of class k.

Noise-Based Augmentation. Since increased data diversity improves density estimation, we augment each incoming point cloud with Gaussian noise and compute the embedding of the perturbed sample $\mathbf{x}^{aug}_t$ alongside the original embedding $\mathbf{x}_t$ (Fig. 2). Both embeddings are used in the M-step updates to enrich the learned distribution. The augmented sample shares the same class-level responsibility $\gamma_{t,k}$ as the original embedding. This effectively increases the local sampling density around each observation, leading to more stable and robust parameter updates during online learning. We show in the ablation study (Sec. 4.4) that this strategy effectively improves performance.

3.3 Residual Text Alignment

To improve cross-modal alignment under distribution shift, we refine each text embedding $\mathbf{w}_k$ with a learnable residual vector $\mathbf{\Delta}_k$ conditioned on the evolving target distribution (Fig. 2):

$$\tilde{\mathbf{w}}_k = \frac{\mathbf{w}_k + \mathbf{\Delta}_k}{\|\mathbf{w}_k + \mathbf{\Delta}_k\|_2}. \tag{11}$$

Using the learned class-conditional distributions, we score each refined prototype by its log-likelihood under each class:

$$s_{i,j} \;=\; \log P^m(\tilde{\mathbf{w}}_i \mid y = j). \tag{12}$$

We then optimize a symmetric InfoNCE-style Oord et al. (2018) alignment objective that encourages $s_{i,i}$ to be large relative to mismatched pairs:

$$\mathcal{L}_{\text{align}} = -\sum_{i=1}^{K} \log \frac{\exp(s_{i,i})}{\sum_{j=1}^{K} \exp(s_{i,j})} \;-\; \sum_{j=1}^{K} \log \frac{\exp(s_{j,j})}{\sum_{i=1}^{K} \exp(s_{i,j})} \tag{13}$$

This bidirectional formulation enforces that each refined text embedding is most likely under its corresponding class density, while each class density assigns the highest likelihood to its matched text prototype. The residuals $\{\mathbf{\Delta}_k\}$ are optimized online, while all VLFM backbone parameters remain fixed. Note that gradients are computed only with respect to the residual vectors, and no backpropagation is performed through the VLFM backbone.

3.4 Logit Fusion

At inference, we represent both the density model and the zero-shot classifier using pre-softmax class logits. Following Eq. 2, the density-model logit is obtained by

$$\ell^m_{t,k} = \lambda\big(\log P^m(\mathbf{x}_t \mid y = k) + \log P^m(y = k)\big), \tag{14}$$

while the zero-shot logit is given by $\ell^{zs}_{t,k} = \tau\,\mathrm{sim}(\mathbf{x}_t, \mathbf{w}_k)$. The scaling factor $\lambda = \min(\rho n, \eta)$ controls the contribution of the density model, where n denotes the number of processed test samples and ρ and η are hyperparameters. This schedule gradually increases the influence of the density model as more target samples are observed, while limiting its impact during the early stages when density estimates are less reliable.

To adaptively fuse the two predictors, we weight their logits according to inverse predictive entropy:

$$\ell^f_{t,k} = \left(\frac{H^m_t}{H^{zs}_t + H^m_t}\right)\ell^{zs}_{t,k} + \left(\frac{H^{zs}_t}{H^{zs}_t + H^m_t}\right)\ell^m_{t,k}, \tag{15}$$

where H^{zs}_t and H^m_t are the entropies of the zero-shot and density-based posteriors, respectively. This confidence-aware fusion assigns greater weight to the more certain predictor.

4 Experiments

4.1 Setup

Datasets. We evaluate our approach under out-of-distribution (OOD) shifts using ModelNet40-C Sun et al. (2022), ShapeNet-C Mirza et al. (2023), and ScanObjectNN-C Mirza et al. (2023). These benchmarks introduce 15 types of synthetic corruptions, including density variations, noise, and geometric transformations, each evaluated across five severity levels. To further assess whether the method can improve alignment even without distribution shift, we also conduct experiments on the clean test splits of ModelNet40 Wu et al. (2015), ShapeNetCore-v2 Chang et al. (2015), and ScanObjectNN Uy et al. (2019). Additionally, we evaluate performance on the large-scale 3D dataset OmniObject3D Wu et al. (2023) (216 classes), which contains diverse object categories.

Baselines. We compare MODE-TTA with classical TTA and vision–language adaptation methods, including LAME Boudiaf et al. (2022), T3A Iwasawa & Matsuo (2021), CloudFixer Shim et al. (2024), 3DD-TTA Dastmalchi et al. (2025), TPT Shu et al. (2022), TDA Karmanov et al. (2024),

Table 1: Top-1 accuracy (%) on ModelNet40-C using Uni3D-Large (batch size = 1). * denotes VLFM-based TTA methods. Source-Only indicates no adaptation. Best and second-best results are marked in **bold** and underline, respectively.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	occlusion	lidar	Avg.
Source-Only	57.37	54.01	70.21	61.91	60.69	51.74	52.39	67.50	74.87	72.40	71.02	63.97	58.95	47.24	22.93	59.15
LAME (CVPR22)	57.33	54.09	70.10	61.63	60.94	51.94	52.27	67.50	75.16	72.45	71.27	63.90	59.04	47.33	22.61	59.17
3DD-TTA (WACV25)	60.53	59.76	63.53	64.71	67.91	50.73	51.09	59.36	66.98	64.26	58.67	59.36	53.77	36.10	32.37	56.06
CloudFixer † (ECCV24)	65.44	65.84	63.90	68.11	72.77	52.80	53.81	52.88	63.21	59.36	61.18	56.16	52.18	29.09	24.60	56.09
T3A (NIPS21)	69.40	**70.26**	41.89	63.33	70.74	61.26	59.27	72.20	79.33	77.59	78.36	71.07	65.51	49.59	32.01	64.12
TPT* (NIPS22)	60.65	57.20	76.32	61.15	63.47	55.39	55.78	71.36	74.77	75.20	73.01	65.41	60.97	47.60	18.10	61.02
TDA* (CVPR24)	62.20	61.63	75.16	65.52	67.18	57.46	57.94	70.95	76.94	74.43	73.26	67.46	63.17	50.69	30.39	63.63
DOTA* (NIPS25)	64.46	63.29	72.97	67.58	73.83	60.13	60.37	70.79	77.63	74.96	74.27	68.84	65.15	51.70	32.37	64.83
Point-Cache* (CVPR25)	64.34	64.87	73.95	68.31	71.68	62.84	65.19	73.22	77.80	77.15	75.77	69.77	68.31	54.78	32.98	66.73
Uni-Adapter* (AAAI26)	66.82	65.52	**78.32**	72.25	72.04	65.60	66.61	**77.51**	80.63	79.05	**79.30**	**75.29**	73.38	56.92	36.26	69.70
MODE-TTA (Ours)	**70.38**	70.14	77.92	**73.06**	**73.85**	**70.83**	**70.70**	76.81	**82.05**	**79.54**	78.68	74.43	**73.70**	**58.79**	**40.31**	**71.41**

Table 2: Top-1 accuracy (%) on ShapeNet-C using Uni3D-Large (batch size = 1). * denotes VLFM-based TTA methods. Source-Only indicates no adaptation. Best and second-best results are marked in **bold** and underline, respectively.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	occlusion	lidar	Avg.
Source-Only	60.33	55.75	65.95	65.02	59.04	59.41	60.23	79.06	71.07	75.62	73.87	76.82	63.22	2.37	1.06	57.92
LAME (CVPR22)	60.43	55.89	66.04	65.12	59.09	59.42	60.33	79.13	71.16	75.74	74.08	76.99	63.35	2.31	1.02	58.01
3DD-TTA (WACV25)	65.78	64.16	55.00	61.75	**68.05**	55.06	56.20	74.20	67.35	68.06	61.19	73.01	56.29	2.50	0.97	55.30
CloudFixer † (ECCV24)	65.57	**65.30**	58.15	69.53	63.65	55.02	56.71	69.89	58.67	65.65	65.64	70.36	55.09	3.75	2.46	57.24
T3A (NIPS21)	60.60	53.12	20.70	44.19	49.31	46.35	44.37	70.31	63.01	64.86	63.83	68.24	52.60	1.00	0.87	46.89
TPT* (NIPS22)	62.87	56.63	69.20	64.70	59.10	58.29	60.43	81.59	75.23	76.93	74.56	80.52	63.02	2.17	1.25	59.10
TDA* (CVPR24)	62.75	58.95	68.33	67.14	62.09	61.28	62.56	79.00	71.44	75.93	74.79	77.17	64.44	3.82	1.81	59.43
DOTA* (NIPS25)	61.13	56.87	66.94	66.45	59.81	61.08	62.09	80.26	72.69	76.54	75.05	78.31	64.79	1.64	1.22	58.99
Point-Cache* (CVPR25)	62.63	56.71	66.51	65.85	61.15	59.79	61.49	75.89	69.47	72.61	70.81	73.82	63.41	3.64	1.67	57.70
Uni-Adapter* (AAAI26)	66.89	62.23	71.38	68.62	64.15	67.42	**67.33**	80.76	75.69	78.11	77.01	79.64	**70.14**	4.36	2.47	62.41
MODE-TTA (Ours)	**66.95**	63.81	**71.67**	**70.62**	66.06	**68.22**	66.79	**82.27**	**75.84**	**79.05**	**78.40**	**81.10**	69.00	**4.74**	**2.88**	**63.15**

DOTA Han et al. (2024), PointCache Sun et al. (2025), and Uni-Adapter Tamjidi et al. (2025). CloudFixer, 3DD-TTA, PointCache, and Uni-Adapter target 3D adaptation, while others are adapted from 2D; TPT, TDA, and DOTA are designed for vision–language models.

Implementation. We use Uni3D-Large Zhou et al. (2024) as the main 3D vision-language backbone and also evaluate on ULIP-2 Xue et al. (2024). Adaptation is online with batch size one. We set $\rho = 0.02$, $\eta = 0.1$, and $M = 8$ for all datasets, with ρ and η selected by validation grid search; sensitivity analyses are provided in Sec. 4.4. Component means are initialized from text embeddings with small random perturbations, and diagonal variances with std 0.0001. All experiments use corruption severity 5 on an RTX PRO 6000. Further details are given in Appendix B.

4.2 Results on OOD Data

Table 1 reports Top-1 accuracy on ModelNet40-C under distribution shifts. Source-Only achieves 59.15% average accuracy, while existing TTA methods provide limited gains. MODE-TTA reaches 71.41%, achieving the best accuracy on most corruptions, including *lidar* (40.31%) and *occlusion* (58.79%). Compared with DOTA Han et al. (2024), which uses a single Gaussian per class, MODE-TTA is more robust through mixture-based density modeling and residual text alignment.

Table 2 reports Top-1 accuracy on ShapeNet-C, where most methods perform better than on ModelNet40-C, leaving less room for improvement. Nevertheless, MODE-TTA achieves the best average accuracy of 63.15%, outperforming both classical TTA and VLFM-specific methods. Table 3 summarizes results on ScanObjectNN-C, which includes severe noise, occlusions, and real-world scanning artifacts. MODE-TTA reaches 50.90% accuracy, exceeding Uni-Adapter (46.33%)

Table 3: Top-1 accuracy (%) on ScanObjectNN-C using Uni3D-Large (batch size = 1). * denotes VLFM-based TTA methods. Source-Only indicates no adaptation. Best and second-best results are marked in **bold** and underline, respectively.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	occlusion	lidar	Avg.
Source-Only	29.78	25.99	40.62	45.96	30.64	33.05	34.42	56.28	47.16	54.22	54.04	56.80	43.55	9.98	8.61	38.07
LAME (CVPR22)	29.60	26.85	51.12	52.32	31.33	33.73	37.18	56.80	45.78	54.56	54.22	56.29	45.27	6.71	3.96	39.05
3DD-TTA (WACV25)	32.19	30.81	27.71	39.59	34.60	25.82	26.51	45.61	38.04	36.14	33.39	45.09	33.05	8.61	4.47	30.78
CloudFixer † (ECCV24)	36.83	33.22	36.14	48.19	37.69	27.54	30.46	45.44	40.28	38.73	38.21	46.13	35.28	10.67	**11.70**	34.43
T3A (NIPS21)	34.94	32.70	34.08	43.37	33.22	36.49	36.32	62.65	55.42	61.46	62.31	64.03	56.97	9.47	8.61	42.14
TPT* (NIPS22)	30.04	28.43	40.95	46.76	32.68	35.55	34.39	56.60	50.66	53.81	54.39	60.30	42.70	11.13	6.07	38.96
TDA* (CVPR24)	31.33	28.40	52.67	53.36	32.36	36.32	40.45	56.80	46.82	55.25	55.76	57.66	49.40	8.09	4.65	40.62
DOTA* (NIPS25)	35.63	29.43	57.32	54.56	35.64	39.59	43.20	58.34	49.57	56.63	57.66	62.31	50.09	7.92	6.20	42.94
Point-Cache* (CVPR25)	30.98	27.19	55.25	56.45	33.73	40.79	43.03	59.50	49.23	60.07	56.63	57.66	49.05	8.26	4.30	42.13
Uni-Adapter* (AAAI26)	35.28	37.69	53.35	59.55	39.07	42.16	**52.49**	60.58	51.97	61.61	60.24	60.92	54.38	**20.82**	4.81	46.33
MODE-TTA (Ours)	**45.09**	**40.27**	**64.89**	**64.20**	**44.40**	**44.41**	50.08	**69.70**	**59.55**	**64.20**	**66.26**	**65.40**	**58.34**	16.87	9.81	**50.90**

Table 4: Generalization to ULIP-2. MODE-TTA consistently improves performance across all datasets and corruption types.

	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	Avg.
ModelNet-C	ULIP-2	20.91	19.12	58.22	56.64	22.40	46.67	48.62	63.08	67.42	69.69	66.73	56.15	53.68	49.95
	ULIP-2 + MODE-TTA	**27.86**	**24.86**	**62.48**	**61.31**	**28.03**	**49.23**	**50.89**	**69.24**	**74.39**	**73.46**	**72.12**	**64.46**	**62.15**	**55.42**
ShapeNet-C	ULIP-2	21.25	18.84	57.67	57.49	25.02	49.20	50.63	70.58	67.86	70.83	67.14	69.34	55.11	52.38
	ULIP-2 + MODE-TTA	**24.12**	**21.35**	**64.27**	**64.75**	**31.26**	**53.68**	**54.89**	**75.27**	**72.05**	**74.20**	**71.53**	**74.30**	**59.84**	**57.04**
SONN-C	ULIP-2	16.52	12.73	36.83	39.07	19.10	23.40	25.81	48.19	39.75	47.67	38.21	44.40	29.43	32.39
	ULIP-2 + MODE-TTA	**17.66**	**14.80**	**40.10**	**41.99**	**20.05**	**24.44**	**26.85**	**52.84**	**47.84**	**52.49**	**42.68**	**46.47**	**32.70**	**35.45**

by 4.57%. Its improvement over DOTA Han et al. (2024) also underscores the limitations of modeling each class with a single Gaussian.

To further evaluate the generality of MODE-TTA beyond Uni3D, we additionally apply it to ULIP-2 Xue et al. (2024). As shown in Table 4, MODE-TTA consistently improves ULIP-2 across all datasets and corruption types, demonstrating the robustness and effectiveness of the proposed adaptation strategy across different 3D vision-language foundation models.

4.3 Results on Clean Data

While TTA methods are typically designed to handle distribution shifts, we investigate whether MODE-TTA also benefits performance on clean data. Fig. 3 compares the zero-shot accuracy of ULIP-2 and Uni3D before and after applying MODE-TTA across four datasets: ModelNet40, ShapeNetCore-v2, ScanObjectNN, and OmniObject3D. Interestingly, MODE-TTA improves performance even without corruption, indicating that beyond robustness to distribution shifts, it also refines alignment between feature representations and class distributions.

4.4 Ablation Studies

Effect of Different Components. Table 5 evaluates the three MODE-TTA components: *posterior distribution modeling*, *noise-based augmentation*, and *residual text alignment*. Mixture density modeling provides substantial gains across datasets, while augmentation further improves density estimation and update stability. Residual text alignment adds further gains by better aligning text embeddings with learned feature distributions.

Effect of Residual Learning. Fig. 4 shows the likelihood matrix $\log P^m(\mathbf{w}_i \mid y = j)$ on ModelNet40-C before adaptation, after adaptation without residual learning, and with residual learn-

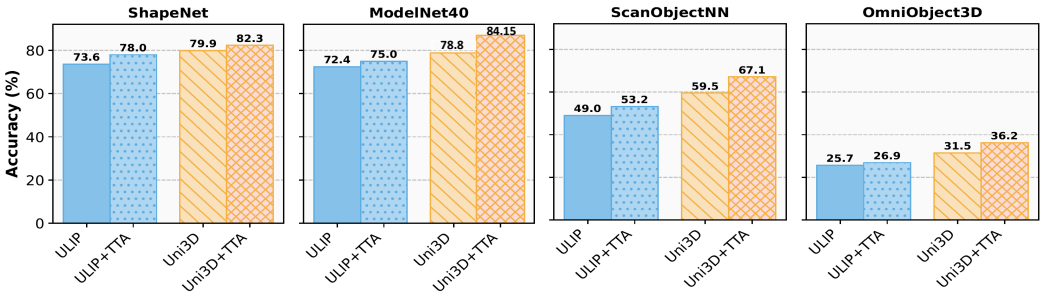

Figure 3: Zero-shot performance before and after adaptation on clean datasets.

Table 5: Effect of MODE-TTA components

GMM	Aug.	Residual	ModelNet-C	ShapeNet-C	SONN-C
✗	✗	✗	59.15	57.92	38.07
✓	✗	✗	69.30	61.20	48.37
✓	✓	✗	70.73	62.68	49.98
✓	✓	✓	**71.41**	**63.15**	**50.90**

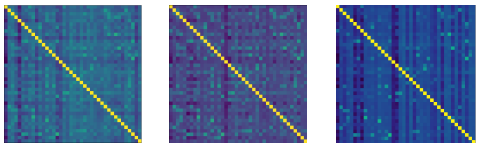

Figure 4: Likelihood matrix before adaptation, without residual learning, and with residual learning.

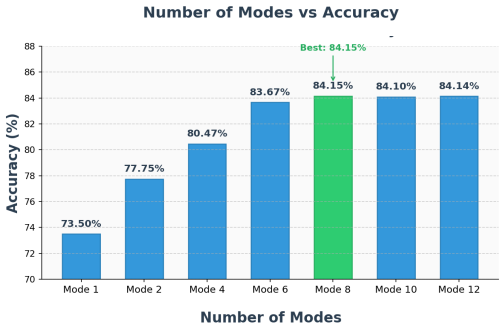

Figure 5: Effect of the number of mixture modes M on accuracy, with the best performance at $M = 8$.

ing. Weak diagonal dominance initially indicates text–feature misalignment; adaptation sharpens the densities but retains off-diagonal responses. Residual learning produces a strongly diagonal matrix, indicating improved class-wise alignment.

Hyperparameter Analysis. We analyze the number of mixture components M on clean ModelNet40 Wu et al. (2015). As shown in Fig. 5, a single Gaussian ($M = 1$) performs substantially worse, underscoring the need to model multi-modal feature structure. Performance improves with larger M and peaks at $M = 8$, used throughout our experiments.

Efficiency Analysis. We evaluate the computational overhead of MODE-TTA by comparing throughput and accuracy on ScanObjectNN-C. As shown in more detail in Appendix D, even the simplest variant based solely on GMM (without augmentation or residual learning) outperforms existing TTA methods on most corruption types while retaining approximately 97% of the original model's throughput, corresponding to only $\sim$3% overhead. The full model further improves accuracy with a modest reduction in speed, demonstrating a favorable accuracy–efficiency trade-off.

5 Conclusion and Future Work

We address the degradation of 3D vision–language foundation models under distribution shift with MODE-TTA, an online test-time adaptation framework based on streaming density modeling. MODE-TTA represents each class with a Gaussian mixture model to capture multi-modal feature structure and updates its parameters online through hierarchical EM guided by soft zero-shot assignments. Residual text adaptation further improves cross-modal alignment by refining text embeddings toward the target distribution. Extensive experiments show that MODE-TTA consistently outperforms prototype-based and single-Gaussian baselines while maintaining a favorable accuracy–efficiency trade-off. These results highlight the importance of explicitly modeling multi-modal feature distributions for robust test-time adaptation. Future work will explore open-world settings with unknown-class detection and more expressive online density models.

AI USE STATEMENT

Generative AI tools were used solely to assist with editing and polishing the manuscript for clarity, grammar, and readability. They were not used to develop the methodology, formulate hypotheses, design experiments, analyze or interpret results, or generate scientific claims. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide implementation details, hyperparameter settings, and additional ablations in the main paper and Appendix. All datasets and backbone models used in our experiments are publicly available.

REFERENCES

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani Oghani, Milad Cheraghalikhani, David Osowiechi, Farzad Beizaee, Gustavo A. Vargas Hakim, Ismail Ben Ayed, and Christian Desrosiers. Test-time adaptation in point clouds: Leveraging sampling variation with weight averaging. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 266–275, February 2025.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, 2022.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7020–7030, 2023.

Zhuoxiao Chen, Junjie Meng, Mahsa Baktashmotlagh, Yonggang Zhang, Zi Huang, and Yadan Luo. Mos: Model synergy for test-time adaptation on lidar-based 3d object detection. *arXiv preprint arXiv:2406.14878*, 2024.

Qiyuan Dai and Sibei Yang. Free on the fly: Enhancing flexibility in test-time adaptation with online em. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9538–9548, 2025.

Hamidreza Dastmalchi, Aijun An, Ali Cheraghian, Shafin Rahman, and Sameera Ramasinghe. Test-time adaptation of 3d point clouds via denoising diffusion models. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 1566–1576, February 2025.

Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2704–2714, 2023.

Zongbo Han, Jialong Yang, Guangyu Wang, Junfan Li, Qianli Xu, Mike Zheng Shou, and Changqing Zhang. Dota: Distributional test-time adaptation of vision-language models. *arXiv preprint arXiv:2409.19375*, 2024.

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*, pp. 16494–16504, 2023.

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.

Hyun-Kurl Jang, Jihun Kim, Hyeokjun Kweon, and Kuk-Jin Yoon. Talos: Enhancing semantic scene completion via test-time adaptation on the line of sight. *Advances in Neural Information Processing Systems*, 37:74211–74232, 2025.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14162–14171, 2024.

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*, 2020.

Hongbin Lin, Yifan Zhang, Shuaicheng Niu, Shuguang Cui, and Zhen Li. Monotta: Fully test-time adaptation for monocular 3d object detection. In *European Conference on Computer Vision*, pp. 96–114. Springer, 2024.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023.

Zichen Liu, Hongbo Sun, Yuxin Peng, and Jiahuan Zhou. Dart: Dual-modal adaptive online prompting and knowledge retention for test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14106–14114, 2024.

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

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 *ICLR*, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

David Osowiechi, Mehrdad Noori, Gustavo Adolfo Vargas Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Watt: Weight average test-time adaption of clip. *arXiv preprint arXiv:2406.13875*, 2024.

Hajin Shim, Changhun Kim, and Eunho Yang. Cloudfixer: Test-time adaptation for 3d point clouds via diffusion-guided geometric transformation. In *European Conference on Computer Vision*, pp. 454–471. Springer, 2024.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

Hongyu Sun, Qiuhong Ke, Ming Cheng, Yongcai Wang, Deying Li, Chenhui Gou, and Jianfei Cai. Point-cache: Test-time dynamic and hierarchical cache for robust and generalizable point cloud analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1263–1275, 2025.

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.

Mehran Tamjidi, Hamidreza Dastmalchi, Mohammadreza Alimoradijazi, Ali Cheraghian, Aijun An, and Morteza Saberi. Adapt-as-you-walk through the clouds: Training-free online test-time adaptation of 3d vision-language foundation models. *arXiv preprint arXiv:2511.15311*, 2025.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23231–23241, 2024.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814, June 2023.

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 (CVPR)*, June 2015.

Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Qi Wang, and Cees GM Snoek. Dynaprompt: Dynamic test-time prompt tuning. *arXiv preprint arXiv:2501.16404*, 2025.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1179–1189, 2023.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27091–27101, 2024.

Jiakang Yuan, Bo Zhang, Kaixiong Gong, Xiangyu Yue, Botian Shi, Yu Qiao, and Tao Chen. Regtta3d: Better regression makes better test-time adaptive 3d object detection. In *European Conference on Computer Vision*, pp. 197–213. Springer, 2024.

Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. *Advances in Neural Information Processing Systems*, 37:32111–32136, 2024a.

Chenyu Zhang, Kunlun Xu, Zichen Liu, Yuxin Peng, and Jiahuan Zhou. Scap: Transductive test-time adaptation via supportive clique-based attribute prompting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30032–30041, 2025a.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022.

Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-Tao Xia. Boostadapter: Improving vision-language test-time adaptation via regional bootstrapping. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Yabin Zhang, Wenjie Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory networks: A versatile adaptation approach for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28718–28728, 2024c.

Youjia Zhang, Youngeun Kim, Young-Geun Choi, Hongyeob Kim, Huiling Liu, and Sungeun Hong. Backpropagation-free test-time adaptation via probabilistic gaussian alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Jichun Zhao, Haiyong Jiang, Haoxuan Song, Jun Xiao, and Dong Gong. Dˆ 3ctta: Domain-dependent decorrelation for continual test-time adaption of 3d lidar segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11864–11874, 2025.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024.

Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen Lei. Bayesian test-time adaptation for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2639–2650, 2023.

Tianpei Zou, Sanqing Qu, Zhijun Li, Alois Knoll, Lianghua He, Guang Chen, and Changjun Jiang. Hgl: Hierarchical geometry learning for test-time adaptation in 3d point cloud segmentation. In *European Conference on Computer Vision*, pp. 19–36. Springer, 2024.