Point-Cache: Test-time Dynamic and Hierarchical Cache for Robust and Generalizable Point Cloud Analysis

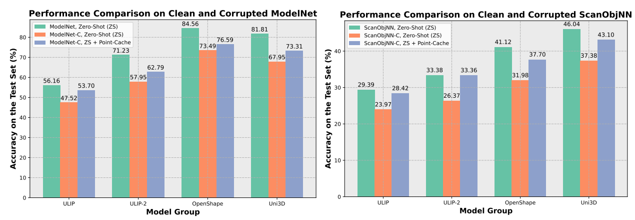
Hongyu Sun[1,2], Qiuhong Ke[2], Ming Cheng[2], Yongcai Wang[1]*, Deying Li[1], Chenhui Gou[2], Jianfei Cai[2]
[1]Department of Computer Science, Renmin University of China, China
[2]Department of Data Science & AI, Monash University, Australia
{sunhongyu,ycw,deyingli}@ruc.edu.cn {qiuhong.ke,ming.cheng,jianfei.cai}@monash.edu

Abstract

*This paper proposes a general solution to enable point cloud recognition models to handle distribution shifts at test time. Unlike prior methods, which rely heavily on training data (often inaccessible during online inference) and are limited to recognizing a fixed set of point cloud classes predefined during training, we explore a more practical and challenging scenario: adapting the model solely based on online test data to recognize both previously seen classes and novel, unseen classes at test time. To this end, we develop **Point-Cache**, a hierarchical cache model that captures essential clues of online test samples, particularly focusing on the global structure of point clouds and their local-part details. Point-Cache, which serves as a rich 3D knowledge base, is dynamically managed to prioritize the inclusion of high-quality samples. Designed as a plug-and-play module, our method can be flexibly integrated into large multimodal 3D models to support open-vocabulary point cloud recognition. Notably, our solution operates with efficiency comparable to zero-shot inference, as it is entirely training-free. Point-Cache demonstrates substantial gains across 8 challenging benchmarks and 4 representative large 3D models, highlighting its effectiveness. Code is available at* https://github.com/auniquesun/Point-Cache.

1. Introduction

In recent years, 3D sensors such as LiDARs and RGB-D cameras have been widely adopted in robotics and electric vehicles for their ability to provide reliable 3D geometry measurements [11, 16, 20, 31, 59, 61–63, 68]. Point clouds are among the most direct data formats produced by these 3D sensors. Although remarkable progress has been made in 3D point cloud recognition, the success is primarily based on the assumption that the test data and the model training data are identically distributed [2, 9, 26, 32, 37, 39, 40, 64, 87]. However, this assumption is frequently violated in real-world scenarios due to complex geometries, as well as sensing and processing errors [43, 50, 51].



(a) ModelNet & ModelNet-C. (b) ScanObjNN & ScanObjNN-C.

Figure 1. **Recognition accuracy comparison on clean and corrupted point cloud datasets**. The suffix -C indicates datasets with corruptions. Models experience a severe performance drop when data corruptions arise. **Point-Cache** effectively narrows down the performance gap between clean and corrupted data. The hardest split of ScanObjNN is used.

In practice, a majority of models remain vulnerable to distribution shifts, such as out-of-distribution (OOD) samples [6, 7, 50, 69], data corruptions [43, 44, 51, 56], and more. As shown in Fig. 1, notable performance gaps (*e.g*., 11+%) occur when models are tested on clean (ModelNet [70]) versus corrupted data (ModelNet-C [43]). These issues drive researchers to enhance the robustness and generalization of point cloud recognition methods. Related efforts can be grouped into three main streams. The first focuses on training models that can transfer to target data. These models learn a shared feature space between source and target domains [10, 41, 45, 67], or apply meta-learning frameworks to handle geometry shifts by training multiple classifiers [19], or design learnable prompts to elicit general knowledge from pre-trained large 3D models [50]. However, possible shifts are unpredictable, and the training-based methods cannot account for all cases during training. As a result, these models are limited when new shifts arise at test time [75]. The second stream addresses the problem

*Corresponding author.

through test-time adaptation strategies [35, 46, 66]. These strategies improve performance with various adaptation techniques, such as masked auto-encoding for point clouds (MATE [35]), static prototype memory (BFTT3D [66]), and diffusion models (CloudFixer [46]). Nevertheless, we find that these approaches rely heavily on training data. For instance, MATE [35] requires pre-training on the entire training set, and BFTT3D [66] employs training data to build a static and offline prototype memory to store point cloud features. This heavy dependence on training data poses challenges, particularly when such data is unavailable or when distribution shifts occur at test time. In these cases, revisiting the training data may not be helpful to adapt the model to online test data. Additionally, these test-time methods can only recognize classes seen during training, which limits their practicality in generalizable point cloud analysis.

Inspired by advancements in language-guided image recognition [22, 38, 42, 79], the third branch of works utilizes contrastive pre-training on large-scale triplets (point, text, image) to connect flexible language descriptions and enable open-vocabulary point cloud recognition [29, 72, 73, 88]. These large multimodal 3D models demonstrate promising zero-shot performance and provide a new paradigm for generalizable point cloud analysis. However, zero-shot predictions are performed independently for each test sample, with the final accuracy simply averaged across the test set. This approach does not fully exploit the distribution of test data, particularly the statistics of reliable predictions from large 3D models, which could further enhance accuracy and generalization. At this point, we pose the following question: *can we take advantages of both the promising zero-shot ability of large 3D models and test-time adaptation techniques to unlock more robust and generalizable point cloud recognition?*

In this work, to fully exploit the data distribution during test time, we develop a cache model to store critical fingerprints of test samples. These fingerprints consist of key-value pairs derived by large multimodal 3D models, where the key represents the sample feature extracted by the point encoder, and the value represents the predicted (pseudo) label from the model. This cache model is entirely constructed from online test data without accessing any training samples. Our approach is thus more practical and challenging than prior methods [10, 19, 35, 41, 45, 46, 50, 66, 67]. Our cache model is hierarchical, as it stores not only the global features of a point cloud in a global cache but also its local-part details in a local cache, which are essential for distinguishing subtle differences among classes. We also empirically demonstrate that this coarse-to-fine design enhances robustness and generalization in the wild. Additionally, our cache model is dynamic, since we need to continuously update the cache with online test data to prioritize high-quality key-value pairs. The constructed cache serves as a powerful 3D knowledge base, capturing key characteristics of online test data. When a new sample arrives, we query the knowledge base, compute the affinity between the sample and cached keys, and produce a prediction by weighting the cached (pseudo) labels accordingly. Our method is training-free and flexible, requiring only a pre-trained 3D model and test data. More details will be provided in Sec. 3.

The superiority of the proposed method is supported by comprehensive comparisons with previous strong baselines across 8 challenging benchmarks, containing up to 1,156 point cloud classes. Moreover, we implement the hierarchical cache as a plug-and-play module and integrate it into representative large 3D models, such as ULIP [72], ULIP-2 [73], OpenShape [29] and Uni3D [88], to enhance test-time point cloud recognition dynamically. The proposed cache model brings notable and consistent gains over the baselines. For instance, our model yields +6.18% and +4.84% absolute accuracy improvements average over 7 types of corruptions in ModelNet-C [43] based on ULIP [72] and ULIP-2 [73]. On Objaverse-LVIS [7] that includes 1,156 categories, Point-Cache boosts the accuracy by +2.10% absolute points compared to the zero-shot ULIP-2 [73]. In summary, the contributions of our paper are threefold.

- To our knowledge, Point-Cache is the first to explore test-time point cloud recognition in a more practical and challenging setting, where only online test data is available and the model needs to generalize to both known classes and unseen new classes, which is beyond the capability of prior test-time methods.
- We propose a hierarchical cache model to capture essential distribution clues of online test samples, creating an accurate profile for various point clouds from global to local perspectives. We implement the hierarchical cache as a plug-and-play component, enabling diverse 3D backbones to achieve enhanced test-time performance.
- Comprehensive experiments and consistent gains across 8 public datasets and 4 large 3D models demonstrate the efficacy of Point-Cache.

2. Related Work

Test-time adaptation in 3D point cloud analysis receives relatively less attention compared to the 2D image community [5, 12, 14, 21, 23, 27, 28, 30, 34, 47, 52, 54, 60, 78, 80]. This technique has recently emerged in 3D point cloud tasks, including recognition [35, 46, 66], registration [17], upsampling [18] and object detection [65, 74, 85]. Among these, MATE [35], BFTT3D [66], and CloudFixer [46] are specifically focused on test-time point cloud recognition and are thus closely related to our work. However, as previously analyzed, these methods depend heavily on training data from the source domain to construct their pipelines,

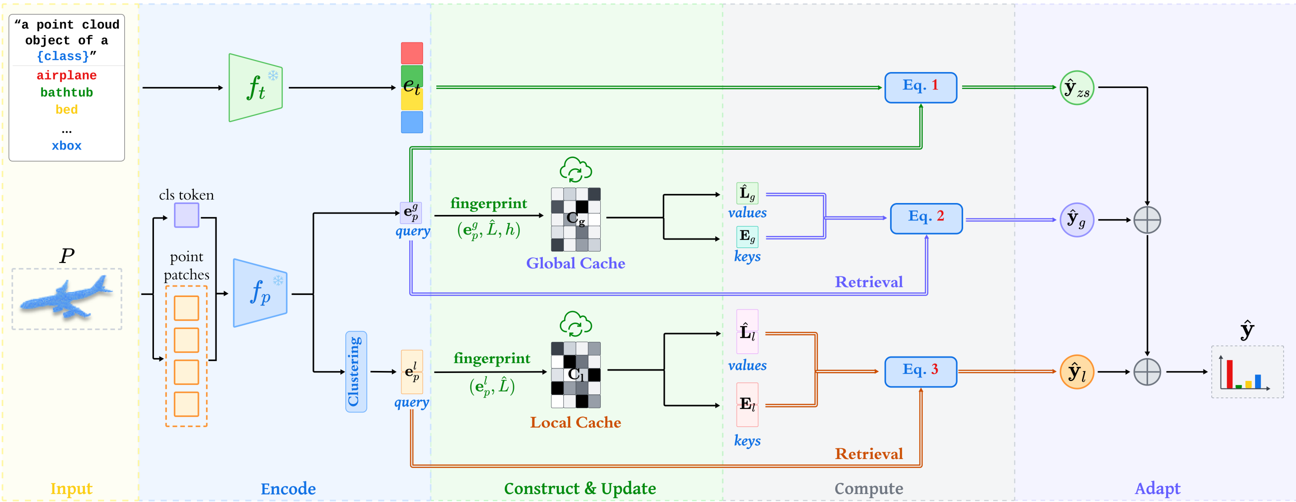


Figure 2. **The overall pipeline of Point-Cache.** The zero-shot predictions $\hat{\mathbf{y}}_{zs}$ of large 3D models are effectively adapted by our global cache logits $\hat{\mathbf{y}}_g$ and local cache logits $\hat{\mathbf{y}}_l$ to handle the distribution shifts, enabling robust and generalizable point cloud analysis.

which may not be available or helpful to understand target data distributions at test time. Furthermore, they lack effective strategies to leverage the distribution of test samples. By contrast, Point-Cache is constructed entirely from online test data without access to any training samples, making our setting both more challenging and more practical. Additionally, our cache model, powered by large 3D models, can generalize to both known classes and unseen new categories, an advancement that prior point cloud adaptation/generalization frameworks have not achieved [10, 19, 35, 41, 45, 46, 66, 67].

Cache models that store sample features in memory for further reference and reuse have been explored in fields like language modeling [15, 24, 33, 58], image classification [3–5, 13, 21, 23, 36, 48, 55, 58, 82, 83] and point cloud recognition [53, 66, 84]. Our proposed cache model stands out from previous ones in several key aspects: (1) Our cache model is constructed based on test samples whereas previous cache models rely on a large training set for construction, as in Tip-Adapter [82], CaFo [83], PointNN [84], Point-PEFT [53] and BFTT3D [66]. (2) Our cache model is hierarchically designed to capture both global and local-part features of a point cloud object, while existing methods store only global representations [3–5, 13, 21, 48, 53, 58, 66, 82–84]. We empirically demonstrate that this coarse-to-fine design improves the capture of target data features, thereby enhancing test-time performance. (3) Our cache model is dynamically updated with online test samples, incorporating a selective mechanism to ensure high-quality key-value pairs replace less reliable ones when the cache reaches its capacity. In contrast, prior cache models, which are constructed offline using the entire training set, are static and consume a substantial amount of memory [53, 66, 84].

Large Multi-modal 3D Models [25, 29, 72, 73, 76, 81, 88, 89] spark a wave of open-vocabulary point cloud understanding by pre-training on large-scale datasets [7] in recent years. For instance, ULIP-2 [73], OpenShape [29] and Uni3D [88] pre-train the 3D encoders on million-scale (point, image, text) triplets using contrastive learning. These pre-trained point cloud encoders are then transferred to downstream 3D tasks like classification and retrieval, demonstrating strong zero-shot performance that surpasses traditional 3D models [32, 39, 40, 64, 71, 87] which are typically tailored to specific datasets [1, 56, 70]. This new paradigm presents a promising direction for generalizable point cloud processing. Our work is complementary to these models: rather than developing another large 3D model, we build upon their capabilities to enhance robustness and generalization for point cloud analysis at test time. Our hierarchical cache, backed by these large 3D models, generalizes to unseen classes beyond training, addressing a significant limitation of current test-time methods for point cloud recognition [35, 46, 66].

3. Method

This section elaborates on the details of Point-Cache, the overall pipeline of our approach is depicted in Fig. 2.

Preliminary. We begin by reviewing the zero-shot inference process of existing large multimodal 3D models [29, 72, 73, 81, 88, 89]. These models use separate encoders to map inputs from a point cloud $P \in \mathbb{R}^{N\times 3}$ and text T (*e.g.*, "a point cloud object of a `{class}`") to a shared feature space. Denoting the point cloud encoder as $f_p(\cdot)$ and the text encoder as $f_t(\cdot)$, we can obtain the global feature $\mathbf{e}_p^g = f_p(P) \in \mathbb{R}^d$ of the point cloud P and the text feature $\mathbf{e}_t = f_t(T) \in \mathbb{R}^d$, where d is the feature dimension. To recognize the input point cloud P in a zero-shot manner, these models replace "`{class}`" in text T with each of possible C category names and compute the class probability $\hat{\mathbf{y}}_{zs}$ using Eq. 1.

$$\hat{\mathbf{y}}_{zs} = \{\hat{y}_i|_{i=1}^{C}\}, \quad \hat{y}_i = \frac{\exp(sim(\mathbf{e}_p^g, \mathbf{e}_t^i)/\tau)}{\sum_{j=1}^{C}\exp(sim(\mathbf{e}_p^g, \mathbf{e}_t^j)/\tau)} \tag{1}$$

Here $\mathbf{e}_t^i$ indicates the text encoding of the i-th category. $sim(\cdot,\cdot)$ measures the cosine similarity between inputs, and τ is a temperature coefficient. Finally, the point cloud P is classified into the category with the highest probability, indexed by $\hat{L} = \arg\max_i\{\hat{y}_i|_{i=1}^{C}\}$.

As described in Sec. 1, our hierarchical cache stores the fingerprints of online test samples, which consists of key-value pairs. The key is the test sample feature $\mathbf{e}_p^g$ extracted by the point encoder of a pre-trained large 3D model, and the value is the class label $\hat{L}$ predicted by the zero-shot inference. In addition, we compute the entropy $h = -\sum_{i=1}^{C}\hat{y}_i \log \hat{y}_i$ of the zero-shot prediction $\hat{\mathbf{y}}_{zs}$, as it is essential to assess the confidence with which the large 3D model produces this prediction. This entropy value is employed to filter out high-quality samples in our cache model.

3.1. Text & Point Cloud Encoding

We can obtain the text encoding $\mathbf{e}_t$ of the input T and the global feature $\mathbf{e}_p^g$ of a point cloud P as introduced in the preliminary. However, it is *non-trivial* to represent the local-part features $\mathbf{e}_p^l$ of the point cloud P due to the following challenges analyzed in Fig. 3:

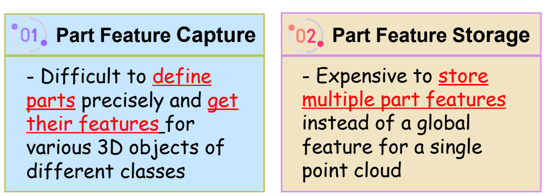


Figure 3. **Challenges in encoding part feature** for various 3D objects of different classes: Part Feature Capture & Storage.

To address these challenges, we made a careful analysis and developed effective solutions, seeing Fig. 9 in the Supplementary for intuitive illustration. (1) A straightforward approach might involve using an additional pre-trained segmentation network to segment point cloud parts, but this would be computationally prohibitive during online inference. Instead, we propose a fuzzy definition of parts. Specifically, since the architecture of the point encoder of large 3D models is based on Transformer [8, 57, 77] and the input is a sequence of point patches, we avoid strict part definitions according to physical semantics (*e.g.*, head, body, wing, and tail in an airplane), and instead treat the output point patches from the last encoder layer as local part representations. (2) A second challenge is that a long point patch sequence can become resource-intensive; for example, the point encoder in ULIP-2 [73] generates 512 point patches per object, which would require substantial memory. To mitigate this problem, we propose to summarize these point patches into m centers $\mathbf{e}_p^l \in \mathbb{R}^{m\times d}$ using K-Means clustering ($m < 10$), reducing memory usage by two orders of magnitude compared to storing all patches.

3.2. Construction & Update of Hierarchical Cache

To thoroughly explore the distribution of test data, we develop a hierarchical cache to record the fingerprints of online test samples. This hierarchical cache includes both a global cache, which stores overall information, and a local cache, which captures detailed part-level information for each point cloud. The coarse-to-fine structure enables more precise profiling of test data and distinguishes our approach from previous cache models used in image recognition [3–5, 13, 21, 23, 36, 48, 55, 58, 82, 83] and point cloud analysis [53, 66, 84]. The complete construction process of hierarchical cache is outlined in Alg. 1 of the Supplementary.

Global cache is represented by $\mathbf{C}_g = (\mathbf{E}_g, \hat{\mathbf{L}}_g, \mathbf{h}_g)$, storing (feature, label, entropy) triplets for a set of point cloud objects. To manage storage efficiently, we design a selection mechanism to identify high-quality test samples to place into the cache.

Construction: We set an upper bound K on the number of samples per class in the cache, allowing at most K fingerprints (shots) can be placed in each category. Initially, the global cache is empty. When a new test sample P arrives, we generate its fingerprint $(\mathbf{e}_p^g, \hat{L}, h)$ and assign it to class $\hat{L}$.

Update: With the number of online samples grows, we check whether the number of shots for class $\hat{L}$ has hit the upper bound. If it is not the case, the fingerprint of current sample is appended to class $\hat{L}$. Otherwise, we locate the fingerprint with highest entropy within class $\hat{L}$ and compare it to the entropy of current test sample. If the current test sample has a lower entropy, we replace the highest-entropy fingerprint with the new one; otherwise, the current sample is skipped. When the cache is full, $\mathbf{E}_g$ has dimensions $CK \times d$, $\hat{\mathbf{L}}_g$ has dimensions $CK \times 1$, and $\mathbf{h}_g$ has dimensions $CK \times 1$.

Local cache. Although the global feature of a point cloud captures expressive information, it lacks the local details of the 3D object, which are critical for point cloud recognition, especially when shapes share an overall appearance but differ in local parts. The local cache addresses this limitation by storing part-level information. We denote the local cache as $\mathbf{C}_l = (\mathbf{E}_l, \hat{\mathbf{L}}_l)$, which store the fingerprints of local parts of point cloud objects. Here $\mathbf{E}_l$ represents part features of the test samples and $\hat{\mathbf{L}}_l$ indicates their labels.

Accordingly, we define the local-part fingerprint of a test sample P as $(\mathbf{e}_p^l, \hat{L})$, where $\mathbf{e}_p^l \in \mathbb{R}^{m\times d}$ corresponds to local-part features and $\hat{L}$ is the class label of P.

Construction & Update: Note that the sample selective mechanism of the local cache follows the same rules as the global cache, meaning the local-part fingerprint of a test sample is only stored if it qualifies for inclusion in the global cache. So there is no need to record the entropy h again in the local cache. When the cache is full, $\mathbf{E}_l$ has dimensions $(m \cdot CK) \times d$ and $\hat{\mathbf{L}}_l$ has dimensions $CK \times 1$.

3.3. Test-time Adaptation by Hierarchical Cache

With the hierarchical cache serving as a rich knowledge base of online point cloud data, we can adapt the zero-shot predictions of large 3D models for new samples, effectively addressing distribution shifts that occur at test time. The adaptation follows a coarse-to-fine process as below.

Test-time adaptation by global cache. During adaptation, a new sample Q is treated as a query to retrieve knowledge from the global cache, which we use to produce the adaptation logits. Specifically, the retrieval process is simulated by calculating the affinity A_g between the query and cached features $\mathbf{E}_g$. This allows the new sample to establish connections with cached high-quality samples, providing it with a contextual understanding of the online test data. We then generate the adaptation logits $\hat{\mathbf{y}}_g$ by weighed ensembling of cache labels $\hat{\mathbf{L}}_g$, as shown in Eq. 2.

$$A_g = \exp(-\beta_g(1 - \mathbf{e}_q^g \mathbf{E}_g^\top)),\ \hat{\mathbf{L}}_g = \mathrm{OH}(\hat{\mathbf{L}}_g),\ \hat{\mathbf{y}}_g = A_g \hat{\mathbf{L}}_g \tag{2}$$

where $\mathbf{e}_q^g \in \mathbb{R}^{1\times d}$ represents the global feature of sample Q, and $\mathbf{E}_g \in \mathbb{R}^{CK\times d}$ denotes the cached keys. The 'OH' operation transforms $\hat{\mathbf{L}}_g$ into a one-hot encoding. The term $\mathbf{e}_q^g \mathbf{E}_g^\top \in \mathbb{R}^{1\times CK}$ measures the cosine similarity between the query and all keys, with β_g acting as a hyper-parameter to modulate the sharpness and the exponential operation ensuring similarity values fall within (0, 1]. The affinity $A_g \in \mathbb{R}^{1\times CK}$, which grasps the relationship between the query and cached keys, is utilized to ensemble the transformed one-hot labels $\hat{\mathbf{L}}_g \in \mathbb{R}^{CK\times C}$, yielding global cache adaptation logits $\hat{\mathbf{y}}_g \in \mathbb{R}^{1\times C}$.

Test-time adaptation by local cache. While global cache adaptation is effective, it may not handle fine-grained distribution shifts at test time. To achieve a more precise adjustment, we exploit local part knowledge. The local-part features $\mathbf{e}_q^l$ of the point cloud Q serve as queries to the local cache. During retrieval, we compute the affinity A_l between $\mathbf{e}_q^l$ and all cached part features $\mathbf{E}_l$. The affinity A_l is then used to weight the cache labels $\hat{\mathbf{L}}_l$, producing local cache adaptation logits $\hat{\mathbf{y}}_l$ as in Eq. 3.

$$A_l = \exp(-\beta_l(1 - \mathbf{e}_q^l \mathbf{E}_l^\top)),\ \hat{\mathbf{L}}_l = \mathrm{OH}(\hat{\mathbf{L}}_l),\ \hat{\mathbf{y}}_l = \phi(A_l \hat{\mathbf{L}}_l) \tag{3}$$

where $\mathbf{e}_q^l \in \mathbb{R}^{m\times d}$, $\mathbf{E}_l \in \mathbb{R}^{(m\cdot CK)\times d}$, and $\hat{\mathbf{L}}_l \in \mathbb{R}^{(m\cdot CK)\times C}$. The function $\phi(\cdot)$ is an average pooling operation over m parts to produce class logits. The term $e_q^l \mathbf{E}_l^\top \in \mathbb{R}^{m\times(m\cdot CK)}$ measures the cosine similarity between the part features of sample Q and all part features in the cache. Likewise, the similarity is modulated by the coefficient β_l and scaled with the exponential function, resulting in the affinity $A_l \in \mathbb{R}^{m\times(m\cdot CK)}$. The term $A_l \hat{\mathbf{L}}_l \in \mathbb{R}^{m\times C}$ represents the class logits for all m parts, which are transformed by a pooling operation $\phi(\cdot)$ to derive the final local cache adaptation logits $\hat{\mathbf{y}}_l \in \mathbb{R}^{1\times C}$.

Overall prediction combines three components as shown in Eq. 4. The zero-shot prediction $\hat{\mathbf{y}}_{zs}$ of large multimodal 3D model is effectively adapted using our global cache prediction $\hat{\mathbf{y}}_g$ and local cache prediction $\hat{\mathbf{y}}_l$, balanced by the coefficients α_g and α_l.

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_{zs} + \alpha_g \hat{\mathbf{y}}_g + \alpha_l \hat{\mathbf{y}}_l \tag{4}$$

4. Experiments

4.1. Experimental Settings

Datasets. To evaluate the *robustness* of point cloud recognition, we choose four public datasets that contain various point cloud corruptions: ModelNet-C [43] and three variants of ScanObjectNN-C [56]. ModelNet-C [43] introduces 7 atomic corruptions, including adding global outliers, adding local outliers, dropping global structure, dropping local parts, rotation, scaling, and jittering. Other corruptions can be derived from these atomic corruptions, as discussed in [43]. We apply these atomic corruptions to the three variants of ScanObjectNN [56] and produce their corrupted versions. To assess *generalization* on unseen new data, we test the proposed method on four challenging benchmarks: OmniObject3D [69] (216 classes), Objaverse-LVIS [7] (1,156 classes), the hardest variant of ScanObjectNN [56] and widely used ModelNet40 [70]. Recognition accuracy (%) is the primary evaluation metric.

Implementation Details. We choose ULIP [72], ULIP-2 [73], OpenShape [29] and Uni3D [88] as the large multimodal 3D models for experiments. Their pre-trained weights are loaded and then frozen. Global and local-part features are generated by modifying the forward pass function without altering the model architecture. The maximum number of samples per class in the cache, K, is set to 3, and each point cloud is clustered into $m = 3$ local parts. Both the balance factors α_g and α_l are set to 4.0 while the coefficients β_g and β_l are set to 3.0. We will examine the design choices in the ablation studies. To optimize memory and computation, we convert the input point clouds and model parameters into 'dtype' `fp16` in Pytorch. All experiments are conducted on two 4090 GPUs. Additional details can be found in the Supplementary.

Table 1. **Comparison of recognition accuracy on ModelNet-C that contains 7 types of corruptions**. Each clean point cloud has 1024 points and the corruption severity level is 2. The last column is the average over 7 corruption types. The best results are in **bold** and the second best are underlined. The setting applies to the following tables unless otherwise specified.

Method	**Clean Data**	**Corruption Type**							**Avg.**
	ModelNet	Add Global	Add Local	Drop Global	Drop Local	Rotate	Scale	Jitter	
ULIP [72]	56.16	33.55	43.92	54.70	50.89	55.27	50.20	44.08	47.52
+**Global Cache**(Ours)	62.12	45.79	**47.98**	56.85	53.89	60.25	54.34	48.91	52.56
+**Hierarchical Cache**(Ours)	**64.22**	**46.15**	47.85	**59.16**	**56.00**	**61.47**	**55.35**	**49.92**	**53.70**
ULIP-2 [73]	71.23	65.15	54.62	68.76	57.98	70.30	67.10	21.76	57.95
+**Global Cache**(Ours)	73.95	67.02	59.32	71.35	61.59	72.37	68.40	28.20	61.18
+**Hierarchical Cache**(Ours)	**74.53**	**68.11**	**61.26**	**73.22**	**63.65**	**73.34**	**70.42**	**29.50**	**62.79**
O-Shape [29]	**84.56**	71.64	67.79	81.56	73.58	82.01	78.48	59.36	73.49
+**Global Cache**(Ours)	84.52	74.72	72.77	**82.41**	75.12	**83.18**	**78.93**	67.91	76.43
+**Hierarchical Cache**(Ours)	84.04	**74.84**	**73.70**	82.21	**76.26**	82.66	78.12	**68.35**	**76.59**
Uni3D [88]	81.81	72.45	56.36	68.15	67.18	79.94	75.36	56.24	67.95
+**Global Cache**(Ours)	83.14	76.13	66.49	71.43	69.81	81.52	75.85	61.43	71.81
+**Hierarchical Cache**(Ours)	**83.87**	**77.51**	**71.15**	**72.16**	**70.75**	**81.77**	**77.31**	**62.52**	**73.31**

Table 2. **Comparison of recognition accuracy across a suite of datasets**. S-PB_RS_T50 is the hardest split of ScanObjectNN. O-LVIS: Objaverse-LVIS. Omni3D: OmniObject3D. The number under each dataset indicates the number of points (pts) for each object. In Omni3D, each point cloud can be represented by a different number of points. The last column is the average accuracy on these datasets.

Method	ModelNet40 (10000 pts)	S-PB_RS_T50 (2048 pts)	O-LVIS (10000 pts)	Omni3D			**Avg.**
				(1024 pts	4096 pts	16384 pts)	
ULIP [72]	58.75	46.44	6.24	8.39	7.75	7.28	22.48
+**Global Cache**(Ours)	61.22	50.21	**7.02**	10.00	9.36	8.43	24.37
+**Hierarchical Cache**(Ours)	**62.93**	**51.80**	**7.02**	**10.47**	**9.75**	**8.90**	**25.15**
ULIP-2 [73]	72.97	47.13	30.26	26.36	29.20	26.58	38.75
+**Global Cache**(Ours)	74.51	51.70	**32.65**	28.51	31.10	28.53	41.17
+**Hierarchical Cache**(Ours)	**75.53**	**54.98**	32.36	**29.37**	**31.24**	**29.44**	**42.15**
Uni3D [88]	88.41	65.19	**55.42**	31.52	41.98	41.86	54.09
+**Global Cache**(Ours)	88.86	**68.51**	53.36	34.97	45.13	45.19	56.00
+**Hierarchical Cache**(Ours)	**89.18**	68.24	55.19	**35.82**	**45.60**	**45.89**	**56.65**

4.2. Test-time Robustness and Generalization

Robustness. The robustness analysis of our training-free test-time hierarchical caches takes place on 4 point cloud corruption datasets [43, 56]. As observed in Tab. 1, the zero-shot accuracy of various large 3D models on ModelNet-C is considerably boosted by our global cache model, with increases such as +5.04% for ULIP [72], +3.23% for ULIP-2 [73], +2.94% for OpenShape [29] and +3.86% for Uni3D [88], averaged across 7 corruption types. By combining global and local cache together, our hierarchical cache model attains further improvements, highlighting the benefits of incorporating a local cache. Notably, these gains extend beyond corrupted data, as our hierarchical cache model also improves recognition accuracy on the clean ModelNet dataset, *e.g.*, +8.06% absolute increase over ULIP's zero-shot performance. An exception is OpenShape, which maintains a slightly higher zero-shot accuracy on clean data than our cache model. However, when considering the 7 corruption types, our global and hierarchical cache models outperform OpenShape by 2.94% and 3.10%, respectively, demonstrating improved robustness against data corruption at test time. Consistent improvements are also showcased on the variants of ScanObjectNN, shown in Tab. 5, 6 and 7 in the Supplementary.

Generalization. We evaluated the generalization capability of Point-Cache on four representative benchmarks that were unseen during the pre-training of the large 3D models. These benchmarks generally do not contain data corruptions. While ModelNet40 [70] and ScanObjectNN [56] contain a limited number of classes, OmniObject3D [69] and Objaverse-LVIS [7] represent a broader range of 3D concepts, covering 216 and 1,156 classes, respectively. As shown in Tab. 2, Point-Cache consistently enhances the recognition accuracy across datasets and 3D backbones. For instance, our hierarchical cache built on ULIP and ULIP-2 yields absolute improvements of 6.31% and 5.51% , respectively, average over 6 scores. On ScanObjectNN, our hierarchical cache significantly raises the zero-shot prediction of ULIP and ULIP-2, from 27.20% to 51.80% for ULIP and from 36.22% to 54.98% for ULIP-2. On the challenging Objaverse-LVIS (O-LVIS), our global cache with ULIP-2 successfully realizes a 2.39% absolute gains in zero-shot accuracy across 1,156 classes, a notable achievement. Furthermore, our cache model enables Uni3D to attain an 89.18% accuracy on ModelNet40 without any training, effectively matching the fully-supervised PointNet (89.20%).

4.3. Memory Usage and Throughput

Memory. We use Uni3D [88] as the baseline and adapt its zero-shot predictions with Point-Cache. The experiments are conducted on 3 datasets as shown in Tab. 3. The batch size is set to 1 because we need to update the cache and adapt the class logits according to every single sample. The results indicate our global cache and hierarchical cache consume similar or slightly higher memory than the baseline on each dataset. Furthermore, as the number of classes increases rapidly, the memory usage grows at a slower rate. This is primarily because the memory consumption is dominated by the large number of parameters in Uni3D (*e.g.*, 1,016.5M). In contrast, the hierarchical cache built on O-LVIS uses only approximately 7.1M parameters, which is negligible compared to Uni3D. Detailed calculations are provided in the Supplementary.

Table 3. **Comparison of memory usage (MB)**. The batch size is set to 1 and the used device is an RTX 4090. The number under each dataset indicates #Classes. Hierar: Hierarchical Cache.

Method	ModelNet-C (40)	Omni3D (216)	O-LVIS (1156)	#Params (M)
Uni3D	**5,062**	**5,062**	**5,062**	1,016.5
+**Global**(Ours)	**5,062**	5,064	5,070	1,016.5
+**Hierar**(Ours)	5,064	5,068	5,090	1,016.5

Throughput. We evaluate the throughput of Point-Cache on ModelNet40, where throughput is measured as the number of test samples processed per second (abbreviated as t/s). The results are reported in Tab. 4. Point-Cache runs slightly slower than zero-shot inference due to the additional overhead introduced by cache updates, logits computation, *etc*. However, the throughput degradation is minimal, *e.g.*, a 0.04 t/s drop for OpenShape with the global cache, and a 0.07 t/s drop for ULIP-2 with the hierarchical cache. These results prove that Point-Cache brings remarkable accuracy improvements with negligible computational cost.

4.4. Ablation Study

We use Uni3D with our hierarchical cache and 4 public datasets for ablation analysis, as shown in Fig. 4.

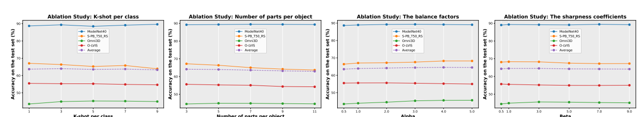
Shot size per class in the global cache. The upper bound K on the number of samples per class is a critical variable

Table 4. **Comparison of throughput (t/s) for different models on ModelNet40**. The batch size is set to 1 and the device is a 4090 GPU. The results are averaged over all test samples.

Method	Zero-shot	+Global Cache	+Hierar Cache
ULIP	**11.25**	11.21	11.18
ULIP-2	**11.18**	11.14	11.11
OpenShape	**9.80**	9.76	9.74
Uni3D	**9.78**	9.75	9.73

as it affects the total size of the hierarchical cache. We analyze its impact on the test-time recognition accuracy on 4 datasets, shown in Fig. 4a. As K increases, accuracy trend differ across datasets; for example, accuracy increases on Omni3D but decreases on S-PB_T50_RS. We adopt $K = 3$ as it produces the best average accuracy (63.98%). The results indicate that our cache model does not require a large number of samples to achieve strong performance.

Number of local parts per 3D object. To optimize memory and computation, we cluster hundreds of point patches of a 3D object into m parts. We investigate the effect of this variable on test-time performance, as exhibited in Fig. 4b. For three out of the four datasets, namely S-PB_T50_RS, Omni3D, and O-LVIS, accuracy generally decreases as the number of parts increases, suggesting that a point cloud does not need to be represented with an excessive number of parts. We choose $m = 3$ by default, as this setting reaches the best average accuracy across the 4 datasets.



(a) shot size K (b) m parts (c) α_g and α_l (d) β_g and β_l

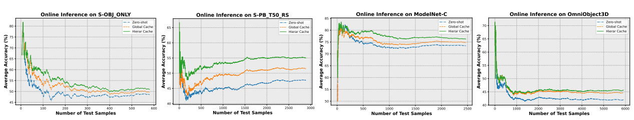
Figure 4. **Ablation studies on the hyper-parameters in the cache design**, including the shot size K per class, the number of parts m per object, the balance factors in the final prediction logits and the sharpness coefficients in affinity computation.

The balance factors. The factors α_g and α_l control the relative weights of the global cache prediction $\hat{\mathbf{p}}_g$ and local cache prediction $\hat{\mathbf{p}}_l$. To simplify experimentation, we set $\alpha_g = \alpha_l$ in the ablation study. Candidate values are $\{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}$. As shown in Fig. 4c, increasing these factors improves performance on S-PB_T50_RS and Omni3D, while it has minimal effect on ModelNet40 and O-LVIS. We set the factors to 4.0, which yields the optimal average accuracy across all four datasets.

The sharpness coefficients. The coefficients β_g and β_l are used to modulate the sharpness of the query-key affinity. Here we analyze their impact on test-time accuracy on 4 benchmarks. Likewise, we let $\alpha_g = \alpha_l$ and the candidate values are $\{0.5, 1.0, 3.0, 5.0, 7.0, 9.0\}$. As Fig. 4d displays, accuracy initially improves slightly with increasing sharpness but then decreases as sharpness becomes too high. We set the sharpness to 3.0 that yields the optimal result average over 4 datasets.

4.5. Visualization

Online Inference. We visualize the online inference process of Point-Cache using different large 3D models and compare three model variants: zero-shot prediction, the model with our global cache, and the model with our hierarchical cache, referring to Fig. 5. During inference, we calculate the average recognition accuracy over accumulated samples. Due to the small number of samples at the initial stage, the average accuracy may fluctuate drastically, so we discard the statistics for the first 5 samples. Overall, methods employing the hierarchical cache consistently outperform other variants by a clear margin across various large 3D models and datasets.

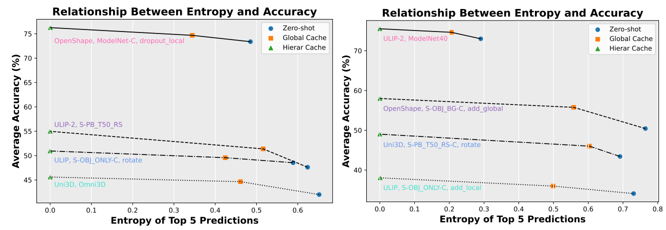


(a) ULIP. (b) ULIP-2. (c) OpenShape. (d) Uni3D.

Figure 5. **The average recognition accuracy of accumulated samples during online inference**. The curve changes significantly in the initial stage due to the small number of samples. Models with our global and hierarchical cache receives perceptible performance gains.

Entropy and Accuracy. Fig. 6 illustrates the relationship between entropy and accuracy for different models on various datasets. During online inference, we first compute the entropy of top 5 predictions for each test sample and then average the entropy across all accumulated samples. Similarly, the corresponding average accuracy can be calculated. The results reveal that Point-Cache effectively reduces the uncertainty of the zero-shot predictions made by large 3D models, enhances prediction confidence, and consequently lifts recognition accuracy. Additionally, we observe that the hierarchical cache model is more effective in reducing entropy compared to the global cache model, further underscoring the importance of incorporating the local cache.

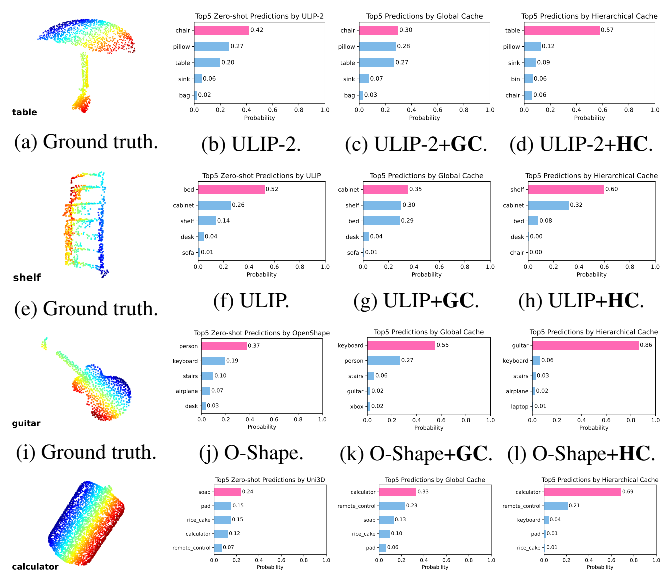
Qualitative Analysis. Fig. 7 presents the step-by-step adaptation process of Point-Cache. The first three rows demonstrate cases where the global cache model attempts to adjust the zero-shot predictions of various large 3D models but fails. Instead, the hierarchical cache model successfully refines the class logits based on the global cache. In the last row, our global cache model corrects the zero-shot prediction in the first step, and the hierarchical cache model subsequently retains the top class while increasing its confidence. The examples showcased here are not cherry-picked.



(a) Group One. (b) Group Two.

Figure 6. **Relationship between entropy and accuracy.** Models with our global and hierarchical cache consistently exhibit lower entropy and higher accuracy across various benchmarks.

Additional quantitative and qualitative results can be found in the Supplementary.



(a) Ground truth. (b) ULIP-2. (c) ULIP-2+**GC**. (d) ULIP-2+**HC**.

(e) Ground truth. (f) ULIP. (g) ULIP+**GC**. (h) ULIP+**HC**.

(i) Ground truth. (j) O-Shape. (k) O-Shape+**GC**. (l) O-Shape+**HC**.

(m) Ground truth. (n) Uni3D. (o) Uni3D+**GC**. (p) Uni3D+**HC**.

Figure 7. **The zero-shot predictions before and after adaptation by Point-Cache. GC**: global cache. **HC**: hierarchical cache.

5. Conclusion

This paper explores test-time point cloud recognition in a practical and challenging setting, where only test data and the pre-trained models are available. We develop Point-Cache, a powerful 3D knowledge base that enhances robustness and generalization at test time. It records the fingerprints of online test samples through a global and a local cache, creating a more accurate profile of 3D data. Designed as a plug-and-play module, Point-Cache can be seamlessly integrated into various large 3D models without compromising efficiency, enabling open-vocabulary point cloud recognition for both known and unseen classes, effectively addressing limitations of prior test-time methods.

Acknowledgement

We thank all anonymous reviewers and area chairs for their time and valuable feedback. This work was partially supported by China Scholarship Council (CSC) under the Grant No. 202306360147 and Australian Research Council funding under the Grant No. DE250100030. Dr. Qiuhong Ke is partially supported by the Australian Research Council funding Discovery Early Career Researcher Award (DE250100030). Dr. Deying Li was partially supported by the National Natural Science Foundation of China under the Grant No. 12071478. Dr. Yongcai Wang was partially supported by the National Natural Science Foundation of China under the Grant No. 61972404, the Public Computing Cloud at Renmin University of China, and the Blockchain Lab, School of Information, Renmin University of China.

References

[1] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 3

[2] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. PointGPT: Auto-regressively generative pre-training from point clouds. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 3, 4

[4] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9042–9051, 2021.

[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. 2, 3, 4

[6] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems*, pages 35799–35813. Curran Associates, Inc., 2023. 1

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2023. 1, 2, 3, 5, 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 4

[9] Lunhao Duan, Shanshan Zhao, Nan Xue, Mingming Gong, Gui-Song Xia, and Dacheng Tao. Condaformer: Disassembled transformer with local structure enhancement for 3d point cloud understanding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[10] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6367–6376, 2022. 1, 2, 3

[11] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and ZhaoXiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2918–2927, 2021. 1

[12] 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 (ICCV)*, pages 2704–2714, 2023. 2

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 3, 4

[14] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, pages 29374–29385. Curran Associates, Inc., 2022. 2

[15] Edouard Grave, Moustapha M Cisse, and Armand Joulin. Unbounded cache model for online language modeling with open vocabulary. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3

[16] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021. 1

[17] 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 (ICCV)*, pages 16494–16504, 2023. 2

[18] Ahmed Hatem, Yiming Qian, and Yang Wang. Test-time adaptation for point cloud upsampling using meta-learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1284–1291, 2023. 2

[19] Chao Huang, Zhangjie Cao, Yunbo Wang, Jianmin Wang, and Mingsheng Long. Metasets: Meta-learning on point sets for generalizable representations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8859–8868, 2021. 1, 2, 3, 4

[20] Zhe Huang, Shuo Wang, Yongcai Wang, Wanting Li, Deying Li, and Lei Wang. Roco: Robust cooperative perception by iterative object matching and pose adjustment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7833–7842, 2024. 1

[21] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*, pages 2427–2440. Curran Associates, Inc., 2021. 2, 3, 4

[22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[23] 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 (CVPR)*, pages 14162–14171, 2024. 2, 3, 4

[24] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. 3

[25] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 1

[27] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 2

[28] 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, 2022. 2

[29] 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. In *Advances in Neural Information Processing Systems*, pages 44860–44879. Curran Associates, Inc., 2023. 2, 3, 5, 6, 1, 4

[30] 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? In *Advances in Neural Information Processing Systems*, pages 21808–21820. Curran Associates, Inc., 2021. 2

[31] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary point-cloud object detection without 3d annotation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1190–1199, 2023. 1

[32] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022. 1, 3

[33] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. 3

[34] M. Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[35] M. Jehanzeb Mirza, Inkyu Shin, Wei Lin, Andreas Schriebl, Kunyang Sun, Jaesung Choe, Mateusz Kozinski, Horst Possegger, In So Kweon, Kuk-Jin Yoon, and Horst Bischof. Mate: Masked autoencoders are online 3d test-time learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16709–16718, 2023. 2, 3

[36] Emin Orhan. A simple cache model for image recognition. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 3, 4

[37] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J. Kim. Self-positioning point-based transformer for point cloud understanding. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21814–21823, 2023. 1

[38] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc Le. Combined scaling for open-vocabulary image classification. 2021. 2

[39] 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 (CVPR)*, 2017. 1, 3

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1, 3

[41] 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. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1, 2, 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[43] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In *In-*

ternational Conference on Machine Learning (ICML), 2022. 1, 2, 5, 6

[44] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions, 2022. 1

[45] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7213–7222, 2022. 1, 2, 3

[46] 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 (ECCV)*, 2024. 2, 3

[47] 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. In *Advances in Neural Information Processing Systems*, pages 14274–14289. Curran Associates, Inc., 2022. 2

[48] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3, 4

[49] Hongyu Sun, Yongcai Wang, Xudong Cai, Xuewei Bai, and Deying Li. Vipformer: Efficient vision-and-pointcloud transformer for unsupervised pointcloud understanding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3

[50] Hongyu Sun, Qiuhong Ke, Yongcai Wang, Wang Chen, Kang Yang, Deying Li, and Jianfei Cai. Point-prc: A prompt learning based regulation framework for generalizable point cloud analysis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[51] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions, 2022. 1

[52] 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 *Proceedings of the 37th International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020. 2

[53] Yiwen Tang, Ray Zhang, Zoey Guo, Xianzheng Ma, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5171–5179, 2024. 3, 4, 6

[54] Yun-Yun Tsai, Fu-Chen Chen, Albert Y. C. Chen, Junfeng Yang, Che-Chun Su, Min Sun, and Cheng-Hao Kuo. Gda: Generalized diffusion for robust test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23242–23251, 2024. 2

[55] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2725–2736, 2023. 3, 4

[56] 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 (ICCV)*, 2019. 1, 3, 5, 6

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 4

[58] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 3, 4

[59] Anran Wang, Jiwen Lu, Jianfei Cai, Tat-Jen Cham, and Gang Wang. Large-margin multi-modal deep learning for rgb-d object recognition. *IEEE Transactions on Multimedia*, 17 (11):1887–1898, 2015. 1

[60] 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

[61] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13520–13529, 2023. 1

[62] Haiyang Wang, Hao Tang, Shaoshuai Shi, Aoxue Li, Zhenguo Li, Bernt Schiele, and Liwei Wang. Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In *ICCV*, 2023.

[63] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8437–8445, 2019. 1

[64] 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, 3

[65] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11720, 2020. 2

[66] 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 (CVPR)*, pages 23231–23241, 2024. 2, 3, 4, 6

[67] Xin Wei, Xiang Gu, and Jian Sun. Learning generalizable part-based feature representation for 3d point clouds. In *Advances in Neural Information Processing Systems*, pages 29305–29318. Curran Associates, Inc., 2022. 1, 2, 3, 4

[68] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7): 5092–5113, 2024. 1

[69] 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)*, pages 803–814, 2023. 1, 5, 6

[70] 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)*, 2015. 1, 3, 5, 6

[71] 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. 3

[72] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5, 6, 1, 4

[73] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4, 5, 6, 1

[74] 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*, 2021. 2

[75] Teresa Yeo, Oğuzhan Fatih Kar, Zahra Sodagar, and Amir Zamir. Rapid network adaptation:learning to adapt neural networks using test-time feedback. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4651–4664, 2023. 1

[76] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *Advances in Neural Information Processing Systems*, pages 32215–32234. Curran Associates, Inc., 2023. 3

[77] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, 2022. 4, 1, 3

[78] Maxime Zanella and Ismail Ben Ayed. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23783–23793, 2024. 2

[79] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18123–18133, 2022. 2

[80] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, pages 38629–38642. Curran Associates, Inc., 2022. 2

[81] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8552–8562, 2022. 3

[82] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXV*, pages 493–510, Berlin, Heidelberg, 2022. Springer-Verlag. 3, 4

[83] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15211–15222, 2023. 3, 4

[84] 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 (CVPR)*, pages 5344–5353, 2023. 3, 4, 6

[85] Weichen Zhang, Wen Li, and Dong Xu. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6765–6775, 2021. 2

[86] Zhimin Zhang, Xiang Gao, and Wei Hu. Invariantoodg: Learning invariant features of point clouds for out-of-distribution generalization, 2024. 2, 4

[87] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021. 1, 3

[88] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 5, 6, 7, 4

[89] 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 (ICCV)*, 2023. 3