DINO Eats CLIP: Adapting Beyond Knowns for Open-set 3D Object Retrieval

Supplementary Material

In the supplementary material, we first provide more implementation details. Next, we conduct more analysis experiments. Then, we provide more qualitative results. Lastly, we evaluate DEC on more challenging and realistic scenarios.

A. More Implementation Details

We set the chunk size to 3, efficiently reducing the number of views by a factor of 6 with each CBR block and pooling layer. Thus, by design, CAM is highly light-weight that comprises only maximum of two CBR blocks to process an initial set of 24 views. Empirically, a single block is optimal for smaller datasets (OS-ESB-core, OS-NTU-core), while two blocks yield stronger performance on larger ones (OS-MN40-core, OS-ABO-core). When only one block is used, we simply aggregate the view features into a compact representation using adaptive average pooling. We also implement a three-layer MLP as our plain adapter for our baseline, which follows an encoder-decoder architecture: the first layer projects the input dimension D down to a bottleneck dimension $D/4$, the second layer operates within this bottleneck, and the third layer projects the features back to the original dimension D ($D = 768$).

B. More Analyses

Impact of View Number. We analyze the impact of view number on OS-MN40-core. As shown in Figure 1, performance remains relatively stable when using only 2 to 4 views, then increases markedly between 4 and 8 views, and continues to improve slightly up to 24 views. This trend indicates that our method can effectively use a moderate number of views for decent performance. Yet we observe diminishing returns once most geometric view information has been observed.

Impact of fusion weight λ. Table 1 summarizes optimal λ across benchmarks and backbones. For OS-ESB-core and OS-NTU-core, the optimal λ remains consistently small (0.11) for both ViT-B/14 and ViT-L/14, suggesting that the pretrained DINO embeddings already encode sufficiently discriminative geometric information for these datasets. Consequently, only a minimal contribution from the adapted feature branch is required to attain peak performance, and excessive adaptation may perturb the pretrained feature structure. Conversely, OS-MN40-core and OS-ABO-core exhibit substantially higher optimal fusion weights, ranging from 0.2 to 0.4. These datasets involve greater intra-class variability and more heterogeneous shape distributions, thereby increasing the necessity for task-specific feature adjustment. In such settings, the adapter contributes more complementary information, and the model benefits from assigning a larger relative weight to the adapted features.

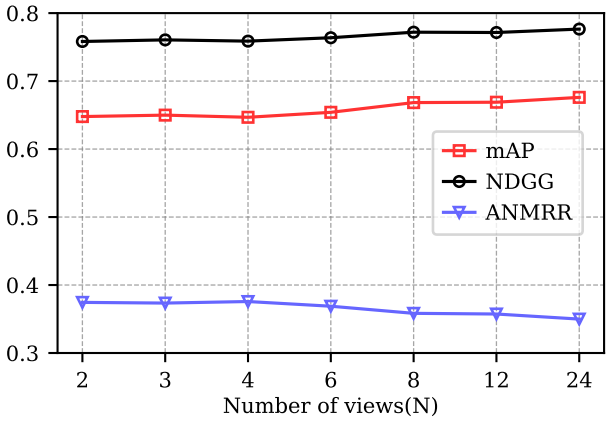

Figure 1. **Effect of view numbers (N) on OS-MN40-core.**

Dataset	Backbone	λ	mAP↑	NDCG↑	ANMRR↓
OS-ESB-core	ViT-B/14	$\lambda = 0.11$	61.82	24.55	42.74
	ViT-L/14	$\lambda = 0.11$	60.59	24.46	43.30
OS-NTU-core	ViT-B/14	$\lambda = 0.11$	61.56	27.26	41.62
	ViT-L/14	$\lambda = 0.11$	65.15	28.24	37.80
OS-MN40-core	ViT-B/14	$\lambda = 0.3$	67.62	77.67	34.99
	ViT-L/14	$\lambda = 0.4$	69.69	79.11	33.23
OS-ABO-core	ViT-B/14	$\lambda = 0.3$	65.04	59.34	37.32
	ViT-L/14	$\lambda = 0.2$	67.91	60.73	34.74

Table 1. **Optimal λ across different datasets and backbones.**

Overlapping vs. Non-overlapping Chunking. Table 2 studies the impact of stride size in CBR. The non-overlapping configuration (stride = 3) yields the highest performance across mAP, NDCG, and ANMRR. It indicates that non-overlapping chunking is not only sufficient but also works better: it successfully captures diverse local contexts while avoiding the computational redundancy and potential feature dilution associated with overlapping strides. Hence, DEC's non-overlapping chunking strategy achieves a much better balance between effectiveness and efficiency.

Combining DINO and CLIP Features To study whether directly combining CLIP and DINO features derives more effective representations, we train an MLP on added DINOv2-ViT/B14 (mapped to the same dim. as CLIP) and CLIP ViT/B16 features with the standard cross-entropy loss. Yet it is worse than the DINOv2 baseline (Table 3) on 3/4 datasets in mAP. It suggests that naive fusion strug-

stride size	mAP↑	NDCG↑	ANMRR↓
1	64.64	76.31	37.34
2	65.52	76.62	36.74
3	**67.62**	**77.67**	**34.95**

Table 2. **Impact of different stride sizes on OS-MN40-core.**

gles to reconcile the two divergent feature spaces, whereas ours uses CLIP to provide semantic, generalized priors that regularize and guide DINO's training, resulting in a more effective synergy.

Methods	OS-ESB	OS-NTU	OS-MN40	OS-ABO
DINOv2	59.19	59.77	62.77	62.17
Base. (DINOv2+CLIP)	59.40	59.28	61.35	60.69
Ours	**61.82**	**61.56**	**67.62**	**65.04**

Table 3. *Base.* fuses DINO and CLIP, while we use DINO only.

C. More Qualitative Results

Figure 2 presents more retrieval examples on OS-MN40-Core. As shown, DEC faithfully retrieves relevant 3D objects for 3D query objects of common classes such as bathtub, door, and bed. However, certain challenge cases (row 4-6) exist for classes such as stool, bowl, and bottle, leading to failures. For instance, in row 5, a bowl query is incorrectly matched with instances from the vase, despite the subtle differences in their geometric profiles (*e.g.*, aperture width and base structure). Notably, the top-ranked false positives typically share strong overall geometric similarities with the 3D query objects, which indicates that our model successfully learns high-level shape features but can be confounded by nuanced inter-class variations. One potential direction for improvement is to enhance fine-grained class discrimination by modeling these fine-grained geometric variations.

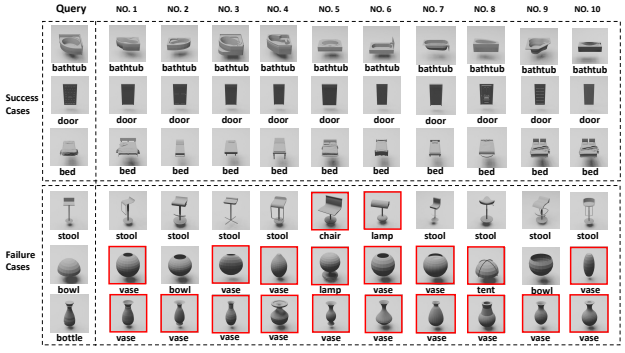

Figure 2. **More retrieval examples of our method on OS-MN40-core.** Incorrect matches are in red boxes.

D. Retrieval on Seen and Unseen Categories

We further evaluate the performance of our method on both seen and unseen categories. Following HGM^{2}R [2], we split the ModelNet40 dataset into two subsets: one for seen categories and one for unseen categories. Each subset contains 20 categories, with 80% of the data used for training (training on seen categories and unseen categories separately) and 20% reserved for retrieval evaluation (evaluation on seen and unseen categories). The models are trained on the seen categories and evaluated separately on both the seen and unseen category sets. As shown in Table 4, for seen categories, our method demonstrates competitive performance on seen categories, relying solely on multi-view images. It achieves a mAP of 94.09 and Recall@100 of 82.63, surpassing other methods in Recall@100. While the mAP is slightly below HGM^{2}R (94.10), which requires multi-modalities and test data for training, our method only relies on visual input alone and does not involve test data for training, highlighting our advantages in 3DOR. For unseen categories, our method outperforms all methods, achieving 86.47 in mAP and 82.00 in Recall@100. These results confirm the model's strong generalization ability, crucial for real-world applications with unseen categories.

Method	On Seen Categories		On Unseen Categories	
	mAP↑	Recall@100↑	mAP↑	Recall@100↑
TCL [4]	93.50	82.14	73.92	71.76
MMJM [8]	91.99	80.78	73.07	71.38
SDML [6]	88.50	78.50	74.69	72.39
CMCL [7]	90.99	79.60	75.21	72.49
MMSAE [13]	88.72	78.61	76.03	72.94
MCWSA [14]	85.70	76.83	72.89	70.56
PROSER [15]	87.71	77.78	74.93	72.56
InfoNCE [9]	93.65	82.19	73.92	71.64
HGM^{2}R [2]	**94.10**	82.47	82.23	78.21
DAC [12]	91.12	80.46	86.27	81.76
Our Method	94.09	**82.63**	**86.47**	**82.00**

Table 4. **Separate retrieval results on seen and unseen classes.**

E. Results on More Realistic Datasets

OS-Objaverse-core. To validate DEC on real-world scenarios, we curate OS-Objaverse-core, a new large-scale open-set 3DOR dataset based on Objaverse-LVIS of Objaverse [1]. Objaverse-LVIS has 1,142 categories in total. We filter out categories with fewer than 20 objects, resulting in 607 categories. We split them into training, query, and target subsets, which have 10,092, 1,458, and 15,274 objects, respectively. The training set comprises 121 seen classes for training, while the query and target contain 486 unseen classes for evaluation. Table 5 compares the performance of DEC with state-of-the-art methods on

Method	mAP↑	NDCG↑	ANMRR↓
CLIP-AdaM [5]	16.97	13.71	80.39
DAC [12]	21.89	16.42	75.86
DINOv3 [11]	32.23	21.34	66.31
Ours	**35.53**	**23.02**	**63.71**

Table 5. **Performance comparison on OS-Objaverse-core.**

Method	mAP↑	NDCG↑	ANMRR↓
DAC [12]	29.40	43.06	68.40
DINOv3 [11]	31.02	45.30	66.54
Ours	**35.70**	**47.29**	**62.96**

Table 6. **Performance comparison on 3DFuture.**

OS-Objaverse-core. We also include a zero-shot baseline based on DINOv3 [11] which mean-pools the view features of ViT-L. As shown, DEC significantly surpasses this zero-shot baseline, increasing the mAP by +3.30%, demonstrating the effectiveness of DEC for adaptation in this challenging large-scale dataset with highly diverse categories. When compared with recent competitors such as CLIP-AdaM [5] and DAC [12], our method consistently achieves remarkable performance gains across all metrics. Specifically, we achieve a notable gain of +13.64% in mAP, +6.60% in NDCG, and +15.15% in ANMRR over DAC.

3DFuture. To further validate DEC's effectiveness on real-world data, we evaluate it on 3DFuture [3], a large-scale 3D object retrieval benchmark comprising 49 object categories. We split them into training, query, and target subsets, which have 3373, 205, and 11234 objects, respectively. Ten classes are designated as seen and used for training, and the remaining 39 classes are reserved for evaluation. As shown in Table 6, our method significantly outperforms the strong baselines, achieving substantial improvements across all evaluation metrics. Compared to DAC [12], we observe a remarkable gain of +6.30% in mAP, +4.23% in NDCG, and a -5.44% reduction in ANMRR, confirming the efficacy of our design in handling the high category diversity and scale of the 3DFuture dataset. Compared with recent DINOv3 [11], we yield gains of +4.68% in mAP, +1.99% in NDCG, and -3.58% in ANMRR. These results demonstrate that our method offers a scalable and effective solution for real-world 3D object retrieval scenarios in open-set conditions.

ScanObjectNN. Our DEC is also *compatible with point clouds* by projecting it online into depth maps. For the experiment, 10 depth maps are projected for each point cloud online following [16]. We experiment on noisy real-world ScanObjectNN. We achieve 30.34% mAP, outperforming recent CLIP-Adam [5] greatly. The zero-shot baseline attains only 23.77% mAP.

Method	mAP
Base.	23.77
CLIP-Adam [5]	24.18
Ours	**30.34**

Table 7. **Performance comparison on ScanObjectNN.**

Method	Top-1 Accuracy
PointCLIP V2 [16]	89.55
PointNet++ [10]	90.7
Ours	**91.49**

Table 8. **Few-shot learning results on ModelNet40.** We report the 16 shot classification accuracy (%).

F. Extending to Few-shot Classification

3D classification. Beyond open-set 3D object retrieval, we further evaluate DEC on diverse downstream tasks to assess its general-purpose representation capability. We conduct a 16-shot 3D classification on ModelNet40. As shown in Table 8, we surpass PointCLIP V2 [16] greatly on Top-1 accuracy, even surpassing the fully supervised PointNet++ [10] without *normal*.

References

[1] 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 *CVPR*, pages 13142–13153, 2023. 2

[2] Yifan Feng, Shuyi Ji, Yu-Shen Liu, Shaoyi Du, Qionghai Dai, and Yue Gao. Hypergraph-based multi-modal representation for open-set 3d object retrieval. *IEEE TPAMI*, 2023. 2

[3] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *IJCV*, 129(12):3313–3337, 2021. 3

[4] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *CVPR*, pages 1945–1954, 2018. 2

[5] Xinwei He, Liang Ma, Yuxuan Cheng, Zhichuan Wang, Yulong Wang, Yang Zhou, and Xiang Bai. Clip-adam: Adapting multi-view clip for open-set 3d object retrieval. In *SIGIR*, pages 1022–1032, 2025. 3

[6] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. Scalable deep multimodal learning for cross-modal retrieval. In *SIGIR*, pages 635–644, 2019. 2

[7] Longlong Jing, Elahe Vahdani, Jiaxing Tan, and Yingli Tian. Cross-modal center loss for 3d cross-modal retrieval. In *CVPR*, pages 3142–3151, 2021. 2

[8] Weizhi Nie, Qi Liang, An-An Liu, Zhendong Mao, and Yangyang Li. Mmjn: Multi-modal joint networks for 3d shape recognition. In *ACM MM*, pages 908–916, 2019. 2

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 3

[11] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 3

[12] Zhichuan Wang, Yang Zhou, Zhe Liu, Rui Yu, Song Bai, Yulong Wang, Xinwei He, and Xiang Bai. Describe, adapt and combine: Empowering clip encoders for open-set 3d object retrieval. In *ICCV*, pages 21026–21036, 2025. 2, 3

[13] Yiling Wu, Shuhui Wang, and Qingming Huang. Multi-modal semantic autoencoder for cross-modal retrieval. *Neurocomputing*, pages 165–175, 2019. 2

[14] Jiahao Zheng, Sen Zhang, Zilu Wang, Xiaoping Wang, and Zhigang Zeng. Multi-channel weight-sharing autoencoder based on cascade multi-head attention for multimodal emotion recognition. *IEEE TMM*, 2022. 2

[15] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021. 2

[16] 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 *ICCV*, pages 2639–2650, 2023. 3