A. Implement Details

A.1. Triplet Proxy Collection

Generally, the triplet proxy collection concludes given text proxies $\mathbf{X}^T$, extracting image proposals as 2D proxies $\mathbf{X}^I$ and constructing 3D proxies $\mathbf{X}^P$ with geometry relations between images and point clouds. According to the application scenarios, we detail the construction process in indoor and outdoor scenes separately.

Indoor scenes. The indoor scenes S usually adopt RGB-D sensors to collect images with corresponding depth maps as I_s^{uvd}, where $s \in |S|$. Specifically, we first provide given text proxy $\mathbf{X}^T$ as the input of pretrained DetCLIP [10] to extract 2D image proposals $I_s^{i,uvd}, i \in |X_s^I|$, where $|X_s^I|$ denotes the amount of 2D proxies in scene s. Then we segment the foreground images with unsupervised GrabCut [8] algorithm as $I_s^{i,uvd'}$, thus the point cloud instances can be reconstructed by the RGB-D pixels with camera calibration G_{IN}, which can be formulated as:

$$\lceil x, y, z \rceil = \mathrm{G}_{\mathrm{IN}}^{-1} \times \lceil u, v, d \rceil,$$

where $\mathrm{G}_{\mathrm{IN}} = \mathrm{I} \times \mathrm{R}_c$ denotes the combination of the intrinsics matrix I and the extrinsics matrix R_c of RGB-D camera.

Outdoor scenes. Considering a much wider perception range, outdoor scenes usually have LiDAR and camera sensors to capture point clouds P_s^{xyz} and camera images I_s^{uv}. Thus point clouds can be projected into camera pixels with sensor transformation matrix $\mathrm{G}_{\mathrm{OUT}}$ as:

$$\lceil u, v, d \rceil = \mathrm{G}_{\mathrm{OUT}} \times \lceil x, y, z \rceil,$$

where $\mathrm{G}_{\mathrm{OUT}} = \mathrm{I} \times \mathrm{R}_c^{-1} \times \mathrm{R}_l$ are the combination of camera intrinsics matrix I, camera extrinsics matrix R_c and the LiDAR extrinsics matrix R_l. Concretely, we first conduct a similar procedure to indoor scenes that produces 2D image proposals as $I_s^{i,uvd}, i \in |X_s^I|$ for 2D proxies X_s^I. Then we extract the 3D frustum $P_s^{i,xyz'}$ by extruding the 2D image proposal into 3D space and conduct DBSCAN clustering within the frustum. Eventually, we obtain the 3D proxy instance by filtering the point cloud cluster $P_s^{i,xyz}$. The whole process of triplet proxy collection is illustrated in Figure A1.

A.2. Contrastive Pretraining

Our main paper applies the popular point cloud classifier PointNet++ [7] as our point cloud encoder. Concretely, we use two set abstraction layers that aggregate multi-scale information and then encode the feature vectors for point cloud instances by three fully convolutional layers. We remove the convolutional head of PointNet++ since the point cloud features of CLIP^2 can be directly referenced to the language embedding for downstream tasks.

We conduct all experiments using Pytorch [4], 8 Tesla V100 cards on a single server. We randomly sample 2048 points on each object both for training and testing. At training time, AdamW optimizer [3] is performed on 8 GPUs with 200 batch sizes on each. The learning rate is set to 0.006, 3e-2 as weight decay, and 0.9 as momentum. And we adopt the cosine decay with 1000 iteration warm-up. For both indoor and outdoor datasets, we train 100 epochs.

A.3. ScanNet Dataset

Considering the ambiguous synonyms in the raw classes, like “handrail”, “stair rail” and “banister”, we involve a data preprocessing step aimed at merging raw classes using WordNet[1] synonyms. Specifically, the official file *scannetv2-labels.combined.tsv* provided by ScanNet is utilized to identify synonyms for the various classes. This process resulted in the merging of 290 classes, which included 86 classes that lacked synonyms. To further refine the merged classes, the 86 classes were subjected to an additional merging step. This step involved merging them into existing synonyms based on the path similarity in WordNet. The decision to merge was guided by a predefined threshold, such that only classes with a path similarity score above the threshold were merged.

The outcome of the above-described process was a final set of 249 classes deemed suitable for open-vocabulary evaluation. These classes represented a more refined and comprehensive set of merged classes, facilitating more reasonable and consistent evaluations.

B. Additional Results

B.1. Different Point Cloud Encoder

We compare three alternatives of point cloud encoder, including PointNet [6], DGCNN [5] and PointNet++ [7], and report the class average Top1 accuracy of zero-shot recognition in Table A1. Specifically, PointNet encodes point cloud features with point-wise MLP and max-pooling, DGCNN applies EdgeConv to extract edge features and then ensemble the point cloud features, while PointNet++ adopts additional hierarchical feature learning based on PointNet to leverage neighborhoods at multiple scales. As illustrated in Table A1, PointNet++ outperforms the other two encoders on all benchmarks, showing its superiority in extracting effective point cloud features. We believe more advanced point cloud encoder architectures can further enhance our learned 3D representation of CLIP^2.

B.2. Impact Analysis of Proxy

Proxy range. As the prior knowledge of open-world vocabularies, we adopt the caption list in 2D open-world dataset LVIS [2] to set text proxies without human annotations. In Table A2, we transfer the text proxies to the

[1] https://wordnet.princeton.edu/

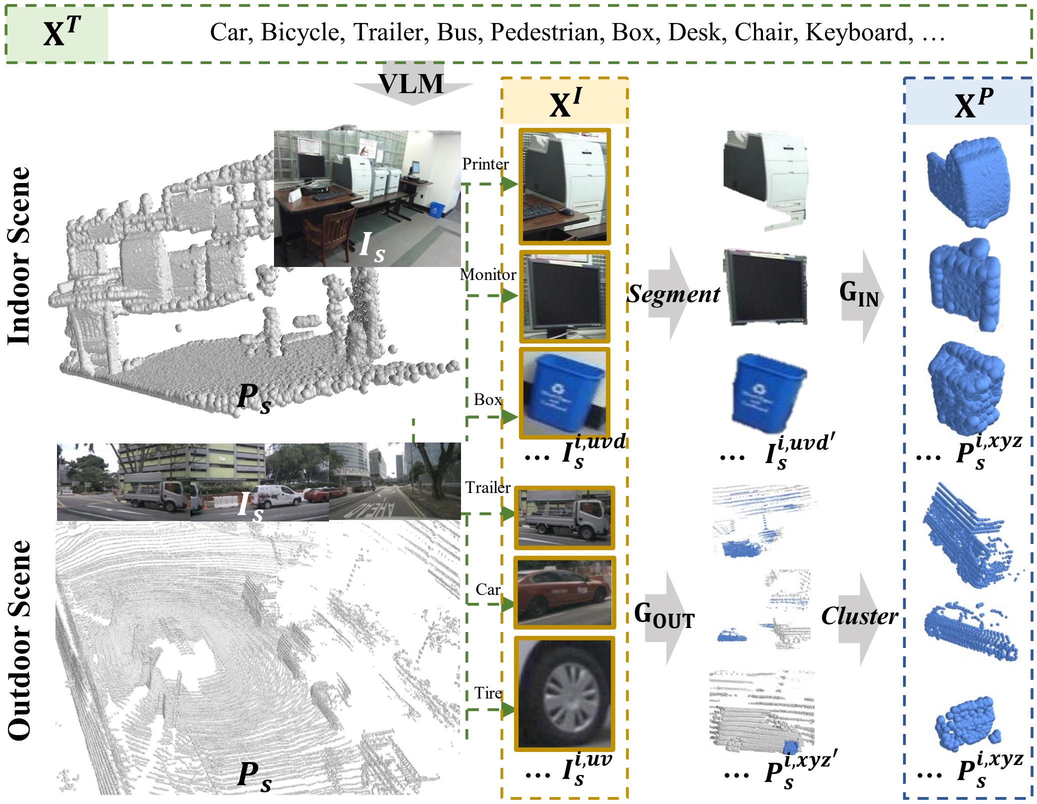

Figure A1. Illustration of triplet proxy generation process.

Encoder	ScanNet	SUN RGB-D	ScanObjectNN
PointNet	22.6	45.3	27.0
DGCNN	26.0	52.7	34.0
PointNet++	38.5	61.3	39.4

Table A1. Comparison of point cloud encoders.

Range	ScanNet		SUN RGB-D	ScanObjectNN
	Main Top1	384 cls. Top5	Main Top1	Top1
SUN=37	36.6	17.1	63.6	34.4
LVIS=1203	38.5	22.0	61.3	39.4
SCAN=384	39.5	23.0	61.6	44.6

Table A2. Comparison with different proxy range.

groundtruth list of segmentation annotations of the SUN RGB-D [9], which presents the congruous vocabulary range of dataset annotations with less noise but a narrow vision of open-vocabulary. Results in Table A2 demonstrate that the groundtruth proxy range can improve the intra-dataset recognition performance on SUN RGB-D by 2.3% average Top1 Acc. However, the inter-dataset performance drops 1.9% on ScanNet and 5.0% on ScanObjectNN, and yields a 4.9% drop of average Top5 Acc on the extended vocabularies of ScanNet. The overall results validate that the open-world vocabulary of text proxies benefits transferable 3D representation learning.

Proxy quantity. In Figure A2, we present the performance curves that demonstrate the consistency between the zero-shot recognition performance and increasing proxy data. Our analysis suggests that increasing the amount of training data in future work has the potential to further improve the upper bound of performance. These findings highlight the importance of scaling proxy data in a cost-effective manner.

B.3. Comparison with Supervised Baselines

We conduct supervised training with popular 3D encoder PointNet [6] and PointNet++ [7] using annotations from SUN RGB-D training set. We consider two different settings as supervised baselines: 1) Traditional logit classification head **L_Head**, which is fixed to the predefined training classes and fails to identify novel classes. 2) Text classification head indicated as **T_Head**. Specifically, we replace the logit classification head with CLIP text embeddings. And the maximum cosine similarities between the 3D feature and text embeddings are the final results. According to the text classification head, we can compare the generalization of flexible categories with supervised training.

The results in Table A3 show that CLIP^2 is comparable

Method	Backbone	SUN	ScanNet
supervised L_Head		48.5	-
supervised T_Head	PointNet [6]	44.2	14.4
CLIP^2		45.3	22.6
supervised L_Head		63.4	-
supervised T_Head	PointNet++	60.3	25.5
PointCLIP [11]	[7]	11.5	6.7
CLIP^2		61.3	38.5

Table A3. Comparisons with supervised baselines. We train supervised baselines on SUN RGB-D (SUN) dataset and evaluate recognition results on SUN and zero-shot performance on ScanNet. **L_Head** and **T_Head** indicate logit classification head and text classification head respectively.

to supervised baselines on SUN RGB-D and outperforms them on ScanNet, illustrating the effectiveness of our unsupervised approach and its superiority in open-vocabulary understanding.

C. More Qualitative Results

Saliency map between text prompt and point cloud. To validate our CLIP^2, we show a saliency map between the given text prompt and the point cloud within one scene in Figure A3. Specifically, we calculate the feature distances between the class texts and the point cloud scene and plot the saliency map, with lighter highlights representing smaller feature distances. The text feature has greater similarity to the point feature of the corresponding class, indicating the feature alignment between text and point cloud.

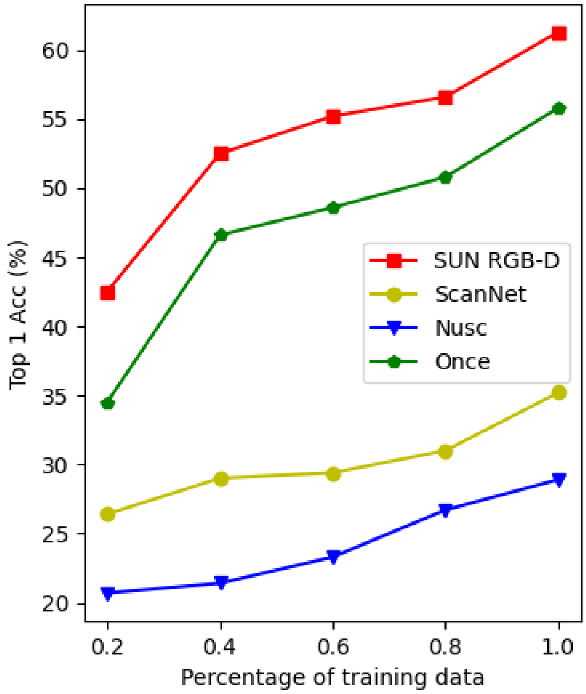

Figure A2. Performance curves for training proxy quantity.

Visualization of zero-shot localization. We show more qualitative results in Figure A4 for indoor scenes SUN RGB-D [9] and Figure A5 for outdoor scenes nuScenes [1]. The visualization results illustrate the zero-shot localization and recognition abilities of CLIP^2. Specifically, the proposed CLIP^2 enables the open-world vocabularies beyond groundtruth annotations without extra human supervision, such as 'Tire' and 'Debris' in Figure A4.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 3

[2] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 1

[5] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 2018. 1

[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1, 2, 3

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 1, 2, 3

[8] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *TOG*, 2004. 1

[9] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 2, 3

[10] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. In *NeurIPS*, 2022. 1

[11] 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 *CVPR*, 2022. 3

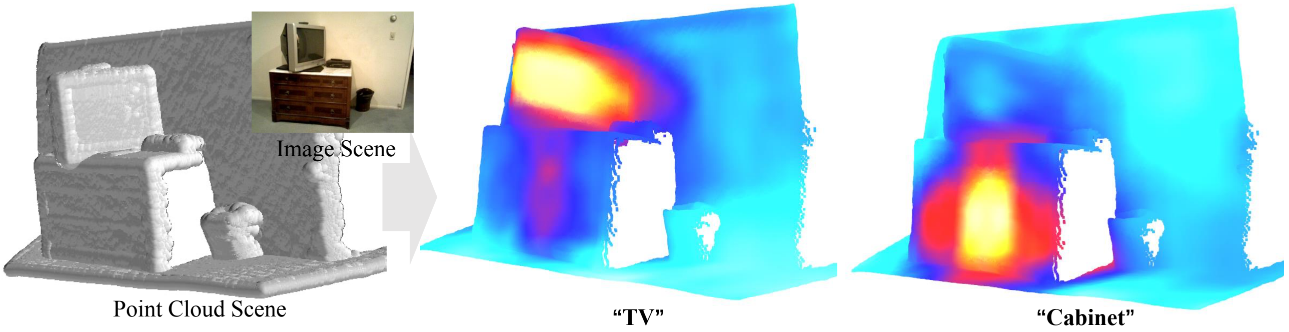

Figure A3. Saliency maps between texts and point cloud scenes.

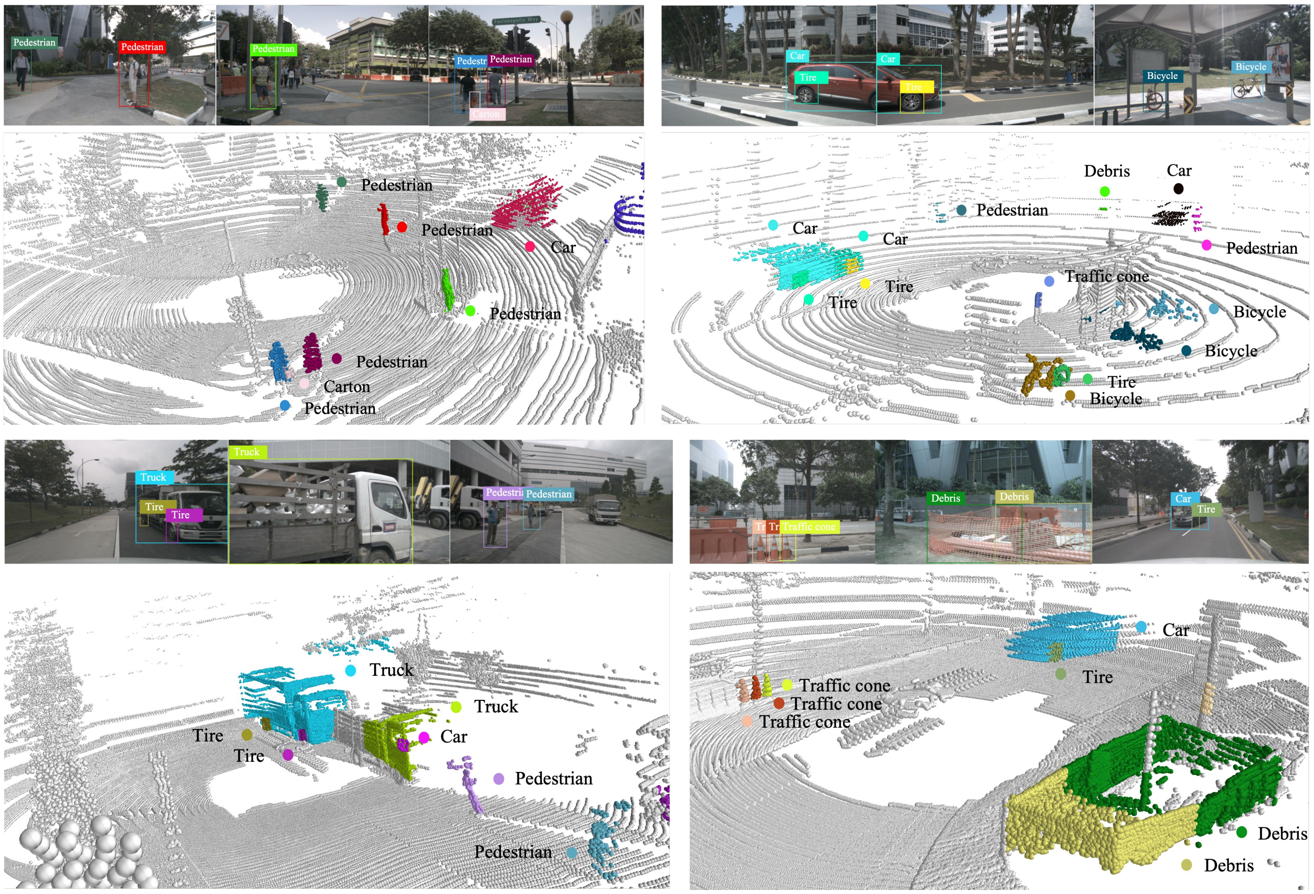

Figure A4. More Visualizations of the zero-shot localization and recognition on the nuScenes dataset. The proposed CLIP^2 enables the open-world vocabularies beyond groundtruth annotations without extra human supervision, such as ’Tire’ and ’Debris’. Best viewed in colors.

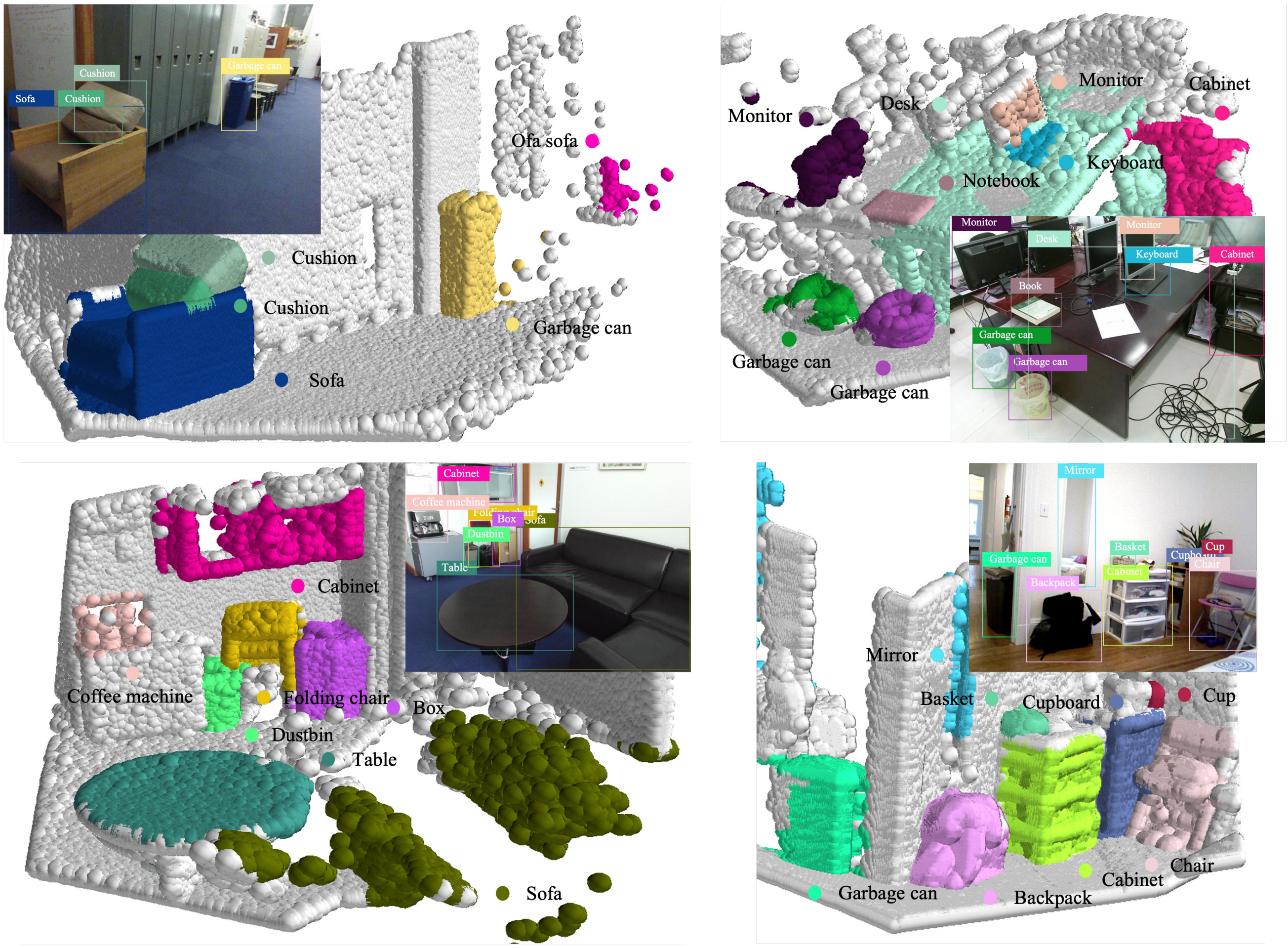

Figure A5. More Visualizations of the zero-shot localization and recognition on SunRGB-D dataset. The proposed CLIP^2 shows open-world recognition ability in realistic scenarios. Best viewed in colors.