Supplemental Material: Low-shot Object Learning with Mutual Exclusivity Bias

Anh Thai[1] **Ahmad Humayun**[2] **Stefan Stojanov**[1] **Zixuan Huang**[1]
Bikram Boote[1] **James M. Rehg**[1,3]
[1]Georgia Institute of Technology, [2]Google Deepmind,
[3]University of Illinois, Urbana-Champaign

1 Data

1.1 Datasets

In our work, we performed experiments and analysis using three datasets: Toys4K [14], ShapeNetCore.v2 [4], ABC [8], and CO3D [12]. In the following section, we provide comprehensive details about each of these datasets.

Toys4K [14]. This dataset consists of 4,179 object instances in 105 categories. We use the base and low-shot splits provided by Stojanov et al. [14]. In particular, the base classes consist of 40 categories while the low-shot classes have 55 categories. Objects in this dataset were collected under Creative Commons and royalty-free licenses. (Please refer to Table 1 for base/low-shot split compositions).

ShapeNetCore.v2 [4]. This dataset consists of 52K objects in 55 categories. We partition these categories into 25 base and 30 low-shot classes (see Table. 1). The terms of use for ShapeNet are specified on their website, which can be accessed at https://shapenet.org/terms.

ABC [8]. For pretraining our representation learning models, we used a subset of 100K object instances from ABC, which contains a total of 750K instances. Note that this dataset lacks categorical structures. The dataset is distributed under the MIT license. More licensing information is available at https://deep-geometry.github.io/abc-dataset/#license.

CO3D [12]. We chose the 13 classes out of 51 classes that overlap with Toys4K for low-shot validation, detailed in Table 1. The terms of use for CO3D are specified at https://ai.facebook.com/datasets/co3d-downloads/.

1.2 Data Generation

Software. We used Blender 2.93 [1] with ray-tracing renderer Cycles for data generation and rendering.

Assets. Objects are placed on top of a plane that simulates the ground/floor with PBR materials and image-based lighting from HDRI environment maps are used to illuminate scenes. We collected these assets from PolyHaven [2]. The list of assets used is shown in Table 2.

Scene Generation. Given any 3D categorical dataset, we first partition these object categories into disjoint sets: base classes and low-shot classes. For each object in the dataset, we preprocess it by simulating a rigid body drop using Blender [1]. This simulation process is repeated 16 times, allowing us to collect metadata and initial rotational poses for each object. These collected data are used in the subsequent stages of scene generation.

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

Figure 1: Rendered scenes for LSME on Toys4k [14]

To generate each scene, we first choose a subset of objects from the dataset. Their initial rotational poses are determined by randomly choosing from the preprocessed poses. Objects are then scaled and placed into the scene at random locations. We ensure that collisions do not occur by maintaining a minimum margin of $\Delta > 0$ between each pair of objects. We randomize the scene background by randomly choosing a pair of PBR material and HDRI environment map from the assets.

Data Rendering. To render each view of the scenes, we first determine the camera position. The camera's position in the scene is specified by three parameters: $\theta \in [0, 2\pi]$, $r \in [r_{min}, r_{max}] > 0$, and $z \in [z_{min}, z_{max}] > 0$ where θ is the rotational angle, r is the distance from the origin in the XY-plane, and z denotes the world Z-coordinate of the camera. Note that $r_{min}, r_{max}, z_{min}, z_{max}$ are preset parameters. The world coordinate of the camera is computed by $(r\cos(\theta), r\sin(\theta), z)$. To determine the camera's orientation, it is set to point towards a location on the XY-plane that is within a small distance ϵ from the mean locations of the objects in the scene. This is done by rotating the camera in the world XY and YZ-planes. We then randomize illumination intensity, consistently for all the views of each scene.

Generated Data for LSME. We generated 1K scenes for each of support and query sets, with each scene consisting of 20 views. The data generated for LSME evaluation can be found at https://tinyurl.com/3a9r83z9. Additionally, the code for data generation is available on our

ShapeNetCore.v2		Toys4k		CO3D
Base	Low-shot	Base	Low-shot	Low-shot
chair	piano	candy	boat	TV
table	train	flower	lion	mouse
bathtub	file	dragon	whale	car
cabinet	pistol	apple	cupcake	toaster
lamp	motorcycle	guitar	train	microwave
car	printer	tree	pizza	donut
bus	mug	glass	marker	orange
cellular	rocket	cup	cookie	sandwich
guitar	skateboard	pig	sandwich	bicycle
bench	bed	cat	octopus	banana
bottle	ashcan	chair	monkey	bowl
laptop	washer	ice-cream	fries	motorcycle
jar	bowl	hat	violin	pizza
loudspeaker	bag	deer mouse	mushroom	
bookshelf	mailbox	penguin	closet	
faucet	pillow	ball	tractor	
vessel	earphone	fox	submarine	
clock	camera	dog	butterfly	
airplane	basket	knife	pear	
pot	remote	laptop	bicycle	
rifle	stove	pen	dolphin	
display	microwave	mug	bunny	
knife	microphone	plate	coin	
telephone	cap	chess piece	radio	
sofa	dishwasher	cake	grapes	
	keyboard	frog	banana	
	tower	ladder	cow	
	helmet	keyboard	donut	
	birdhouse	sofa	stove	
	can	trashcan	sink	
		dinosaur	orange	
		bottle	saw	
		elephant	chicken	
		pencil	hamburger	
		key	piano	
		monitor	light bulb	
		hammer	spade	
		screwdriver	crab	
		robot	sheep	
		bread	toaster	
			lizard	
			motorcycle	
			mouse	
			pc mouse	
			bus	
			helicopter	
			microwave	
			cell battery	
			drum	
			panda	
			TV	
			car	
			helmet	
			fridge	
			bowl	

Table 1: Split composition of ShapeNetCovre.v2, Toys4K and CO3D

GitHub repository at https://github.com/rehg-lab/LSME. Detailed parameters for scene generation can be found in Table 3.

1.3 Data Augmentation for Contrastive Training

To augment the data, we applied various transformations, including random horizontal flips and brightness and color jittering. Following [13], we employed random object masking, where the object instance mask was used to eliminate the background. Additionally, we applied rotations and translations to the foreground object and incorporated background randomization techniques.

PBR	HDRI
Carpet001	Aft Lounge
Carpet005	Anniversary Lounge
Carpet006	Balcony
Carpet007	Cabin
Carpet008	Cayley Interior
Carpet009	Children's Hospital
Carpet013	Colorful Studio
Carpet014	Entrance Hall
Fabric024	Fireplace
Fabric025	Hotel Room
Fabric028	Kiara Interior
Marble012	Lapa
Planks001	Lebombo
Planks009	Lythwood Lounge
Planks011	Lythwood Room
Planks013	Moonlit Golf
Planks014	Music Hall
Planks018	Photo Studio
Terrazzo001	Reading Room
Tiles001	Roof Garden
Tiles027	Small Empty House
Tiles071	Spiaggia Di Mondello
Tiles072	St Fagans Interior
WoodFloor005	Umhlanga Sunrise
WoodFloor028	Wooden Lounge

Table 2: List of assets used in data generation.

Parameter	Value
Camera r	$[1.0, 1.1)$
Camera z	$[0.3, 0.5)$
Camera jittering ϵ	0.01
Object scale	$[0.35, 0.45)$
Object location	$[-0.5, 0.5)$
Illumination intensity	$[0.6, 0.8)$
Object margin Δ	0.4

Table 3: Data rendering parameters.

1.4 More Data Visualizations

Figure 1 showcases additional examples of rendered scenes from the Toys4K dataset [14]. These examples highlight the diversity found in the background, illumination conditions, and object poses within the scenes.

In Figure 2, we demonstrate the instance mask prediction of the FreeSOLO [15] model finetuned on 1K scenes of ABC. The quality of the predicted masks is essential to solving LSME.

2 Additional Experiments

2.1 Evaluation Metric Details

We evaluate the performance of the baselines using the following metrics: 1) support assignment accuracy (SA) which quantifies the percentage of accurately identifying the novel instance within the scene, and 2) low-shot accuracy (LSA) for measuring low-shot performance, and 3) mean intersection-over-union (mIoU) for instance segmentation as detailed below. For each episode,

$$SA = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbb{1}\{\hat{o}_i = o_i\}$$

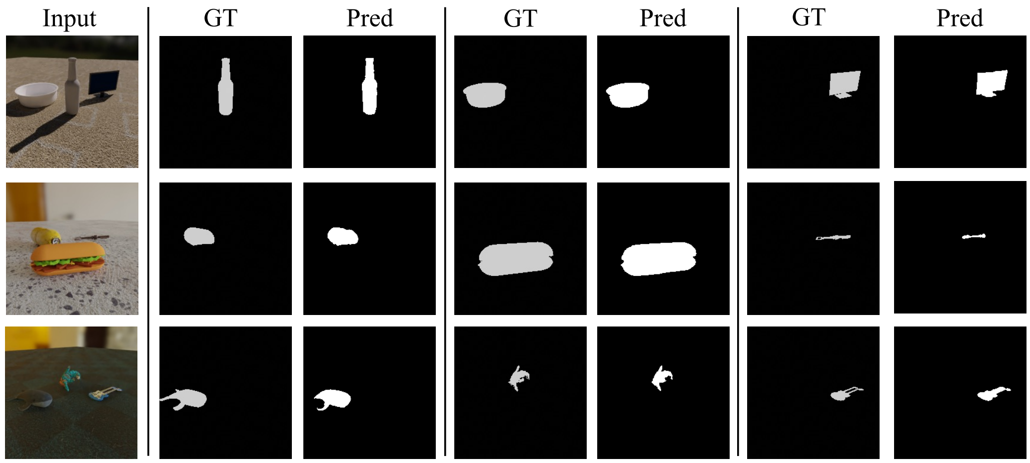

Figure 2: Segmentation prediction results on Toys4K [14] using FreeSOLO [15] fine-tuned on ABC model

Table 4: Results on low-shot recognition on the Toys4k dataset in single object setting. All methods consistently experience a significant drop in accuracy when being evaluated on the harder data variants.

	DINOv1-S/8		DINOv2-S/14		DINOv2-B/14	
Variants	1-shot 5-way	1-shot 10-way	1-shot 5-way	1-shot 10-way	1-shot 5-way	1-shot 10-way
Inst-SObj	95.80±0.46	92.37±0.42	95.75±0.44	93.06±0.41	96.50±0.43	94.22±0.37
Categ-SObj	73.06±0.96	60.73±0.76	77.11±0.89	66.62±0.78	79.69±0.99	69.55±0.77
Categ-SObj-PoseVar	68.84±1.04	57.45±0.77	73.07±1.03	61.44±0.80	75.18±1.04	66.30±0.79

where o, $\hat{o}$, and N_s are ground truth object, predicted object, and the number of support objects respectively (e.g. in the 1-shot-5-way setup $N_s = 5$ since there are 5 support objects in the episode.)

$$LSA = \frac{1}{N_q} \sum_{i=1}^{N_q} \sum_{k=1}^{N_w} \mathbb{1}\{\hat{y}_{ik} = y_{ik}\}$$

where $\hat{y}$ and y are predicted and ground truth labels respectively. The number of query objects is denoted as N_q while N_w is the number of classes (e.g. in the 1-shot-5-way setup, $N_w = 5$ since there are 5 novel classes.)

$$mIoU = \sum_{i=1}^{N} \frac{\hat{m}_i \cap m_i}{\hat{m}_i \cup m_i}$$

where m, $\hat{m}$, and N denote the ground truth mask, predicted mask, and number of objects respectively.

2.2 Main Manuscript Results

In this section, we report the confidence intervals of the experiment results in the main manuscript (Please see Tables 4, 5, 6, 7, and 8). We evaluate our models with 500 episodes and 15 query scenes for each episode.

2.3 Other Low-shot Setups

Table 9 presents the results of DINOv2 ViT B/14, both pre-trained and fine-tuned on ABC, in various low-shot setups, including 1-shot-5-way, 5-shot-5-way, 1-shot-10-way, and 5-shot-10-way under LSME setting on Toys4k.

While the support assignment accuracy (SA) remains consistent across all low-shot setups, the low-shot accuracy shows a notable improvement in the 5-shot scenarios with an approximate 16% increase in low-shot accuracy in both 5-way and 10-way setups.

Table 5: Results on low-shot recognition on the Toys4k dataset in multi-object setting. All methods consistently experience a significant drop in low-shot accuracy when mutual exclusivity is required, and further decrease when instance segmentation is involved.

	DINOv1 ViT S/8		DINOv2 ViT S/14		DINOv2 ViT B/14		CLIP-Img ViT B/16		ImageBind ViT H/16	
Variants	LSA	SA	LSA	SA	LSA	SA	LSA	SA	LSA	SA
Categ-MObj	56.99 ±0.97	N/A	56.95 ±0.99	N/A	57.92 ±1.04	N/A	56.76 ±1.01	N/A	60.49 ±1.00	N/A
Categ-MObj -SuppAssign	40.21 ±1.10	51.68 ±1.95	41.26 ±1.15	52.28 ±1.86	43.21 ±1.21	54.96 ±1.89	41.22 ±1.16	51.64 ±1.87	45.91 ±1.25	58.58 ±2.00
LSME	36.44 ±1.08	46.92 ±2.04	37.08 ±1.05	48.16 ±1.87	39.24 ±1.17	50.88 ±1.91	38.25 ±1.14	48.96 ±2.03	38.85 ±1.14	50.24 ±1.98

Table 6: Performance of DINOv2 and our method fine-tuned on Toys4k and ABC on Toys4k under LSME setting. All methods use ViT B/14 as the backbone and our method is initialized with pretrained DINOv2 weights. Training on ABC improves the performance significantly, surpassing the model that was trained on the base classes of Toys4k with the same number of scenes.

Method	LSA	SA
DINOv2	39.24±1.17	50.88±1.91
Ours-DINOv2-Toys	43.62±1.29	53.44±1.89
Ours-DINOv2-ABC	**47.70±1.26**	**61.32±1.86**

Representation Learning Models: We use pre-trained backbones, (e.g. DINOv1 [3], DINOv2 [11]) contrastive training strategy with a momentum encoder[7]. Given two views of the same scene, v_1 and v_2, we first use the instance mask associated with each object in the scene to eliminate the background and other objects. Subsequently, we extract the query object feature by performing a forward pass of the image encoder on v_1. For each query feature, we minimize the InfoNCE [10] loss function.

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\exp(q \cdot k_+/\tau) + \sum_{k_-} \exp(q \cdot k_-/\tau)}$$

The positive sample k_+ is the feature of the same object in v_2 while the negative set $\{k_-\}$ consists of object features from the memory queue as in MoCo-v2 [6] and different objects from the same scene. For each input view pair, we ensure to only train on objects that are visible in both views (e.g. with instance segmentation area greater than some threshold $\sigma = 30$ pixels).

In our approach, we omit the projector and predictor components present in most contrastive learning approaches [7, 7, 5] since we found empirically that this gave better performance. We trained our model using AdamW optimizer with initial learning rate $5e^{-6}$ and weight decay 0, batch size 32 on 3 RTX 2080 GPUs for 50 epochs. Training took approximately 5 hours in clock time. Our pretrained weights can be found at https://tinyurl.com/3a9r83z9 and the training code is on our GitHub repository at https://github.com/rehg-lab/LSME. All pre-trained weights for other models are directly loaded from the corresponding released codebases.

Segmentation Models: We finetuned the pretrained FreeSOLO [15] model on 1K scenes of ABC dataset with instance mask annotations. To obtain the predicted instance masks for low-shot, we performed a forward pass of the fine-tuned model on our low-shot data. From the output masks, we retained the ones with a confidence score above 0.5. To handle overlapping masks, we merged those with an IoU greater than 0.7. Finally, we employed the Hungarian matching algorithm [9] to associate each predicted mask with its corresponding ground truth mask. We finetuned FreeSOLO with batch size 6 on 3 RTX 2080 GPUs for 30K epochs.

Table 7: Performance of different methods on Toys4k under Categ-SObj-PoseVar and Categ-MObj settings. These settings solve a similar problem, with Categ-MObj having object occlusions present in both support and query objects. Performance of all methods drops significantly when faced with occlusion cases.

Method	DINOv1 S/8	DINOv2 S/14	DINOv2 B/14
Categ-SObj-PoseVar	68.84 ±1.04	73.07 ±1.03	75.18 ±1.04
Categ-MObj	56.99 ±0.97	56.95 ±0.99	57.92 ±1.04

Table 8: The performance of different methods under LSME setting on Toys4k with two object segmenters. The quality of the instance masks plays a significant role in the low-shot and shot assignment performance for all methods.

Method	mIoU		DINOv1 S/8		DINOv2 S/14		DINOv2 B/14	
	Support	Query	LSA	SA	LSA	SA	LSA	SA
FreeSOLO [15]	0.74	0.76	30.05 ±0.84	38.932 ±1.92	32.03 ±0.90	41.72 ±2.01	33.22 ±0.99	44.04 ±1.90
FreeSOLO-ABC	0.85	0.86	36.44 ±1.08	46.92 ±2.04	37.08 ±1.05	48.16 ±1.87	39.24 ±1.17	50.88 ±1.91

Table 9: Results on low-shot recognition on the Toys4k dataset in multi-object setting. All methods consistently experience a significant drop in low-shot accuracy when mutual exclusivity is required, and further decrease when instance segmentation is involved.

	DINOv2 ViT B/14		DINOv2 ViT B/14-ABC	
Low-shot Setup	LSA	SA	LSA	SA
1-shot-5-way	39.24±1.17	50.88±1.91	47.70±1.26	61.32±1.86
5-shot-5-way	55.03±0.99	50.22±0.99	63.52±1.02	60.60±1.13
1-shot-10-way	28.32±0.73	51.32±1.46	35.66±0.82	61.10±1.30
5-shot-10-way	43.26±0.70	50.62±0.69	51.72±0.75	60.85±0.74

References

[1] Blender, https://blender.org/.

[2] Poly haven, https://polyhaven.com/.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

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

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[8] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019.

[9] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

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

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[12] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[13] Stefan Stojanov, Anh Thai, Zixuan Huang, and James M. Rehg. Learning dense object descriptors from multiple views for low-shot category generalization. In *Advances in Neural Information Processing Systems*, pages 12566–12580, 2022.

[14] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021.

[15] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022.