SUPPLEMENTAL MATERIALS: CONTINUAL LEARNING IN OPEN-VOCABULARY CLASSIFICATION WITH COMPLEMENTARY MEMORY SYSTEMS

S-1 ALGORITHMIC DESCRIPTIONS OF TREEPROBE

Algorithm 1: Training Procedure of TreeProbe

Input: Training set X, Tree T, Leaf capacity ψ
Output: Trained classifiers in each leaf node of T

foreach $\mathbf{v}_i \in X$ **do**
  l = NEARESTLEAF($\mathbf{v}_i$, T)
  if COUNT(l) $< \psi$ **then**
    l = INSERTDATA($\mathbf{v}_i$, l)
    TRAINCLASSIFIER(l)
  end
  else
    SPLITNODE(l, $\mathbf{v}_i$)
    l = NEARESTLEAF($\mathbf{v}_i$, T)
    l = INSERTDATA($\mathbf{v}_i$, l)
    TRAINCLASSIFIER(l)
  end
end

Algorithm 2: Inference Procedure of TreeProbe

Input: Image embedding $\mathbf{v}_I$, Tree T, Number of nearest nodes k, Exemplar set M,
Output: Exemplar embedding $\mathbf{v}_e$ for $\mathbf{v}_I$

$\mathbf{v}_\mathcal{K}$ = FINDNEARESTSAMPLES($\mathbf{v}_I$, M)
$\mathbf{v}$ = list()
foreach $\mathbf{v}_i \in \mathbf{v}_\mathcal{K}$ **do**
  l = NEARESTLEAF($\mathbf{v}_i$, T)
  c = GETCLASSIFIER(l)
  $\mathbf{v} \leftarrow$ CLASSIFY($\mathbf{v}_I$, c)
end
$\mathbf{v}_e$ = COMPUTEEMBEDDING($\mathbf{v}$, $\mathbf{v}_I$)

Algorithm 1 and Algorithm 2 contain psuedocode for the training and inference of our TreeProbe method. Definitions of the involved functions are provided below:

- **NEARESTLEAF**($\mathbf{v}_i$, T): Returns the nearest leaf node to the data point x_i in tree T.
- **COUNT**(l): Returns the current number of data points in leaf node l.
- **INSERTDATA**($\mathbf{v}_i$, l): Inserts data point $\mathbf{v}_i$ into leaf node l and returns the updated node.
- **SPLITNODE**(l, $\mathbf{v}_i$): Splits leaf node l into two child nodes when it reaches capacity, distributes data points using KMeans clustering, and adds new data point $\mathbf{v}_i$ to the appropriate child node.
- **TRAINCLASSIFIER**(l): Trains a linear classifier on the data points in leaf node l.
- **FINDNEARESTSAMPLES**($\mathbf{v}_I$, M): Finds the k nearest neighbors in the exemplar set to the image embedding $\mathbf{v}_I$.
- **GETCLASSIFIER**(l): Return the classifier for node l.
- **CLASSIFY**($\mathbf{v}_i$, c): Classifies $\mathbf{v}_i$ using the linear classifier c, returning the label embedding of the most likely label.
- **COMPUTEEMBEDDING**($\mathbf{v}$, $\mathbf{v}_I$): Computes the exemplar embedding for $\mathbf{v}_I$ by applying a similarity-weighted average of the text embeddings of the most likely class labels from a temporal list $\mathbf{v}$.

The notation of Algorithm 1 and Algorithm 2 may differ from the main paper.

S-2 IMPLEMENTATION DETAILS

We conduct our experiments on a setup featuring an RTX 3090 GPU and an AMD Ryzen 9 5950X CPU, using PyTorch as our primary framework. We adhere to the CLIP code example, using sklearn

`LogisticRegression` to implement linear classifiers and setting the sklearn regularization strength to 0.316. The maximum iteration is set to 5k. Our tree probe's node capacity is set at 50k. For efficient retrieval from large-scale exemplar sets, we use FAISS Johnson et al. (2019), specifically using the IndexFlatIP class for its precision and performance. Model performances are gauged via Top-1 accuracy, with the officially released ViT-B/32 CLIP checkpoint serving as our memory or zero-shot model. We select $k = 9$ based on a hyperparameter sweep. Our approach is not sensitive to k, with very similar performance in a range from 6 to 30.

S-3 More experiment setting details

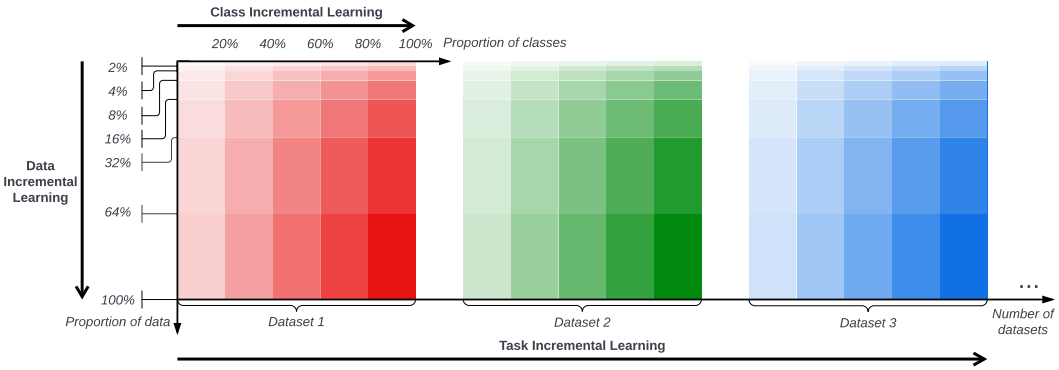


Figure S1: Illustration of continual learning scenarios. Data incremental learning includes seven stages, each comprising 2%, 4%, 8%, 16%, 32%, 64%, and 100% of task data respectively. Class incremental learning divides a task into five stages, each containing 20% of classes. In task incremental learning, each task is considered a stage.

We present an illustration of data, class, and task incremental learning scenarios in Fig. S1. When evaluating different methods in data and class incremental learning scenarios, we ensure fairness by randomly selecting an identical portion of data/class for all methods, achieved by setting the same seed. Models are separately built for each target task. The performance of each stage is averaged across all target tasks. In task-incremental learning, each stage is embodied by a distinct task. The task order is randomly arranged as CIFAR100, SUN397, FGVCAircraft, EuroSAT, OxfordIIITPets, StanfordCars, Food101, and Flowers102, consistent for all methods for fair comparison. As shown in ZSCL Zheng et al. (2023), task order has little impact on the relative performance comparison of different models. For all learning scenarios, we make the assumption that training data accumulates across all stages in all scenarios, with a similar spirit to Prabhu et al. (2023). This assumption is based on the fact that real-world applications are often more limited by computational and time budgets than by storage. If data privacy is not a concern, building a continual learning system without losing data access is more effective than assuming past data is non-accessible. Furthermore, we enhance storage efficiency by saving samples as condensed feature vectors, a significant improvement over some earlier works.

S-4 Descriptions of tasks

We perform experiments on various commonly used visual datasets to demonstrate the generalization capabilities of our method. These datasets encompass a broad range of image categories and application scenarios, including both fine-grained and generalized datasets. We briefly introduce all used tasks in this paper in the following.

S-4.1 General tasks

ImageNet ImageNet Russakovsky et al. (2015) contains 1,281,167 training images, 50,000 validation images and 100,000 test images. The categories represent a wide variety of objects, animals, scenes, and even abstract concepts. This dataset has served as a fundamental dataset to evaluate performances of classification models, or as a pretraining dataset.

CIFAR100 The CIFAR100 dataset Krizhevsky & Hinton (2009) consists of object images and is a subset of the 80 million tiny images dataset. It contains 60,000 32×32 color images from 100 object categories, with 600 images per category. The dataset has 100 fine-grained classes, grouped into 20 coarse-grained classes.

SUN397 The SUN397 dataset Xiao et al. (2010) consists of scene images, containing 108,754 images across 397 scene categories, with each category having between 100 and 500 images. This dataset is commonly used for scene understanding tasks. Since there is no official dataset split for this dataset, we randomly select 60% of images as training data, 20% as validation data, and the rest as test data. We use NumPy random permutation to split with the seed set to 0.

S-4.2 FINE-GRAINED TASKS

FGVCAircraft The FGVCAircraft dataset Maji et al. (2013) serves as a benchmark for fine-grained visual categorization of aircraft. It contains 10,200 images from 102 distinct categories. Each category includes approximately 100 images, annotated with the aircraft model, variant, and manufacturer.

DTD The Describable Textures Dataset (DTD) Cimpoi et al. (2014) consists of 5,640 images across 47 texture categories, with each category featuring 120 real-world texture images such as fabrics, rocks, and surfaces. The dataset poses a challenge for texture classification due to subtle differences between textures within the same category and large variations in texture appearance caused by scale, orientation, and lighting.

Food101 The Food-101 dataset Bossard et al. (2014) comprises 101,000 images across 101 food categories, each with 1,000 images. This dataset challenges fine-grained image classification due to high intra-class variation and visual similarities across categories. It serves as a rigorous benchmark for evaluating computer vision models in food recognition and provides a robust platform for training machine learning models in understanding culinary aesthetics and preferences.

StanfordCars The StanfordCars dataset Krause et al. (2013) is a benchmark dataset containing 16,185 images from 196 different car classes, divided into a 50-50 training and testing split. The classes correspond to specific car makes, models, and years, such as the 2012 Tesla Model S or 2012 BMW M3 coupe.

Flowers102 The 102 Category Flower Dataset Nilsback & Zisserman (2008) is a compilation of flower images. It includes 8,189 images across 102 flower categories, with each category containing between 40 and 258 images. The dataset's images vary in size and aspect ratio, captured using different cameras, lighting conditions, and backgrounds.

OxfordIIITPets The OxfordIIITPets dataset Parkhi et al. (2012) is a collection of pet images, featuring 7,349 images from 37 different cat and dog breeds. Each breed has between 100 and 200 images. The dataset is challenging because the appearance of the same breed can vary significantly, and different breeds may have similar-looking features.

EuroSAT The EuroSAT dataset Helber et al. (2019) is a remote sensing image dataset comprising Sentinel-2 satellite data. It contains 27,000 images that cover 13 spectral bands and consist of 10 different land use and land cover categories, including forests, urban areas, and water bodies. This dataset is commonly employed for remote sensing and land cover classification tasks. Since there is no official dataset split for this dataset, we randomly select 70% of images as training data and the rest as validation data. We use NumPy random permutation to perform splitting with the seed set to 0.

UCF101 The UCF101 dataset Soomro et al. (2012) is a commonly used benchmark for action recognition. It consists of 13,320 videos from 101 action categories, with each category containing at least 100 videos. The actions include a wide range of human activities such as basketball shooting, horse riding, and juggling. The dataset is unique in its focus on complex, naturalistic action sequences, with videos varying in length from a few seconds to a minute. Since there is no official dataset split for this dataset, we randomly select 70% of images as training data and the rest as validation data. We use NumPy random permutation to perform splitting with the seed set to 0.

S-4.3 LONG-TAILED TASK

Places365LT Places365LT Liu et al. (2019) a synthetic long-tail derivative of Places2 dataset Zhou et al. (2018). The image resolution is 256×256. It contains 365 scene classes with at least 5 samples each. The classes are not uniformly distributed, forming a long-tailed distribution. It contains some label noises, making classification even harder on this dataset.

S-5 PROMPT TEMPLATES FOR TASKS

Task(s)	Prompt template
ImageNet Russakovsky et al. (2015), CIFAR100 Krizhevsky & Hinton (2009), SUN397 Xiao et al. (2010)	"a photo of a {label}."
FGVCAircraft Maji et al. (2013)	"a photo of a {label}, a type of aircraft."
DTD Cimpoi et al. (2014)	"a photo of a {label} texture."
StanfordCars Krause et al. (2013)	"a photo of a {label}, a type of car."
Food101 Bossard et al. (2014)	"a photo of {label}, a type of food."
Flowers102 Nilsback & Zisserman (2008)	"a photo of a {label}, a type of flower."
OxfordIIITPets Parkhi et al. (2012)	"a photo of a {label}, a type of pet."
EuroSAT Helber et al. (2019)	"a centered satellite photo of {label}."
UCF101 Parkhi et al. (2012)	"a video of a person doing {label}."
Places365LT Liu et al. (2019)	"a photo of the {label}, a type of place."

Table 1: Prompts of tasks

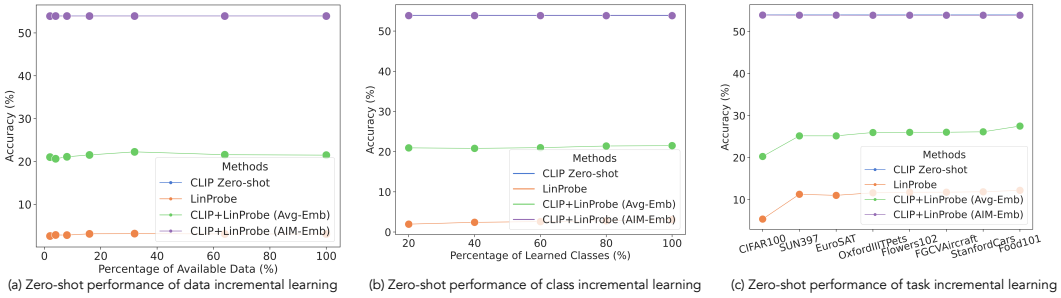
CLIP Radford et al. (2021) suggests utilizing a sentence template (e.g., ``A photo of a {label}.''), as input to the text decoder instead of a plain text label, due to its training data being primarily full sentences describing images. Consistent with this paper's focus, we employ a simple prompt template for each task. Most of these templates are based on CLIP's recommendations[3] and are summarized in Tab. 1.

S-6 ZERO-SHOT PERFORMANCES ON DIFFERENT TASKS

Task	Zero-shot Acc (%)	Official ZS Acc (%)
ImageNet Russakovsky et al. (2015)	59.7	63.2
CIFAR100 Krizhevsky & Hinton (2009)	62.3	65.1
SUN397 Xiao et al. (2010)	59.2	63.2
FGVCAircraft Maji et al. (2013)	18.1	21.2
DTD Cimpoi et al. (2014)	42.0	44.5
StanfordCars Krause et al. (2013)	58.6	59.4
Food101 Bossard et al. (2014)	82.6	84.4
Flowers102 Nilsback & Zisserman (2008)	67.9	66.7
OxfordIIITPets Parkhi et al. (2012)	87.5	87.0
EuroSAT Helber et al. (2019)	45.4	49.4
UCF101 Parkhi et al. (2012)	60.1	64.5
Places365LT Liu et al. (2019)	40.0	/

Table 2: Zero-shot performances of CLIP ViT-B/32 pretrained model on different tasks. "ZS" is for zero-shot and "Acc" is for accuracy. Results of column "Official ZS Acc" are taken from the CLIP original paper Radford et al. (2021). "/" represents lack of official results.

Tab. 2 shows the zero-shot performance of our implementation in different tasks. We conjecture that the main difference of official zero-shot performances comes from the ensemble prompt trick as mentioned in CLIP Radford et al. (2021) and randomness in dataset splits of several tasks (e.g., SUN397).



(a) Zero-shot performance of data incremental learning (b) Zero-shot performance of class incremental learning (c) Zero-shot performance of task incremental learning

Figure S2: Zero-shot performances on data (a), class (b), and task (c) incremental learning scenarios. x-axis represents stages when the evaluation is performed.

S-7 Additional Results

S-7.1 Zero-shot task performances of every stage

We further append the results of zero-shot tasks evaluated after each stage on the data, class, and task incremental learning in Fig. S2. Across all scenarios, AIM-Emb helps maintain the zero-shot performances on all stages.

S-7.2 More results of the comparison to previous methods

Method	**Transfer**	△	**Avg.**	△	**Last**	△
CLIP Zero-shot	69.4	0.0	65.3	0.0	65.3	0.0
LwF Li & Hoiem (2016)	56.9	-12.5	64.7	-0.6	74.6	+9.3
iCaRL Rebuffi et al. (2017)	50.4	-19.0	65.7	+0.4	80.1	+14.8
WiSE-FT Wortsman et al. (2022a)	52.3	-17.1	60.7	-4.6	77.7	+12.4
ZSCL Zheng et al. (2023)	68.1	-1.3	75.4	+10.1	83.6	+18.3
KNN	69.3	-0.1	72.7	+7.4	78.4	+13.1
LinProbe	69.3	-0.1	77.1	+11.8	86.0	+20.7
TreeProbe (50k)	69.3	-0.1	75.9	+10.6	85.5	+20.2

Table 3: Comparison of different methods on MTIL in Order I from ZSCL Zheng et al. (2023). KNN, LinProbe , and TreeProbe (50k) are complementary methods with AIM-Emb as the fusing approach.

We also supplement the results obtained by KNN and LinProbe while comparing to previous methods in Tab. 3. As shown, all of the approaches of complementary systems with AIM-Emb achieve good Transfer, reiterating the effectiveness of AIM. LinProbe excels at all metrics with the cost of efficiency, which is predictable from the results shown in our main manuscript.

To give more details on the accuracies we achieve on every task under all stages for TreeProbe (50k), we follow ZSCL Zheng et al. (2023) to give all numbers in Tab. 4 for further reference.

S-8 Additional ablation experiments

Effect of different fusing operations. We describe several forms of the fusing operations in Sec. 3.3, including: Avg-Prob, AIM-Prob, Avg-Emb, and AIM-Emb. We compare using TreeProbe (50k) under the task incremental learning scenario. Fig. S3 shows the results on target and zero-shot tasks. The figure shows that AIM-Emb and AIM-Prob have similar performance on target tasks, surpassing the other two fusing operations. Combined with the zero-shot performance, the results suggest the effectiveness of AIM in adaptively choosing the better prediction model. The probabilistic prediction is better than the embedding prediction when performing averaging in both target tasks and zero-shot tasks. But when combined with AIM, the embedding version has a reasonably better performance in zero-shot tasks. Therefore, we choose AIM-Emb as the default fusing operation.

[3] https://github.com/openai/CLIP/blob/main/data/prompts.md

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	
Transfer		87.90	68.22	45.32	54.61	71.08	88.86	59.45	89.07	64.61	64.05	69.3
Aircraft	52.45	87.90	68.22	45.32	54.61	71.08	88.86	59.45	89.07	64.61	64.05	
Caltech101	52.48	96.89	68.22	45.32	54.61	71.08	88.86	59.45	89.07	64.61	64.05	
CIFAR100	52.48	96.89	68.22	45.32	54.61	71.08	88.86	59.45	89.07	64.61	64.05	
DTD	52.42	96.83	81.98	70.32	54.61	71.08	88.86	59.45	89.07	64.61	64.05	
EuroSAT	52.45	89.92	81.99	66.65	95.74	71.08	88.86	59.45	89.07	64.61	64.05	
Flowers	52.51	90.55	81.93	66.44	95.74	54.12	88.86	59.45	89.07	64.61	64.05	
Food	52.48	90.78	81.94	67.23	95.78	65.49	92.25	59.45	89.07	64.61	64.05	
MNIST	52.09	93.95	81.96	69.73	94.33	95.69	92.27	98.59	89.07	64.61	64.05	
OxfordPet	52.63	95.10	82.05	70.05	95.63	95.69	92.29	98.58	92.91	64.61	64.05	
Cars	52.54	95.22	81.94	67.98	94.15	95.59	92.29	98.58	93.02	86.27	64.05	
SUN397	52.48	95.56	81.94	66.91	95.59	95.59	92.21	98.60	93.19	86.15	81.76	85.5
Avg.	52.45	93.59	79.47	61.93	80.49	77.96	90.40	73.68	90.15	68.53	65.66	75.9

Table 4: Accuracy (%) of our TreeProbe (50k) model on the MTIL benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color. We follow the same table arrangement as in ZSCL Zheng et al. (2023).

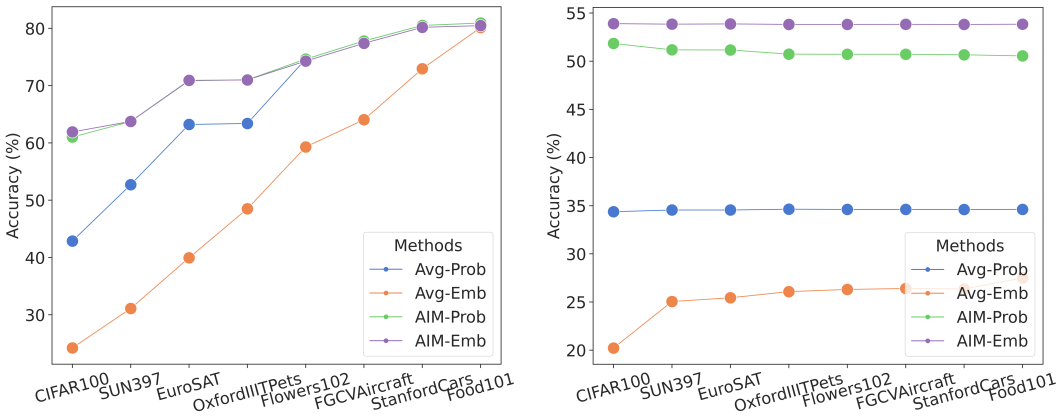


Figure S3: Target (left) and zero-shot (right) accuracies of different fusing operations.

Different versions of KNN and choice of k. As indicated in Sec. 3.2, we can get the embedding by taking the one attached to the most likely label (MV−KNN), or averaging the text embeddings of the k nearest neighbors (AVG−KNN), or performing a weighted-averaging over the embeddings of all k nearest neighbors where weights come from the similarities between the image embedding and the k neighbors' image embeddings (WAVG−KNN). We also compare these approaches under different ks to further choose a suitable k for experiments. As indicated in Fig. S4, from the curve, $k = 9$ gives reasonable performances of all approaches on both the target and zero-shot tasks. Compared to AVG−KNN, larger k is more beneficial for MV−KNN since larger k is more stable for MV−KNN, and it is more likely to include more mismatches from the nearest neighbors forAVG−KNN . From the plot, we can clearly read that WAVG−KNN is consistently better than AVG−KNN and MV−KNN across different ks, making it our default option of the prediction approach for fast learning system.

Effect of ensemble classifiers in TreeProbe inference. Referencing Sec. 3.2, we observe that ensemble predictions from multiple classifiers associated with k retrievals slightly enhance performance. Fig. S5 presents these results under the task incremental learning setting for eight target tasks. Our final model, TreeProbe, is better than its variant without the ensemble classification function in both target and zero-shot performance. The additional inference cost for ensemble predictions is negligible, so we choose it as the default setting for its better performance.

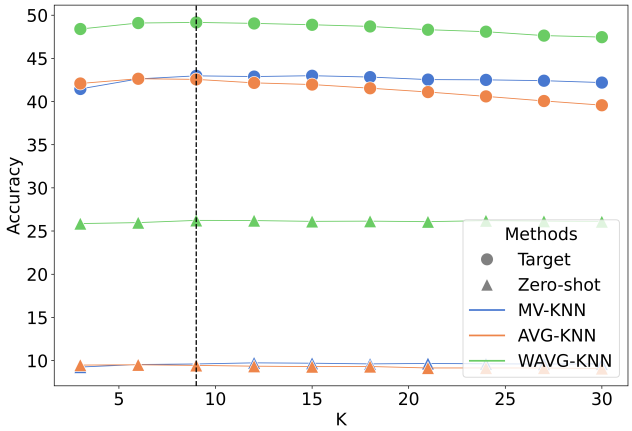


Figure S4: Results of different versions of KNN on target tasks after finishing all stages under the task incremental learning scenario. We further ablate on k selection so choose k as the x-axis. The vertical dashed line represents $k = 9$.

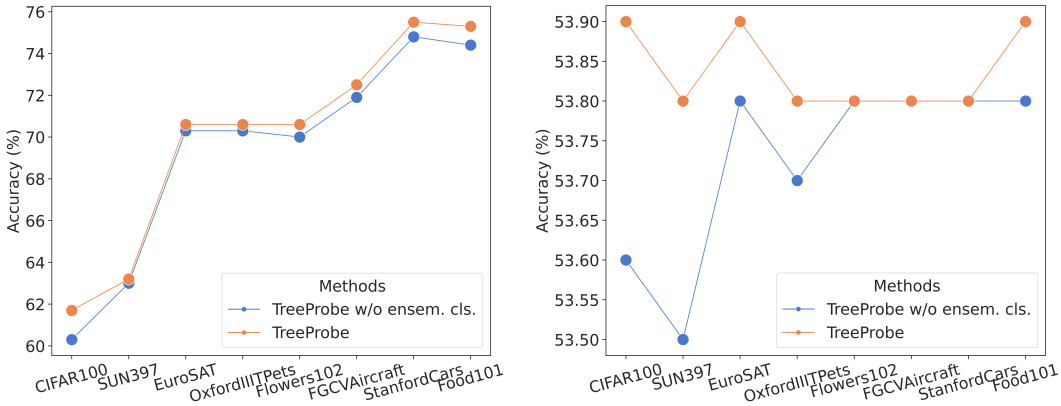


Figure S5: Target and zero-shot task performance comparison w.r.t. ensemble classifiers. "TreeProbe w/o ensem. cls." is the version of TreeProbe by finding the cluster most similar to the input and using the corresponding classifier to predict labels. The reported accuracy is the average across tasks after incrementally receiving training data from the datasets shown on the x-axis.

S-9 Evaluation on long-tailed classification

Method	CLIP Zero-shot	KNN	LinProbe	TreeProbe	KNN*	LinProbe*	TreeProbe*	PaCo	RAC
Accuracy (%)	40.0	35.5	37.0	30.5	40.4	42.7	41.3	41.2	47.2

Table 5: Comparison of long-tailed classification on Places365LT Liu et al. (2019). * means + AIM-Emb. For this experiment, we use CLIP ViT-L/14@336px as the backbone network.

In long-tailed classification, some test labels are rare or unobserved in training, so blending exemplar-based models with consolidated models can be beneficial, as shown by Long et al. (2022). To accommodate this setting, we adjust our AIM-Emb method by considering the 2/3 rarest labels as not being present in the exemplar set and the remainder as being present, to calculate $\mathbf{v}_{\text{out}}$. In this experiment, the node capacity of TreeProbe methods is 10k. In Tab. 5, we present results on the Places365LT dataset Liu et al. (2019). Our AIM-Emb method with LinProbe and TreeProbe outperform the zero-shot baseline. We also compare to PaCo Cui et al. (2021) and RAC Long et al. (2022), which are specifically designed for long-tail classification. PaCo incorporates learnable class centers to account for class imbalance in a contrastive learning approach. RAC trains an image encoder augmented with label predictions from retrieved exemplars. Although not specifically

designed for long-tailed classification, our method performs similar to PaCo, but RAC performs best of all.

REFERENCES

Charu C. Aggarwal. Instance-based learning : A survey. In *Data Classification: Algorithms and Applications*, chapter 6. CRC Press, 2014.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pp. 7120–7129, 2017.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *ICLR*, 2022.

Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

Jihwan Bang, Heesu Kim, Youngjoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pp. 8218–8227, 2021.

Gianluca Bontempi, Mauro Birattari, and Hugues Bersini. Lazy learning for local modelling and control design. *International Journal of Control*, 72(7-8):643–658, 1999.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. *ArXiv*, abs/2102.02779, 2021.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proc. CVPR*, 2014.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *ICCV*, pp. 695–704, 2021.

C. Domeniconi and D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. In *NeurIPS*, 2002.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pp. 986–996. Springer, 2003.

Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. Towards general purpose vision systems: An end-to-end task-agnostic vision-language architecture. In *CVPR*, 2022.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. Webly supervised concept expansion for general purpose vision models. In *ECCV*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Z. Li and D. Hoiem. Learning without forgetting. *PAMI*, 40(12):2935–2947, 2018.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Proc. ECCV*, 2016.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pp. 388–404. Springer, 2022.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pp. 2537–2546, 2019.

Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proc. CVPR*, 2022.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proc. NeurIPS*, 2017.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *ArXiv*, abs/2206.08916, 2022.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

Randall C. O'Reilly, Rajan Bhattacharyya, Michael D. Howard, and Nicholas Ketz. Complementary learning systems. *Cogn. Sci.*, 38(6):1229–1248, 2014.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012.

Ameya Prabhu, Philip Torr, and Puneet Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *The European Conference on Computer Vision (ECCV)*, August 2020.

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K. Dokania, Philip H. S. Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? *CoRR*, abs/2303.11165, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proc. CVPR*, 2017.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley D. Edwards, Nicolas Manfred Otto Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *ArXiv*, abs/2205.06175, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proc. ICML*, pp. 4555–4564, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, pp. 2990–2999, 2017.

Yunsheng Song, Jiye Liang, Jing Lu, and Xingwang Zhao. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251:26–34, 2017.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygXPaEYvH.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, pp. 7949–7961, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022b.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: dynamically expandable representation for class incremental learning. In *Proc. CVPR*, 2021.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proc. ICML*, pp. 3987–3995, 2017.

Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 2126–2136. IEEE, 2006.

Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas J. Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *ECCV*, pp. 698–714, 2020.

Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3):1–19, 2017.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. *CoRR*, abs/2303.06628, 2023.

Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2018.