Few-Shot Recognition via Stage-Wise Retrieval-Augmented Finetuning

Supplementary Material

Outline

This document supports our main paper with detailed results and comprehensive analyses. The document is organized as below:

- **Section A.** We provide a detailed summary of benchmarking datasets used in our experiments.
- **Section B.** We provide details of hyperparameters used in our work.
- **Section C.** We report detailed results of comparing SWAT with previous FSR methods for each benchmark dataset.
- **Section D.** We provide details on how we retrieve pretraining data and compare different retrieval and filtering methods.
- **Section E.** We compare different mixed sample data augmentation methods and analyze the impact of the mixing ratio within a batch.
- **Section F.** We validate the design of our SWAT by ablating different stage-2 training strategies and comparing SWAT with recent state-of-the-art finetuning methods.
- **Section G.** We provide further analyses on SWAT, including the impact of training epochs, different classifier initialization methods, and more detailed experimental results.
- **Section H.** We provide analysis of the imbalance of retrieved data and the impact of retrieval size.
- **Section I.** We provide code and instructions for replicating our experiments.

A. Summary of Datasets

We summarize the nine fine-grained datasets used in our experiments in Table 6. Following [36, 59], we sample few-shot data from the official training set and evaluate model performance on the official test set, except for ImageNet where we report performance on its validation set. We repeat each experiment three times with three random seeds. Note that we strictly follow the validation-free protocol [59] that we do not use any validation data for hyperparameter tuning or model selection.

Table 6. **Statistics of nine fine-grained datasets repurposed in our work.** We list the number of images in the official training, validation, and test sets for each dataset. The protocol of few-shot recognition samples few-shot data from the official training set; we use them as *our train set*. We repeat the sampling and training three times for each method with three random seeds. To evaluate methods, we repurpose their official test set as *our test set* (except on ImageNet where we use its official validation set as our test set). We benchmark methods on *our test sets*. Note again that we *do not* use any validation examples for model selection or hyperparameter tuning; instead, we strictly adhere to the realistic validation-free protocol for few-shot research [59].

dataset	# cls	official-train	official-val	official-test	task
Semi-Aves [61]	200	3,959	2,000	4,000	recognize birds
Flowers [41]	102	4,093	1,633	2,463	recognize flowers
Aircraft [39]	100	3,334	3,333	3,333	recognize aircrafts
EuroSAT [19]	10	13,500	5,400	8,100	classify satellite images
DTD [11]	47	2,820	1,128	1,692	recognize textures
OxfordPets [46]	37	2,944	736	3,669	recognize pets
Food101 [4]	101	50,500	20,200	30,300	recognize food
StanfordCars [32]	196	6,509	1,635	8,041	recognize cars
ImageNet [12]	1,000	1.28M	50,000	N/A	large scale recognition

B. Hyperparameter Setting

Stage-1 End-to-End Finetuning. We follow previous work in other lines [33, 36, 43] to set hyperparameters in our work. Specifically, for stage-1 end-to-end finetuning of SWAT, we follow suggestions from [33, 69] to use a smaller learning rate (1e-6) for updating the visual encoder and a larger learning rate (1e-4) for the linear classifier. We initialize the classifier weights using the text embedding following [43] (cf. Table 14). For other hyperparameters, we adopt the values reported in [36, 43] which include the AdamW optimizer, a batch size of 32, weight decay of 1e-2, and a cosine-annealing learning rate schedule with 50 warm-up iterations. We do not do early stopping as we strictly follow the validation-free protocol [59]. Instead, we train for 50 epochs. The only exception is for ImageNet, we train for 10 epochs due to the large amount of retrieved data for its 1,000 classes. The temperature factor is learned during the finetuning process with an initial learning rate of 1e-4 and the same cosine-annealing learning rate schedule. For data augmentation, we mix retrieved data with few-shot data using CutMix [76], following [45] to sample the mixing ratio from a uniform distribution ($\alpha = 1.0$ for beta distribution) and apply CutMix with a probability of 0.5. Our few-shot finetuning (FSFT) adopts the same set of training recipe.

Stage-2 Classifier Retraining. We use the same set of hyperparameters and follow the practice in [43] to train for 10 epochs with a fixed temperature of 0.01. We initialize the classifier in stage 2 using the learned classifier weights from stage-1 end-to-end finetuning, following [33].

Baselines. For baseline methods, we reimplement Cross-Modal Linear Probing [36] using the same hyperparameters as in stage-2 classifier retraining and training for 50 epochs. We obtain the results of CLAP [59] using OpenCLIP models with its default hyperparameters. For other baseline methods

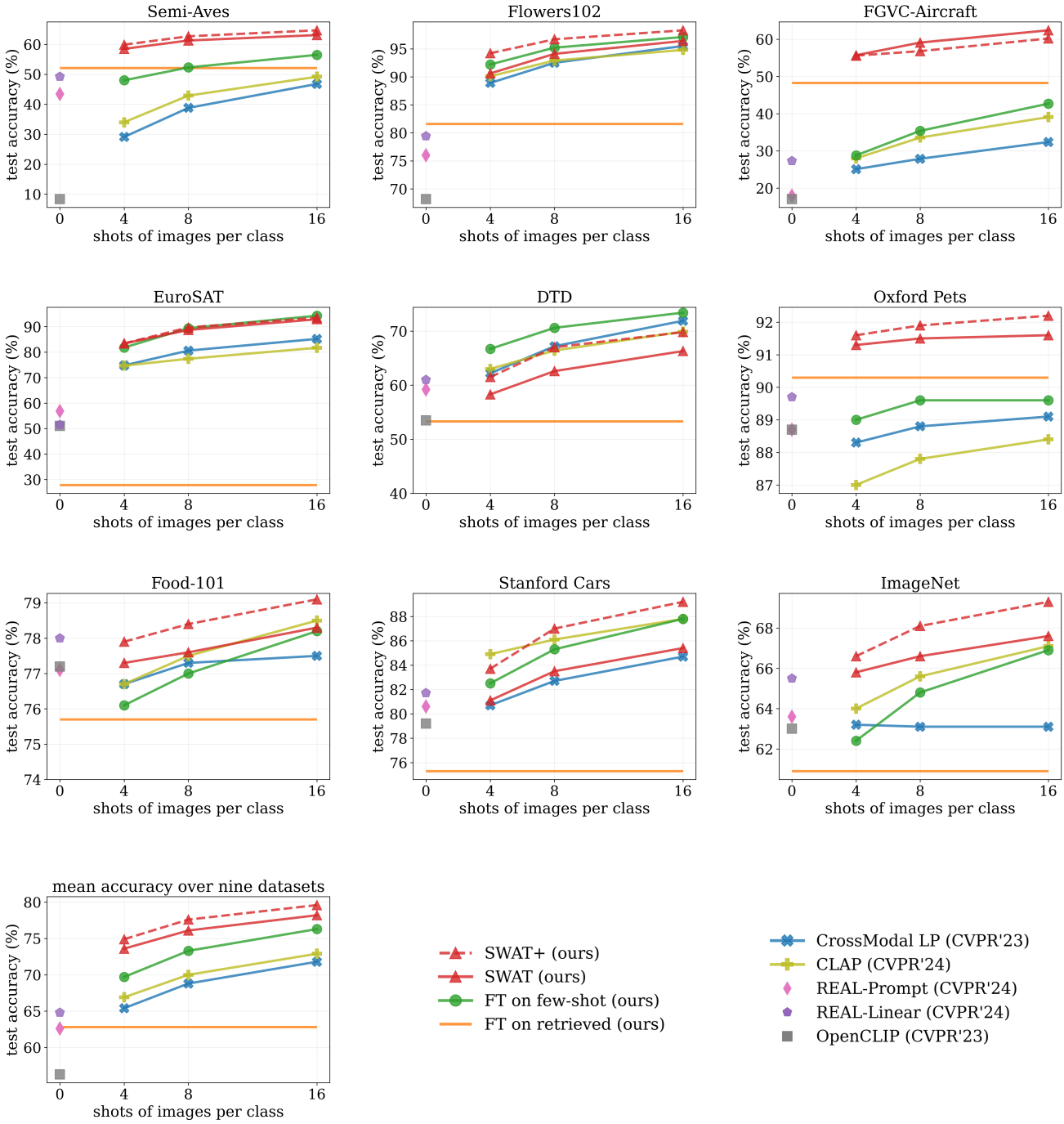

Figure 5. **Comparison of SWAT with state-of-the-art zero-shot and few-shot methods.** We show that simply finetuning the whole visual encoder on few-shot data (our few-shot finetuning, green line) outperforms previous FSR methods while finetuning on retrieved data (orange line) underperforms zero-shot methods (e.g., ImageNet, EuroSAT, Food, DTD, and Stanford Cars) due to the large domain gap and imbalanced distributions of retrieved data. Our SWAT (red line) outperforms previous methods by >6% w.r.t accuracy over nine datasets, with significant improvements (20-30%) on challenging datasets like Semi-Aves and Aircraft. The results validate the effectiveness of our SWAT in mitigating the domain gap and imbalanced distribution issues. We also show that our SWAT+ (red dashed line) which finetunes both visual encoder and classifier on few-shot data in stage 2 improves further over SWAT (cf. Section F). Detailed performance on each dataset is provided in Table 7. For Flowers, EuroSAT, DTD, and Stanford Cars datasets, we show that SWAT can be further improved by 1-6% of accuracy with proper filtering on the retrieved data (cf. Table 9).

Table 7. **Detailed comparison of our SWAT and few-shot finetuning (FSFT) with state-of-the-art zero-shot and few-shot recognition methods using OpenCLIP ViT/B-32 model.** SWAT significantly outperforms previous few-shot recognition methods by 6% across nine datasets. We also include the results of FSFT with and without CutMix, as well as SWAT+ where we finetuning the whole model rather than only the classifier on few-shot data in the second stage (cf. Section F). We highlight the best number in **bold** and underline the second best. Superscripts mark improvements compared to previous state-of-the-art FSR method CLAP [59].

shots	strategy	methods	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet	average
0	prompting	OpenCLIP [10]	8.4	68.2	17.1	51.1	53.5	88.7	77.2	79.2	63.0	56.3
	retrieval-augmented	REAL-Prompt [43]	43.4	76.0	18.0	56.9	59.2	88.7	77.1	80.6	63.6	62.6
		REAL-Linear [43]	49.2	79.4	27.3	51.5	61.0	89.7	78.0	81.7	65.5	64.8
4	prompt-learning	CoOp [83]	38.1	86.1	20.6	68.6	53.9	86.7	73.5	62.7	58.5	61.0
		PLOT [8]	37.2	87.8	22.4	72.4	56.0	88.6	77.2	63.4	61.5	62.9
	adapter-based	CLIP-Adapter [15]	39.2	85.3	23.0	72.5	47.2	80.0	72.1	61.0	55.7	59.6
		TIP-Adapter [78]	37.4	69.8	19.6	54.3	53.5	82.3	74.7	57.7	60.2	56.6
		TIP-Adapter (f) [78]	42.4	74.4	21.9	66.8	58.0	85.5	75.3	61.1	61.5	60.8
		TaskRes(e) [75]	43.2	89.4	25.9	73.0	58.4	84.6	74.5	64.7	58.0	63.5
		CrossModal-LP [36]	29.1	88.9	25.1	74.8	62.2	88.3	76.7	80.7	63.2	65.4
		CLAP [59]	34.0	90.1	28.0	74.7	63.0	87.0	76.7	**84.9**	64.0	66.9
	finetuning-based	FSFT (ours)	47.5$^{+13.5}$	92.5$^{+1.4}$	27.9$^{-0.1}$	81.6$^{+6.9}$	66.6$^{+3.6}$	88.7$^{+1.7}$	75.8$^{-0.9}$	81.5$^{-3.4}$	62.3$^{-1.7}$	69.4$^{+2.5}$
		FSFT w/ CutMix (ours)	48.0$^{+14.0}$	92.2$^{+1.1}$	28.8$^{+0.8}$	81.8$^{+7.1}$	**66.7**$^{+3.7}$	89.0$^{+2.0}$	76.1$^{-0.6}$	82.5$^{-2.4}$	62.4$^{-1.6}$	69.7$^{+2.8}$
		SWAT (ours)	58.5$^{+24.5}$	90.6$^{+0.5}$	**55.7**$^{+27.7}$	83.2$^{+8.5}$	58.3$^{-4.7}$	91.3$^{+4.3}$	77.3$^{+0.6}$	81.1$^{-3.8}$	65.8$^{+1.8}$	73.5$^{+6.6}$
		SWAT+ (ours)	**59.9**$^{+25.9}$	**94.2**$^{+4.1}$	55.6$^{+27.6}$	**83.4**$^{+8.7}$	61.5$^{-1.5}$	**91.6**$^{+4.6}$	**77.9**$^{+1.2}$	83.7$^{-1.2}$	**66.6**$^{+2.6}$	**74.9**$^{+8.0}$
8	prompt-learning	CoOp [83]	42.0	91.3	26.6	77.1	59.7	85.4	71.6	67.6	60.4	64.6
		PLOT [8]	41.4	92.4	26.2	78.2	61.7	87.4	75.3	67.0	61.9	65.7
	adapter-based	CLIP-Adapter [15]	41.2	91.9	27.9	78.5	61.4	83.4	72.1	66.8	57.0	64.5
		TIP-Adapter [78]	39.8	73.8	19.4	62.3	51.5	82.3	73.9	57.6	59.4	57.8
		TIP-Adapter (f) [78]	46.2	84.3	23.8	70.3	59.8	85.6	75.0	64.4	61.8	63.5
		TaskRes(e) [75]	47.1	94.3	30.9	78.8	63.5	85.7	74.4	69.7	59.1	67.1
		CrossModal-LP [36]	38.8	92.5	27.9	80.6	67.2	88.8	77.3	82.7	63.1	68.8
		CLAP [59]	42.9	92.9	33.6	77.4	66.4	87.8	77.5	86.1	65.6	70.0
	finetuning-based	FSFT (ours)	51.2$^{+8.3}$	95.4$^{+2.5}$	33.1$^{-0.5}$	**90.3**$^{+12.9}$	**71.0**$^{+4.6}$	89.3$^{+1.5}$	76.0$^{-1.5}$	83.5$^{-2.6}$	64.4$^{-1.2}$	72.7$^{+2.7}$
		FSFT w/ CutMix (ours)	52.3$^{+9.4}$	95.2$^{+2.3}$	35.4$^{+1.8}$	89.4$^{+12.0}$	70.6$^{+4.2}$	89.6$^{+1.8}$	77.0$^{-0.5}$	85.3$^{-0.8}$	64.8$^{-0.8}$	73.3$^{+3.3}$
		SWAT (ours)	60.8$^{+17.9}$	94.1$^{+1.2}$	**59.1**$^{+25.5}$	89.2$^{+11.8}$	62.6$^{-3.8}$	90.8$^{+3.0}$	77.5$^{+0.0}$	83.5$^{-2.6}$	66.6$^{+1.0}$	76.0$^{+6.0}$
		SWAT+ (ours)	**62.7**$^{+19.8}$	**96.7**$^{+3.8}$	56.8$^{+23.2}$	89.7$^{+12.3}$	67.0$^{+0.6}$	**91.9**$^{+4.1}$	**78.4**$^{+0.9}$	**87.0**$^{+0.9}$	**68.1**$^{+2.5}$	**77.6**$^{+7.6}$
16	prompt-learning	CoOp [83]	46.1	94.5	31.4	83.7	62.5	87.0	74.5	73.6	61.9	68.4
		PLOT [8]	44.4	94.8	31.5	82.2	65.6	87.2	77.1	72.8	63.0	68.7
	adapter-based	CLIP-Adapter [15]	43.6	94.6	34.2	83.2	65.7	84.9	74.0	73.5	59.0	68.1
		TIP-Adapter [78]	42.0	78.4	22.0	67.9	54.8	81.1	73.0	58.8	57.8	59.5
		TIP-Adapter (f) [78]	50.1	91.2	29.3	76.6	64.6	85.4	74.7	69.6	62.3	67.1
		TaskRes(e) [75]	48.5	96.1	36.5	83.7	65.9	86.3	75.4	75.4	60.9	69.9
		CrossModal-LP [36]	46.8	95.5	32.4	85.2	71.9	89.1	77.5	84.7	63.1	71.8
		CLAP [59]	49.2	94.8	39.1	81.7	69.9	88.4	78.5	87.8	67.1	72.9
	finetuning-based	FSFT (ours)	55.3$^{+6.1}$	97.0$^{+2.2}$	37.0$^{-2.1}$	94.0$^{+12.3}$	73.3$^{+3.4}$	89.5$^{+1.1}$	77.1$^{-1.4}$	85.7$^{-2.1}$	66.7$^{-0.4}$	75.1$^{2.2}$
		FSFT w/ CutMix (ours)	56.5$^{+7.3}$	97.1$^{+2.3}$	42.7$^{+3.6}$	**94.3**$^{+12.6}$	**73.4**$^{+3.5}$	89.6$^{+1.2}$	78.2$^{-0.3}$	87.8$^{+0.0}$	66.9$^{-0.2}$	76.3$^{+3.4}$
		SWAT (ours)	63.1$^{+13.9}$	96.4$^{+1.6}$	**62.4**$^{+23.3}$	92.6$^{+10.9}$	66.3$^{-3.6}$	91.6$^{+3.2}$	78.3$^{-0.2}$	85.4$^{-2.4}$	67.6$^{+0.5}$	78.2$^{+5.3}$
		SWAT+ (ours)	**64.7**$^{+5.5}$	**98.3**$^{+3.5}$	60.2$^{+21.1}$	93.5$^{+11.8}$	69.8$^{-0.1}$	**92.2**$^{+3.8}$	**79.1**$^{+0.6}$	**89.2**$^{+1.4}$	**69.3**$^{+2.2}$	**79.6**$^{+6.7}$

that originally used an unrealistically large validation set for hyperparameter tuning, we copy their results from [59].

C. Detailed Benchmarking Results

We compare our SWAT and few-shot finetuning (FSFT) with prior state-of-the-art zero-shot [21, 43] and few-shot recognition methods [36, 59] using the OpenCLIP ViT-B/32 model in Fig. 5 and list the detailed performance in Table 7. We also include the performance of our few-shot finetuning without CutMix. Results show that SWAT outperforms previous FSR methods by >6% accuracy over nine datasets, with substantial gains (20-30%) on challenging datasets where prior FSR accuracy [36, 59] was below 50% (e.g., Semi-Aves and Aircraft). Additionally, SWAT with OpenCLIP ViT-B/16 model (Table 8) yields even higher gains of 8% over [59] across nine datasets.

Further Analysis. Our experiments show that SWAT underperforms prior state-of-the-art FSR method [59] on DTD and Stanford Cars. We conjecture that this is due to the significant domain gaps in the retrieved data, finetuning on which could hurt the model's performance. This motivates us to apply a filtering technique to remove excessively out-of-domain retrieved data. Indeed, as shown in Table 9, applying

Table 8. **Detailed comparison of SWAT with state-of-the-art zero-shot and few-shot recognition methods using OpenCLIP ViT-B/16 model.** Results show that SWAT achieves larger performance gains (∼8%) over CLAP [59] with a larger backbone of ViT-B/16. We also include the results of FSFT with and without CutMix, as well as SWAT+ where we finetuning the whole model rather than only the classifier on few-shot data in the second stage (cf. Section F). We highlight the best number in **bold** and underline the second best. Superscripts mark improvements compared to previous state-of-the-art CLAP [59].

shots	strategy	methods	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet	average
0	prompting	OpenCLIP [10]	8.5	68.3	17.9	50.1	49.2	91.0	82.7	83.6	67.2	57.6
	retrieval-augmented	REAL-Prompt [43]	51.2	76.0	19.4	51.2	56.7	91.0	82.8	84.4	67.6	64.5
		REAL-Linear [43]	57.1	80.3	29.2	46.8	60.3	91.4	83.3	85.5	69.8	67.1
4	adapter-based	CrossModal-LP [36]	37.7	90.1	27.9	74.8	62.4	90.6	82.2	85.6	67.8	68.8
		CLAP [59]	40.0	91.0	29.9	76.7	64.6	88.9	80.4	86.8	66.9	69.5
	finetuning-based	FSFT (ours)	57.7$^{+17.7}$	93.6$^{+2.6}$	33.0$^{+3.1}$	**85.5**$^{+8.8}$	**69.1**$^{+4.5}$	91.4$^{+2.5}$	81.9$^{+1.5}$	86.1$^{-0.7}$	67.4$^{+0.5}$	74.0$^{+4.5}$
		FSFT w/ CutMix (ours)	58.8$^{+18.8}$	93.4$^{+2.4}$	33.4$^{+3.5}$	83.4$^{+7.7}$	68.6$^{+4.2}$	91.8$^{+2.9}$	82.7$^{+2.3}$	87.0$^{+0.2}$	67.8$^{+0.9}$	74.1$^{+4.6}$
		SWAT (ours)	69.2$^{+29.2}$	93.8$^{+2.8}$	**66.5**$^{+36.6}$	84.2$^{+8.5}$	62.6$^{-2.0}$	92.9$^{+4.0}$	83.3$^{+2.9}$	85.2$^{-1.6}$	70.6$^{+3.7}$	78.7$^{+9.2}$
		SWAT+ (ours)	**70.5**$^{+30.5}$	**96.0**$^{+5.0}$	64.5$^{+34.6}$	84.4$^{+7.7}$	64.7$^{+0.1}$	**93.4**$^{+4.5}$	**83.9**$^{+3.5}$	**88.5**$^{+1.7}$	**71.8**$^{+4.9}$	**79.7**$^{+10.2}$
8	adapter-based	CrossModal-LP [36]	49.4	93.6	32.5	81.8	67.8	90.9	82.9	87.4	68.0	72.7
		CLAP [59]	49.1	93.4	36.1	79.0	67.7	89.6	81.5	88.4	68.5	72.6
	finetuning-based	FSFT (ours)	61.9$^{+12.8}$	96.6$^{+3.2}$	39.6$^{+3.5}$	**90.9**$^{+11.9}$	73.3$^{+6.6}$	91.4$^{+1.8}$	82.0$^{+0.5}$	87.8$^{-0.6}$	69.4$^{+0.9}$	77.0$^{+4.4}$
		FSFT w/ CutMix (ours)	63.0$^{+13.9}$	96.4$^{+3.0}$	42.9$^{+6.8}$	90.3$^{+11.3}$	**73.5**$^{+6.8}$	92.1$^{+2.5}$	83.2$^{+1.7}$	89.6$^{+1.2}$	69.8$^{+1.3}$	77.9$^{+5.3}$
		SWAT (ours)	71.4$^{+22.3}$	96.5$^{+3.1}$	**69.1**$^{+33.0}$	88.8$^{+4.6}$	66.3$^{-1.4}$	93.2$^{+3.6}$	83.8$^{+0.5}$	87.2$^{-1.2}$	71.5$^{+3.0}$	80.9$^{+8.3}$
		SWAT+ (ours)	**73.2**$^{+24.1}$	**98.2**$^{+4.8}$	67.3$^{+31.2}$	88.9$^{+9.9}$	68.5$^{+0.8}$	**93.9**$^{+4.3}$	**84.3**$^{+2.8}$	**90.7**$^{+2.3}$	**73.2**$^{+4.7}$	**82.0**$^{+9.4}$
16	adapter-based	CrossModal-LP [36]	57.7	96.5	38.9	84.5	73.3	90.7	83.3	88.8	68.0	75.7
		CLAP [59]	56.9	95.2	42.4	82.2	71.4	90.3	82.3	89.8	70.0	75.6
	finetuning-based	FSFT (ours)	66.3$^{+9.4}$	98.0$^{+2.8}$	45.6$^{+3.2}$	94.1$^{+11.9}$	75.8$^{+4.4}$	91.5$^{+1.2}$	82.5$^{+0.2}$	89.7$^{-0.1}$	70.2$^{+0.2}$	79.3$^{+3.7}$
		FSFT w/ CutMix (ours)	67.3$^{+10.4}$	98.2$^{+3.0}$	51.2$^{+8.8}$	**94.2**$^{+12.0}$	**76.1**$^{+4.7}$	92.3$^{+2.0}$	84.0$^{+1.7}$	91.3$^{+1.5}$	72.1$^{+2.1}$	80.7$^{+5.1}$
		SWAT (ours)	73.9$^{+17.0}$	98.2$^{+3.0}$	**72.6**$^{+30.2}$	93.0$^{+10.8}$	69.0$^{-2.4}$	93.3$^{+3.0}$	84.4$^{+2.1}$	89.0$^{-0.8}$	72.3$^{+2.3}$	82.9$^{+7.3}$
		SWAT+ (ours)	**75.0**$^{+18.1}$	**99.0**$^{+3.8}$	69.8$^{+27.4}$	93.0$^{+10.8}$	72.5$^{+1.1}$	**94.1**$^{+3.8}$	**85.0**$^{+2.7}$	**92.3**$^{+2.5}$	**74.2**$^{+4.2}$	**83.9**$^{+8.3}$

proper filtering on the retrieved data significantly boosts the performance of SWAT, allowing it to outperform CLAP [59]. We also find filtering improves SWAT on other datasets, including Semi-Aves, Flowers, and EuroSAT (cf. Table 10).

Moreover, the improved SWAT still underperforms our few-shot finetuning (FSFT) on DTD datasets. We hypothesize that the discrepancy is because of DTD's strict data collection rules, which include only images that are almost entirely filled with a texture [11]. In contrast, the retrieved images often have only part of the region depicting the texture (Fig. 11 and 12). This suggests future work to explore better retrieval or filtering methods to find images that are better aligned with downstream distribution, e.g., by referring to the data collection rules provided in the data annotation guidelines. We explore different retrieval methods in Section D below.

Table 9. **Comparison of SWAT's performance with prior state-of-the-art FSR method CLAP [59].** SWAT underperforms CLAP on DTD and Cars datasets due to the significant domain gaps. However, with proper filtering on retrieved images (by keeping the top-10 retrieved images for each class that are ranked by prompt-to-caption or T2T similarity and discarding others), SWAT outperforms CLAP. We show results of different retrieval sizes in Table 17. Subscripts mark the performance difference compared with CLAP.

dataset	methods	4-shot	8-shot	16-shot
DTD	CLAP [59]	63.0	66.4	69.9
	SWAT	58.3$^{-2.0}$	62.6$^{-3.8}$	66.3$^{-3.6}$
	SWAT+filtering	63.5$^{+0.5}$	69.1$^{+2.7}$	72.9$^{+3.0}$
Cars	CLAP [59]	84.9	86.1	87.8
	SWAT	81.1$^{-3.8}$	83.5$^{-2.6}$	85.4$^{-2.4}$
	SWAT+filtering	83.5$^{-1.4}$	86.8$^{+0.7}$	88.6$^{+0.8}$

D. Analysis of Retrieval and Filtering Methods

Retrieval-augmented learning has been extensively studied for zero-shot recognition [37, 43, 67]. Previous work [37] utilizes text-to-text (T2T) or text-to-image (T2I) similarity to retrieve images relevant to each downstream concept. However, as noted by [43], such similarity-based retrieval requires significant storage for downloading all the source images (e.g., >10TB for LAION-400M) and high compute costs for computing image and text features (>250 T4 GPU hours). In addition, [67] points out the challenge of threshold selection in similarity-based retrieval: setting it too low includes irrelevant images, which can negatively impact training. Moreover, the proper threshold varies for different concepts, making it infeasible to search at scale. Given the above limitations, in this study, we adopt the *string-matching-based retrieval* by [43], detailed in the following two steps.

Step 1: String Matching with Synonyms. We use string matching to retrieve images whose captions contain any of the downstream concepts' synonyms. This circumvents the

Table 10. **Comparison of SWAT using different retrieval methods.** We conduct experiments on six datasets by first conducting string matching following [43] to download images whose captions contain any of the concepts' synonyms, then ranking the images using different text (few-shot concepts or database captions) and image (database images or few-shot images) features for selecting the images most relevant to downstream concepts. The top-ranking 500 images for each class are selected for running SWAT with 16 few-shot data. Results show that despite all methods outperforming random sampling by <1% in average accuracy, no single method is the best for all datasets. We highlight the best number in **bold** and underline the second best. We further explore adding text-to-image filtering before text-to-text ranking to remove noisy images with image-to-FS-concept similarity of less than 0.25. Results show that T2I filtering improves SWAT's performance significantly, especially for the DTD dataset (6% improvement). We show examples of T2I filtered images in Fig. 13.

retrieval/ranking method	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Cars	average
random sampling	62.8	96.0	62.2	92.6	64.9	84.7	77.2
text-to-text: FS-concept & DB-caption	**63.4**	96.4	62.7	93.0	65.8	85.4	77.8
image-to-image: FS-image & DB-image	63.0	**97.1**	62.8	92.7	64.9	84.9	77.6
image-to-text: FS-image & DB-caption	63.2	96.8	62.8	**93.4**	**66.7**	**86.9**	**78.3**
image-to-text: FS-concept & DB-image	62.9	96.8	**63.3**	**93.4**	65.7	83.7	77.6
text-to-image filtering (0.25) + text-to-text	63.8	97.5	62.6	93.7	71.6	85.8	79.2

large storage cost, as now we only need to download the text (60GB for LAION texts) for string matching and then the images with matching captions (50GB for all nine datasets). Additionally, [43] shows that using concept synonyms helps retrieve diverse images which benefits retrieval-augmented learning.

Step 2: Selection by Ranking. To select images that are most relevant to downstream concepts, we rank the retrieved images based on prompt-to-caption (T2T) similarities and select the top-ranking 500 images for each downstream concept. We compare different ranking methods using text (image captions or downstream concepts) and image (pre-training images or few-shot images) features in Table 10. The results show that, despite all ranking methods outperforming the random sampling, no single ranking method is the best across all datasets. This suggests future work to design retrieval methods customized to each downstream task.

T2I Filtering Improves SWAT Performance. To explore better retrieval methods, we follow the practice in the curation of the LAION dataset to apply text-to-image (T2I) filtering, excluding noisy retrieved images with T2I (few-shot concepts and retrieved images) similarities below 0.25. Despite that adding T2I filtering increases the imbalance of retrieved data, it notably improves SWAT's performance (cf. Table 10), especially on the DTD dataset (>6%). We show examples of T2I-filtered noisy images in Fig. 13. This suggests future retrieval methods to explore better filtering techniques. By default, our SWAT does not apply T2I filtering as post-processing, because determining a proper threshold for each class requires a large validation set which is not allowed in our realistic FSR setup.

E. Analysis of Data Augmentation Methods

We show examples of various mixed sample data augmentation (MSDA) methods in Fig. 6 and compare their performance using SWAT across five datasets (Semi-Aves, Flowers, Aircraft, EuroSAT and DTD) in Table 11. Results show that CutMix performs the best with minimal computation overhead, while SaliencyMix [64] performs similarly but incurs significant overhead due to the extraction of saliency maps.

Table 11. **Comparison of using different Mixed Sample Data Augmentation (MSDA) methods in SWAT.** Compared with no mixing, all mixing methods increase accuracy by 1-2%. MixUp [77] slightly underperforms other CutMix variants, likely because it creates unnatural artifacts that could confuse the model [76]. We also find that randomly applying CutMix regardless of few-shot and retrieved images performs better than strictly cutting few-shot patches and pasting them into retrieved images (CutMix-strict), likely because doing so limits the diversity of data augmentation. By default, SWAT uses CutMix [76], which achieves the best performance and low computation overhead among all the compared MSDA methods. **Bold** and underlined numbers mark the best and second best numeric metrics; superscripts denote improvements over no mixing. See visual examples of different MSDA methods in Fig. 6.

MSDA method	compute overhead	mean accuracy of five datasets		
		4-shot	8-shot	16-shot
No mixing	None	68.3	71.9	75.6
MixUp [77]	Low	69.1$^{+0.8}$	73.0$^{+1.1}$	76.6$^{+1.0}$
SaliencyMix [64]	High	70.1$^{+1.8}$	**74.4**$^{+2.5}$	77.7$^{+2.1}$
CMO [45]	Med	69.9$^{+1.6}$	74.1$^{+2.2}$	77.1$^{+1.5}$
ResizeMix [47]	Med	69.6$^{+1.3}$	74.1$^{+2.2}$	77.2$^{+1.6}$
CutMix-strict	Med	70.1$^{+1.8}$	73.8$^{+1.9}$	77.6$^{+2.0}$
CutMix [76]	Low	**70.5**$^{+2.2}$	74.2$^{+2.3}$	**77.8**$^{+2.2}$

Impact of Mixing Ratio. We further explore the impact of the mixing ratio between retrieved and few-shot data within a batch when applying CutMix augmentation. Results in Fig. 7 shows that SWAT achieves the best performance when applying a "natural ratio" by combining retrieved data and few-shot annotated data without sophisticated resam-

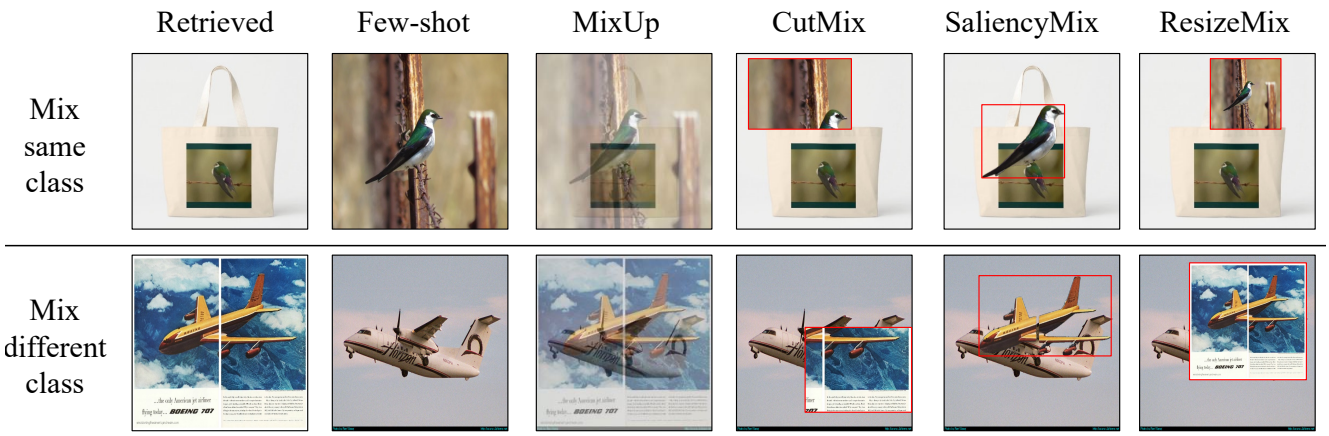

Figure 6. **Examples of different mixed sample data augmentation (MSDA) methods.** We show two examples where the first row shows mixing the retrieved and few-shot images from the same class in Semi-Aves dataset [61], and the second row shows mixing images from different classes in the FGVC-Aircraft dataset [39]. These MSDA methods encourage the model to learn from small discriminative parts of the object or details in the background (e.g. part of a bird or airplane), thereby improving the performance. Compared to CutMix [76] and its variants (SaliencyMix [64], ResizeMix [47]), MixUp [77] augments data by simply interpolating two images, which may create unnatural artifacts that could confuse the model [76].

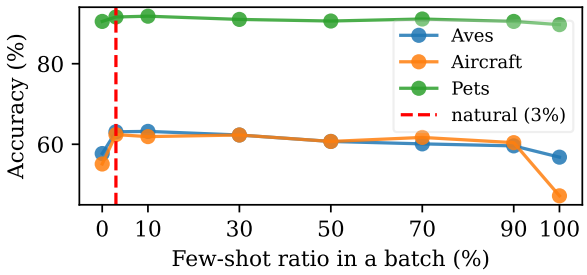

Figure 7. **Comparison of final accuracy with varying few-shot ratio in a batch.** Our SWAT adopts a "natural ratio" by combining retrieved data and few-shot annotated data without sophisticated resampling methods. The natural ratio is 3%, meaning 3% data in each batch is from the few-shot data. Results show that the "natural ratio" (red dashed line) performs better than either increasing the ratio (which reduces data diversity) or decreasing it (which increases domain gap).

Table 12. **Comparison of ImageNet accuracy and training time cost of different stage-2 training strategies.** We experiment by finetuning the stage-1 trained model on 16-shot data from ImageNet following different training strategies. Results show that finetuning only the classifier (as done in SWAT) improves the rare class accuracy significantly more than finetuning the visual encoder only. In addition, the training time cost of retraining the classifier is much less than finetuning the visual encoder. Moreover, finetuning both the visual encoder and classifier achieves further improvement over SWAT, likely due to the insufficient representation learning in stage 1 with only 50 training epochs. We denote this scenario as **SWAT+** and report its performance across all datasets in Fig. 5, Table 7 and Table 8.

FT encoder	FT classifier	Avg	common	rare	time
acc after stage-1		67.1	68.3	56.1	
	✓	67.6$^{+0.5}$	68.3$^{+0.0}$	<u>61.2</u>$^{+5.1}$	**0.5** mins
✓		67.4$^{+0.3}$	68.5$^{+0.2}$	57.3$^{+1.2}$	15 mins
✓	✓	69.3$^{+2.2}$	70.1$^{+1.8}$	**62.0**$^{+5.9}$	15 mins

pling methods. This encourages future work to explore better mixed sample data augmentation methods.

F. Validating the Design of SWAT

Ablation of Stage-2 Training Strategy. To validate the design of stage-2 classifier retraining in SWAT, we compare the performance of different stage-2 training strategies in Table 12. Results show that retraining only the classifier achieves significantly larger accuracy improvement on rare classes than retraining only the visual encoder, validating its effectiveness in mitigating imbalanced distribution. Furthermore, we find that retraining both the visual encoder and classifier improves further over SWAT by 1∼2%. We hypothesize that this is due to the insufficient representation learning in stage 1 with only 50 training epochs (recall that we follow realistic evaluation protocol that do not use validation set to tune hyperparameters). A supporting evidence is found in Fig. 8 where we show that longer training in stage 1 generally yields better final accuracy. We denote this strategy as **SWAT+** and report its performance across all datasets in Fig. 5, Table 7 and Table 8. Considering the comparable performance and much less training time cost, we adopt classifier retraining for stage 2 in our SWAT.

Comparison with SOTA Finetuning Methods. Table 13 shows that our SWAT outperforms recent probing-based

Table 13. **Comparison of different finetuning methods.** We compare SWAT with state-of-the-art probing-based and finetuning-based methods using the same training data (a mix of retrieved and few-shot data). We experiment with the T2I-filtered retrieved data for each dataset (cf. Table 10). We use the same set of hyperparameters in Section B for all methods except using a larger batch size of 256 for FLYP following [17]. Results show that finetuning-based methods largely outperform probing-based methods, indicating the necessity of finetuning the visual encoder to learn better representation. In addition, ensembling the finetuned model with the zero-shot model (WiSE-FT with $\alpha = 0.5$ [69]) leads to much worse accuracy than standard finetuning, likely because the zero-shot OpenCLIP model struggles to recognize these fine-grained concepts [52]. Finally, SWAT outperforms other finetuning methods, validating its effectiveness in mitigating domain gaps and imbalanced distribution issues in retrieved data. We highlight the best number in **bold** and underline the second best.

method	Semi-Aves			Flowers			Aircraft			EuroSAT			DTD			mean accuracy		
(shots)	4	8	16	4	8	16	4	8	16	4	8	16	4	8	16	4	8	16
linear probing [49] ICML'24	49.8	52.4	54.4	86.9	89.4	92.8	34.6	35.8	38.2	68.0	78.2	82.4	61.7	65.5	68.9	60.2	64.3	67.3
CMLP [36] CVPR'23	49.2	51.9	53.6	87.0	89.3	92.9	34.1	35.4	37.8	70.1	79.4	83.5	61.3	64.8	68.6	60.3	64.2	67.3
REAL-Linear [43] CVPR'24	51.0	52.5	54.3	85.0	86.4	88.7	31.2	31.8	33.8	66.5	73.4	76.2	62.2	64.7	67.4	59.2	61.8	64.1
standard FT [49] ICML'24	55.2	57.6	60.4	89.4	92.8	95.5	48.9	51.2	53.0	83.3	88.3	92.8	61.5	65.6	70.3	67.7	71.1	74.4
WiSE-FT [69] CVPR'22	51.7	53.2	56.1	82.1	84.6	87.0	32.2	33.2	34.0	77.4	85.2	87.4	**64.1**	66.7	69.4	61.5	64.6	66.8
FLYP [17] CVPR'23	56.0	57.7	59.6	88.1	91.1	94.4	47.9	49.7	51.2	75.4	83.3	90.6	63.1	67.4	70.3	66.1	69.2	72.6
SWAT (ours)	**58.6**	**61.3**	**63.8**	**91.0**	**94.7**	**97.5**	**55.5**	**58.1**	**62.6**	**84.6**	**89.2**	**93.7**	63.0	**67.6**	**71.6**	**70.5**	**74.2**	**77.8**

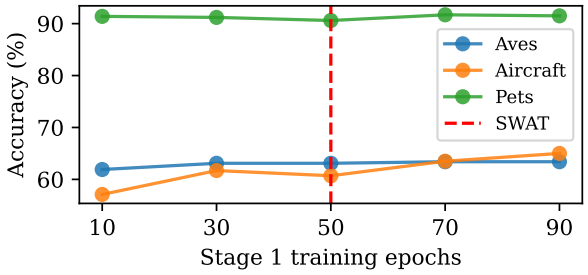

Figure 8. **Comparison of final accuracy with increasing stage-1 training epochs.** Results show that increasing stage-1 training epochs generally increases final accuracy slightly, without overfitting issues. This is likely due to the improved representation learning. We set stage-1 training epochs to 50 for all datasets, following the realistic FSR setup that does not tune hyperparameters using a large validation set.

or finetuning-based methods using the same retrieved and few-shot data. SWAT also outperforms recent ensembling-based [69] and contrastive finetuning [17] methods, highlighting the effectiveness of our proposed stage-wise training in mitigating domain gap and imbalanced distribution issues.

G. Further Analyses on SWAT

Impact of Stage-1 Training Epochs. We compare the final accuracy with a varying number of epochs for stage-1 end-to-end finetuning in Fig. 8. Results show that longer training generally yields better performance due to improved representation learning. Please note that our realistic FSR setup does not allow using a validation set to tune training epochs. Our paper sets the number of training epochs to 50 for all datasets (cf. Section B).

Table 14. **Comparison of classifier initialization methods in SWAT.** We compare the final test accuracy by initializing the classifier before stage-1 end-to-end finetuning in different ways. Initializing classifier weights with text embedding features leads to better performance than random initialization. [33] explains that using randomly initialized classifier weights to finetune the model can distort the features of pretrained model, leading to worse finetuning performance. Throughout this work, we use prompts in [43] to initialize classifier weights in SWAT. Subscripts mark the performance improvement compared with random initialization.

classifier initialization	mean accuracy of nine datasets		
	4-shot	8-shot	16-shot
random	72.7	75.1	77.5
text embedding [43]	$73.6^{+0.9}$	$76.1^{+1.0}$	$78.2^{+0.7}$

Classifier Initialization. We compare different classifier initialization methods for SWAT (Table 14). Results show that initializing with text embedding yields better performance than random initialization.

Retraining Classifier does not Overfit. In Fig. 9, we show the final test accuracy after retraining the classifier across varying epoch numbers. For all datasets, accuracy remains stable with increasing epochs. The small standard deviations across three runs with different random seeds confirm that stage-2 classifier retraining with few-shot data does not suffer from overfitting.

More Detailed Experimental Results. In addition. we show the detailed performance of classifier retraining for each dataset in Table 15. The rare classes of the Aircraft dataset show significant performance gains (>10%) after classifier retraining, demonstrating the efficacy of classifier retraining with few-shot data in mitigating domain gaps and imbalanced distribution. In addition, we include the detailed

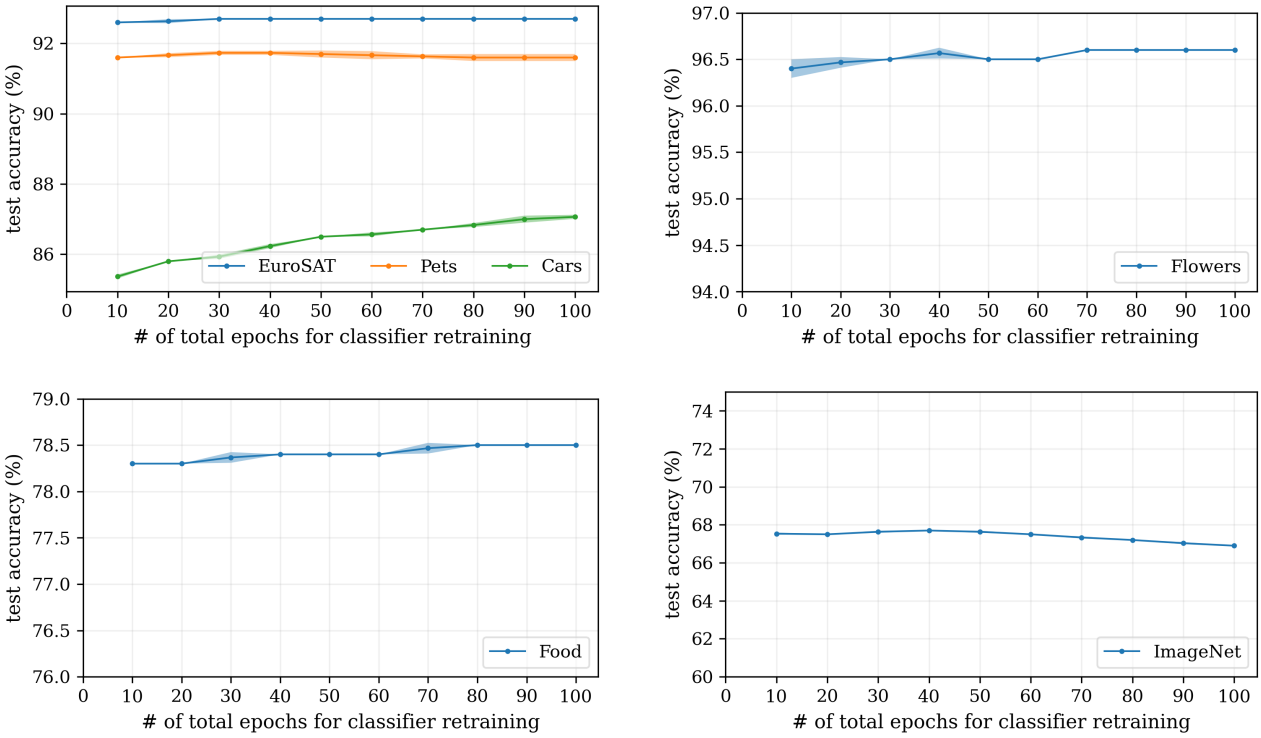

Figure 9. **Retraining the classifier on the few-shot data does not suffer from overfitting.** We show the final test accuracies by retraining the classifier on the few-shot data for different epoch numbers. For each dataset, we perform three runs of training with different random seeds. Results show that testing accuracy remains stable with more epochs and shows small standard deviations, indicating classifier retraining does not suffer from overfitting.

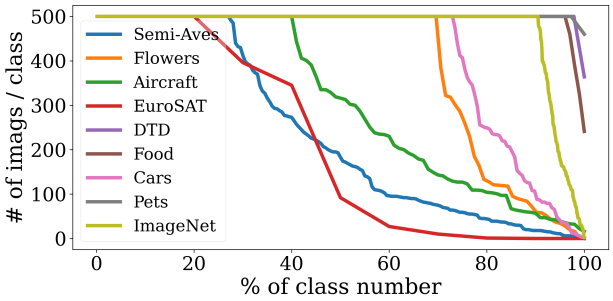

Figure 10. **Retrieved data follows imbalanced distribution for all nine datasets.** The retrieved data for ImageNet, Food, DTD, and Pets datasets are less imbalanced than other datasets, likely because the concepts from these datasets naturally appear more frequently on the Internet [43].

ablation of SWAT components on each dataset in Table 16. Results show that applying CutMix [76] and classifier retraining effectively mitigate the domain gap and imbalanced distribution problem, verifying the design of SWAT.

H. Analysis of Retrieved Data

Imbalances of Retrieved Data. We show the imbalanced distribution of retrieved data for all nine datasets in Fig. 10. We report the total number of retrieved images per dataset with increasing retrieval size (images per class) in Table 18. With increasing retrieval size, the total number of retrieved images increases less significantly due to the limited presentation of many downstream concepts in the pretraining datasets (e.g. LAION [55, 56]). To address this issue, we suggest future work to retrieve relevant images from diverse data sources, e.g. other datasets or the Internet [34]. Fig. 11 shows more examples of retrieved images for each dataset.

Impact of Retrieval Sizes. Additionally, we compare SWAT's performance on different retrieval sizes in Table 17. Results show that SWAT saturates at 500 images per class for 4-shot and 8-shot cases and at 300 for 16-shot. Notably, for Flowers, EuroSAT, DTD, and Cars, retrieving only 10 images per class yields the best results, likely due to improved data balance and the exclusion of noisy images (Fig. 13). Future work can study how to enhance the balance and quality of retrieved data.

Table 15. **Detailed comparison of the accuracy of common and rare classes after stage-1 and stage-2 training.** We define the rare classes as the 10% least frequent classes in retrieved data and the rest as the common classes. Results show that stage-2 classifier retraining clearly improves recognition accuracy on both common and rare classes in all methods, including finetuning on few-shot data only, on retrieved data only, and on mixed data with or without CutMix data augmentation. Importantly, the improvement on rare classes is more significant than that on common classes, confirming that classifier retraining mitigates the issue of imbalanced distribution in the retrieved data. We report the accuracy for each dataset using 16-shot examples.

data used in stage-1: finetuning	stage	classes	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet	average
few-shot only (balanced)	stage-1 finetuning	common	56.1	97.6	43.2	94.9	73.5	90.6	78.8	88.6	68.0	76.8
		rare	63.5	100.0	34.6	87.1	76.7	84.9	74.8	84.4	56.8	73.6
		average	56.9	97.4	42.4	94.1	73.9	90.0	78.4	88.0	66.9	76.4
	stage-2 classifier retraining	common	56.0	97.5	47.3	95.0	72.9	90.0	78.9	88.5	67.1	77.0
		rare	63.8	100.0	46.4	87.2	76.7	86.9	74.5	83.3	57.3	75.1
		average	56.8	97.4	47.2	94.3	73.3	89.7	78.4	87.9	66.1	76.8
retrieved only (imbalanced)	stage-1 finetuning	common	56.2	84.4	52.5	30.4	52.4	90.8	76.0	78.1	62.5	64.8
		rare	15.0	54.4	10.2	0.0	61.1	85.5	73.3	51.8	46.4	44.2
		average	52.1	81.6	48.3	27.9	53.3	90.3	75.7	75.3	60.9	62.8
	stage-2 classifier retraining	common	60.0	90.2	57.5	32.2	54.6	90.9	76.8	82.9	64.7	67.8
		rare	36.9	77.8	33.7	0.0	62.8	86.6	74.2	66.8	58.8	55.3
		average	57.7	88.6	55.1	29.4	55.4	90.5	76.6	81.2	64.1	66.5
retrieved + few-shot	stage-1 finetuning	common	61.4	94.6	57.4	93.4	62.3	91.4	77.9	81.8	64.8	76.1
		rare	49.4	96.8	26.2	87.5	69.4	87.4	75.9	68.8	52.7	68.2
		average	60.2	94.7	54.3	92.8	63.1	91.0	77.7	80.3	63.6	75.3
	stage-2 classifier retraining	common	61.6	95.4	60.6	93.4	62.8	91.3	78.0	84.0	65.7	77.0
		rare	52.2	98.0	44.3	87.5	68.9	87.1	76.0	73.1	57.6	71.6
		average	60.6	95.4	59.0	92.8	63.5	91.0	77.8	82.8	64.9	76.4
retrieved + few-shot w/ CutMix	stage-1 finetuning	common	63.7	96.4	61.3	93.4	64.8	91.5	78.3	83.9	68.3	78.0
		rare	55.8	100.0	34.7	83.9	72.2	89.2	77.4	78.0	56.1	71.9
		average	62.9	96.3	58.7	92.5	65.6	91.3	78.2	83.2	67.1	77.3
	stage-2 classifier retraining	common	64.0	96.4	63.7	93.7	65.6	91.9	78.4	86.1	68.3	78.7
		rare	54.9	100.0	50.9	82.0	72.2	88.6	77.5	79.9	61.2	74.1
		average	63.1	96.4	62.4	92.6	66.3	91.6	78.3	85.4	67.6	78.2

I. Code and Instructions

We release open-source Python code at https://github.com/tian1327/SWAT.

Requirements. Running our code requires some common packages. We installed Python and most packages through Anaconda. A few other packages might not be installed automatically, such as clip, open_clip_torch, img2dataset, torchvision, and PyTorch, which are required to run our code. We provide detailed instructions for building the environment in file `ENV.md`. Below are the versions of Python and PyTorch used in our work.

- Python version: 3.8.19
- PyTorch version: 2.0.1

We suggest assigning >50GB storage space and >5GB GPU RAM to reproduce our experiments.

License. We release open-source code under the MIT License to foster future research in this field.

Instructions. We provided detailed step-by-step instructions for running our code in the following markdown files.

- `ENV.md`
  Create a conda environment and install the required packages.
- `DATASETS.md`
  We provide detailed steps for setting up the benchmarking datasets and sampling few-shot data from the official training sets with three random seeds.
- `RETRIEVAL.md`
  We provide step-by-step instructions on how to use string-matching [43] to retrieve relevant images from OpenCLIP's pretraining dataset LAION-400M [55, 56]. Examples of different ranking and filtering methods for selecting the images that are most relevant to downstream concepts are also provided.
- `README.md`
  We provide instructions on how to run the provided code for few-shot finetuning (FSFT) and SWAT. In addition, we provide guidelines on how to reproduce the baseline methods Cross-Modal Linear Probing [36] and CLAP [59].

Table 16. **Ablation study on important components in our SWAT.** We show the detailed performance improvements by each component for each dataset in our SWAT. Finetuning on simply combined retrieved and few-shot data underperforms finetuning solely on few-shot data (8-shot and 16-shot, with or without CutMix), due to the large domain gap and imbalanced distribution in retrieved data. However, further applying CutMix and classifier retraining improves the test accuracy significantly, confirming their effectiveness in mitigating the domain gap and imbalanced distributions. We also compare the performance of few-shot finetuning with and without CutMix data augmentation. The results indicate more few-shot data yields more improvements, likely due to stronger data augmentation.

shots	method	finetune model	retrieve data	apply CutMix	retrain classifier	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet	average
4	CLAP [59]					34.0	90.1	28.0	74.7	63.0	87.0	76.7	84.9	64.0	66.9
	FTFS (ours)	✓				47.5	92.5	27.9	81.6	66.6	88.7	75.8	81.5	62.3	69.4
	FTFS (ours)	✓		✓		48.0	92.2	28.8	81.8	66.7	89.0	76.1	82.5	62.4	69.7
		✓	✓			54.7	89.7	50.1	80.2	56.3	90.7	76.4	76.9	61.8	70.8
		✓	✓	✓		57.9	90.2	53.8	83.2	58.7	91.0	77.2	79.8	65.2	73.0
	SWAT (ours)	✓	✓	✓	✓	58.5	90.6	55.7	83.2	58.3	91.3	77.3	81.1	65.8	73.5
8	CLAP [59]					42.9	92.9	33.6	77.4	66.4	87.8	77.5	86.1	65.6	70.0
	FTFS (ours)	✓				51.2	95.4	33.1	90.3	71.0	89.3	76.0	83.5	64.4	72.7
	FTFS (ours)	✓		✓		52.3	95.2	35.4	89.4	70.6	89.6	77.0	85.3	64.8	73.3
		✓	✓			57.3	91.9	52.4	87.0	59.2	91.1	76.8	78.9	62.5	73.0
		✓	✓	✓		60.6	93.7	55.7	89.1	61.8	90.8	77.6	81.3	65.8	75.2
	SWAT (ours)	✓	✓	✓	✓	60.8	94.1	59.1	89.2	62.6	90.8	77.5	83.5	66.6	76.0
16	CLAP [59]					49.2	94.8	39.1	81.7	69.9	88.4	78.5	87.8	67.1	72.9
	FTFS (ours)	✓				55.3	97.0	37.0	94.0	73.3	89.5	77.1	85.7	66.7	75.1
	FTFS (ours)	✓		✓		56.5	97.1	42.7	94.3	73.4	89.6	78.2	87.8	66.9	76.3
		✓	✓			60.2	94.7	54.3	92.8	63.1	91.0	77.7	80.3	63.6	75.3
		✓	✓	✓		62.9	96.3	58.7	92.5	65.6	91.3	78.2	83.2	67.1	77.3
	SWAT (ours)	✓	✓	✓	✓	63.1	96.4	62.4	92.6	66.3	91.6	78.3	85.4	67.6	78.2

Table 17. **Impact of retrieval size (number of images per class) on the performance of SWAT.** We show the performance of SWAT on each dataset using different numbers of retrieved images. We highlight the best number in **bold** and underline the second best. Importantly, we find that retrieving 10 images per class works best for Flowers, EuroSAT, DTD, and Cars datasets. This is probably because LAION-400M contains limited images that match these downstream concepts and simply retrieving more will include more noisy images and more imbalanced distributions, which hurt the training performance. We list the performance of the previous state-of-the-art few-shot recognition method CLAP [59] in the table for comparison with our SWAT.

shots	Retrieval size	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet	average
4	CLAP [59]	34.0	90.1	28.0	74.7	63.0	87.0	76.7	84.9	64.0	66.9
	10	52.4	**91.8**	37.0	**84.7**	**63.5**	89.3	75.9	**83.5**	64.8	71.4
	100	57.4	90.7	47.0	82.1	62.1	89.9	76.6	**83.5**	**66.1**	72.8
	300	**58.7**	91.4	54.1	82.2	59.3	91.1	77.1	81.7	65.9	73.5
	500	58.5	90.6	55.7	83.4	58.3	91.3	**77.3**	81.1	65.8	**73.6**
	1,000	58.3	89.6	**58.1**	84.1	57.7	**91.4**	76.2	81.1	65.2	73.5
8	CLAP [59]	42.9	92.9	33.6	77.4	66.4	87.8	77.5	86.1	65.6	70.0
	10	55.7	**95.2**	42.2	**90.0**	**69.1**	89.4	76.9	**86.8**	65.8	74.6
	100	59.2	94.6	49.9	88.6	65.2	90.2	77.2	85.3	67.0	75.2
	300	60.6	94.3	56.5	89.3	63.1	90.9	**77.6**	83.9	**67.3**	75.9
	500	**61.3**	94.1	59.1	88.7	62.6	**91.5**	**77.6**	83.5	66.6	**76.1**
	1,000	60.9	92.9	**60.6**	88.9	59.8	91.4	76.7	83.6	66.2	75.7
16	CLAP [59]	49.2	94.8	39.1	81.7	69.9	88.4	78.5	87.8	67.1	72.9
	10	58.4	**97.0**	48.6	**94.0**	**72.9**	89.6	78.5	**88.6**	66.9	77.2
	100	61.8	96.8	54.5	93.4	69.4	90.2	**78.6**	87.1	**67.9**	77.7
	300	63.2	96.8	60.8	93.1	67.0	91.3	**78.6**	86.0	67.8	**78.3**
	500	63.1	96.4	62.4	92.9	66.3	91.6	78.3	85.4	67.6	78.2
	1,000	**63.6**	96.4	**64.2**	93.0	63.0	**91.8**	77.2	85.6	67.2	78.0

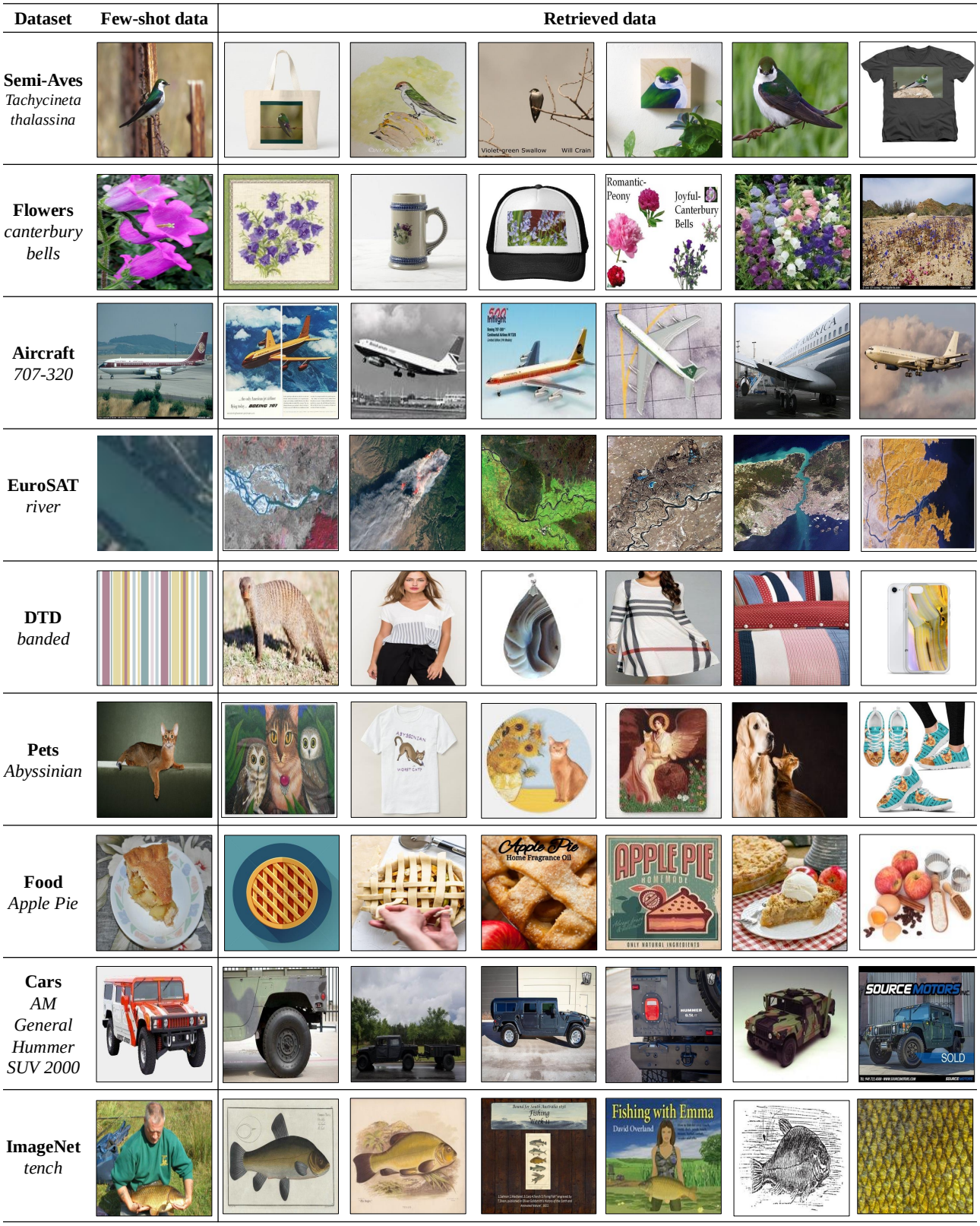

Figure 11. **Comparison of downstream few-shot data with retrieved pretraining images (from LAION-400M [55]) for nine fine-grained datasets**. We present more examples of retrieved images for randomly selected classes from each dataset. Compared to downstream few-shot images, the retrieved data exhibits diverse styles, backgrounds, resolutions, and even semantics, demonstrating significant domain gaps.

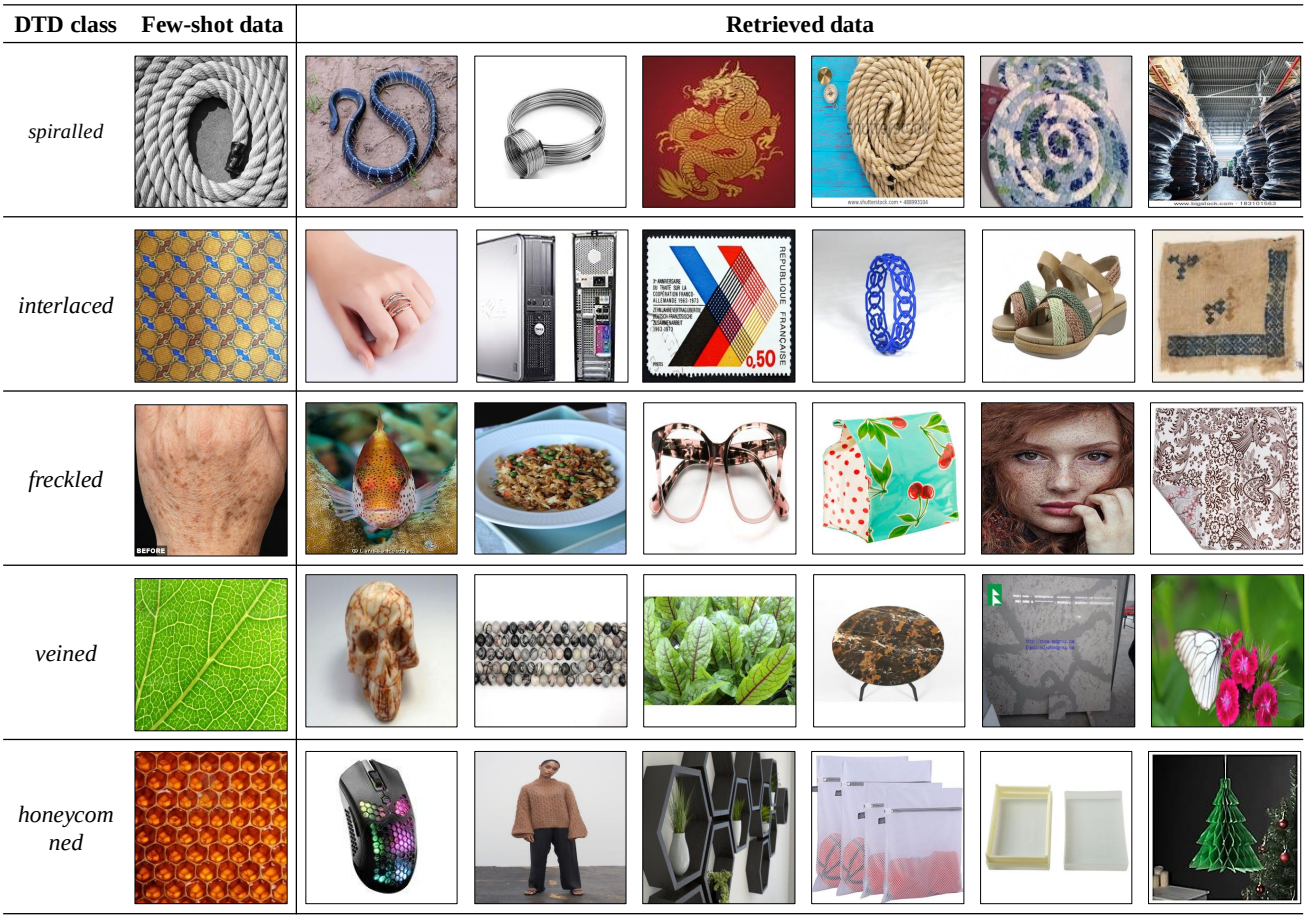

Figure 12. **Visual comparison between downstream DTD images and the retrieved images (from LAION-400M [55]) for various DTD concepts**. Clearly, a large domain gap exists between the two data resources regarding styles, backgrounds, semantics, etc. In addition, the retrieved images only have a partial region depicting the texture, contrasting to the few-shot images which are "almost entirely filled with a texture" according to DTD's strict data collection rules [11]. We suggest future work to explore better retrieval methods that are closely aligned with downstream data distribution, e.g., by referring to the data collection/annotation rules provided in the data annotation guidelines of a downstream task.

Table 18. **Total number of retrieved images for each dataset under different retrieval sizes.** With a larger retrieval size (number of retrieved images per class), we observe a diminished increase in the total number of retrieved images. This is because many downstream concepts have limited presence in the pretraining set (LAION-400M [55, 56]). See the imbalanced distribution of each dataset in Fig. 10.

images / class	Semi-Aves	Flowers	Aircraft	EuroSAT	DTD	Pets	Food	Cars	ImageNet
10	1,940	1,002	1,000	71	470	370	1,010	1,939	9,989
100	15,687	9,376	9,120	530	4,700	3,700	10,100	18,494	98,753
300	34,685	25,140	21,774	1,330	14,100	11,100	30,241	51,251	288,532
500	47,006	39,465	30,429	1,871	23,364	18,460	49,914	80,648	471,876
1,000	67,418	71,332	44,519	2,387	45,978	36,105	96,697	147,568	901,902

Dataset	Few-shot data	Noisy retrieved images (w/ captions) filtered by T2I thresholding (< 0.25)					
Semi-Aves *Tachycineta thalassina*							
T2I similarity		0.1575	0.1601	-0.0459	0.0344	0.1930	0.1363
synonyms:	*Violet-green Swallow*	Violet Green Swallow Nest with Four Eggs, *Tachycineta Thalassina*	NestWatch Focal Species Western North America *Violet green swallow*	Iittala BIRDS by TOIKKA *Violet Green Swallow*	I spent a bit more time with … *Violet Green Swallow* in the early morning	*Tachycineta thalassina* distribution map	Violet green Swallows (*Tachycineta thalassina*)
Flowers *canterbury bells*							
T2I similarity		0.0995	-0.0136	0.0885	0.0669	0.0350	0.2150
synonyms:	*bellflowers*	Vase Of Peonies And *Canterbury Bells* Poster by Albert Williams	Case (Inrō) with Design of … Chinese *Bellflowers* in a Pot (reverse)	Portrait of Kitten Among Dwarf Roses and *Bellflowers*	Look at this arch! Pampas grass, protea, garden roses, *canterbury bells*	*Canterbury Bells* Wine Red Long Sleeve Dress at Lulus.com!	Ladybug And *Bellflowers* iPhone Case by Nailia Schwarz
Aircraft *707-320*							
T2I similarity		0.1286	0.1883	0.0424	0.2269	0.2344	0.0691
synonyms:	*Boeing 707-320* *Boeing 707* *B707*	*Boeing 707* Stainless Steel Strap Watches Pilot Eyes Store	John Travolta is pilot of his very own jumbo jet, a 1964 *Boeing 707* 100 series	Pan am *Boeing 707* recycled cufflinks	*BOEING 707* AIRPLANE LAPEL TAC PIN PILOT	Tintin Lavion Le *Boeing 707* Quantas N°15 29535 (2014)	*B707* No FTAA, Fair Trade Not Free Trade Button
EuroSAT *river*							
T2I similarity		0.0925	0.0387	0.0853	0.2351	0.0758	0.0352
synonyms:	*River*	The wonderful sunset at our camping site on the Zambesi *river*	A bald eagle migration study of eagles that visit the Chilkat *River*	A Buddhist temple (wat) on the west bank of the Chao Phraya *River*	Flex Maslan kayakfari … canoe shark *river* slough camp	High *River* Denture & Implant Clinic	Acheson Business Park, Edson, Whitecourt, Peace *River*
DTD *banded*							
T2I similarity		0.0852	0.1139	0.1665	0.1318	0.0922	0.1589
synonyms:	*Vertical striped*	Lenox Opal Platinum *Banded* Bone China Sugar Bowl	3/5vx1000 optibelt oem equivalent cogged wedge *banded* v belt	Cinch Men's FR Lightweight *Vertical Striped* Long Sleeve Work Shirt	Colorful, *Banded* (Rainbow) Fluorite Skull For Sale	*Banded* mongoose juvenile sleeping on adult escort	Stockfoto: Three *banded* plover

Figure 13. **Examples of noisy retrieved images (from LAION-400M [55]) filtered by T2I thresholding.** We show that string-matching-based retrieval (by searching image captions that contain any *concept synonyms*) can retrieve noisy images that could compromise the learning of downstream concepts, e.g., the bird eggs or the distribution map of bird species (first row). Using text-to-image (T2I) filtering helps remove such noisy images and improve the performance of SWAT (Table 10). We choose a T2I threshold of 0.25 for our experiment, similar to that used in the curation of LAION [55, 56]. We highlight the T2I cosine similarity and concept synonyms for each image.