Concept-Aware Batch Sampling Improves Language-Image Pretraining

Supplementary Material

A. DATACONCEPT Curation: Further Details

A.1. Vocabulary Construction

Scaling Concept Vocabulary: We scale up the tag generation pipeline of RAM++ (Recognize Anything) [41, 92] by incorporating more long-tailed concepts. In the original work, RAM++ extracts the top $4,585$ concepts by parsing 14 million sentences from their pool of pretraining datasets and then extracting tags using a SceneGraph Parser [84], hence attempting to focus on more common concepts. However, our work focuses more on open-vocabulary recognition and localization, hence we scale up the concept vocabulary to include objects that may be found in-the-wild in image-text pretraining datasets. We include the concepts collected in [78] as well as 200 classes from the rare classes subset of OpenImages [47]. Finally, we also adopt and filter the vocabulary pool from V3Det [81], a state-of-the-art open-vocabulary dataset which observes and encodes the relationship between categories by defining a hierarchy tree of concepts.

Systematic Concept Curation and Redundancy Resolution: Curating this concept pool comes with redundancies, which need to be systematically resolved. We first establish a set of pre-defined heuristics that comprise grounds for removing concepts from the vocabulary. Then we first automate the concept removal process, followed by a manual inspection of the collected vocabulary to remove concepts that violate these heuristics. This ensures a very thorough curation, which we detail below:

1. **Morphological Redundancies.** We perform a normalization step to remove morphological variants of the same concept (*e.g* singular and plural forms) into a single entity using lemmatization. In practise, we canonicalize noun such that entries like `dogs`, `dog`, and `dog's` are collapsed to the same lemma. Addressing morphological redundancies early in the pipeline reduced spurious multiplicity caused by simple variations.
2. **Syntactical Redundancies**. We identify spelling/spacing artifacts and remove them if they are duplicated (`" cat"` and the correct `"cat"`). This normalization is deterministic and involves collapsing repeated whitespace, lowercasing capital letters, and replacing underscores with spaces. This step reduces accidental duplicates which were caused by formatting differences, occurring due to the collection of concepts from different sources, as highlighted above. Since the following heuristics involve embedding computations, this step prevents unnecessary computations.
3. **Semantic Redundancies**. We remove semantic redundancies using WordNet (formalized through synsets) to detect synonyms in addition to semantic embeddings of concepts using a pretrained SentenceTransformer model [69]. This phase is conservative, we only want to remove near identical concepts (such as `tv` and `television`) rather than loosely related terms. This design choice is particularly important as we deal with a lot of concepts that could be considered similar in a relaxed definition (such as different editions of car models). WordNet synsets serve as an initial lightweight signal for detecting synonyms and the SentenceTransformer embeddings are used for more robust coverage. We use `all-MiniLM-L6-v2` to compute vector embeddings, followed by comparing concept pairs using the cosine similarity and only merging/removing concepts if the similarity is higher than 0.95. This ensures only near-identical concepts are collapsed (for example, British and USA English spellings of the same concept) and separate but related concepts (for example `hedgehog`/`porcupine` and `crayfish`/`spiny lobster` are preserved.
4. **Unsafe Concepts**. We identify unsafe concepts (*e.g.* racially motivated concepts like `white man` and `black man`) through thorough manual inspection and remove them. Additionally, we build a lightweight safety classifier by encoding a curated list of race-related and NSFW terms using the SentenceTransformer model from before. A concept is flagged as unsafe if the cosine similarity between the concept and the encoded list of unsafe terms exceeds 0.7 for race-related terms and 0.65 for NSFW terms. These thresholds were determined iteratively to prevent false positives (for example `black cat`).

With these steps, we obtain our final concept vocabulary of $18,884$.

A.2. Object Tagging

Motivation. Previous attempts to annotate pretraining datasets have used object tagging to return a list of probable objects in a sample, above a specified threshold. For example, Udandarao et al. [78] used RAM++ [41, 92] to annotate visual concepts in many large image-text datasets. However, as discussed in Sec. 2, the expanded vocabulary (from $4,029$ to $19,261$) introduces miscalibrations and overestimations in the model predictions. For example, abiding by the confidence threshold of 0.7 image resolution of (384,384) from Udandarao et al. [78], we note that RAM++ tends to overestimate classes when the vocabulary is expanded. This arises from the increased semantic similarity among real-world concepts in the visual space, as a factor of a large vocabulary. An increase in the hierarchy for common and long-tailed classes (there are several sub-species of snakes in the vocabulary as we see in Fig. 5) is to be expected with an increase in the vocabulary of visual concepts, which leads to inherent uncertainty of making predicting for images that induce visual uncertainty.

Optimal RAM++ Threshold. One simple solution is to increase the threhsold, which highlights the flexibility of open-set image tagging - the RAM++ model easily adapts to a larger vocabulary despite being trained on $\sim 4,000$ concepts.As a sanity check, we apply RAM++ under three different confidence thresholds: 0.7, 0.75, and 0.8, still processing each image at a resolution of $(384, 384)$. We choose this resolution as it is the default chosen by RAM++. This multi-threshold setup allows us to explicitly study how sensitive the predicted tag set is to the choice of threshold, and to quantify the extent to which miscalibration persists even under stricter filtering regimes. Note that the tags generated at a threshold of 0.75 is a strict subset of 0.7 and tags generated at a threshold of 0.8 is a strict subset of 0.75 and 0.7.

Increasing the confidence threshold to 0.75 still results miscalibrations in some form (see Fig. 5), although some low-confidence noise seems to be removed. It is to be noted that increasing the threshold to 0.8 significantly increased the proportion of samples with no generated tags. Hence, we opt for using 0.75 as our final threshold for object tagging using RAM++.

Why Object Detection? Simply generating concept tags can lead to mistakes as highlighted above, especially for images with high levels of visual uncertainty. Tagging lacks spatial grounding and cannot differentiate between multiple instances or object-level relationships. Additionally, concept tags injects only one form of added metadata: other tasks like object detection can add richer and more valuable fine-grained information into these large datasets. Hence, we advocate for the conducting an additional step to annotate image-text pretraining datasets.

	RAM++ Threshold 0.7	0.75	0.8
	boa constrictor | iguana | snake | African chameleon | cobra | fence | cage | zoo | museum | animal | closeup | picture | display | close-up | python | display device | burmese python | brown snake | crotalus oreganus | pantherophis guttatus | crotalus ornatus | moa | barosaur | hoop snake | hognose snake | leaf-nosed snake | horseshoe whipsnake | masticophis lateralis | sonoran whipsnake | chicken snake | indian rat snake | glossy snake | viperine grass snake | banded sand snake | black-headed snake | sonoran lyre snake | carpet snake | reticulated python | indian python | rock python | amethystine python | black mamba | death adder | notechis scutatus | taipan | vipera berus | puff adder | gaboon viper | horned viper | crotalus adamanteus | western diamondback | rock rattlesnake | snake charmer	boa constrictor | iguana | snake | cobra | fence | zoo | animal | display | python | display device | hoop snake | hognose snake | horseshoe whipsnake | masticophis lateralis | sonoran whipsnake | chicken snake | glossy snake | viperine grass snake | banded sand snake | sonoran lyre snake | carpet snake | reticulated python | indian python | rock python | amethystine python | notechis scutatus | taipan | gaboon viper | horned viper | crotalus adamanteus | western diamondback	iguana | snake | cobra | hoop snake | hognose snake | horseshoe whipsnake | sonoran whipsnake | chicken snake | glossy snake | banded sand snake | sonoran lyre snake | carpet snake | reticulated python | indian python | rock python | amethystine python | taipan | gaboon viper | horned viper
	bicycle | man | road | white | guy | cyclist | ride | bike | race | shirt | road |helmet | bicycle helmet | biker | cycling | list | professional | wear | yellow | Bicycle model | bicycle-built-for-two | pedelec | tall bike | road bicycle | Road cycling	bicycle | man | guy | cyclist | ride | bike | race | | shirt | road | bicycle helmet | biker | yellow | Bicycle model | pedelec | tall bike | road bicycle	bicycle | man | guy | cyclist | ride | race | bicycle helmet | yellow | pedelec
	Highway or Road | street | motorbike | motorbikes | car | motorcycle | road | crowd | traffic | vehicle | parade | flag | ride | red | motorcyclist | drive | police | roadway | biker | carry | catch | city street | crowded | march | protester	street | motorbikes | car | motorcycle | road | crowd | traffic | vehicle | parade | flag | ride | roadway | city street | march	car | motorcycle | road | crowd | traffic | parade | flag | roadway
	fly | small white butterfly | gossamer-winged butterfly | drawing | butterfly | white | blue | flower | picture | beautiful | hydrangea | sit | hydrangea macrophylla | butterfly flower | celastrina | hesperia (butterfly) | celastrina lucia | celastrina echo | pierid | large white	butterfly | white | blue | flower | picture | beautiful | hydrangea | hydrangea macrophylla | celastrina	butterfly | white | blue | flower | hydrangea | celastrina
	human | ocean | man | sea | couple | pose | fish | guy | boat | red | water | catch | fisherman | fishing | sit | tuna | cyprinus rubrofuscus | leather carp | reef squirrelfish | soldierfish | boarfish | coelacanth | armored catfish | cusk | round whitefish | opah | oarfish | brotula | ambloplites rupestris | creole-fish | jewfish | crevalle jack | threadfish | moonfish | amberjack | rudderfish | kingfish | florida pompano | bigeye scad | round scad | red snapper | grey snapper | mutton snapper | lutjanus apodus | red porgy | sheepshead | striped drum | sciaenops ocellatus | mulloway | yellowfin croaker | spadefish | pigfish | hogfish | puddingwife | oilfish | wahoo | king mackerel | bluefin | bonito | blue marlin | striped marlin | spearfish | palometa | barrelfish | yellowfin mojarra | vermillion rockfish | red rockfish | rosefish | lumpsucker | pogge | queen triggerfish | ocean sunfish | atlantic halibut | pacific halibut | sand dab | tonguefish | saltwater fish | sunfish | panfish | redfish | rockfish | angler	man | pose | fish | guy | boat | water | catch | fisherman | leather carp | reef squirrelfish | boarfish | opah | jewfish | crevalle jack | moonfish | amberjack | rudderfish | kingfish | florida pompano | bigeye scad | round scad | red snapper | grey snapper | mutton snapper | lutjanus apodus | red porgy | sciaenops ocellatus | mulloway | spadefish | pigfish | hogfish | oilfish | bluefin | bonito | spearfish | barrelfish | yellowfin mojarra | vermillion rockfish | red rockfish | rosefish | queen triggerfish | atlantic halibut | pacific halibut | sand dab | tonguefish | sunfish | redfish | angler	man | pose | fish | guy | catch | fisherman | leather carp | opah | jewfish | moonfish | amberjack | kingfish | bigeye scad | grey snapper | mutton snapper | lutjanus apodus | red porgy | sciaenops ocellatus | mulloway | spadefish | pigfish | hogfish | oilfish | spearfish | barrelfish | red rockfish | rosefish | atlantic halibut | pacific halibut | sand dab | tonguefish | sunfish | redfish

Figure 5. **Qualitative Results with different RAM++ thresholds**. While [78] found 0.7 to be the suitable RAM++ threshold, we We show qualitative examples across three different thresholds: 0.7, 0.75, 0.8 on a much larger concept bank. We find the most suitable pool of concepts at the 0.75 confidence threshold.

A.3. Object Detection

Benefits of Localized Annotations. Object tagging using RAM++ provides great insights into the object composition of images in image-text datasets. However, relevant factors for the holistic understanding of pretraining data such as the number of instances of the same concept in an image(count) and the localization of these concepts(spatial awareness) are confounded away by simply tagging an image with objects. To mitigate this, we incorporate bounding box information into the pipeline, which resolves both the issues identified.

GroundingDINO. Given an image, our model of choice, GroundingDINO [51] returns localized concept information, such as bounding boxes, detected concepts, confidence scores of each box, etc. Since, we use a detection model grounded in natural language, GroundingDINO can effectively detect objects from an image when provided an input text and each detection is tagged with a similarity score across the individual input text tokens.

How to provide text for an image is a design choice. Since Datacomp is an image-text dataset, one approach could be to provide the caption for the image as the input text. However, the alt-text captions are of low quality and do not always correspond to the visual concepts in the image. This artifact of web-scale image-text datasets have been well-studied and works such as [49, 58] have proposed methods to improve the text distribution. Another potential input involves providing the entire pool of concepts as the text input. Doing so leads to over-representation of objects being detected which are not visually present in the image, thus leading to some form of hallucination. This is especially true since we have $19,261$ concepts in our pool, significantly increasing the probability of hallucinations and reducing the processing speed of the model.

Our Approach. Our solution involves providing RAM++ object tags at a 0.75 confidence threshold as prompts to GroundingDINO. By reducing the vocabulary pool, we mitigate hallucinations and errors while also improving the detection model's processing speed. Through manual inspection, to remove low-confidence predictions to prevent a second degree of over-representation, we set a text threshold by only extracting concepts with a box-concept similarity score higher than 0.27. We set the same threshold for bounding box confidence scores too. With this configuration, we can now annotate each image of a pretraining dataset with the concept tags, per-concept logit scores from RAM++ and the set of bounding boxes, detected classes and their corresponding confidence scores.

Ensembling: An Introduction. An additional confounder is that DataComp-128M is available in multiple resolutions. To leverage this and increase the trustworthiness of DATACONCEPT, we apply Weighted Box Fusion (WBF) [73] for bounding box ensembling. WBF generates the final set of bounding boxes by using the confidence scores of the proposed bounding boxes of multiple object detection models/various configurations of the same object detection model. This approach is in contrast to Non-maximum suppression(NMS) which just removes part of the predictions instead of aggregating them. Ensembling has proven to be an effective strategy in complex object detection tasks [77]. Specifically, we ensemble across image resolutions $384, 512, 800, 1000$ to obtain more robust final detection results, refer to Fig. 6 for visual inspection. We provide more details in Appx. A.4.

Final Annotations. As we have demonstrated, DATACONCEPT has been curated using high confidence thresholds and stricter annotation protocols, with localization requiring bounding boxes to be generated for the precise regions of objects. This added difficulty has led to extremely rare concepts being underrepresented in the annotations. Nevertheless, DATACONCEPT-M contains $12,253$ unique concepts, which we define as $\mathcal{C}$, the concept pool for CABS.

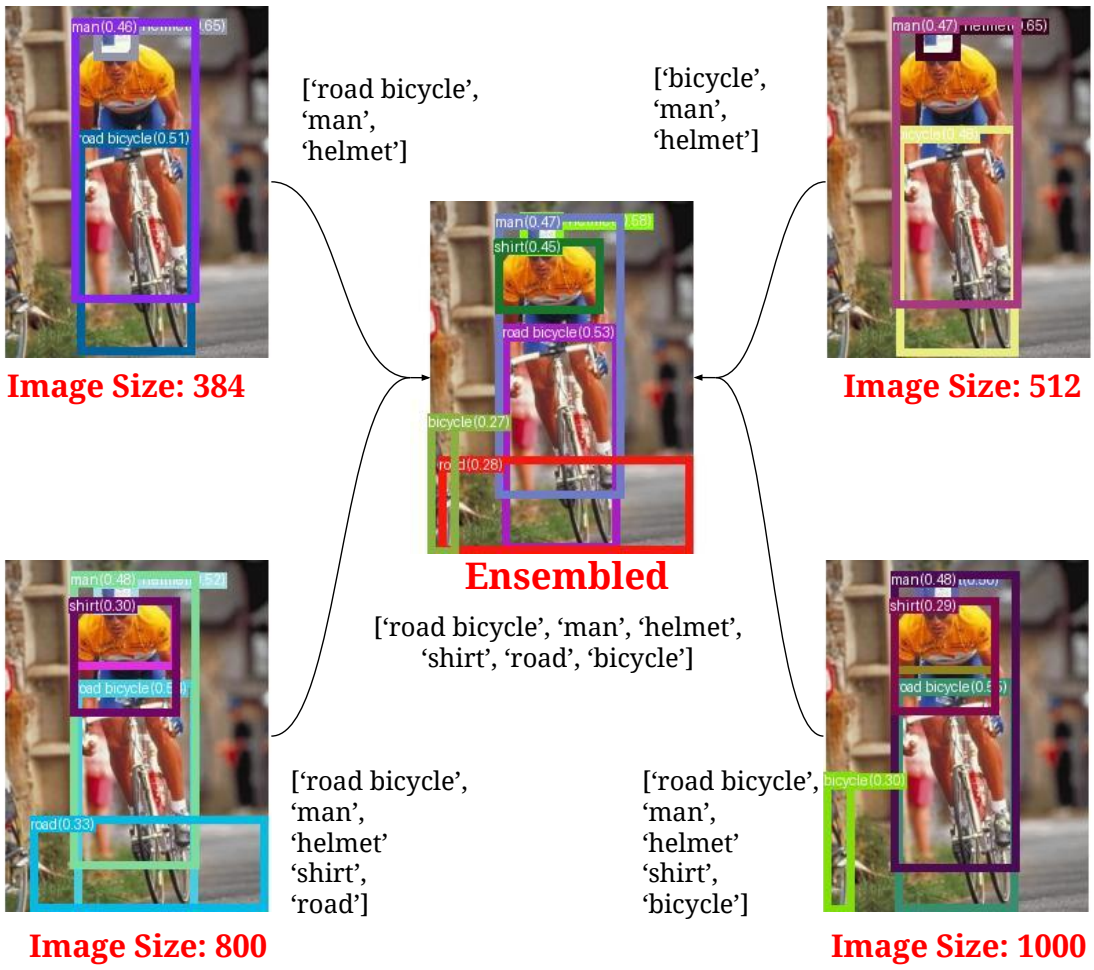

Figure 6. **Ensembling bounding boxes provides the best detection predictions for DATACONCEPT.** Using Weighted Box Fusion, we are able to detect single instances(no overlap of bounding boxes) of all relevant objects in an image.

A.4. Weighted Box Fusion for Ensembling Bounding Boxes

Weighted Box Fusion(WBF) [73] is a post-processing measure within object detection, generally used when there are multiple bounding boxes predicted by different models, or the same model performed on image with different augmentations. Our approach involves the latter, with one GroundingDINO model producing 4 bounding box predictions for one sample across different image resolutions. While other approaches, such as in traditional Non-Maximum Suppression (NMS), may remove detection with a lower score when multiple boxes overlap, WBF forms clusters of overlapping boxes, as long as it belongs to the same class, and produces a single box by taking a confidence-weighted average of coordinates. This preserves geometric evidence from different resolutions and often yields tighter, better-centered localization.

Notation. We start with a set of bounding boxes across the n image resolutions $\{384, 512, 800, 1000\}$ and their associated confidence scores

$$\mathcal{B} = \{b_i = (x_1^{(i)}, y_1^{(i)}, x_2^{(i)}, y_2^{(i)})\}_{i=1}^{n}$$
$$S = \{s_i\}_{i=1}^{n}, \quad s_i \in [0, 1]$$

Each box is also assigned a *class label* (concept) predicted by the model at that resolution:

$$C = \{c_i\}_{i=1}^{n}, \quad c_i \in \mathcal{V},$$

where $\mathcal{V}$ is our concept vocabulary (e.g., `person`, `car`, `flower`). We define resolution weights

$$\alpha_{m(i)}, \quad i = 1, \ldots, n$$

where $m(i)$ is resolution for box i. The fusion weight for each box is defined as

$$w_i = \alpha_{m(i)} \cdot s_i.$$

In our setup, we do not upweight any specific resolution, hence $\alpha_{m(i)}$ is always 1 and we do not use $\alpha_{m(i)}$ in future definitions and formulae. Note that the set of bounding boxes at each resolution are first sorted in decreasing order of confidence scores before the following steps are implemented.

Clustering. Bounding boxes need to be grouped into clusters to implement WBF. The heuristic is simple, two bounding boxes belong to the same cluster iff there is a significant overlap spatially and the classes of the two boxes are the same.

A cluster $\mathcal{K}$ associated with a reference box j is defined as

$$\mathcal{K}(j) = \{\, i \in \{1, \ldots, n\} \mid \mathrm{IoU}(b_i, b_j) > T,\ c_i = c_j \,\}$$

where T is a predefined IoU threshold. The IoU threshold is used as the metric for spatial overlap. In our experiments T is set to 0.29.

IoU Definition. For two boxes A and B, the Intersection-over-Union (IoU) is defined as

$$\mathrm{IoU}(A, B) = \frac{A \cap B}{A \cup B}$$

Ensembling. For a cluster $\mathcal{K}$ containing k bounding boxes corresponding to the same class, the final coordinates are computed as follows:

$$\hat{x}_1 = \frac{\sum_{i \in \mathcal{K}} w_i\, x_1^{(i)}}{\sum_{i \in \mathcal{K}} w_i}, \qquad \hat{y}_1 = \frac{\sum_{i \in \mathcal{K}} w_i\, y_1^{(i)}}{\sum_{i \in \mathcal{K}} w_i},$$
$$\hat{x}_2 = \frac{\sum_{i \in \mathcal{K}} w_i\, x_2^{(i)}}{\sum_{i \in \mathcal{K}} w_i}, \qquad \hat{y}_2 = \frac{\sum_{i \in \mathcal{K}} w_i\, y_2^{(i)}}{\sum_{i \in \mathcal{K}} w_i}.$$

The fused confidence score for the fused box as the average confidence of all boxes that form the cluster as is denotes as follows:

$$\hat{s} = \frac{\sum_{i \in \mathcal{K}} w_i \, s_i}{\sum_{i \in \mathcal{K}} w_i}$$

This is in stark contrast with other bounding box selection methods like NMS [56], Non-Maximum Weighted (NMW) method [62], etc. NMS completely excludes boxes that have a lower IoU than the threshold, while NMW does not change confidence scores. On the other hand, WBF uses all boxes provided and determines the final coordinates by means of confidence scores of the specific prediction.

Two-stage post-filtering. Following closely the original WBF formulation [73], the fused confidence scores are rescaled to reflect model agreement:

$$\hat{s} \leftarrow \hat{s} \cdot \frac{\min(T, n)}{n} \qquad \text{or} \qquad \hat{s} \leftarrow \hat{s} \cdot \frac{T}{n},$$

where T is the number of boxes in the cluster and n is the number or resolutions. This reduces the score of boxes supported by only a small subset of resolutions. Essentially, if any of i fails to predict a bounding box belonging to a cluster, we reduce the score of the fused box as opposed to a cluster with predictions from all $i \in n$.

After WBF, we apply an optional second-stage filter to remove near-duplicate boxes of the same class. We do this for an added level of rigor to the final annotations. For each class, boxes with IoU above a stricter threshold T_{post} (e.g., 0.5) are re-clustered, and only the highest-confidence box in each cluster $\mathcal{K}$ is retained.

Summary. We adopt a rigorous approach to ensemble bounding boxes across a variety of resolutions and in this section we demonstrate why WBF is the most robust method to achieve this. Ensembling results in a list of bounding boxes, concepts and confidence scores which have been re-calibrated via weighted averaging (producing smoother, more meaningful scores). We provide all of these annotations in DATACONCEPT.

A.5. Ensembling: Quantitative Results

Motivation. In this section, we ask: *How do we quantify ensemble quality?* Since we do not have ground-truth information when dealing with DataComp, we refer to evaluations on benchmarks aligned with our task: obtaining a proxy for open-vocabulary object localization and detection. This is aligned with the takeaways from recent benchmarking works such as Ghosh et al. [33], which proposes granular evaluations into semantically related domains to determine the quality of machine learning models.

With this motivation, we test our ensembling approach using ODinW [48], a rigorous benchmark of 13 and 35 class variants comprisinng several varieties of image resolutions designed to assess model performance within real-world contexts [95]. GroundingDINO obtains an mAP of 26.1% on the 35 class variant of ODinW while more recent works using GroundingDINO as a base model obtain an mAP of 28% [95]. This difficulty of the task (ODinW approximates the long-tail, open-vocabulary distribution of internet-scale pre-training data) and the multitude of image resolutions align with DataComp and demonstrates that ODinW is a suitable benchmark to test our ensembling approach for bounding box annotations.

Evaluation Protocol. Given an image from the ODinW test set, we generate bounding box predictions for single resolutions (among $\{384, 512, 800, 1000\}$), as well as all combinations of ensembling (two resolutions, three resolutions and all resolutions). Taking from the ODinW test classes, we report average precision results of 10 classes, chosen which provide variance in performance across our resolutions and ensembles, as this provides the most insight into which method should be adopted. Results with single resolutions are shown in Appx. 9 and combinations of resolutions in Appx. 10. We show consistently that ensembling across all 4 resolutions provides the best bounding boxes for annotating DATACONCEPT.

Table 9. **Performance across resolutions and WBF ensembling on ODinW datasets.**We show that ensembling across all 4 resolutions gives the best detection predictions.

	Resolution					
Dataset	**384**	**512**	**800**	**1000**	**Ensembled (All)**	**Image Size (W×H)**
AerialMaritimeDrone_large	0.19	0.23	0.39	0.31	**0.41**	1000×750
AerialMaritimeDrone_tiled	0.44	0.47	0.35	0.23	**0.55**	800×600
ChessPieces	0.07	0.16	**0.18**	0.17	0.17	2048×1732
DroneControl	0.43	0.42	0.45	**0.47**	0.46	300×300
EgoHands_generic	0.95	0.95	0.97	0.97	**1.00**	1280×720
MountainDewCommercial	0.06	0.07	0.07	0.09	**0.11**	1290×896
North_American_Mushrooms	**0.73**	0.63	0.63	0.63	0.70	416×416
PKLot	0.45	0.45	0.46	0.44	**0.62**	640×640
brackishUnderwater	0.17	0.25	0.33	0.39	**0.59**	960×540
Self-driving car	0.29	0.37	0.36	**0.37**	0.36	1920×1200
mAP	0.39	0.40	0.42	0.41	**0.49**	–

Table 10. **Performance across various WBF ensembling combinations on ODinW datasets**. Ensembling across all 4 resolutions yields the best overall detection accuracy.

Dataset	**Resolution**					**Ensembled**	**Image Size**
	384 + 512	512 + 800	800 + 1000	384 + 512 + 800	512 + 800 + 1000		
AerialMaritimeDrone_large	0.29	0.40	<u>0.41</u>	0.40	<u>0.41</u>	<u>0.41</u>	1000×750
AerialMaritimeDrone_tiled	0.48	0.48	0.40	0.52	0.49	**0.55**	800×600
ChessPieces	0.12	0.16	<u>0.17</u>	0.16	<u>0.17</u>	<u>0.17</u>	2048×1732
DroneControl	0.33	0.38	0.41	0.33	0.38	**0.46**	300×300
EgoHands_generic	1.00	1.00	1.00	1.00	1.00	**1.00**	1280×720
MountainDewCommercial	0.06	0.07	0.10	0.07	0.10	**0.11**	1290×896
North_American_Mushrooms	<u>0.70</u>	0.64	0.60	0.66	0.60	<u>0.70</u>	416×416
PKLot	0.60	**0.63**	0.62	0.62	0.61	0.62	640×640
brackishUnderwater	0.41	0.55	0.56	0.55	0.58	**0.59**	960×540
Self-driving car	0.29	0.35	0.38	0.34	**0.38**	0.36	1920×1200
mAP	0.43	0.47	0.46	0.46	0.47	**0.49**	–

A.6. Concept Distribution

Having created DataConcept, we run a few analyses into the concept distribution of the dataset. We are particularly interested in two axes of inspection, ① DataConcept-wide concept count distribution (Appx. A.6.1) and ② Sample-level concept count distribution (Appx. A.6.2. Both these inspections inform different CABSvariants while curating online batches.

A.6.1. Dataset-wide Concept Count

As mentioned above, the final vocabulary $\mathcal{V}$ of DATACONCEPTcomprises $12,253$ unique concepts after GroundingDINO bounding box annotations, from the $19,261$ concepts in the concept bank. This means that in the complete 128M sample pool of DATACONCEPT, $12,253$ concepts occur at least once. We ask: *how are these concepts represented in the dataset?*

Fig. 7 demonstrates the extreme long-tailed nature of DATACONCEPT, a by-product of web-scaled distributions captured in DataComp. There is a total of $486,303,998$ annotations in DATACONCEPT, the lowest number of annotations being 1 and the highest being $20,974,722$ for `man`. We also find the median concept count to be 489. The figure shows an immense long-tail in the concept distribution, which is aligned with the findings in Parashar et al. [63], Udandarao et al. [78]. Given this extreme long-tailed nature, it is easy to estimate the biased concept distribution of an IID sampled batch during training and why concept-balancing as done in CABS-DM is critical to address this bias. For a better understanding of the concept distribution, we also provide the top 100 concepts with their respective counts as well as release the counts of all concepts as an artifact.

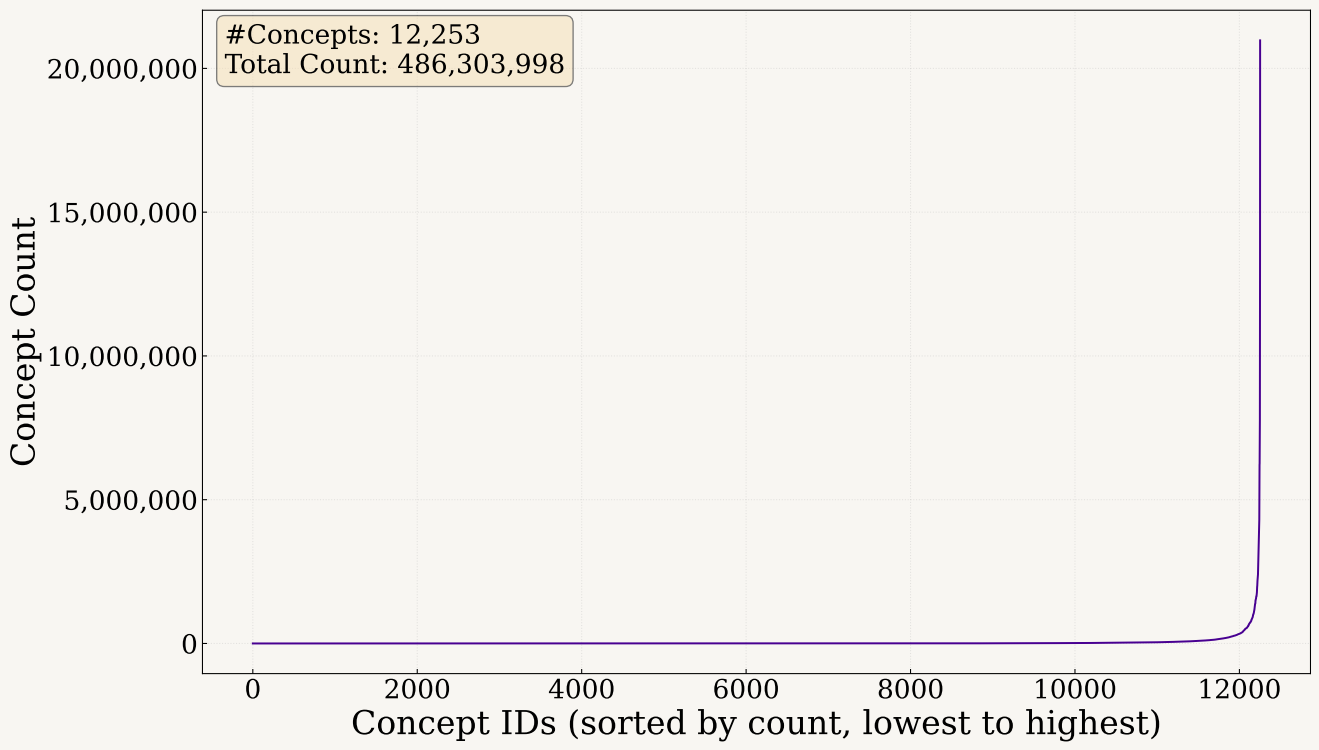

Figure 7. **What is the distribution of concepts in web-scale pretraining datasets?** We demonstrate the distribution of concept counts in DATACONCEPTafter annotations using GroundingDINO. Indeed, DATACONCEPTis strongly long-tailed with 86 concepts having more than 1 million annotations, 685 concepts having more than 100,000 annotations, 2670 concepts having more than 10,000 annotations and 5326 concepts having more than 1,000 annotations.

1. man: 20,974,722
2. woman: 13,264,330
3. flower: 9,397,706
4. chair: 7,770,596
5. wall: 6,939,361
6. hand: 6,760,215
7. car: 6,260,366
8. white: 6,212,499
9. poster: 5,647,604
10. shirt: 5,393,204
11. house: 4,308,600
12. floor: 4,250,178
13. tree: 4,136,099
14. smile: 3,943,812
15. brand: 3,608,960
16. sign: 3,597,012
17. water: 3,497,612
18. text: 3,375,403
19. picture: 3,318,294
20. building: 3,054,316
21. plate: 2,994,394
22. grass: 2,955,611
23. window: 2,727,034
24. dress: 2,712,616
25. box: 2,535,939
26. drawer: 2,506,757
27. cup: 2,401,762
28. plant: 2,376,292
29. child: 2,355,394
30. blue: 2,321,991
31. bottle: 2,268,065
32. girl: 2,215,856
33. road: 2,181,933
34. door: 2,149,296
35. light: 2,096,164
36. room: 1,991,748
37. paper: 1,981,447
38. eye: 1,884,913
39. smartphone: 1,882,529
40. table: 1,779,574
41. flag: 1,759,935
42. blanket: 1,699,679
43. circle: 1,682,581
44. sky: 1,659,255
45. bed: 1,637,075
46. crowd: 1,635,165
47. wheel: 1,634,712
48. hair: 1,634,382
49. guy: 1,608,147
50. dog: 1,606,644
51. pillow: 1,555,904
52. bowl: 1,550,462
53. cocktail table: 1,542,248
54. suit: 1,539,916
55. palm tree: 1,531,113
56. head: 1,504,124
57. necktie: 1,501,698
58. couch: 1,493,784
59. screenshot: 1,396,054
60. microphone: 1,395,115
61. document: 1,381,569
62. boat: 1,379,477
63. bag: 1,362,313
64. pillar: 1,356,866
65. cabinet: 1,310,379
66. number: 1,267,679
67. bird: 1,267,290
68. kitchen: 1,239,892
69. necklace: 1,238,552
70. logo: 1,214,333
71. shoe: 1,162,959
72. counter: 1,159,415
73. illustration: 1,143,293
74. vase: 1,134,215
75. bathroom: 1,102,494
76. living room: 1,095,075
77. fruit: 1,081,347
78. arm: 1,061,739
79. jacket: 1,056,604
80. truck: 1,026,752
81. image: 1,020,059
82. beard: 1,014,506
83. mirror: 1,013,644
84. fence: 1,005,264
85. stone: 1,003,367
86. goggles: 1,001,454
87. map: 995,526
88. faucet: 956,533
89. ball: 948,458
90. star: 945,154
91. carrot: 920,049
92. sink: 909,876
93. armchair: 899,612
94. bench: 899,012
95. face: 888,038
96. apple: 879,642
97. cartoon: 870,921
98. tower: 867,593
99. furniture: 865,619
100. skyscraper: 855,711

A.6.2. Sample-level Concept Count

To the best of our knowledge, previous works have not quantified *image complexity* using visual concepts in web-scale image-text pretraining datasets. Our GroundingDINO annotations are particularly useful here as we can leverage sample-level annotations to measure concept-multiplicity, i.e, *how many concepts are there in a sample?* Object detection annotations are more advantageous than the object tagging approach from Udandarao et al. [78] as RAM++ only tags a specific concept once to a sample, not taking into consideration if that concept is present multiple times in the image. Hence, our approach is the only publicly available resource to conduct a study of this scale.

Fig. 8 demonstrates that samples in DATACONCEPTgenerally have few concepts in them, a reflection of web-scale data, with a median of 3 concepts per-sample. We can infer that the bias towards lower concept counts or lower image complexity is rampant in IID batches during training and that models trained this way do not generalize to complex scenes that are common in retrieval datasets. This bias necessitates the need for CABS-FM and curation with sample complexity in mind.

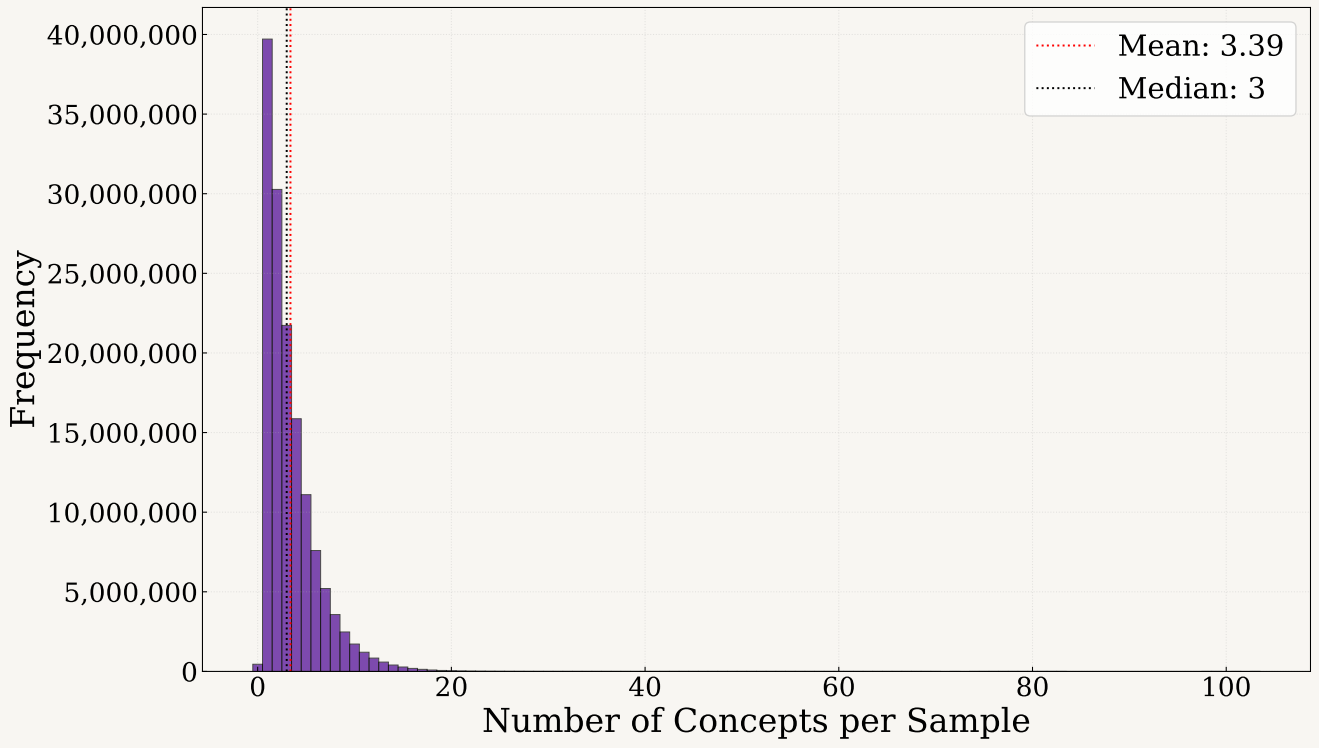

Figure 8. **What is the complexity of DATACONCEPTsamples based on visual concepts?** We demonstrate the distribution of concept counts per sample after annotations using GroundingDINO. Note that GroundingDINO can predict a concept many times, hence these numbers reflect the total number of concepts detected in an image, not unique concepts, hence acting as a suitable measure of image complexity.

B. Concept-aware Recaptioning

B.1. Selecting the Recaptioning VLM

Approach. Open-source VLMs have recently caught up with proprietary models in quality text generation given a prompt and an image. Hence, we opt for choosing a VLM that is optimal for both fidelity (adherence to the prompt and quality of output) and processing speed (we are annotating 128 million image-text pairs).

Our initial model pool includes `Molmo-7B-D-0924` [20], `moondream2` and `Qwen2-VL-7B` [82]. We test these models on a random subset of $10,000$ samples to check both fidelity and processing speed, providing all of them the following prompt:

> Generate a brief and concise image caption using relevant details from alt-text and classes present in the image.
> Alt-Text: {alt-text}
> Classes: {classes}.

We incorporate the raw caption from the sample as well as the list of detected classes for richer and concept-aware captions. Contrary to recent works [5], simply prompting performant open-weight VLMs with alt-text ensures it gets incorporated into the synthetic caption. Additionally, VLMs such as Molmo and Qwen2-VL also discard low quality alt-text, which suits our requirements. We observe that `moondream2` has the fastest processing speed but returns low fidelity captions. `Molmo-7B-D-0924` returns high quality captions but is often quite verbose and prone to hallucinations, on top of being the slowest VLM of the three. Hence, we choose `Qwen2-VL-7B` due to its ability to adhere to the prompt, generate high quality captions with relatively low hallucinations and a moderate processing speed. We admit that these models are not the current state-of-the-art: they were at the time of experimentation and annotation. Please refer to Fig. 9 for more qualitative comparisons between the 3 models.

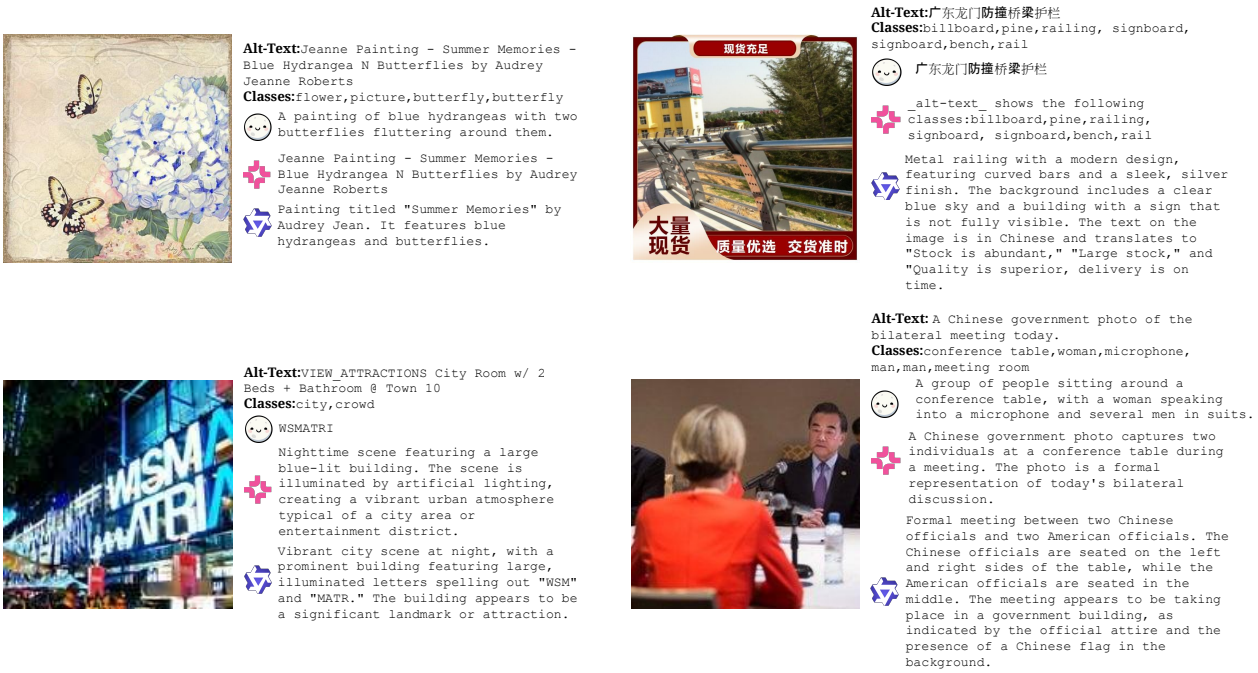

Figure 9. **Comparing 3 state-of-the-art open-weight VLMs on concept-aware captioning for pimage-text pretraining datasets.** We compare Moondream2, Molmo-7B and Qwen2-VL-7B across a random subset of DataComp-128M and select Qwen2-VL for a combination of its higher quality captions and appropriate processing speed.

B.2. Caption Quality

To understand the richness of information in the synthetic captions generated by Qwen2-VL-7B [82], we adopt a similar analysis as Nguyen et al. [58] and measure ① the number of words and ② the concept adherence of our new captions compared to the original raw captions.

Number of Words In Fig. 10, we observe the distributional difference between the raw captions used in DataComp and our synthetically generated captions. While the raw captions have median word count of 6 with a standard deviation of 9.51, Qwen2-VL-7B recaptions have a median word count of 33.56 with a standard deviation of 16.45. Please note that the raw captions, though much shorter generally, contain 214,787 samples with a word count higher than 80 which are included in the mean and standard deviation measurement but are not presented in this plot.

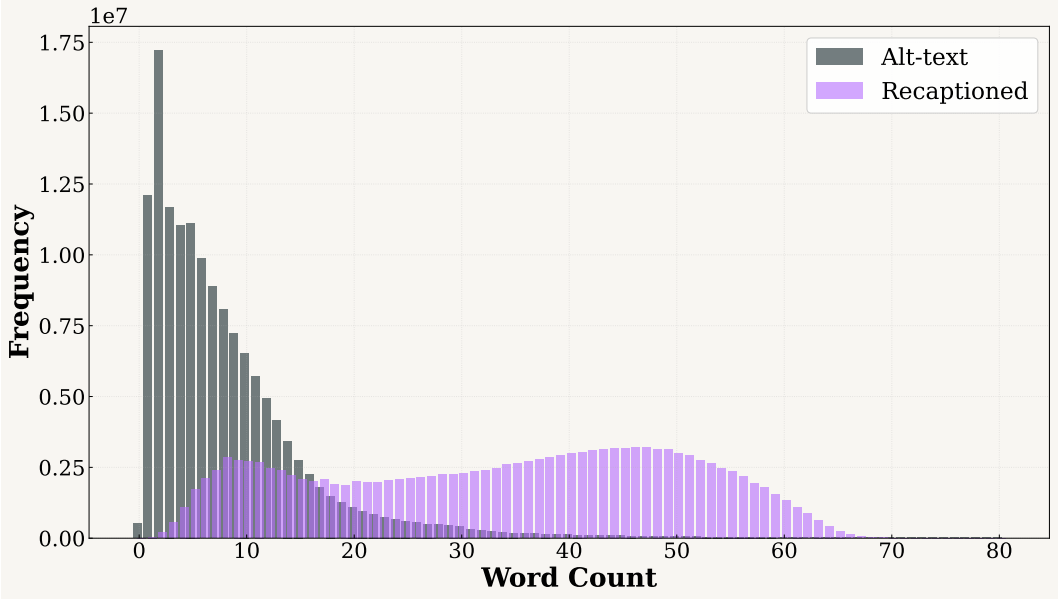

Figure 10. **Word Count Distribution.** Comparison of DataComp alt-text captions and Qwen2-VL-7B recaptions. Alt-text remains short-form, while recaptions are substantially longer. Extremely long alt-text outliers are excluded from the plot for clarity.

Concept Adherence. We sample a 1M random subset from DATACONCEPTto estimate how frequently the alt-text or the synthetic caption contains the concepts the sample has been annotated with. Firstly, since the raw captions are multilingual and our concepts are in English, we translate our raw captions to English. Then we measure the exact match percentage which measures if the exact concept word is found in the text. We then do a partial match with a search over various forms of a concept (lemmatized, plurals, gerunds, synonyms). The concept is found in the text if the best fuzzy match between any concept form and any token in the caption exceeds a similarity threshold τ. We show our results in Tab. 11, By sweeping $\tau \in \{0.6, 0.7, 0.8\}$, we quantify how robust the alignment is under progressively more difficult thresholds of semantic similarity. We show the staggering improvements in concept adherence when using our synthetic recaptions.

Table 11. **Exact and partial concept adherence between alt-text and Qwen2-VL recaptions.**

Caption	Exact Match (%)	Partial Match(%)		
		$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$
Alt-text	3.89	32.65	15.63	9.82
Qwen2-VL Recaptions	51.17	86.69	79.15	67.46

B.3. Qualitative Evaluation: Visualization Results

Concept-Aware Image Caption

Prompt: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.

Alt-text: Jeanne Painting - Summer Memories - Blue Hydrangea N Butterflies by Audrey Jeanne Roberts
Classes: flower, picture, butterfly, butterfly
Synthetic Caption: Painting titled "Summer Memories" by Audrey Jeanne Roberts. The picture is a beautiful representation of summer, with the hydrangea flower and butterflies as the main subjects.

Alt-text: LÅNGFJÄLL Konferenční židle s područkami, Gunnared béžová/bílá
Classes: clock, swivel chair, office desk, office, plant, picture frame, picture frame
Synthetic Caption: Modern office setup with a white office desk, a beige office chair with armrests, and a beige rug on the floor. There is a whiteboard on the wall, a clock, and some green plants in the background.

Alt-text: 台北家鴻燒鵝興安店：千人推薦4顆星港式燒臘店，吃肥美燒鵝飯秒殺便當、鮮蝦雲吞
Classes: plate, cup, twice-cooked pork, soy sauce, rice, salad, noodles, vegetable
Synthetic Caption: Plate of food that includes a serving of rice, noodles, and a piece of twice-cooked pork. There are also some vegetables, such as cucumbers and lettuce, on the plate. A cup of tea is placed next to the plate.

Alt-text: Bolivia Travel, Brazil Travel, Argentina Travel, South America Destinations, South America Travel, Travel Destinations, Travel Advice, Travel Guides, Travel Tips
Classes: picture, tree, river, river
Synthetic Caption: Travel poster for hiking in Tierra del Fuego National Park in Argentina. The poster features a scenic landscape with a river, a tree, and a mountain in the background. The text on the poster reads "Hiking in Tierra del Fuego National Park, Argentina" and "TravelYesPlease.com".

Figure 11. Some examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

Concept-Aware Image Caption

Prompt: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.

Alt-text: 国际音标英语教程练习册 小学英语音标入门指导教程国际音标教材书知行健英语音标发音教材英语发音零基础入门教材指导附音频资料
Classes: book, poster
Synthetic Caption: Cover of a book titled "国际音标英语教程练习册" which translates to "Workbook for the International Phonetic Alphabet" in English. The cover features a green and yellow color scheme with a prominent image of a globe, symbolizing a global or international context.

Alt-text: Rule
Classes: bull, pavement, person, street sign
Synthetic Caption: Person standing in front of a large, bronze bull statue on a cobblestone street. The statue is located in a city setting, and there is a green street sign visible in the background. The person is wearing a dark jacket and appears to be posing for the photo.

Alt-text: Test: Red Dead Redemption 2 im Test (Xbox One X & PC)
Classes: man, hat, fire, gun
Synthetic Caption: Promotional graphic for the video game "Red Dead Redemption 2." It features a man in a Western setting, with a fiery explosion in the background. The text "Test: Red Dead Redemption 2 im Test (Xbox One X & PC)" suggests that this is a test or review of the game for Xbox One X and PC platforms.

Alt-text: Веревочный парк «Активная страна» (Фото) - новости Днепра
Classes: woman, rope, playground, kid
Synthetic Caption: Kid hanging from a rope in a park called "Active Country" (Фото). The park is located in Dnepropetrovsk, as indicated by the text in the image.

Figure 12. Some more examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

Concept-Aware Image Caption

Prompt: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.

Alt-text: Bild
Classes: jar, bottle, plate, kitchen counter, mouse, kitchen, cup, silky pocket mouse, microwave
Synthetic Caption: A small animal, possibly a mouse or a rat, is sitting inside a glass jar on a kitchen counter. There is a wine bottle and another jar of what appears to be pickles or a similar condiment in the background.

Alt-text: Máscara del carnaval - verde/caso de Droid RAZR Carcasa Para Motorola Droid RAZR
Classes: mask, phone, masquerader
Synthetic Caption: The image is of a phone case designed for a Motorola Droid RAZR. The case features a vibrant and intricate design that includes a green and gold color scheme, with a prominent image of a carnival mask. The word "Mystique" is written in a stylized font at the top of the case.

Alt-text: 国家公务员
Classes:
Synthetic Caption: Advertisement for a live course titled "事途先知" (Shì Tú Xiān Zhī), which translates to "Know the Path First" in English. The course is focused on preparing for the "陕西事业单位" (Shǎnxī Shìyè Yùánjì) or "Shaanxi Public Institutions" in English.

Alt-text: 的照片 El Tiemblo y sus Puentes Medievales
Classes: water
Synthetic Caption: Picturesque scene of a stone bridge over a calm body of water, surrounded by trees and foliage. The bridge appears to be of medieval design, with an arched structure. The reflection of the bridge and trees is clearly visible in the water, creating a serene and symmetrical view..

Figure 13. Some more examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

Concept-Aware Image Caption

Prompt: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.

Alt-text: AS Monaco Championship
Classes: postage stamp, handstamp, football
Synthetic Caption: Postage stamp from Monaco featuring a football player in mid-air, celebrating a goal. The stamp commemorates the Monaco Football Club's championship victories in 1963, 1972, 1986, and 1998.

Alt-text: 玉華R媽ˇ笑容超好看
Classes: woman, blackboard, smile
Synthetic Caption: Woman standing in front of a blackboard with Chinese characters written on it. The woman is wearing a white shirt and appears to be holding a book or a piece of paper. The blackboard has some additional markings and writings on it

Alt-text: バンクシー７
Classes: decker bus, wall, woman, balloon, road
Synthetic Caption: Street scene with a red double-decker bus on the left side of the frame. On the right side, there is a mural of a girl holding a red balloon. The background appears to be an urban setting with buildings and a wall.

Alt-text: GRATUIT GRATUITEMENT KOTOB FALSAFA TÉLÉCHARGER
Classes: poster
Synthetic Caption: The image contains text in Arabic and English. The Arabic text reads "ليونارد," which translates to "Leonard" in English. The English text reads "GRATUIT GRATUITEMENT KOTOB FALSAFA TÉLÉCHARGER," which translates to "FREE FREE KOTOB FALSAFA DOWNLOAD" in English.

Figure 14. Some more examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

C. CABS: More Details

C.1. CABS-DM

We provide the full PyTorch style code for the heuristic function used in CABS-DM below.

Algorithm 2 PyTorch-style code for CABS-DM heuristic function

```
# h_DM: CABS for Diversity-Maximization
# C_i = concept set for sample i
# D = (I, T, C) = full super-batch
# theta = (b, F, heap_state) where:
# b = target batch size
# F = maximum frequency per concept in batch
# heap_state = (selected, n_c, heap) for iterative selection
def h_DM(C_i, D, theta):
  b, F, heap_state = theta # unpack parameters
  I, T, C = D # unpack super-batch
  # Step1: initialize on first call
  if heap_state is None:
    global_freqs = gather_all_concept_frequencies(C)
    t_c = concept_balancing_targets(global_freqs, b, F)
    selected, n_c = [], zeros(global_freqs.size)
    heap = init_max_heap()
    # Step2: compute initial gains for all samples
    for i in range(len(C)):
      gain_i = compute_marginal_gain(C[i], t_c, n_c, global_freqs, F)
      heap.push((gain_i, i))
    heap_state = (selected, n_c, heap, t_c, global_freqs)
  selected, n_c, heap, t_c, global_freqs = heap_state
  # Step3: select top sample from heap. Greedy selection: pop best, update counts, refresh heap
  if len(selected) < b and heap:
    idx = heap.pop()
    selected.append(idx)
    update_counts(n_c, C[idx])
    refresh_heap(heap, idx, C, n_c, t_c, global_freqs, F)
  return selected if len(selected) == b else None

# Helper: compute gain from adding concepts in C_i
def compute_marginal_gain(C_i, t_c, n_c, global_freqs, F):
  gain = 0
  for c in C_i:
    if n_c[c] < F: # respect frequency cap
      deficit = max(0, t_c[c] - n_c[c])
      gain += deficit / (global_freqs[c] + 1e-8)
  return gain
```

C.2. CABS-FM

We provide the full PyTorch style code for the heuristic function used in CABS-FM below.

Algorithm 3 PyTorch-style code for CABS-FM heuristic function

```
# h_FM: CABS for Frequency-Maximization
# C_i = concept set for sample i
# D = (I, T, C) = full super-batch
# theta = [] where:
def h_FM(C_i, D, theta):
  I, T, C = D # unpack super-batch

  # Step 1: count number of concepts in sample
  concepts = C_i
  num_concepts = len(concepts)

  # Step 2: compute frequency-maximization score
  # Higher score = more diverse concepts
  score = num_concepts

  return score
```

C.3. Hyperparameters

We adopt the `open_clip` [42] codebase to train CLIP and SigLIP models and incorporate CABSdirectly into the codebase, thus making it easily reproducible for practioners accustomed to the code. We also consider the hyperparameters fixed by Datacomp[31] to ensure that IID results are easily reproducible and that all the performance boosts occur due to CABS. Appx. 12 shows the general hyperparameters used for training as well as CABS-specific hyperparameters.

Table 12. General pretraining and CABS-specific hyperparameters.

Hyperparameter	**IID**	**CABS-DM**	**CABS-FM**
batch_size	1024	5120	5120
beta1	0.9	0.9	0.9
beta2	0.98	0.98	0.98
epochs	1	5	5
eps	1e-06	1e-06	1e-06
force_quick_gelu	False	False	False
gather_with_grad	True	True	True
lr	0.0005	0.0005	0.0005
lr_scheduler	cosine	cosine	cosine
opt	adamw	adamw	adamw
precision	amp	amp	amp
warmup	500	500	500
wd	0.2	0.2	0.2
CABS-specific			
filter_ratio	–	0.8	0.8
max_concept_frequency	–	40	–
min_samples_concept	–	1	–

D. Extended Benchmark Performance

D.1. Evaluation Suite: Further Details

Testing contrastively trained VLMs on a diverse set of benchmarks, such as the set of evaluation test sets suggested by [31] is critical to understand their zero-shot generalization properties. However, recent probes into the reliability of these benchmarks such as [3, 79] have exposed several noisy, error-prone and high variability test sets in this set. We decide to omit these benchmarks, resulting in a final pool of 28 benchmarks, spanning 26 zero-shot classification and 2 image-text retrieval detailed below:

Table 13. Datasets used in Zero-Shot Classification and Image-Text Retrieval Tasks

Task Type	Dataset	Test Set Size	Number of Classes
Classification	Caltech-101 [29]	6,085	102
	Camelyon17	85,054	2
	CIFAR-10 [46]	10,000	10
	CIFAR-100 [46]	10,000	100
	Country211 [66, 76]	21,100	211
	Dollar Street [32]	3,503	58
	DTD [18]	1,880	47
	FGVC Aircraft [53]	3,333	100
	Food-101 [10]	25,250	101
	FMoW [17, 44]	22,108	62
	GeoDE [67]	12,488	40
	ImageNet [21]	50,000	1,000
	ImageNet-A [39]	7,500	200
	ImageNet-O [39]	2,000	200
	ImageNet-R [38]	30,000	200
	ImageNet-Sketch [80]	50,889	1,000
	ImageNet-V2 [68]	10,000	1,000
	Let-it-Wag! [78]	130,000	290
	ObjectNet [6]	18,574	113
	Oxford Flowers-102 [61]	6,149	102
	Oxford-IIIT Pets [64, 90]	3,669	37
	Pascal VOC 2007 [26]	14,976	20
	RESISCS45 [16, 90]	6,300	45
	Stanford Cars [45]	8,041	196
	STL-10 [19]	8,000	10
	SUN-397 [85]	108,754	397
Retrieval	Flickr30k [88]	31,014	N/A
	MSCOCO [14]	5,000	N/A

We make several categories of datasets while presenting them such as **IN-shift** which comprises `imagenet-a`, `imagenet-r`, `imagenet_sketch`, `imagenetv2`, `imagenet-o` and `objectnet`, **Scene** which comprises `vtab-resisc45`, `sun397` and `geode` and **Obj** which comprises the remaining classification datasets.

D.2. Full Model Suite

To provide a more in-depth analysis of the trends seen when comparing IID sampling and CABS-DM and CABS-FM, we conduct experiments on two additional models, CLIP ViT-S-16 and SigLIP ViT-SO400M. We arrive at the same conclusions as discussed in Sec. 4.2, we see the strong performance boosts with CLIP ViT-S-16 and SigLIP ViT-SO400M as we see with CLIP ViT-B-32 and SigLIP ViT-B-16/256. Please refer to Tab. 14 for CABS-DM performance and Tab. 15 for CABS-FM performance. We make the conclusion that CABS *is effective and provides state-of-the-art performance across varied model architectures and varied model sizes and may be adopted as the de-facto online batch sampling algorithm for contrastive pretraining.*

Table 14. **Extended Classification Results** including CLIP ViT-S-16 and SigLIP ViT-SO400M. CABS-DM delivers consistent improvements with these variants as well.

Method	Captions	Zero-shot Classification				Let-it-Wag!	Avg (Clf)
		IN-Val	IN-shift	Obj	Scene		
ViT-S-16							
IID	alt	16.9	15.0	30.3	35.4	6.1	26.6
CABS-DM	alt	**24.6**	**20.6**	**34.8**	**39.0**	**8.3**	**31.5**
IID	recap	24.8	22.8	39.4	44.4	6.3	35.4
CABS-DM	recap	**30.0**	**27.4**	**40.6**	**45.0**	**8.0**	**37.8**
ViT-B-32							
IID	alt	17.3	15.2	32.3	36.4	5.1	28.2
CABS-DM	alt	**21.9**	**18.6**	**34.5**	**38.0**	**7.5**	**30.7**
IID	recap	21.7	20.8	36.4	43.1	5.9	33.0
CABS-DM	recap	**26.7**	**25.4**	**39.6**	**42.8**	**7.1**	**35.5**
ViT-B-16-SigLIP-256							
IID	alt	17.2	15.3	29.6	35.9	5.2	26.4
CABS-DM	alt	**24.1**	**20.8**	**33.5**	**39.6**	**7.0**	**30.9**
IID	recap	28.8	27.4	41.5	48.9	6.6	38.6
CABS-DM	recap	**34.7**	**32.3**	**43.2**	**50.6**	**7.6**	**41.1**
ViT-SO400M-14-SigLIP							
IID	alt	15.5	13.7	27.5	34.7	4.7	24.5
CABS-DM	alt	**22.6**	**18.8**	**33.4**	**40.0**	**6.2**	**30.2**
IID	recap	34.1	31.8	46.3	55.9	7.6	42.2
CABS-DM	recap	**39.6**	**36.1**	**45.1**	**57.5**	**9.4**	**44.2**

Table 15. **Retrieval Results** (COCO and Flickr30K) with averaged retrieval score.

Method	**Captions**	COCO	Flickr	**Avg(Ret)**
		ViT-S-16		
IID	alt	9.6	17.4	13.5
CABS-FM	alt	**11.3**	**23.8**	**17.6**
IID	recap	28.7	47.2	38.0
CABS-FM	recap	**32.4**	**56.2**	**44.3**
		ViT-B-32		
IID	alt	9.7	16.2	12.9
CABS-FM	alt	**11.0**	**21.9**	**16.5**
IID	recap	24.0	41.3	32.6
CABS-FM	recap	**30.4**	**52.9**	**41.6**
		ViT-B-16-SigLIP-256		
IID	alt	11.1	18.9	15.0
CABS-FM	alt	**12.3**	**23.9**	**18.1**
IID	recap	37.1	57.0	47.0
CABS-FM	recap	**39.7**	**63.5**	**51.6**
		ViT-SO400M-14-SigLIP		
IID	alt	8.8	13.7	11.2
CABS-FM	alt	**11.3**	**15.9**	**13.6**
IID	recap	37.7	53.8	45.7
CABS-FM	recap	**39.2**	**57.9**	**48.6**

E. Continual Pretraining

All the experiments we conducted so far in the main paper and previous supplementary sections were operating in the pretraining from scratch regime. Now, we wish to see if CABS is a strong batch sampling algorithm on other pretraining regimes as well, beyond standard pretraining. To this end, we adopt a continual pretraining paradigm [70], where checkpoints trained at the same scale (128M samples seen) are used to initialize the model that we wish to train. Concretely, we initialize from a CLIP ViT-B/32 model trained using IID-sampling on DataComp-128M. We then conduct continued pretraining for 128M more samples (so in total, we the final checkpoint is trained for 256M samples seen) using IID sampling, CABS-DM and CABS-FM. Our results are presented in Tabs. 16 and 17. Across both alt-text and concept-aware synthetic re-captions, CABS-DM and CABS-FM continues to outperform IID sampling on all benchmarks, even in the continual pretraining regime.

Our results hence demonstrate that CABS variants can also be utilized as a strong continual pretraining method that can utilize strong pretrained vision encoders. This has connections to similar results observed in mid-training and annealing of language models [9, 30]. We can further draw a faint connection to data curriculums [7, 93]—where we first start with a standard data-mixture (as induced by IID sampling), followed by a more targeted "mid-training" mixture (as induced by CABS variants). In the future, we can more closely explore finer-grained curriculums using different CABS variants.

Table 16. **Continual Pretraining: Zero-shot Classification Performance.** We isolate the zero-shot classification benchmarks from the continual-pretraining experiment to more clearly highlight the impact of CABS-DM. We observe that CABS-DM consistently outperforms IID sampling when continually pretraining from the same IID initialization, demonstrating stronger concept coverage and more robust generalization under distribution shift.

Method	Captions	IN-Val	IN-shift	Obj	Scene	Let-it-Wag!	Avg (Clf)
			ViT-B-32				
IID	alt	23.7	20.0	37.7	42.3	7.9	33.4
CABS-DM	alt	**27.8**	**23.9**	37.4	**42.7**	**8.9**	**34.4**
IID	recap	27.7	25.8	41.7	47.7	7.7	38.1
CABS-DM	recap	**31.7**	**29.1**	**43.4**	46.8	**8.9**	**40.0**

Table 17. **Continual Pretraining: Cross-modal Retrieval Performance.** This table isolates retrieval metrics to examine how CABS-FM performs in the continual pretraining setting. We report COCO and Flickr30K retrieval scores along with their mean. Similar to CABS-DM on classification, we observe significant performance boosts when comparing CABS-FM to IID sampling.

Method	Captions	COCO	Flickr	Avg (Ret)
		ViT-B-32		
IID	alt	13.7	24.5	19.1
CABS-FM	alt	**14.9**	**28.7**	**21.8**
IID	recap	30.5	49.0	39.8
CABS-DM	recap	**32.7**	**54.2**	**43.5**

F. Ablation on Filter Ratios

In this section, we show how the filter ratio f, defined as the parameter that determines the size of a sub-batch b given super-batch of size B. For example, a filter ratio of $f = 0.5$ would correspond to a super-batch of size 8192 for a sub-batch of size 4096. In most of our experiments, we fix the filter ratio to 0.8. Fig. 15 provides an ablation over various other filter ratios for a ViT-B/32 CLIP model, tested on ImageNet across filter ratios $\{0.5,0.75,0.8,0.9\}$. Performance trends over the set of filter ratios indicate that 0.8 is indeed the optimal filter ratio at the 128M sample scale.

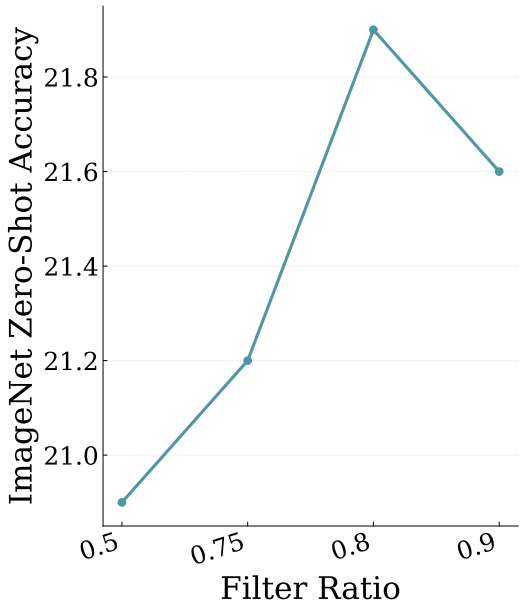

Figure 15. **CABS-DM filtering ratio ablation**. We choose $f = 0.8$ based on ImageNet validation performance. For simplicity, we maintain this filter ratio for CABS-FM as well, and still see strong performance gains on image-text retrieval benchmarks.

G. Stability and Diversity of CABS-DM

In this section, we verify that the performance gains of CABS-DM are not an artefact of a particular random seed and simultaneously that CABS-DM generates diverse sub-batches. For the former, we train 3 independent CABS-DM models and 3 IID models with different random seeds and report mean and standard deviation of downstream performance in Tab. 18. CABS-DM consistently outperforms IID across all seeds and benchmarks, with notably low variance, confirming the robustness of our method.

Next, we report pairwise Jaccard similarity between sub-batches drawn from the same super-batch in Tab. 19. Across 3 seeds and 2 super-batches, mean overlaps range from 0.20 to 0.24, confirming that CABS-DM produces substantially diverse sub-batches across training steps.

Table 18. **IID vs. CABS-DM across 3 random seeds (ViT-B-32-CLIP).** Mean and standard deviation reported.

Method	IN-Val	IN-shift	Obj	Scene	Let-it-Wag	Clf (Avg)
IID	17.4 ± 0.01	15.2 ± 0.08	32.3 ± 0.25	36.1 ± 0.19	5.0 ± 0.0	27.9 ± 0.22
CABS-DM	21.9 ± 0.03	18.9 ± 0.09	34.6 ± 0.27	37.7 ± 0.19	7.5 ± 0.0	30.7 ± 0.23

Table 19. **Pairwise Jaccard similarity between CABS-DM sub-batches** drawn from the same super-batch, across 3 seeds and 2 super-batches (SB).

Super-batch	0 vs 1	0 vs 2	1 vs 2	Mean $\pm$ Std
SB 0	0.239	0.243	0.238	0.240 ± 0.002
SB 1	0.202	0.201	0.202	0.201 ± 0.001
Overall				$\mathbf{0.221 \pm 0.019}$

H. CABS-DM is a Better Vision Encoder for Autoregressive VLMs

To assess the transferability of CABS-DM beyond classification and retrieval performance, we adopt the LLaVA framework [50] to evaluate our vision encoders on VQA and captioning tasks. We use a ViT-B-32 CLIP backbone pretrained with either IID or CABS-DM, and train a LLaVA model for 1 epoch on the standard LLaVA SFT mix. Note that we adopt the exact same training protocol as [50] (including using Vicuna-7B as the language encoder) to attribute the performance gains to the vision encoder. Results are reported in Tab. 20. The CABS-DM vision encoder consistently and strongly outperforms the IID encoder across all benchmarks, demonstrating that the improved visual representations transfer to downstream generative settings.

Table 20. **LLaVA-based captioning and VQA evaluations.** CABS-DM vision encoders consistently outperform IID encoders across VQA (GQA, MME) and captioning (COCO, NoCaps, Flickr) benchmarks. Note: we adopt the quasi-exact match metric for VQA benchmarks and the ROGUE score for captioning benchmarks.

Method	GQA ↑	COCO ↑	MME ↑	NoCaps ↑	Flickr ↑
IID	22.7	11.2	138/410	12.8	11.4
CABS-DM	37.8	13.5	253/725	15.1	13.2

I. Fine-grained Benchmark Performance

Motivation While it is common practice to report the aggregated performance across multiple benchmarks to demonstrate the capabilities of machine learning models, a deeper probe into the benchmarks that comprise the complete suite of evaluation is often necessary to have a deeper understanding of the true capabilities of the model. This is studied in Ghosh et al. [33] for language models and autoregressive vision-language models but the principle may be applied to CLIP as well.

I.1. Expanded Analysis

To that end, we provide an expanded probe into the specific benchmarks where CABS-DM outperforms IID sampling (it is relatively straightforward to observe dataset-specific performance gains for CABS-FM as models are evaluated on 2 benchmarks, MSCOCO and Flickr30k). For example, in Fig. 18, we specifically show performance boosts for CABS-DM over IID-sampling in 23 out of 26 benchmarks. With this, we can ascertain that despite maximizing for concept diversity, CABS-DM shows strong gains on datasets that test for long-tailed concepts as well as for more common concepts. This confirms that CABS-DM is an all-round performant batch sampling algorithm for classification tasks. The per-benchmark breakdown of Tab. 14 is shown below.

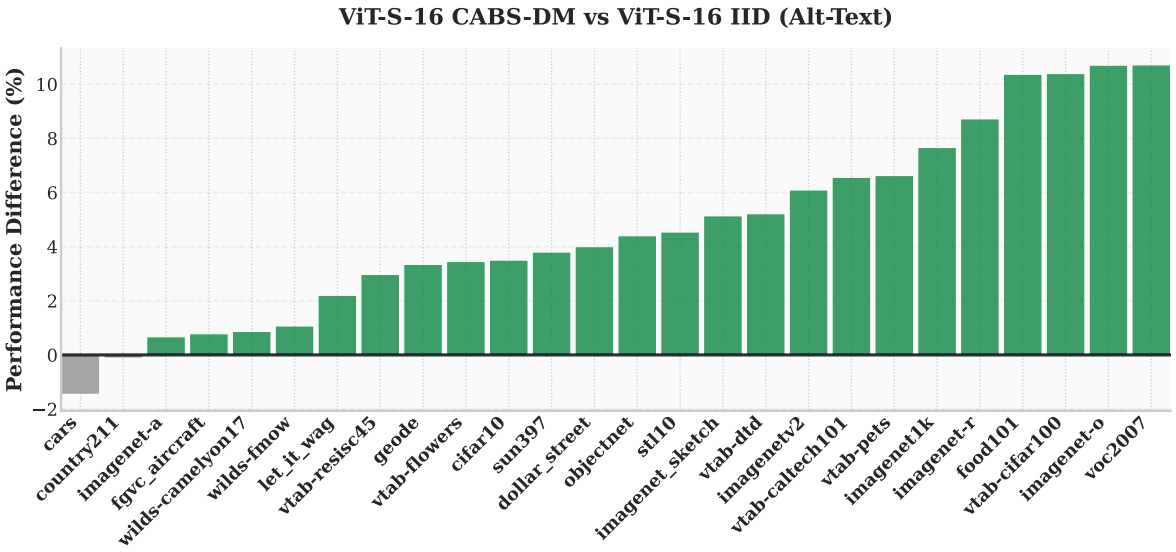

Figure 16. Dataset-wise comparisons for all benchmarks for CLIP ViT-S/16 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

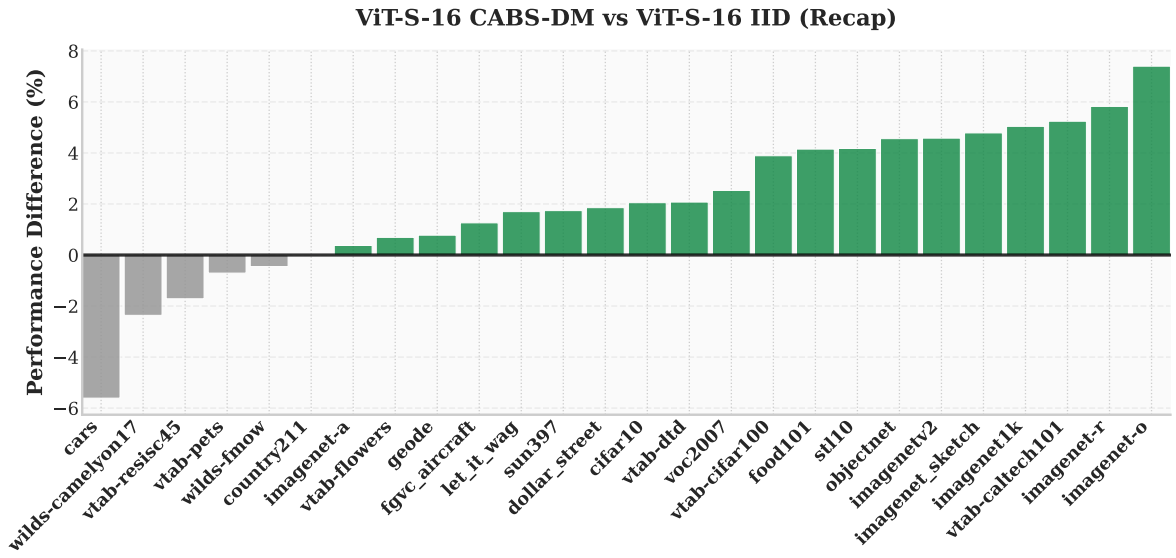

Figure 17. Dataset-wise comparisons for all benchmarks for CLIP ViT-S/16 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

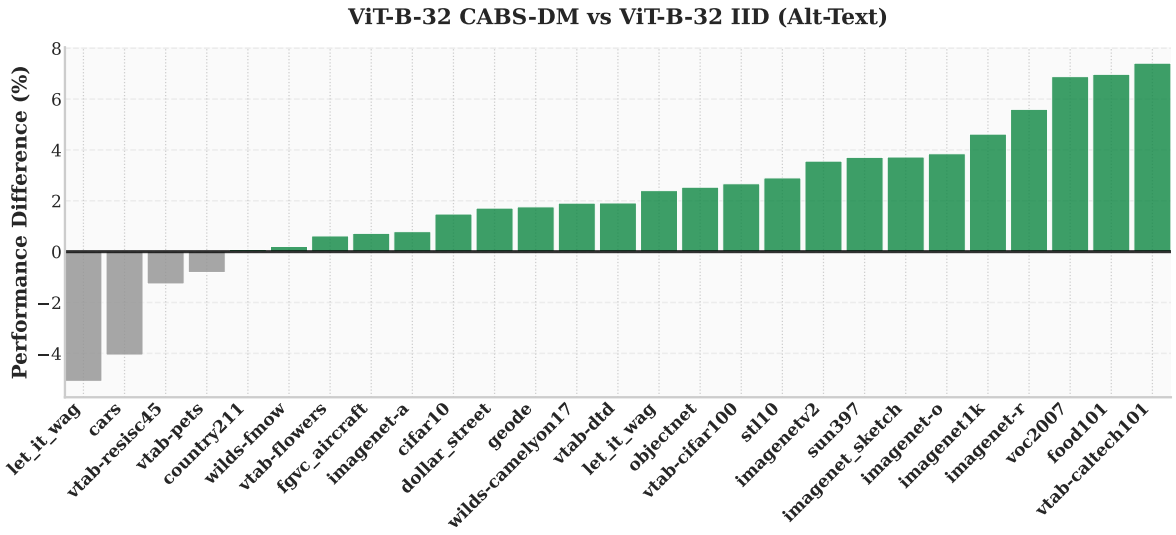

Figure 18. Dataset-wise comparisons for all benchmarks for CLIP ViT-B/32 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

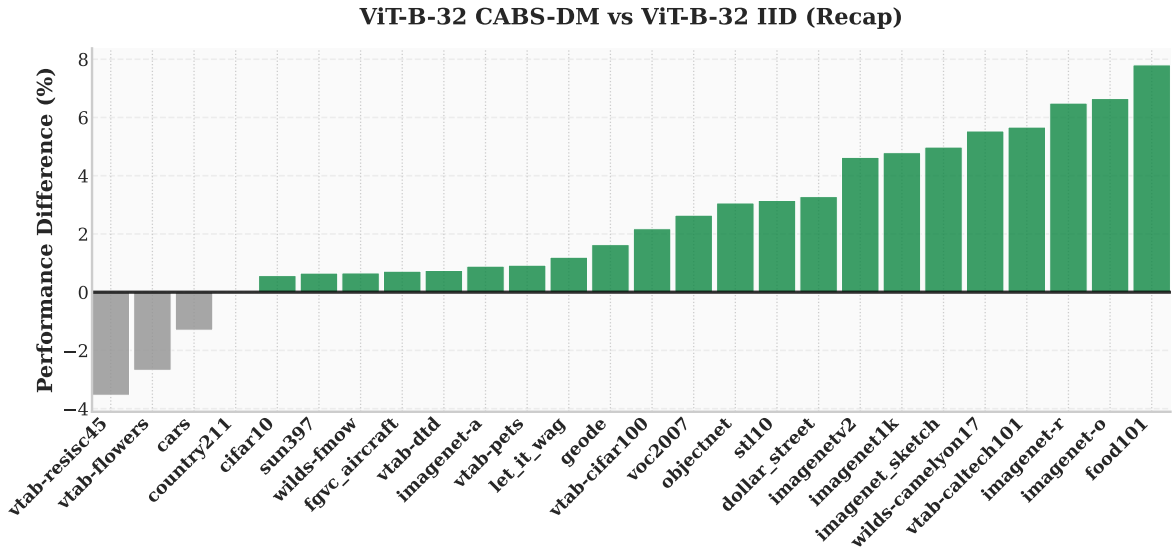

Figure 19. Dataset-wise comparisons for all benchmarks for CLIP ViT-B/32 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

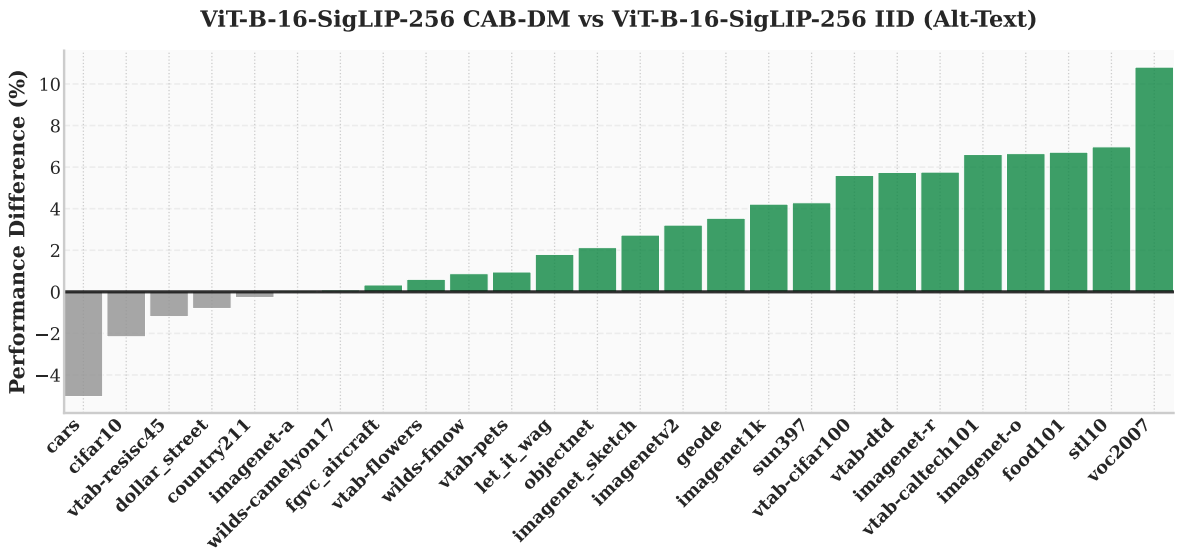

Figure 20. Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

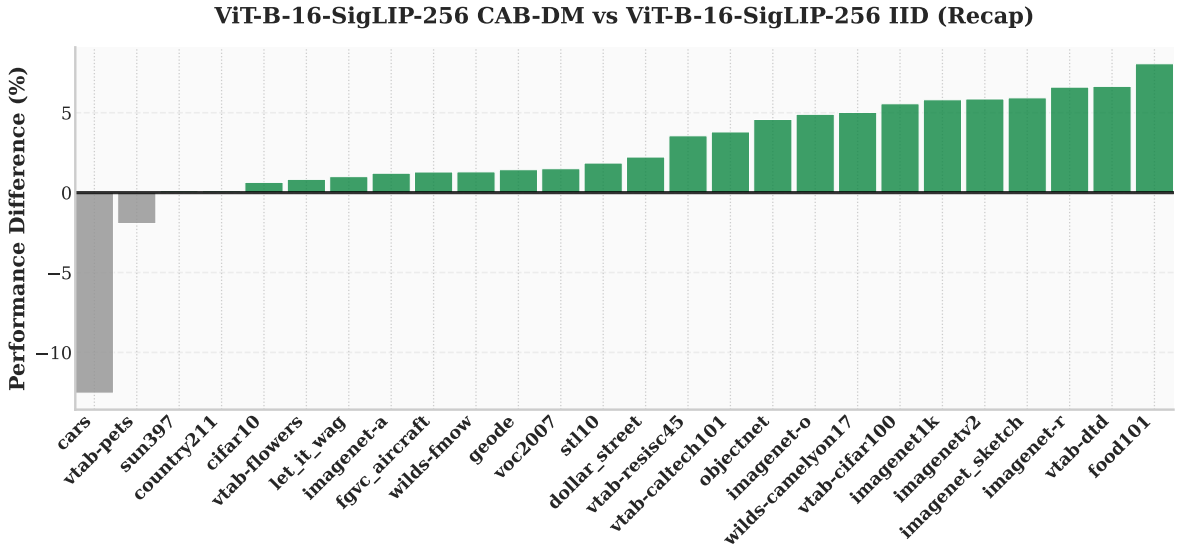

Figure 21. Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

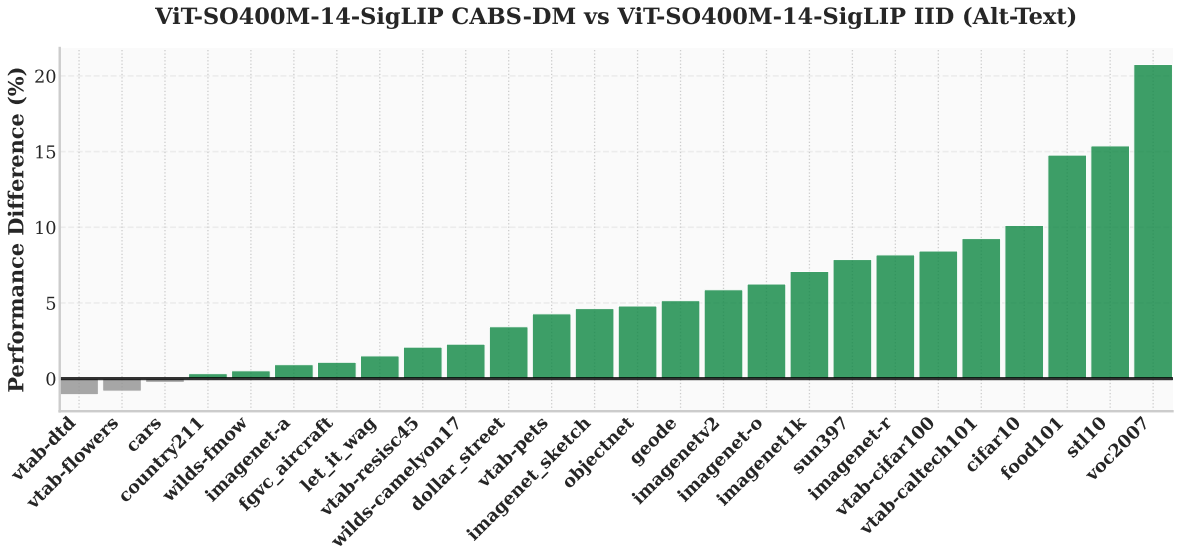

Figure 22. Dataset-wise comparisons for all benchmarks for SigLIP ViT-SO400M-14 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

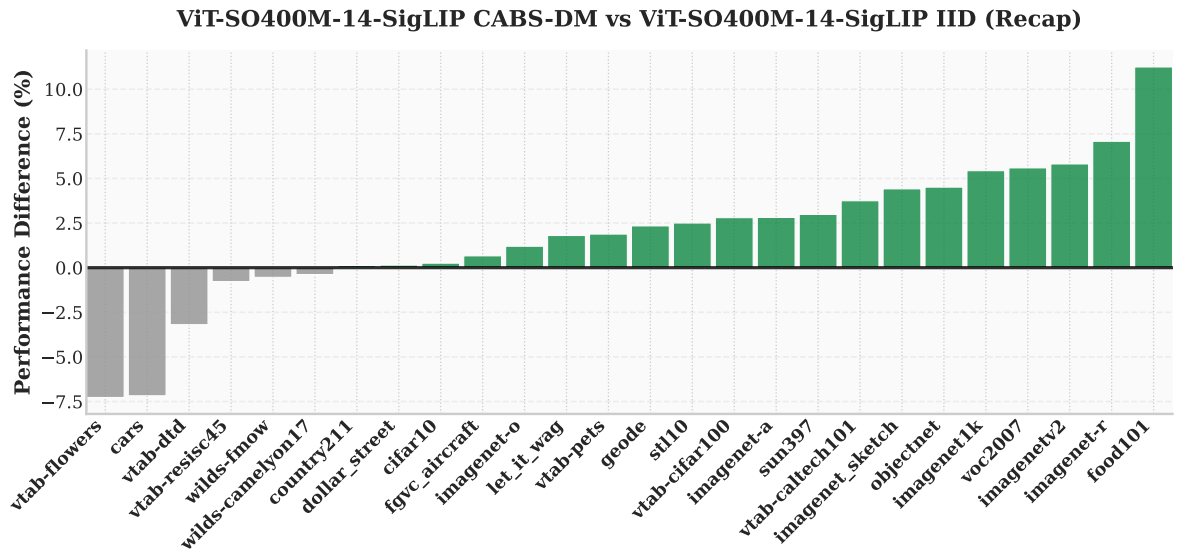

Figure 23. Dataset-wise comparisons for all benchmarks for SigLIP ViT-SO400M-14 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

I.2. MetaCLIP: Further Details

In this section, we extend our analysis on MetaCLIP offline data curation from Sec. 4.3. We first adopt the concept balancing threshold of $20,000$ from [87] and filter DataConcept accordingly. Note, again, that we do not adopt the concepts curated by the original work, instead we use the $12,253$ concept vocabulary $\mathcal{V}$. This results in a 14M filtered dataset.

Training a ViT-B/32 CLIP model with IID sampling on this filtered pool for a total of 128M samples seen results in an ImageNet accuracy of 15.1%, which underperforms standard IID training over the unfiltered pool. Thus, we adopt a modified curation strategy to match the filtered dataset size of worst-case repeats of CABS over various filter ratios. Using this strategy, for filter ratio $f = \{0.5, 0.75, 0.8\}$, we obtain an effective per-epoch samples-seen count of $\mathrm{D}_{\mathrm{filter}} = \{64\mathrm{M}, 32\mathrm{M}, 25.6\mathrm{M}\}$. We obtain the above datasets based on concept balancing using thresholds of $\tau_{\mathrm{MetaCLIP}} = \{600\mathrm{K}, 110\mathrm{K}, 70\mathrm{K}\}$.

We compare CLIP ViT-B/32 models trained using these filtered datasets with CABS-DM, with an additional probe into SigLIP ViT-B-16/256 at $f = 0.8$ (25.6M samples). Finally, even though MetaCLIP is the appropriate baseline to compare CABS-DM with, we also show that CABS-FM outperforms MetaCLIP for both CLIP ViT-B/32 and SigLIP ViT-B-16/256 at $f = 0.8$ (25.6M samples).

Table 21. **Retrieval Results.** Comparing IID sampling, MetaCLIP curation and CABS-FM on MSCOCO and Flickr30k with averaged retrieval score.

Method	**Captions**	MSCOCO	Flickr30k	**Avg(Ret)**
		ViT-B-32		
IID	`alt`	9.7	16.2	12.9
MetaCLIP	`alt`	8.7	11.6	9.7
CABS-FM	`alt`	**11.0**	**21.9**	**16.5**
		ViT-B-16-SigLIP-256		
IID	`alt`	11.1	18.9	15.0
MetaCLIP	`alt`	8.1	12.3	10.2
CABS-FM	`alt`	**12.3**	**23.9**	**18.1**

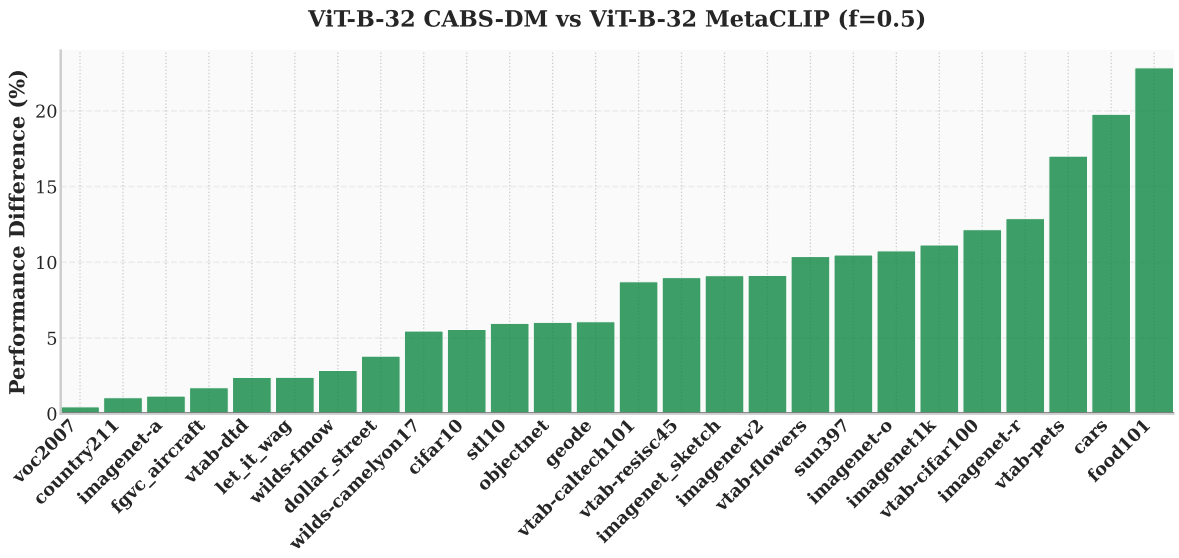

Figure 24. Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.5$) and MetaCLIP curation on alt-text.

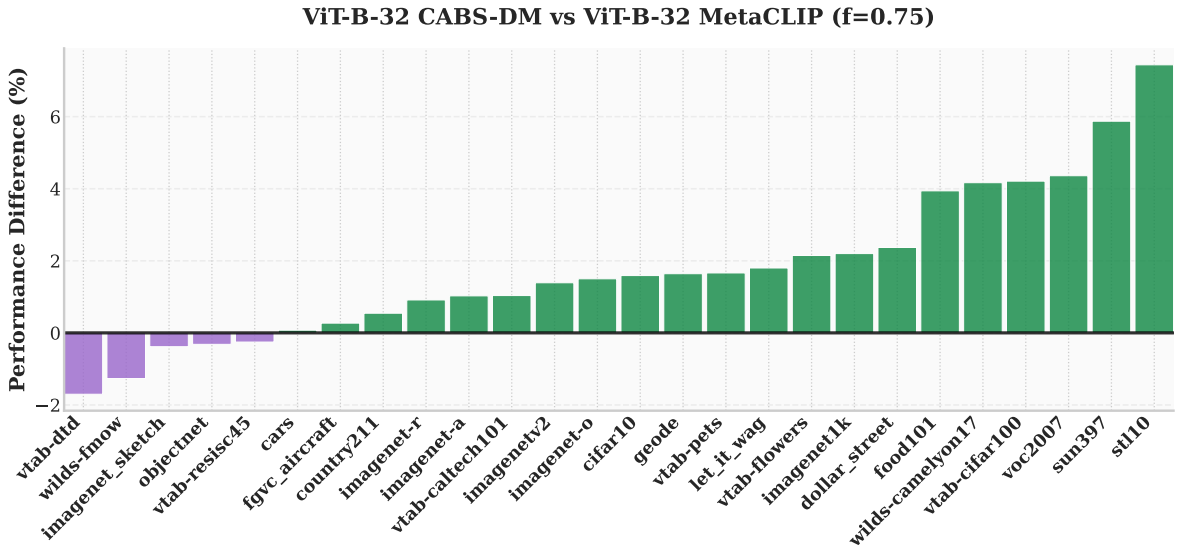

Figure 25. Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.75$) and MetaCLIP curation on alt-text.

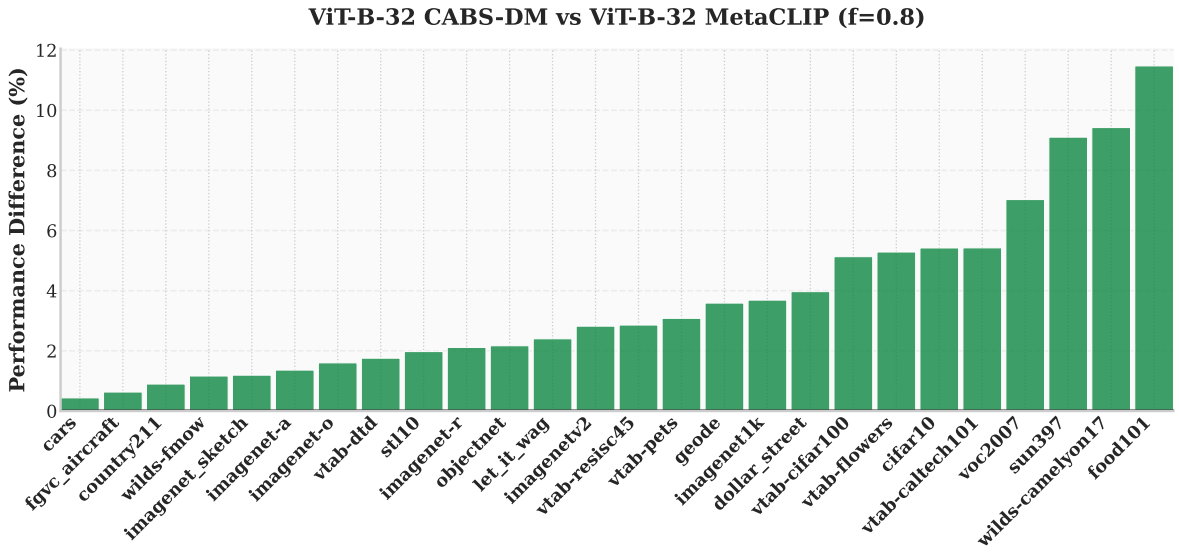

Figure 26. Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.8$) and MetaCLIP curation on alt-text.

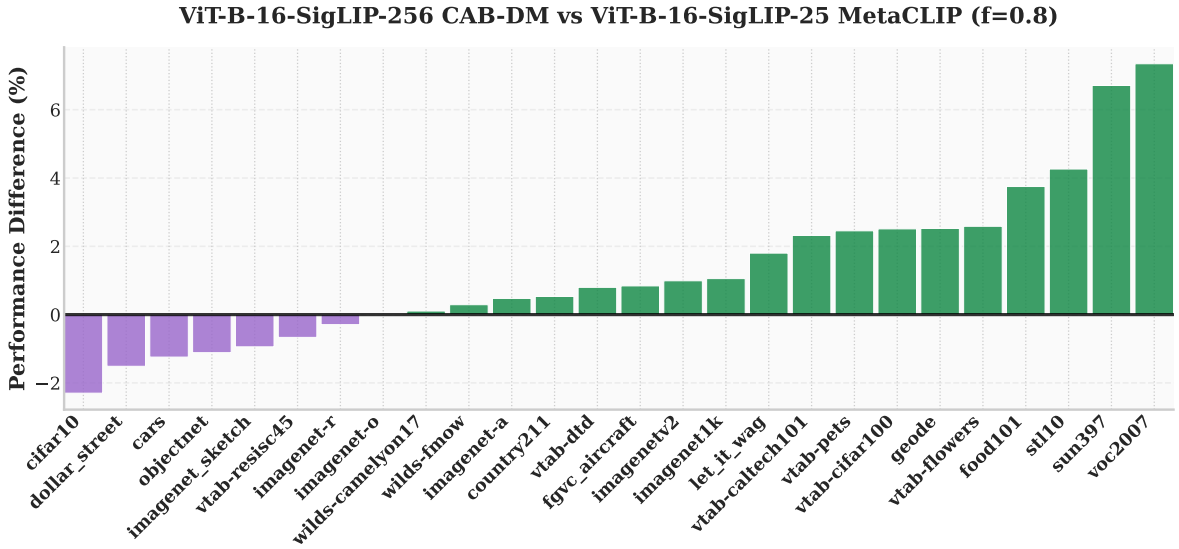

Figure 27. Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM ($f = 0.8$) and MetaCLIP curation on alt-text.