arXiv:2511.20643v1 [cs.CV] 25 Nov 2025

# Concept-Aware Batch Sampling Improves Language-Image Pretraining

Adhiraj Ghosh[1] Vishaal Udandarao[1,2*] Thao Nguyen[3*] Matteo Farina[1,4*] Mehdi Cherti[5]
Jenia Jitsev[5] Sewoong Oh[3] Elisa Ricci[4] Ludwig Schmidt[6] Matthias Bethge[1]
[1]Tübingen AI Center, University of Tübingen [2]University of Cambridge [3]University of Washington
[4]University of Trento [5]LAION [6]Stanford University

ProjectPage Code DataConcept

## Abstract

What data should a vision-language model be trained on? To answer this question, many data curation efforts center on the *quality* of a dataset. However, most of these existing methods are (i) offline, i.e. they produce a static dataset from a set of predetermined filtering criteria, and (ii) concept-agnostic, i.e. they use model-based filters which induce additional data biases. In this work, we go beyond such offline, concept-agnostic methods and advocate for more flexible, task-adaptive *online concept-based curation*. Our first contribution is DataConcept, a collection of 128M web-crawled image-text pairs annotated with fine-grained details about their concept composition. Building on DataConcept, we introduce **C**oncept-**A**ware **B**atch **S**ampling (CABS), a simple yet effective batch-sampling framework that flexibly constructs batches on-the-fly based on specific target distributions. We propose two variants: (i) Diversity Maximization (CABS-DM) to curate batches with a broad coverage of available concepts, and (ii) Frequency Maximization (CABS-FM) to curate batches with high object multiplicity. Through extensive evaluations across 28 benchmarks, we demonstrate that our CABS method significantly benefits CLIP/SigLIP model classes and yields highly performant models. Overall, CABS represents a strong open-source alternative to proprietary online data curation algorithms, enabling practitioners to define custom concept distributions that optimize for specific downstream tasks.

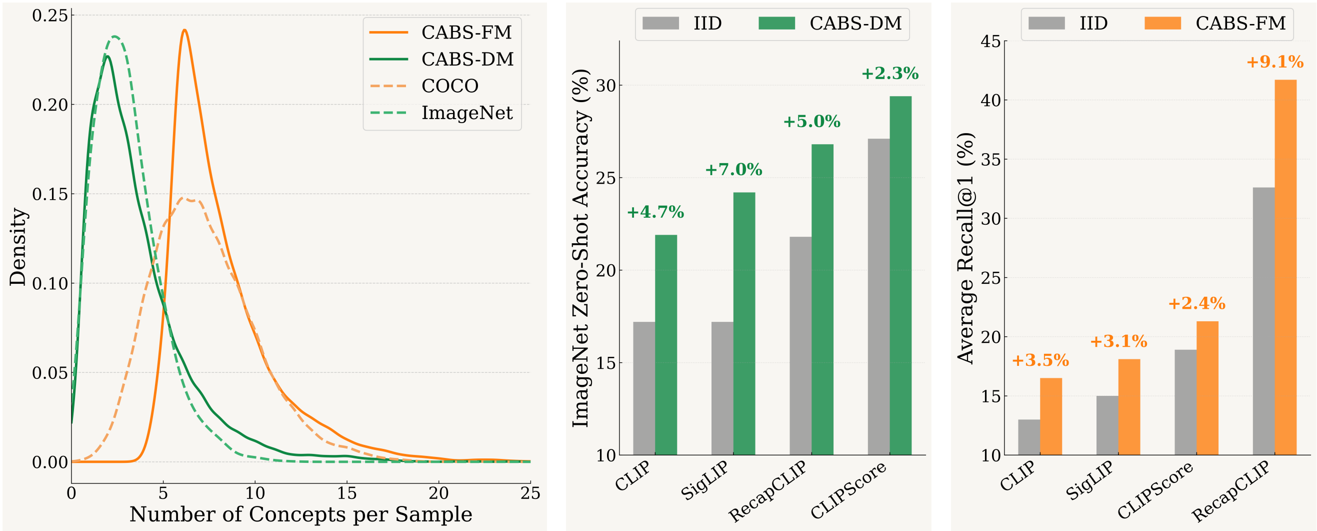


Figure 1: **Task-adaptive, steerable, Concept-Aware Batch Sampling (CABS).** The per-sample concept multiplicities *(left)* of MSCOCO retrieval and ImageNet classification train sets depict their divergent distributional properties. By only modifying a simple scoring function, CABS can flexibly adapt to different target tasks (details in Sec. 3). Both our classification-optimized (CABS-DM, see Sec. 4) and retrieval-optimized (CABS-FM, see Sec. 5) variants outperform IID sampling by large margins, across several experimental configurations.

---

[*] Equal contribution. Order decided by increasing performance on GSM8K.

## 1 Introduction

Web-scale pretraining datasets underlie the impressive generalization capabilities of vision-language models (VLMs). The advent of CLIP (Radford et al., 2021), trained on 400M image-text pairs, motivated the open development of billion-scale datasets like LAION-5B (Schuhmann et al., 2022) or DataComp-12.8B (Gadre et al., 2023). Although dataset size is an influential factor, their *quality* is equally important, if not more (Nguyen et al., 2022; Gadre et al., 2023; Goyal et al., 2024). To improve quality, current curation methods range from filtering according to well-defined metrics (*e.g.*, CLIP score) (Gadre et al., 2023) to synthetically augmenting the captions to be more descriptive (Nguyen et al., 2023; Li et al., 2024). However, most of the widely adopted curation strategies (*e.g.*, those benchmarked by DataComp (Gadre et al., 2023)), focus on quality only at the level of individual samples, overlooking the finer, *concept-level distribution*[1] within web-scale datasets. In other words, existing curation methods tend to be *concept-agnostic* (MetaCLIP (Xu et al., 2024) is a notable exception). Additionally, these methods operate in an *offline* manner, filtering out large portions of data, thus enforcing a *fixed* design choice: once data is discarded, it is difficult, if not impossible to repurpose the resulting subset for other curation strategies. The offline filtering regime also accelerates the depletion of available training samples, creating data scarcity that ultimately imposes a "data wall" on pretraining (Nguyen et al., 2025). Finally, concept-agnostic filtering methods often rely on state-of-the-art, but black-box, models to guide curation. This not only reduces transparency in selection criteria but also risks propagating the model's biases into the curated dataset (Hong et al., 2024; Girrbach et al., 2025). In contrast, concept-aware curation provides transparency and direct control over the composition of the final dataset.

In this work, we depart from such offline sample-level curation protocols, and instead advocate for more flexible *online concept-based curation*. Our rationale is simple: there is no "universal" notion of quality (Gururangan et al., 2022; Longpre et al., 2024), and importantly, as shown in Fig. 1 *(left)*, different downstream evaluations might bias what the optimal concept distribution should look like (Mizrahi et al., 2025; Abbas et al., 2024b). Therefore, we aim to show that incorporating concept-level information *during* pretraining, without discarding any data *a priori*, provides a complementary and effective avenue for multimodal data curation. This aligns with recent works advocating for data reuse over filtering (Nguyen et al., 2024; Pouget et al., 2024).

To achieve this goal, we introduce DataConcept: a multimodal pretraining dataset with 128M image-text pairs fully annotated with grounded concept information. In DataConcept, each sample comes with ① semantic concepts, ② bounding boxes, ③ per-concept confidence scores, and ④ concept-driven synthetic captions. With DataConcept, we ask: *How can we effectively modulate different visual concepts during vision-language pretraining?*

In order to answer this question, we introduce a new training framework: **C**oncept-**A**ware **B**atch-**S**ampling (CABS). In contrast to offline, static curation, we do *not* impose a fixed, predetermined data distribution, but rather enable flexible, task-adaptive control over *online concept-based batch creation*. Our classification-optimized variant, CABS-Diversity Maximization (CABS-DM), selects samples based on *concept-diversity*. This scheme is in line with MetaCLIP's approach and significantly benefits zero-shot classification (see Fig. 1 (middle)), especially over *long-tailed* evaluations. Our second variant, specifically tailored to benefit image-text retrieval tasks (see Fig. 1 (right)), is CABS-Frequency Maximization (CABS-FM). It optimizes for *concept-multiplicity*—selecting samples that encompass a higher number of objects. To our best knowledge, these CABS-variants represent the first reproducible demonstration of task-adaptive online batch sampling. Taken together, our **contributions** are:

1. DataConcept: a new, *concept-centric* pretraining dataset for VLMs comprising 128M samples. Each sample comes with fine-grained concept annotations and a concept-grounded synthetic caption. This helps enable further exploration of concept-centric data curation, a relatively underexplored avenue.

2. CABS: a new framework for vision-language pretraining that involves *online data curation* through *concept-aware batch sampling*. Paired with DataConcept, CABS enables flexible control over the concept distribution of the data used throughout training.

3. Extensive experiments with 28 tasks, 4 visual backbones, and 2 training objectives (CLIP vs SigLIP), demonstrate that CABS variants are highly effective for vision-language pretraining (up to 7% gain on ImageNet zero-shot classification and up to 9.1% gain on image-text retrieval, over strong baselines), while being complementary to existing offline data curation recipes.

---

[1] We adopt the definition of *concepts* from (Udandarao et al., 2024), i.e. objects that can be found in the wild, that we can identify and locate in image samples.

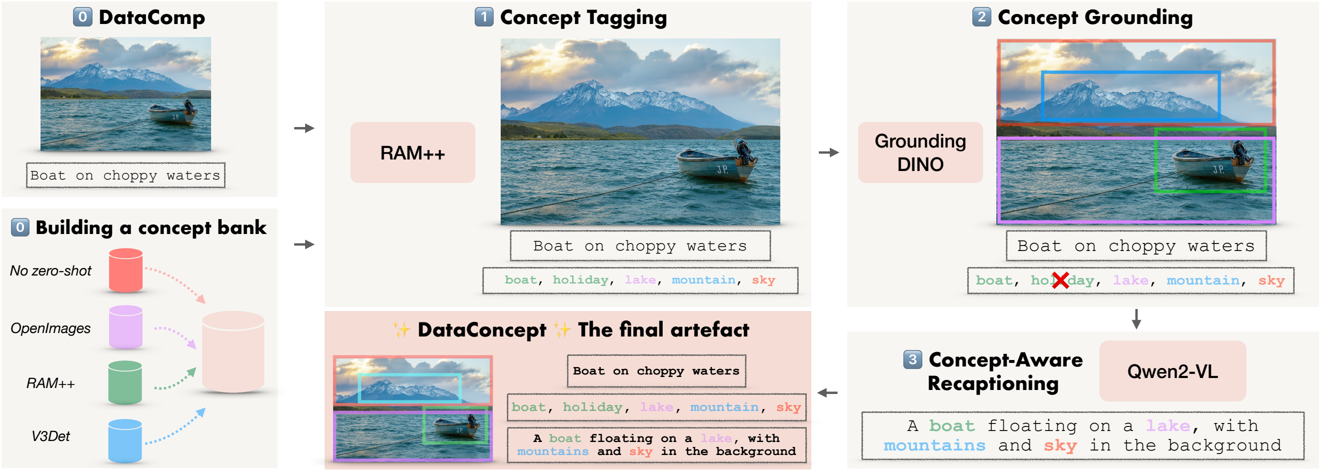

0 DataComp
Boat on choppy waters
0 Building a concept bank
No zero-shot
OpenImages
RAM++
V3Det
1 Concept Tagging
RAM++
Boat on choppy waters
boat, holiday, lake, mountain, sky
2 Concept Grounding
Grounding DINO
Boat on choppy waters
boat, holiday, lake, mountain, sky
3 Concept-Aware Recaptioning
Qwen2-VL
A boat floating on a lake, with mountains and sky in the background
✨ DataConcept ✨ The final artefact
Boat on choppy waters
boat, holiday, lake, mountain, sky
A boat floating on a lake, with mountains and sky in the background

Figure 2: **DataConcept.** We start with images from DataComp (Gadre et al., 2023) and build a concept bank $\mathcal{V}$ by merging, deduplicating, and filtering various concept sources. In ① *First-order tagging*, we assign a preliminary list of concepts (from $\mathcal{V}$) to each sample. ② We then *ground* each concept in the image, removing noise in the initial candidates. ③ Lastly, we use a model to transform alt-texts into *concept-aware captions*.

## 2 Concept-Aware Dataset Augmentation

We introduce DataConcept, our large-scale, *concept-annotated* pool of 128M image-text pairs. We will demonstrate the utility of our annotations by describing how they fit into CABS framework in the next Sec. 3.

**Initial pool.** We start with DataComp's unfiltered medium pool consisting of 128M image-text pairs (Gadre et al., 2023). We denote each sample $i$ as $(\mathcal{I}_i, \mathcal{T}_i)$. The standard protocol for downloading the dataset suffers from significant link-rot.[2] Hence, we opt for randomly sampling a 128M subset from Datacomp's XLarge pool (which consists of 12.8B samples).

**Building a concept bank.** The first step for annotating our pool is determining a *concept bank*, *i.e.*, the set of concepts that we seek to detect and tag. Previous work (Udandarao et al., 2024) curated a concept bank but it is rather limited ($4,029$) due to being constructed from 27 evaluation datasets. For broader coverage, we further source concepts from the class labels used in RAM++ (Huang et al., 2025), V3Det (Wang et al., 2023), and OpenImages (Kuznetsova et al., 2020), resulting in $19,261$ concepts, after de-duplication and safety removal (specific details and methods are provided in Sec. A.1).

**Concept tagging.** Equipped with an expansive concept bank, following Udandarao et al. (2024), we employ the RAM++ model to provide multiple concept tags for each sample in our data pool (more details in Sec. A.2).

**Concept grounding.** While RAM++ annotations provide fine-grained concept annotations per sample, we find that *(i)* RAM++ can be miscalibrated in its confidence predictions due to the extreme diversity of our concept bank, and *(ii)* RAM++ only provides a list of concept tags, without localising them in the image, which can lead to incorrect grounding. Thus, we use GroundingDINO (Liu et al., 2024) to additionally provide concept-specific bounding boxes (see Sec. A.3).

To enable precise localization of concepts, we propose two methods: *(i) Confidence seeding*: we feed RAM++ concept tags per sample (only those with at least 0.75 confidence) as seed prompts to GroundingDINO, and *(ii) Resolution ensembling*: we use Weighted Box Fusion (Solovyev et al., 2021) (see Secs. A.4 and A.5) to ensemble GroundingDINO predictions over multiple image resolutions of $\{384, 512, 800, 1000\}$. This helps reduce hallucinations without significantly affecting latency. With the two aforementioned steps, we obtain a list of concepts and their corresponding bounding boxes and confidence scores for each sample. Across all samples in the pool, we end up with $12,253$ concepts, i.e. $\mathcal{V}$, the final concept vocabulary for CABS. Each sample $i$ is now tagged with a concept set $\mathcal{C}_i$.

**Concept-aware recaptioning.** Lastly, we augment each sample $i$ with a *concept-aware* synthetic caption. Synthetic re-captions have been shown to improve training data quality by reducing noise in alt-texts (Nguyen et al.,

---

[2] We successfully downloaded only 79% of the medium-scale pool, as of September 2024.

**Algorithm 1** PyTorch-style code for CABS

```
# D=(I,T,C)=super-batch(images,texts,concepts)
# f=filter-ratio
# h=concept-aware heuristic gain function
# theta=parameter for heuristic gain function
def cabs(D, f, h):
  I, T, C = D # unpack super-batch
  B = I.size(0)
  b = (1-f)*B
  # Step1: compute heuristic scores
  scores = []
  for i in range(B):
    s_i = h(C[i], D, theta) # scoring
    scores.append(s_i)
  # Step2: select top-k samples by score
  selected_indices = topk(scores, k=b)
  # Step3: construct target batch
  I_target = I[selected_indices]
  T_target = T[selected_indices]
  return (I_target, T_target)
```

2023; Faghri et al., 2025; Fan et al., 2023). We use Qwen2-VL-7B (Wang et al., 2024) to recaption each image in a *concept-aware manner*: for each sample $i$, we provide the list of detected concepts $\mathcal{C}_i$ and the original alt-text caption $\mathcal{T}_i$ in the prompt to the model for recaptioning. The resulting generated caption is denoted as $\mathcal{R}_i$. More insights into recaption distribution and sample visualizations can be found in Sec. B

**DataConcept**. Our multi-stage pipeline, fully summarised in Fig. 2, yields our final dataset. Each image-text sample in our dataset consists of concept metadata, including concept tags with confidence scores, localised bounding-boxes, and concept-aware synthetic captions. For ease of notation, we denote each sample $i$ as $(\mathcal{I}_i, \mathcal{T}_i, \mathcal{R}_i, \mathcal{C}_i)$.

## 3 Concept-Aware Batch Sampling

Having described DataConcept, we now discuss how to leverage its concept-centric annotations to improve language-image pretraining in a task-adaptive manner.

### 3.1 Formulation

We formalize CABS as a parameterized sampling framework. Given superbatch $\mathcal{B}$ of size $B$ drawn IID from the data-pool, we define a target batch size $b < B$ controlled by filter ratio $f \in [0, 1)$, such that $b = (1 - f)B$. For each sample with concept annotations $\mathcal{C}_i$, CABS computes a score $s_i = h(\mathcal{C}_i; \mathcal{B}, \theta_h)$, where $h(.)$ is a concept-aware heuristic gain function parameterized by $\theta_h$(a set of parameters relevant to the sampling strategy), and selects sub-batch $\mathcal{B}_{\text{sub}} \subset \mathcal{B}$ of size $b$ based on these scores. The target sub-batch is constructed as $\mathcal{B}_{\text{sub}} = \text{TopK}_{i \in \mathcal{B}}(s_i, k{=}b)$. For example, if the target is IID sampling, $h(i)$ would be set to 1 for all samples in $\mathcal{B}$ and $\theta_h = \varnothing$. Sampling the top-k in this way would be equivalent to IID sampling. By allowing $h(.)$ and $\theta_h$ to be flexible, practitioners can flexibly instantiate different batch sampling strategies and induce different concept distributions in $\mathcal{B}_{\text{sub}}$ *on-the-fly* during training. This flexibility is powerful as it enables *task-adaptive* batch curation. We provide PyTorch-style pseudocode for CABS in Alg. 1.

### 3.2 Task-Adaptivity of CABS

We now demonstrate two specific cases, *zero-shot classification* and *image-text retrieval*, where the flexibility of our CABS framework enables modifying the concept distributions to be task-aware. Prior work (Abbas et al., 2024b) argues that classification and retrieval benefit from distinct data curation strategies. However, they only perform offline curation and do not disclose details of their methods. This motivates us to develop concrete instantiations of CABS that can accommodate different capabilities tested by classification and retrieval benchmarks.

Table 1: **Parameters of CABS variants.** We indicate the heuristic function $h(\cdot)$ and parameters $\theta_h$ used for CABS-DM (see Sec. 4) and CABS-FM (see Sec. 5), and if the score is dependent on current state of sub-batch.

| Method | $h(.)$ | $\theta_h$ | Dependent? |
|---|---|---|---|
| IID | 1 | $\varnothing$ | ✗ |
| CABS-DM | Eq. (1) | $t_c$ | ✓ |
| CABS-FM | $\lvert\mathcal{C}_i\rvert$ | $\varnothing$ | ✗ |

Zero-shot classification assesses whether a model has learned discriminative features for different classes. Under IID sampling and concept-imbalanced training batches, common concepts are over-represented, resulting in under-optimization for rare concepts, and consequently, poor long-tailed performance (He and Garcia, 2009; Zhao and Zhang, 2014). By constructing batches with more uniform distributions, a model would learn stronger representations for rare concepts, yielding improved generalization on long-tailed classification. In contrast, retrieval benchmarks test multi-object compositional understanding, requiring models to align rich textual descriptions to complex visual scenes (images with multiple concepts). By constructing batches enriched with similarly complex samples, each encompassing multiple concepts, models would generalize better to the compositional nature of retrieval. Given this, we develop two CABS algorithms (Tab. 1):

- **Diversity Maximization**: balance the concept distribution, focusing on uniform concept coverage (Sec. 4).
- **Frequency Maximization**: prioritize samples with the highest concept counts (Sec. 5).

**Empirical Justification.** To validate that classification and retrieval tasks exhibit substantially different concept distributions, we collect 4,096 random samples from MSCOCO (retrieval) and ImageNet (classification) and visualize their per-sample concept counts, following the same protocol used to construct DataConcept, by generating sample-level tags from RAM++ and then prompting GroundingDINO with them to obtain the final annotations. From Fig. 1 (left), we observe that ImageNet images tend to contain single objects, while MSCOCO naturally exhibits multi-object scenes. These characteristics are then approximated by the samples selected by our two CABS variants, further demonstrating the power and flexibility of task-adaptive batch curation. This also highlights the potential of analyzing salient task characteristics and shaping training distributions accordingly, as shown in Mizrahi et al. (2025).

### 3.3 Experimental Setup

**Models.** We train a ViT-B-32 (Dosovitskiy et al., 2020) CLIP using 224 image-resolution and ViT-B-16 SigLIP (Zhai et al., 2023) at 256 resolution. We further test CABS by training ViT-S-16 CLIP and ViT-SO400M-14 SigLIP (Alabdulmohsin et al., 2023) models in Sec. D.2.

**Data.** We experiment with two variants of DataConcept: one using noisy alt-texts ($\mathcal{T}_i$) and another with our *concept-aware re-captions* ($\mathcal{R}_i$). Note that IID sampling with alt-text captions corresponds to DataComp's default setup (Gadre et al., 2023).

**Evaluation Benchmarks.** Following Udandarao et al. (2025), we consider a diverse pool of 25 classification and 2 image-text retrieval benchmarks, spanning fine-grained, object-centric, and scene-centric categories. Additionally, to assess the effectiveness of our models in long-tailed settings, we evaluate on the "Let-It-Wag!" test set from (Udandarao et al., 2024).

**Training.** We fix the training budget to be 128M samples seen; additional findings for higher budgets (1.28B samples seen) are described in Sec. 6. Note that we closely follow the hyperparameters set by DataComp for fair comparison, including a batch-size of 4096. The sample-level concepts $\mathcal{C}_i$ are used only for batch curation and do not contribute to the contrastive objective. Unless specified, we set the filter ratio to $f$=0.8, sampling from superbatches of size $B$=20, 480. We show performance for other filter ratios in Sec. F.

**Baselines.** We compare CABS performance with two popular online batch sampling methods, GRIT-VLP (Byun et al., 2022) and MAFA (Byun et al., 2024). Both GRIT-VLP and MAFA sample hard negatives based on embedding similarity. The key difference lies in how these similarities are computed: GRIT uses the *current* model's embeddings, while MAFA relies on those from a *pretrained* model. MAFA used BLIP for this purpose, but its smaller training budget makes comparisons unfair. To ensure parity, we instead pretrain CLIP and SigLIP on 128M samples and use their embeddings to compute MAFA similarities. Additionally, we note that, although JEST (Evans et al., 2024a) and ACID (Udandarao et al., 2025) are also relevant baselines, they are proprietary algorithms with no public implementation.

# 4 CABS with Diversity Maximization

## 4.1 Formulation

As motivated in Sec. 3.1, zero-shot classification tasks benefit from balanced concept-level supervision across batches. Given the general formulation detailed previously, we instantiate CABS with diversity maximization (CABS-DM) and its corresponding heuristic function $h_{DM}$, which scores samples iteratively such that the top-k-filtered batch approximates a uniform concept distribution. For a superbatch $\mathcal{B}$, CABS-DM assigns higher scores to samples containing under-represented concepts in $\mathcal{B}_{\text{sub}}$ and selects the top $b = (1-f)B$ samples until the frequency of each concept reaches an upper bound $t_c$, a tunable hyperparameter.

CABS-DM constructs a sub-batch by iteratively selecting samples that maximize a gain function $h_{DM}(i)$ and updating the sub-batch concept count $n_c$ (how many times concept $c$ has been selected) for all $c \in \mathcal{C}_i$. This process continues until the desired batch size for training is obtained, which is vastly different from an IID-sampled batch, as illustrated in Fig. 3. An average CABS-DM sub-batch contains 1.5× more concepts than an IID-sampled batch, in addition to exhibiting a mostly flat concept distribution. This helps increase diversity at the batch level. CABS-DM includes the following components:

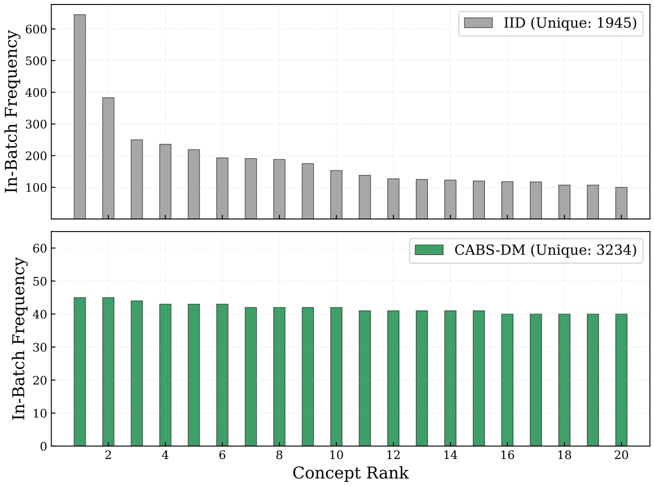


Figure 3: **Sub-batch compositions.** CABS-DM induces a near-uniform concept frequency distribution, de-biasing the distributional skew induced by IID-sampling. **Unique** indicates total unique concepts in the sub-batch: CABS-DM incorporates nearly double the concepts in the curated sub-batch, compared to IID.

**Pooling Concepts and Target Count.** For each $\mathcal{B}$, we first compute the global frequency $\mathcal{F}_c$ of each concept. We next fix the target count $t_c$ for concept $c$, *i.e.* the maximum number of times $c$ should appear in the sub-batch, to enforce approximate uniformity (following our prior notation, $\theta_h$ consists of $t_c$ in this case). In a simplified setting, if each sample comprises 1 concept, $\sum_c t_c \approx b = (1-f)B$.

**Gain Function.** The sample-level gain function is based on the current state of the sub-batch's concept distribution. Given sample $i$ with concept set $\mathcal{C}_i$, we define the gain as

$$h_{\text{DM}}(i) = \frac{1}{|\mathcal{C}_i|} \sum_{c \in \mathcal{C}_i} \begin{cases} \dfrac{t_c - n_c}{t_c} + \dfrac{1}{\mathcal{F}_c}, & \text{if } n_c < t_c, \\ 0, & \text{if } n_c \geq t_c. \end{cases} \tag{1}$$

Each concept contributes two components: a `balance gain` $({}^{(t_c - n_c)}/_{t_c})$ that prioritises under-represented concepts and a `rarity bonus` $({}^{1}/_{\mathcal{F}_c})$ that upweights long-tailed concepts. At each step, we sort all remaining super-batch samples by this score, deterministically select sample $i^\star = \arg\max_i h_{DM}(i)$, append $i^\star$ to the sub-batch, and update $n_c \leftarrow n_c + 1$ for all $c \in \mathcal{C}_{i^\star}$. If concept $c$ exceeds $t_c$, all remaining samples containing $c$ are rendered invalid. Scores $s_i$ are then generated for all relevant samples remaining in $\mathcal{B}$ using $h_{DM}$ and the sample with the highest score is incorporated into the sub-batch for the next iteration.

**Sample Selection.** CABS-DM *proceeds through a sequence of greedy maximizations to yield a balanced and diverse sub-batch.* At every iteration, it deterministically selects the sample with the highest gain, conditioned on the current sub-batch composition, without randomness, akin to an Expectation-Maximization alternating optimization between sample selection and score update. Benefits of $h_{DM}$ include (i) reproducibility across runs for the same super-batch due to deterministic selection, and (ii) gain terms jointly enforce uniform concept coverage and higher batch diversity. We provide PyTorch-style pseudocode in Alg. 2.

Table 2: **CABS-DM improves over IID.** Our method substantially outperforms IID sampling, across settings. Importantly, gains from CABS-DM extend to the long-tailed "Let-It-Wag!" test set too.

| **Method** | **Caption** | **Zero-shot Classification** | | | | **Let-it-Wag!** | **Avg (Clf)** |
|---|---|---|---|---|---|---|---|
| | | IN-Val | IN-shift | Obj | Scene | | |
| **ViT-B-32-CLIP** | | | | | | | |
| IID (Gadre et al., 2023) | `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 (Gadre et al., 2023) | `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 (Gadre et al., 2023) | `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 (Gadre et al., 2023) | `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** |

Table 3: **CABS-DM beats MetaCLIP-style curation**. Despite having similar curation objectives, we show our online concept-balanced batch sampling significantly outperforms offline curation.

| **Method** | **Zero-shot Classification** | | | | **Let-it-Wag!** | **Avg (Clf)** |
|---|---|---|---|---|---|---|
| | IN-Val | IN-shift | Obj | Scene | | |
| **ViT-B-32-CLIP** | | | | | | |
| IID (Gadre et al., 2023) | 17.3 | 15.2 | 32.3 | 36.4 | 5.1 | 28.2 |
| MetaCLIP (Xu et al., 2024) | 18.2 | 16.9 | 30.3 | 32.9 | 5.3 | 26.9 |
| CABS-DM | **21.9** | **18.6** | **34.5** | **38.0** | **7.5** | **30.7** |
| **ViT-B-16-SigLIP-256** | | | | | | |
| IID (Gadre et al., 2023) | 17.2 | 15.3 | 29.6 | 35.9 | 5.2 | 26.4 |
| MetaCLIP (Xu et al., 2024) | 20.3 | 18.9 | 30.7 | 35.3 | 5.3 | 28.0 |
| CABS-DM | **24.1** | **20.8** | **33.5** | **39.6** | **7.0** | **30.9** |

## 4.2 Improvements on Zero-shot Classification

We now comprehensively evaluate the effectiveness of CABS-DM against standard IID sampling for multimodal pretraining. As shown in Tab. 2, CABS-DM consistently delivers improvements across four different test settings. On ImageNet, CABS-DM yields substantial gains over IID sampling, with an absolute improvement of +5.0% for CLIP ViT-B-32 and +6.9% for SigLIP B-16-256. Similar trends are observed across the broader suite of benchmarks and model variants (refer to Sec. D.2), where CABS-DM boosts average accuracy. Beyond standard benchmarks, CABS-DM also enhances long-tailed recognition on Let-It-Wag! (Udandarao et al., 2024), with boosts of $1.0-2.4$%. This demonstrates CABS-DM's ability to improve both general and long-tailed performance.

Notably, we also observe consistent improvements from using our concept-aware re-captions compared to alt-texts, even with standard IID sampling. With CLIP-ViT-B/32, our re-captions lead to a +4.3% boost on ImageNet, and +4.8% for zero-shot classification. For SigLIP-ViT-B/16, the accuracy gains are as large as +11.6% and +12.2%. These results quantify the benefits of both DATACONCEPT and CABS, showcasing that *concept-aware recaptions and task-aware online curation provide the strongest gains.*

## 4.3 Improvements over State-of-the-art Methods

**MetaCLIP.** We compare CABS-DM with MetaCLIP (Xu et al., 2024), an offline curation method that aims at concept-balanced curation by first collecting 500,000 queries from WordNet synsets and Wikipedia titles, followed by matching these queries to a pool of image–text pairs via substring search in alt-texts, capping each query at 20,000 samples. To provide a fair baseline, we re-implement MetaCLIP curation based on image content, using our concept

Table 4: **CABS-DM outperforms SOTA open-source batch sampling methods.** With both CLIP-ViT-B/32 and SigLIP-ViT-B/16, CABS-DM provides significant benefits to LIP compared to GRIT-VLP and MAFA, making it more suitable for modern LIP.

| **Method** | **Zero-shot Classification** | | | | **Let-it-Wag!** | **Avg (Clf)** |
|---|---|---|---|---|---|---|
| | IN-Val | IN-shift | Obj | Scene | | |
| **ViT-B-32-CLIP** | | | | | | |
| IID (Gadre et al., 2023) | 17.3 | 15.2 | 32.3 | 36.4 | 5.1 | 28.2 |
| GRIT-VLP (Byun et al., 2022) | 17.6 | 15.0 | 31.7 | 35.6 | 6.3 | 27.5 |
| MAFA (Byun et al., 2024) | 17.0 | 15.0 | 32.2 | 35.9 | 5.6 | 27.9 |
| CABS-DM | **21.9** | **18.6** | **34.5** | **38.0** | **7.5** | **30.7** |
| **ViT-B-16-SigLIP-256** | | | | | | |
| IID (Gadre et al., 2023) | 17.2 | 15.3 | 29.6 | 35.9 | 5.2 | 26.4 |
| GRIT-VLP (Byun et al., 2022) | 17.3 | 15.1 | 30.7 | 37.3 | 5.0 | 27.2 |
| MAFA (Byun et al., 2024) | 17.2 | 15.2 | 30.7 | 36.2 | 5.1 | 27.1 |
| CABS-DM | **24.1** | **20.8** | **33.5** | **39.6** | **7.0** | **30.9** |

vocabulary $\mathcal{V}$ as the query pool and approximating the concept threshold based on the desired curated dataset size. To align with CABS-DM at $f = 0.8$ (where the full dataset is repeated 5× to match our 128M samples-seen regime), we construct a 25.6M MetaCLIP-subset and train with 5× repeats. This is achieved by using a per-concept threshold of 70,000.

Tab. 3 shows comparisons with CABS-DM, MetaCLIP and IID sampling. CABS-DM substantially outperforms MetaCLIP in zero-shot classification (+3.8%/ +2.9% gains on ImageNet and the average classification set respectively) as well as long-tailed evaluations, highlighting the performance boosts achieved with online batch curation.

**Online Batch Sampling.** After demonstrating benefits of online sampling compared to offline curation, we next compare CABS-DM to other online approaches such as GRIT-VLP (Byun et al., 2022) and MAFA (Byun et al., 2024). Tab. 4 highlights that both methods lag behind CABS-DM. We note GRIT and MAFA also struggle to outperform the IID baseline (with CLIP), but offer modest improvements with SigLIP. These observations are in line with recent works suggesting that SigLIP models benefit more from active batch sampling (Evans et al., 2024a; Udandarao et al., 2025). With SigLIP-ViT-B/16, CABS-DM improvements are up to +6.8% on ImageNet and +3.7% on average.

## 5 CABS with Frequency Maximization

### 5.1 Formulation

As described previously in Sec. 3.1, we next focus on retrieval. Retrieval benchmarks like MSCOCO and Flickr30k often consist of images with multiple objects and complex scenes (Fig. 1), necessitating changes to the design of scoring function compared to CABS-DM. This leads us to instantiate CABS with frequency maximization (CABS-FM) and its corresponding heuristic function $h_{FM}$, which scores samples based on concept count. As a result, filtered sub-batches contain samples from super-batch $\mathcal{B}$ with maximal object multiplicity, exhibiting higher scene complexity overall.

**Gain Function.** We define a simple sample-level gain function $h_{FM}(i) = |\mathcal{C}_i|$, which denotes the number of annotated classes present in sample i in DATACONCEPT. CABS-FM scores every $i \in \mathcal{B}$ by $h_{FM}(i)$, sorts samples by this value, constructs a top-k sub-batch $\mathcal{B}_{\text{sub}} = \text{TopK}_{i \in \mathcal{B}}(|\mathcal{C}_i|, k = b)$ (PyTorch-style pseudocode can be found in Alg. 3). $h_{FM}$ thus provides the model with the most concept-dense sub-batch.

### 5.2 Experiment Results

**Improvements on Image-Text Retrieval.** Following our previous CABS-DM evaluation protocol, we test CABS-FM across the full model suite using alt-text and concept-aware re-captions. As shown in Tab. 5a, CABS-FM

Table 5: **CABS-FM improves over IID and outperforms state-of-the-art online batch sampling methods.** Performance on both Flickr and MSCOCO significantly improved, demonstrating that concept multiplicity curation indeed benefits retrieval. We show significant benefits in using CABS-FM compared to other online batch sampling methods such as GRIT-VLP (Byun et al., 2022) and MAFA (Byun et al., 2024).

| Method | Captions | COCO | Flickr | Avg(Ret) |
|---|---|---|---|---|
| **ViT-B-32-CLIP** | | | | |
| IID | `alt` | 9.7 | 16.2 | 12.9 |
| CABS-FM | `alt` | **11.0** | **21.9** | **16.4** |
| 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** |

(a) Comparison with IID sampling.

| Method | COCO | Flickr | Avg(Ret) |
|---|---|---|---|
| **ViT-B-32-CLIP** | | | |
| IID | 9.7 | 16.2 | 12.9 |
| GRIT-VLP | 9.6 | 15.6 | 12.6 |
| MAFA | 9.6 | 15.5 | 12.5 |
| CABS-FM | **11.0** | **21.9** | **16.5** |
| **ViT-B-16-SigLIP-256** | | | |
| IID | 11.1 | 18.9 | 15.0 |
| GRIT-VLP | 11.6 | 19.6 | 15.6 |
| MAFA | 10.5 | 19.4 | 14.9 |
| CABS-FM | **12.3** | **23.9** | **18.1** |

(b) Comparison with SoTA batch sampling algorithms.

consistently outperforms IID sampling across all configurations, yielding gains of +3.5% and +3.1% for ViT-B-32-CLIP and ViT-B-16-SigLIP-256 (alt-text), averaged over MSCOCO and Flickr30k. These improvements further widen to +9.0% and +4.6% when training on the re-captions.

**Online Batch Sampling Methods.** In Tab. 5b, we find that CABS-FM outperforms GRIT-VLP and MAFA. Similar to the classification case, both baselines fail to surpass IID sampling for ViT-B-32-CLIP and offer only modest improvements for ViT-B-16-SigLIP-256. In contrast, CABS-FM offers large boosts, improving over GRIT-VLP by +3.9% (ViT-B-32-CLIP) and +2.5% (ViT-B-16-SigLIP-256).

## 6 Data- & Compute-Constrained Experiments

Having explored the efficacy of our CABS variants across both classification and image-text retrieval tasks, in this section, we study the benefits of CABS along another axis: *data-* vs *compute-constrained* pretraining.

**Definition.** Let $C$ denote the target compute (FLOPs), $\mathcal{D}$ the pretraining dataset, and $C_{\mathcal{D}}$ the required compute for one epoch over $\mathcal{D}$. If $C \leq C_{\mathcal{D}}$, then training is compute-constrained, i.e., the compute budget is insufficient to consume all the data. If $C > C_{\mathcal{D}}$, then training is data-constrained, i.e., samples must be repeated.

**Experimental Design.** Due to the sampling mechanism of CABS, going from a larger superbatch to a training sub-batch, all the experiments in Secs. 4.2, 4.3 and 5.2 operate under a data-constrained setting for both CABS variants. This occurs since a fraction $f$=0.8 of samples are filtered out online during training, making the *effective* samples-seen-per-epoch for CABS 5× less than IID, which instead operates with $C = C_{\mathcal{D}}$.

Table 6: **CABS-FM is also compatible with CLIP-Score filtering.** Despite the same repeat protocol as CABS-DM, we show unanimous performance gains across all benchmarks and models tested.

| Method | MSCOCO | Flickr30k | Avg(Ret) |
|---|---|---|---|
| **ViT-B-32-CLIP** | | | |
| IID | 13.8 | 24.1 | 18.9 |
| CABS-FM | **15.9** | **26.5** | **21.2** |
| **ViT-B-16-SigLIP-256** | | | |
| IID | 18.7 | 34.7 | 26.7 |
| CABS-FM | **20.1** | **36.3** | **28.2** |

Following common practices in pretraining, we increase the constraints further with two experiments: ① *less data, but higher quality*, where we keep the 128M sample budget, but filter DataConcept via CLIP Score (Schuhmann et al., 2022; Hessel et al., 2021). We keep the top 30% samples as in Gadre et al. (2023), thereby reducing the starting dataset to ∼38M samples[3]. To prevent high repeat rates, we set $f$=0.5, yielding 6.67× worst-case repeats for CABS, which are comparable to the 5× worst-case repeats induced by $f$=0.8 in Sec. 4. Note that IID sampling yields 3.33 worst-case repeats after CLIP-Score filtering. ② *long training*, where we do not filter, but rather increase the training budget to 1.28B samples seen, matching the *large* scale of DataComp. This

---

[3] We use OpenAI's CLIP ViT-L/14 model for scoring cosine similarities.

Table 7: **CABS-DM is compatible with CLIPScore filtering.** Although CABS-DM leads to more repeats, which yield diminishing returns on already curated data (Goyal et al., 2024), we generally improve over IID even with 2× more repeats across model architectures.

| **Method** | **Zero-shot Classification** | | | | **Let-it-Wag!** | **Avg (Clf)** |
|---|---|---|---|---|---|---|
| | IN-Val | IN-shift | Obj | Scene | | |
| **ViT-B-32-CLIP** | | | | | | |
| IID (Gadre et al., 2023) | 27.3 | 23.0 | 39.8 | 43.1 | 10.7 | 35.7 |
| CABS-DM | **30.1** | **25.6** | **41.8** | **44.8** | **12.7** | **37.8** |
| **ViT-B-16-SigLIP-256** | | | | | | |
| IID (Gadre et al., 2023) | 34.7 | 29.5 | <u>46.2</u> | **48.9** | 11.9 | 42.0 |
| CABS-DM | **37.5** | **32.2** | <u>46.2</u> | 48.5 | **12.6** | **42.7** |

training regime corresponds to 10× repeats for IID training and 50× worst-case repeats for CABS, given a filter ratio of $f = 0.8$.

**Less data, but higher quality.** In this regime, both CABS variants remain effective even with CLIP-score-filtered data (see Tab. 7 for CABS-DM and Tab. 6 for CABS-FM). Notably, while repeating curated data has been shown to yield diminishing returns (Goyal et al., 2024), CABS still trumps IID sampling despite using a 2× more data repeat rate.

**Long Training.** Next, we study the training dynamics under the regime where we train both IID and CABS variants with a CLIP ViT-B/32 backbone for 1.28B samples seen. As illustrated in Fig. 4, we find that as long as IID training is compute-constrained (dashed gray line), CABS significantly outperforms the vanilla IID recipe, displaying impressive 3.2× and 2× compute multipliers, which means that IID training requires 3.2× more training steps to reach CABS-DM's ImageNet performance and 2× more training steps to reach CABS-FM's average retrieval performance.

The performance gains only slightly diminish when training is far into the data-constrained regime, with CABS yielding a worst-case of 50 repeats (over 25.6M samples) and IID yielding only 10. We hypothesize this is due to the combination of a large number of repeats (50×) over a comparatively small sample pool: the original CLIP model (Radford et al., 2021), in comparison, used 32× repeats over 400M samples. However, the overall performance is still competitive to the IID baseline, even under this extreme repeat regime. These experiments confirm that our CABS method is fully compatible with ① *smaller, highly curated datasets, and* ② *pre-training on web-crawled corpora for multiple epochs.*

## 7 Related Work

**Sampling Approaches for Training Multimodal Models.** Training web-scale foundation models typically uses uniform, IID mini-batch sampling, which assigns equal weights to each sample in the training set. However, in multimodal corpora, examples differ drastically in quality (Gadre et al., 2023; Schuhmann et al., 2022; Xu et al., 2023), are possibly redundant (Abbas et al., 2023; Elazar et al., 2023; Abbas et al., 2024a; Sorscher et al., 2022; Webster et al., 2023), and exhibit skewed, long-tailed distributions across concepts (Udandarao et al., 2024; Parashar et al., 2024). Moreover, for contrastive objectives like CLIP (Radford et al., 2021), batch composition heavily shapes the learning process. In this context, uniform sampling is not neutral: it can overexpose trivial or spurious correlations and under-represent rare but informative cases. Hence, several recent approaches try to apply better batch sampling schemes to ensure more effective cross-modal learning. Early works like RHO-Loss (Mindermann et al., 2022) and Bad-Students (Evans et al., 2024b) move away from IID sampling, but they select data samples independently without considering the overall batch composition. This issue is then addressed by methods such as GRIT-VLP (Byun et al., 2022), MAFA (Byun et al., 2024), JEST (Evans et al., 2024a), B3 (Thirukovalluru et al., 2025), Falcon (Kim et al., 2025) and ACID (Udandarao et al., 2025). Our paper builds on this line of work by incorporating concept diversity into the training batch construction, an aspect missing from previous methods.

**Analyzing Concepts in Multimodal Datasets.** Understanding the composition of multimodal datasets is important for building better batch sampling methods. Early image-text datasets like CC-3M (Sharma et al., 2018),

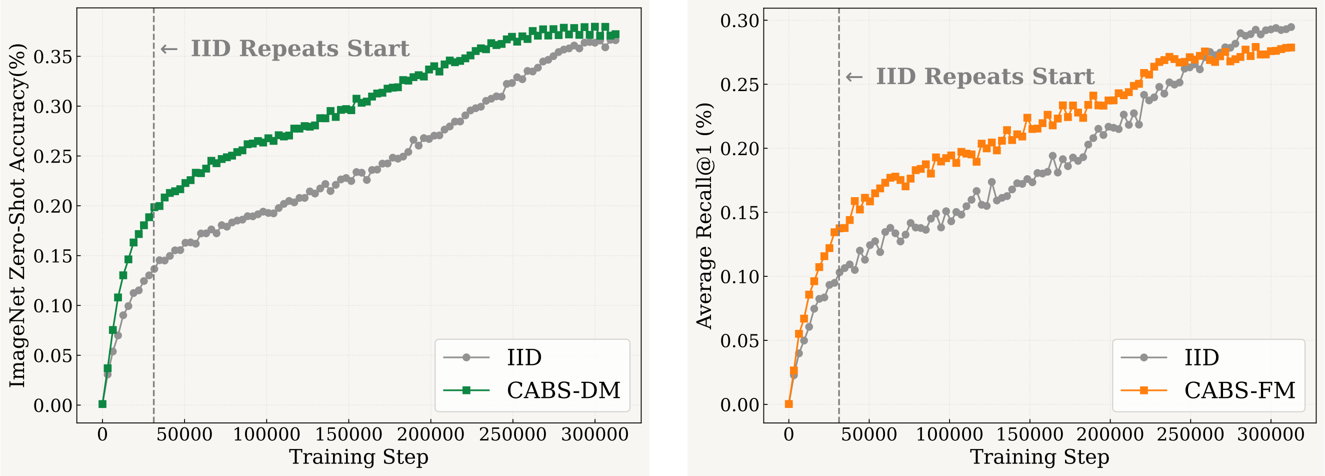


Figure 4: **CABS with longer training (1.28B samples seen).** Both CABS-DM and CABS-FM show significant boost over IID for ViT-B-32-CLIP in both compute-constrained and data-constrained regimes, the grey dashed line being the point where compute-constraint shift to data-constraint in an IID sampling regime.

CC-12M (Changpinyo et al., 2021) and YFCC-100M (Thomee et al., 2016) partially characterize their inherent concept distributions using metadata from the web sources where images are scraped from. The WebLI (Chen et al., 2022) dataset (used for training models like PaliGemma (Beyer et al., 2024) and SigLIP (Zhai et al., 2023)) was annotated using OCR models to detect objects in images. However, due to the scale of compute required for annotating recent open datasets like LAION-5B (Schuhmann et al., 2022) and DataComp-1B (Gadre et al., 2023), very few works have studied their concept distribution. Udandarao et al. (2024) tag each sample in LAION-400M with its constituent concepts by using a pretrained image-tagging model (Huang et al., 2025) and text search. Other works have proposed improving concept coverage in various ways, e.g. considering multilingual data (Nguyen et al., 2024) or recaptioning Yu et al. (2024). Our DataConcept also augments samples with fine-grained concept annotations and is designed specifically to enable explicit control over online, concept-based batch construction.

## 8 Conclusion

We investigate the role of incorporating concept-level information during vision-language pretraining, which is relatively underexplored by prior data-centric work. To this end, we introduce DataConcept, a large-scale, fully annotated pretraining dataset designed to expose concept-level annotations, and CABS, a flexible framework leveraging this information to perform online, concept-aware batch sampling during pretraining. Our extensive evaluations demonstrate the benefits of CABS over IID and other curation strategies (including existing batch sampling algorithms) across both classification and retrieval tasks, highlighting its versatility. By making DataConcept and CABS publicly available, we hope to motivate future work to incorporate concept-awareness into their data pipelines for building better VLMs.

**Limitations.** One disadvantage of CABS is the cost of concept annotations. However, this cost is amortizable as the annotated data can be re-used for training different models to do well on different tasks. It is also worth noting that the runtime of CABS increases as we increase the filtering ratio $f$ from the superbatch. Our experiments show that CABS can still offer performance benefits at low filtering ratios, where the runtime overhead is more manageable. Besides, we have not experimented with more complex multimodal architectures or large-scale training runs that mirror current state-of-the-art training setups.

**Future Work.** Our proposed framework motivates several directions to study concept-centric data curation further. One avenue could be applying CABS to fine-tuning data. In addition, future work could look into other score functions that will work well with a wide range of tasks, balancing both retrieval and classification performance. This balance could potentially be achieved through curriculum learning as well. In our experiments, we pick a score function at the start and apply it to all samples across all superbatches. One could study how to best update the score function throughout the course of training, e.g. by first prioritizing single-object images and then moving on to selecting complex scenes.

## Acknowledgements

The authors thank (in alphabetical order): Sebastian Dziadzio, Andreas Hochlehnert, Benno Krojer, Hilde Kuehne, Ameya Prabhu, Thaddäus Wiedemer, Konstantin Wilkin, Jiajun Zhang, for helpful feedback at different stages of this project. AG gratefully acknowledges LAION and the Gauss Centre for Supercomputing e.V. for funding this work by providing computing time on the JUWELS Booster at Jülich Supercomputing Centre (JSC). AG receives funding from the European Union's Horizon Europe research and innovation program under ELLIOT - Grant Agreement No 101214398. AG and VU thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support. VU was supported by a Google PhD fellowship in Machine Intelligence. MF acknowledges travel support from ELIAS (GA no 101120237). MB acknowledge financial support by the Federal Ministry of Education and Research (BMBF), FKZ: 011524085B and Open Philanthropy Foundation funded by the Good Ventures Foundation.

## References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. 10

Amro Abbas, Evgenia Rusak, Kushal Tirumala, Wieland Brendel, Kamalika Chaudhuri, and Ari S Morcos. Effective pruning of web-scale datasets based on complexity of concept clusters. *arXiv preprint arXiv:2401.04578*, 2024a. 10

Amro Abbas, Josh Wills, Haoli Yin, Paul Burstein, Ning Cao, Aldo Carranza, Alvin Deng, Priya Goyal, Pratyush Maini, Joshua McGrath, Fan Pan, Jack Urbanek, Vineeth Kada, Muhammed Razzak, Vishwa Shah, Vishruth Veerendranath, Bogdan Gaza, Ari Morcos, and Matthew Leavitt. DatologyAI Technical Deep-Dive: Image-Text Data Curation at the Billion-Sample Scale. Technical report, DatologyAI, 2024b. 2, 4, 40

Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36:16406–16425, 2023. 5

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019. 40

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 43

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 11

Cody Blakeney, Mansheej Paul, Brett W Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*, 2024. 43

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 40

Jaeseok Byun, Taebaek Hwang, Jianlong Fu, and Taesup Moon. Grit-vlp: Grouped mini-batch sampling for efficient vision and language pre-training. In *European Conference on Computer Vision*, pages 395–412. Springer, 2022. 5, 8, 9, 10

Jaeseok Byun, Dohoon Kim, and Taesup Moon. Mafa: Managing false negatives for vision-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27314–27324, 2024. 5, 8, 9, 10

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 11

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 40

Xi Chen, Xiao Wang, Soravit Changpinyo, Anthony J Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 11

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 40

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018. 40

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 40

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 40

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025. 31

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 40

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. What's in my big data? *arXiv preprint arXiv:2310.20707*, 2023. 10

Talfan Evans, Nikhil Parthasarathy, Hamza Merzic, and Olivier Henaff. Data curation via joint example selection further accelerates multimodal learning. *Advances in Neural Information Processing Systems*, 37:141240–141260, 2024a. 5, 8, 10

Talfan Evans, Shreya Pathak, Hamza Merzic, Jonathan Schwarz, Ryutaro Tanno, and Olivier J Henaff. Bad students make great teachers: Active learning accelerates large-scale visual understanding. In *European Conference on Computer Vision*, pages 264–280. Springer, 2024b. 10

Mark Everingham. The pascal visual object classes challenge 2007. In *http://www. pascal-network. org/challenges/VOC/voc2007/workshop/index. html*, 2009. 40

Fartash Faghri, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, Alexander Toshev, Oncel Tuzel, and Hadi Pouransari. Mobileclip2: Improving multi-modal reinforced training. *arXiv preprint arXiv:2508.20691*, 2025. 4

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36:35544–35575, 2023. 4

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 40

Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining. *arXiv preprint arXiv:2412.15285*, 2024. 43

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3, 5, 7, 8, 9, 10, 11, 39, 40

William Gaviria Rojas, Sudnya Diamos, Keertan Kini, David Kanter, Vijay Janapa Reddi, and Cody Coleman. The dollar street dataset: Images representing the geographic and socioeconomic diversity of the world. *Advances in Neural Information Processing Systems*, 35:12979–12990, 2022. 40

Adhiraj Ghosh, Sebastian Dziadzio, Ameya Prabhu, Vishaal Udandarao, Samuel Albanie, and Matthias Bethge. Onebench to test them all: Sample-level benchmarking over open-ended capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32445–32481, 2025. 26, 45

Leander Girrbach, Stephan Alaniz, Genevieve Smith, Trevor Darrell, and Zeynep Akata. Person-centric annotations of laion-400m: Auditing bias and its transfer to models. *arXiv preprint arXiv:2510.03721*, 2025. 2

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22702–22711, 2024. 2, 10

Suchin Gururangan, Dallas Card, Sarah Dreier, Emily Gade, Leroy Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A Smith. Whose language counts as high quality? measuring language ideologies in text data selection. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 2562–2580, 2022. 2

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 5

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021a. 40

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021b. 40

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021. 9

Rachel Hong, William Agnew, Tadayoshi Kohno, and Jamie Morgenstern. Who's in and who's out? a case study of multimodal clip-filtering in datacomp. In *Proceedings of the 4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–17, 2024. 2

Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4117–4126, 2025. 3, 11, 19, 20

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 39

Myunsoo Kim, Seong-Woong Shim, and Byung-Jun Lee. Falcon: False-negative aware learning of contrastive negatives in vision-language pretraining. *arXiv preprint arXiv:2505.11192*, 2025. 10

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 40

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 40

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 40

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020. 3, 19

Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems*, 35:9287–9301, 2022. 26

Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, et al. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*, 2024. 2, 22

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 3, 22

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, 2024. 2

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 40

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022. 10

David Mizrahi, Anders Boesen Lindbo Larsen, Jesse Allardice, Suzie Petryk, Yuri Gorokhov, Jeffrey Li, Alex Fang, Josh Gardner, Tom Gunter, and Afshin Dehghan. Language models improve when pretraining data matches target tasks. *arXiv preprint arXiv:2507.12466*, 2025. 2, 5

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, pages 850–855. IEEE, 2006. 25

Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *Advances in Neural Information Processing Systems*, 35: 21455–21469, 2022. 2

Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *Advances in neural information processing systems*, 36:22047–22069, 2023. 2, 3, 22, 32

Thao Nguyen, Matthew Wallingford, Sebastin Santy, Wei-Chiu Ma, Sewoong Oh, Ludwig Schmidt, Pang Wei W Koh, and Ranjay Krishna. Multilingual diversity improves vision-language representations. *Advances in Neural Information Processing Systems*, 37:91430–91459, 2024. 2, 11

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. Recycling the web: A method to enhance pre-training data quality and quantity for language models. *arXiv preprint arXiv:2506.04689*, 2025. 2

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 40

Chengcheng Ning, Huajun Zhou, Yan Song, and Jinhui Tang. Inception single shot multibox detector for object detection. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 549–554. IEEE, 2017. 25

Shubham Parashar, Zhiqiu Lin, Tian Liu, Xiangjue Dong, Yanan Li, Deva Ramanan, James Caverlee, and Shu Kong. The neglected tails in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12988–12997, 2024. 10, 28

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 40

Angéline Pouget, Lucas Beyer, Emanuele Bugliarello, Xiao Wang, Andreas Steiner, Xiaohua Zhai, and Ibrahim M Alabdulmohsin. No filter: Cultural and socioeconomic diversity in contrastive vision-language models. *Advances in Neural Information Processing Systems*, 37:106474–106496, 2024. 2

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 10, 40

Vikram V Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. Geode: a geographically diverse evaluation dataset for object recognition. *Advances in Neural Information Processing Systems*, 36:66127–66137, 2023. 40

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 40

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019. 19

Karsten Roth, Vishaal Udandarao, Sebastian Dziadzio, Ameya Prabhu, Mehdi Cherti, Oriol Vinyals, Olivier Hénaff, Samuel Albanie, Matthias Bethge, and Zeynep Akata. A practitioner's guide to continual multimodal pretraining. *arXiv preprint arXiv:2408.14471*, 2024. 43

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 2, 9, 10, 11

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 10

Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021. 3, 22, 24, 25

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 10

Raghuveer Thirukovalluru, Rui Meng, Ye Liu, Mingyi Su, Ping Nie, Semih Yavuz, Yingbo Zhou, Wenhu Chen, Bhuwan Dhingra, et al. Breaking the batch barrier (b3) of contrastive learning via smart batch mining. *arXiv preprint arXiv:2505.11293*, 2025. 10

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 11, 40

Lukas Tuggener, Raphael Emberger, Adhiraj Ghosh, Pascal Sager, Yvan Putra Satyawan, Javier Montoya, Simon Goldschagg, Florian Seibold, Urs Gut, Philipp Ackermann, et al. Real world music object recognition. *Transactions of the International Society for Music Information Retrieval*, 7(1):1–14, 2024. 22

Vishaal Udandarao, Ameya Prabhu, Adhiraj Ghosh, Yash Sharma, Philip Torr, Adel Bibi, Samuel Albanie, and Matthias Bethge. No" zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 3, 5, 7, 10, 11, 19, 20, 21, 28, 30, 40

Vishaal Udandarao, Nikhil Parthasarathy, Muhammad Ferjad Naeem, Talfan Evans, Samuel Albanie, Federico Tombari, Yongqin Xian, Alessio Tonioni, and Olivier J Hénaff. Active data curation effectively distills large-scale multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14422–14437, 2025. 5, 8, 10, 40

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019. 40

Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19844–19854, 2023. 3, 19

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 4, 31, 32

Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. On the de-duplication of laion-2b. *arXiv preprint arXiv:2303.12733*, 2023. 10

Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6609–6618, 2019. 19

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, 2016. 40

Hu Xu, Saining Xie, Po-Yao Huang, Licheng Yu, Russell Howes, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Cit: Curation in training for effective vision-language data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15180–15189, 2023. 10

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 7, 50

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2: 67–78, 2014. 40

Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14022–14032, 2024. 11

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019. 40

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 5, 11

Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024. 19, 20

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond random sampling: Efficient language model pretraining via curriculum learning. *arXiv preprint arXiv:2506.11300*, 2025. 43

Peilin Zhao and Tong Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*, 2014. 5

Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haian Huang. An open and comprehensive pipeline for unified object grounding and detection. *arXiv preprint arXiv:2401.02361*, 2024. 26

# Concept-Aware Batch Sampling Improves Language-Image Pretraining

Supplementary Material

**A DataConcept Curation: Further Details** **19**
A.1 Vocabulary Construction . . . 19
A.2 Object Tagging . . . 20
A.3 Object Detection . . . 22
A.4 Weighted Box Fusion for Ensembling Bounding Boxes . . . 24
A.5 Ensembling: Quantitative Results . . . 26
A.6 Concept Distribution . . . 28

**B Concept-aware Recaptioning** **31**
B.1 Selecting the Recaptioning VLM . . . 31
B.2 Caption Quality . . . 32
B.3 Qualitative Evaluation: Visualization Results . . . 33

**C CABS: More Details** **37**
C.1 CABS-DM . . . 37
C.2 CABS-FM . . . 38
C.3 Hyperparameters . . . 39

**D Extended Benchmark Performance** **40**
D.1 Evaluation Suite: Further Details . . . 40
D.2 Full Model Suite . . . 41

**E Continual Pretraining** **43**

**F Ablation on Filter Ratios** **44**

**G Fine-grained Benchmark Performance** **45**
G.1 Expanded Analysis . . . 45
G.2 MetaCLIP: Further Details . . . 50

# A DataConcept Curation: Further Details

## A.1 Vocabulary Construction

**Scaling Concept Vocabulary:** We scale up the tag generation pipeline of RAM++ (Recognize Anything) (Zhang et al., 2024; Huang et al., 2025) 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 (Wu et al., 2019), 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 Udandarao et al. (2024) as well as 200 classes from the rare classes subset of OpenImages (Kuznetsova et al., 2020). Finally, we also adopt and filter the vocabulary pool from V3Det (Wang et al., 2023), 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, occuring 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 (Reimers and Gurevych, 2019). 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 US 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 $19,261$.

## 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. (2024) used RAM++ Zhang et al. (2024); Huang et al. (2025) 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. (2024), 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.

<table>
<tr><th></th><th colspan="3">RAM++ Threshold</th></tr>
<tr><th></th><th>0.7</th><th>0.75</th><th>0.8</th></tr>
<tr><td> </td><td>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</td><td>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</td><td>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</td></tr>
<tr><td> </td><td>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</td><td>bicycle | man | guy | cyclist | ride | bike | race | | shirt | road | bicycle helmet | biker | yellow | Bicycle model | pedelec | tall bike | road bicycle</td><td>bicycle | man | guy | cyclist | ride | race | bicycle helmet | yellow | pedelec</td></tr>
<tr><td> </td><td>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</td><td>street | motorbikes | car | motorcycle | road | crowd | traffic | vehicle | parade | flag | ride | roadway | city street | march</td><td>car | motorcycle | road | crowd | traffic | parade | flag | roadway</td></tr>
<tr><td> </td><td>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</td><td>butterfly | white | blue | flower | picture | beautiful | hydrangea | hydrangea macrophylla | celastrina</td><td>butterfly | white | blue | flower | hydrangea | celastrina</td></tr>
<tr><td> </td><td>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</td><td>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</td><td>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</td></tr>
</table>

Figure 5: **Qualitative Results with different RAM++ thresholds**. While Udandarao et al. (2024) found 0.7 to be the suitable RAM++ threshold, 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 (Liu et al., 2024) 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 (Nguyen et al., 2023; Li et al., 2024) 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) Solovyev et al. (2021) 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 (Tuggener et al., 2024). 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 Sec. 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.

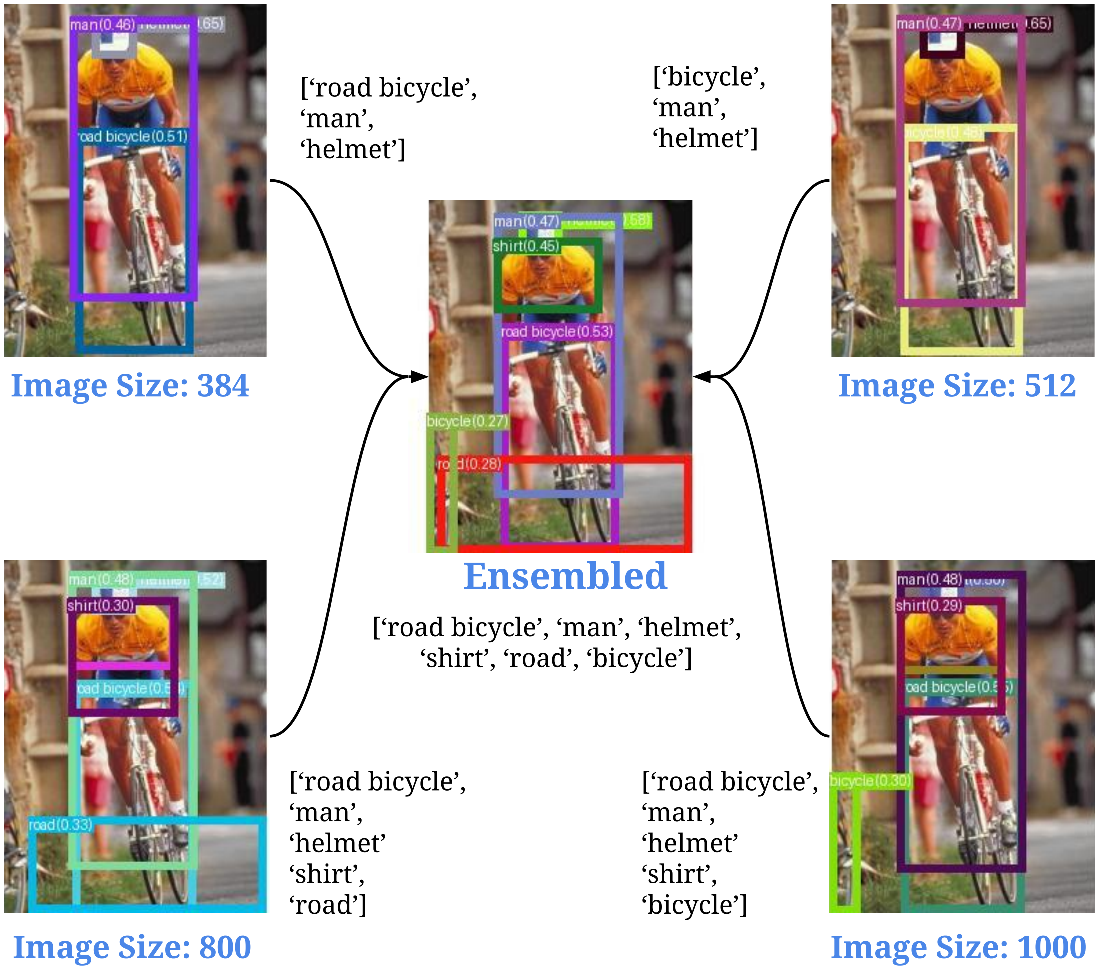


Figure 6: **Ensembling bounding boxes provides the best detection predictions for DataConcept.** Using Weighted Box Fusion, we are able to detect single instances(no overlap of bounding boxes) of all relevant objects in an image.

## A.4 Weighted Box Fusion for Ensembling Bounding Boxes

**Weighted Box Fusion**(WBF) (Solovyev et al., 2021) 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 (Neubeck and Van Gool, 2006), Non-Maximum Weighted (NMW) method (Ning et al., 2017), etc. NMS completly exclude 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 (Solovyev et al., 2021), 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 $T_{\text{post}}$ (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. (2025), 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 (Li et al., 2022), a rigorous benchmark of 13 and 35 class variants comprisinng several varieties of image resolutions designed to assess model performance within real-world contexts (Zhao et al., 2024). 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% (Zhao et al., 2024). 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 Tab. 8 and combinations of resolutions in Tab. 9. We show consistently that ensembling across all 4 resolutions provides the best bounding boxes for annotating DataConcept.

Table 8: **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 9: **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 | 0.41 | 0.40 | 0.41 | 0.41 | 1000×750 |
| AerialMaritimeDrone_tiled | 0.48 | 0.48 | 0.40 | 0.52 | 0.49 | **0.55** | 800×600 |
| ChessPieces | 0.12 | 0.16 | 0.17 | 0.16 | 0.17 | 0.17 | 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 | 0.70 | 0.64 | 0.60 | 0.66 | 0.60 | 0.70 | 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 (Sec. A.6.1) and ② Sample-level concept count distribution (Sec. 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 Udandarao et al. (2024); Parashar et al. (2024). 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.

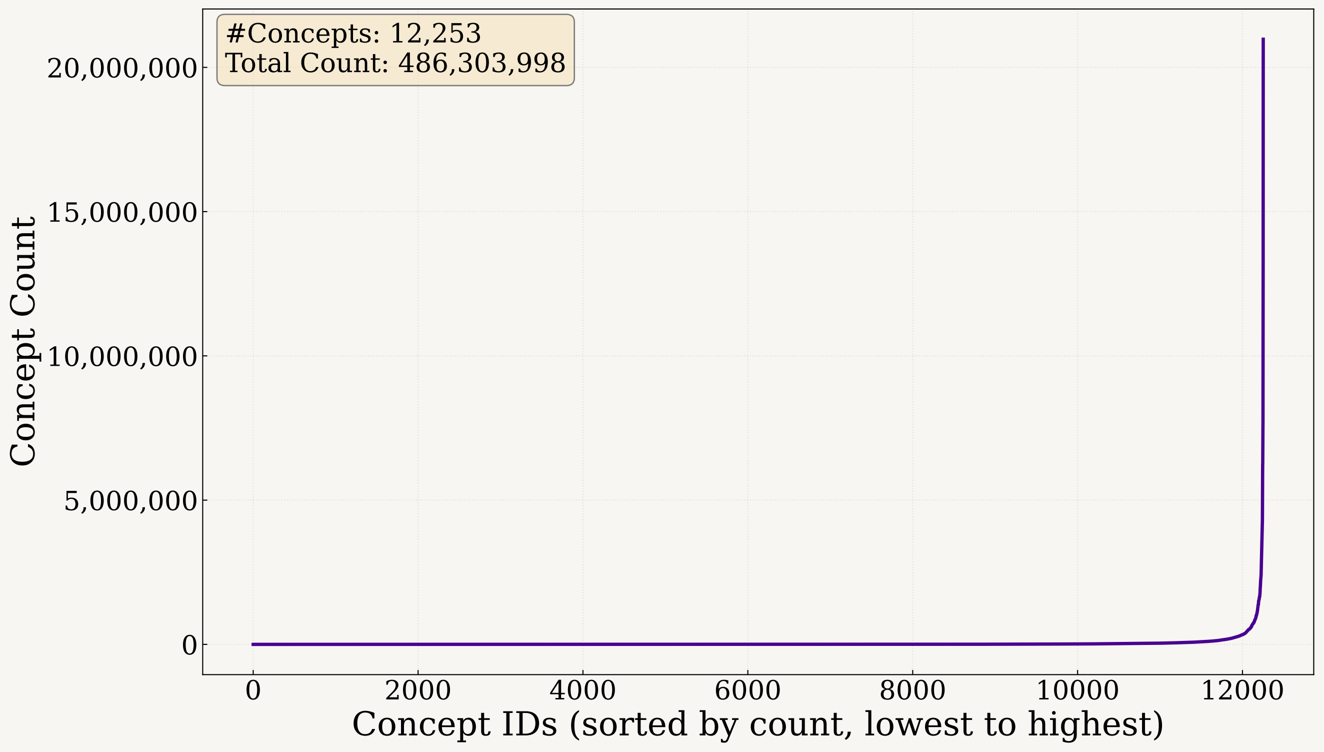


Figure 7: **What is the distribution of concepts in web-scale pretraining datasets?** We demonstrate the distribution of concept counts in DataConceptafter annotations using GroundingDINO. Indeed, DataConceptis strongly long-tailed with 86 concepts having more than 1 million annotations, 685 concepts 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. (2024) 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.

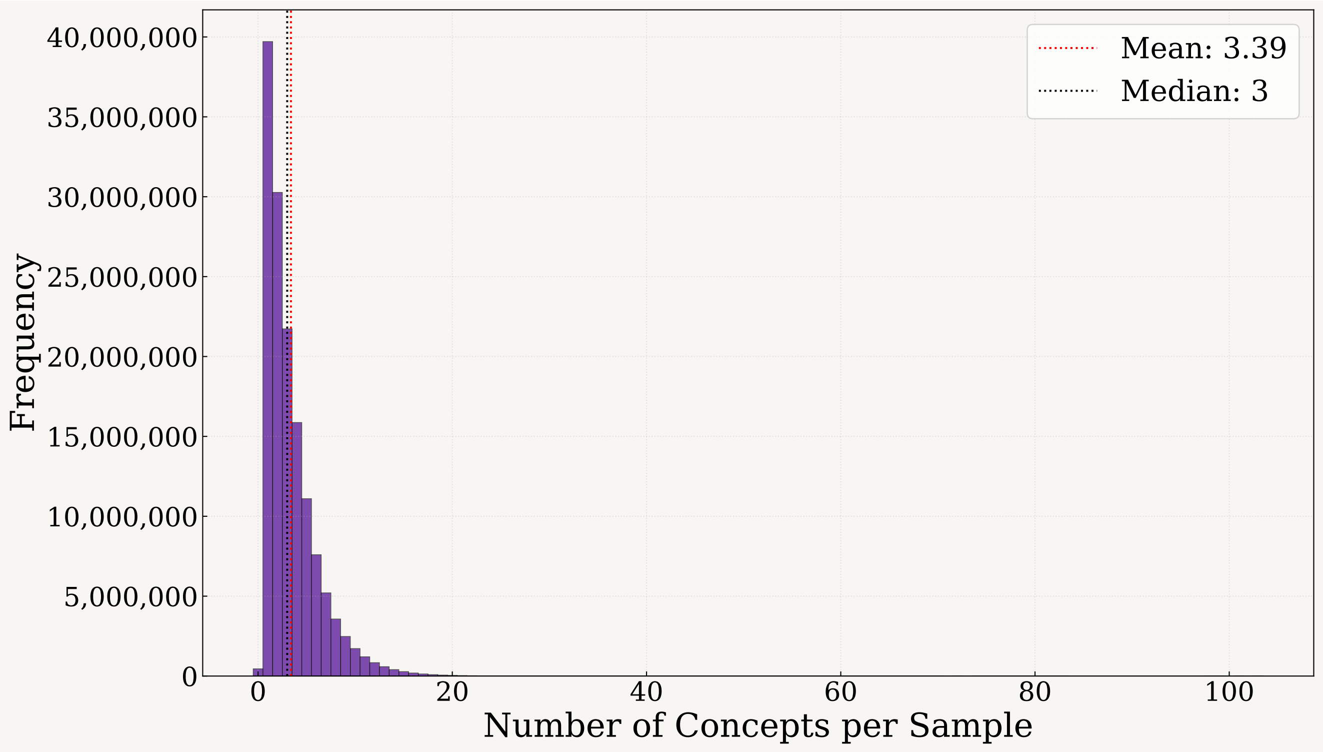


Figure 8: **What is the complexity of DataConceptsamples based on visual concepts?** We demonstrate the distribution of concept counts per sample after annotations using GroundingDINO. Note that GroundingDINO can predict a concept many times, hence these numbers reflect the total number of concepts detected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

# B Concept-aware Recaptioning

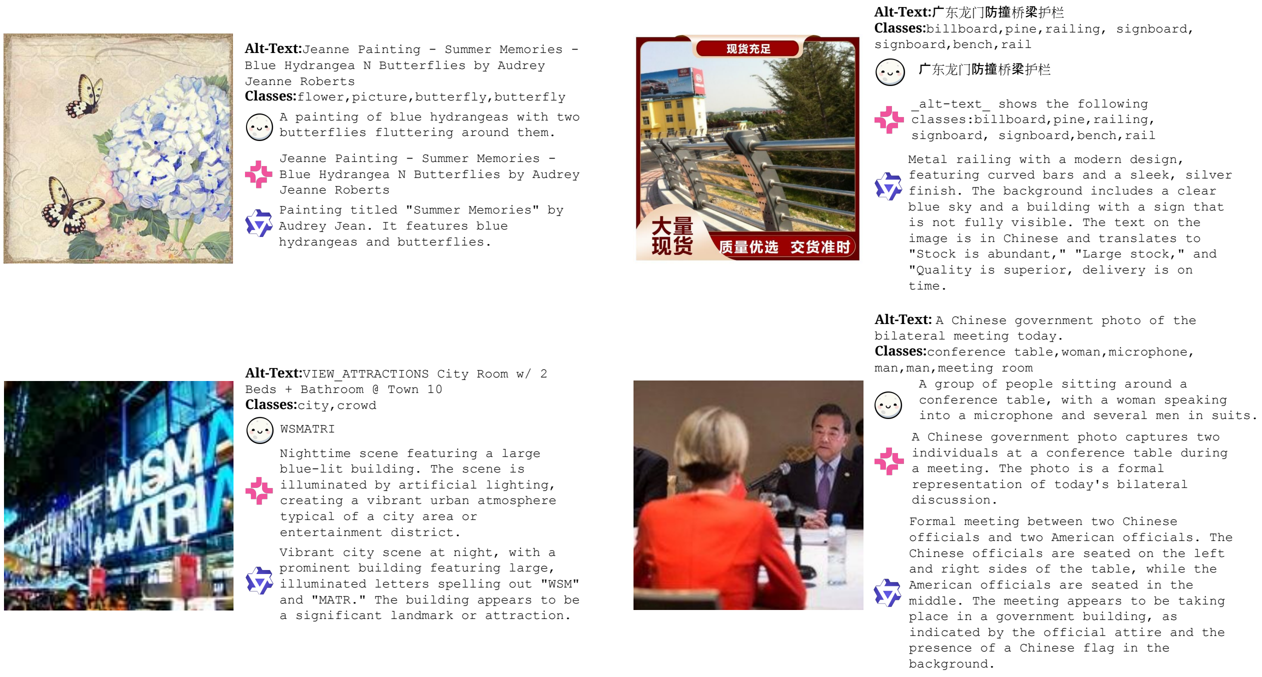
## B.1 Selecting the Recaptioning VLM

**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` Deitke et al. (2025), `moondream2` and `Qwen2-VL-7B` (Wang et al., 2024). 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. Qualitatively, we find that simply prompting performant open-weight VLMs with alt-text results in relevant information getting 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.



**Alt-Text:**Jeanne Painting - Summer Memories - Blue Hydrangea N Butterflies by Audrey Jeanne Roberts
**Classes:**flower,picture,butterfly,butterfly

A painting of blue hydrangeas with two butterflies fluttering around them.

Jeanne Painting - Summer Memories - Blue Hydrangea N Butterflies by Audrey Jeanne Roberts

Painting titled "Summer Memories" by Audrey Jean. It features blue hydrangeas and butterflies.

**Alt-Text:**广东龙门防撞桥梁护栏
**Classes:**billboard,pine,railing, signboard, signboard,bench,rail

广东龙门防撞桥梁护栏

_alt-text_ shows the following classes:billboard,pine,railing, signboard, signboard,bench,rail

Metal railing with a modern design, featuring curved bars and a sleek, silver finish. The background includes a clear blue sky and a building with a sign that is not fully visible. The text on the image is in Chinese and translates to "Stock is abundant," "Large stock," and "Quality is superior, delivery is on time.

**Alt-Text:**VIEW_ATTRACTIONS City Room w/ 2 Beds + Bathroom @ Town 10
**Classes:**city,crowd

WSMATRI

Nighttime scene featuring a large blue-lit building. The scene is illuminated by artificial lighting, creating a vibrant urban atmosphere typical of a city area or entertainment district.

Vibrant city scene at night, with a prominent building featuring large, illuminated letters spelling out "WSM" and "MATR." The building appears to be a significant landmark or attraction.

**Alt-Text:** A Chinese government photo of the bilateral meeting today.
**Classes:**conference table,woman,microphone, man,man,meeting room

A group of people sitting around a conference table, with a woman speaking into a microphone and several men in suits.

A Chinese government photo captures two individuals at a conference table during a meeting. The photo is a formal representation of today's bilateral discussion.

Formal meeting between two Chinese officials and two American officials. The Chinese officials are seated on the left and right sides of the table, while the American officials are seated in the middle. The meeting appears to be taking place in a government building, as indicated by the official attire and the presence of a Chinese flag in the background.

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 (Wang et al., 2024), we adopt a similar analysis as Nguyen et al. (2023) 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.

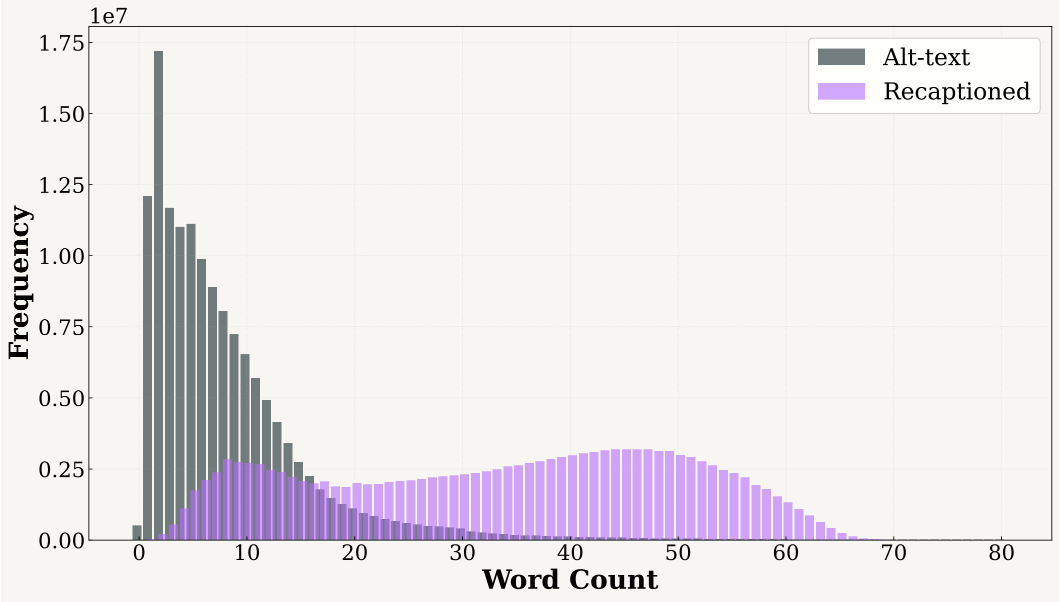


Figure 10: **Word Count Distribution.** Comparison of DataComp alt-text captions and Qwen2-VL-7B recaptions. Alt-text remains short-form, while recaptions are substantially longer. Extremely long alt-text outliers are excluded from the plot for clarity.

**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 $\tau$. We show our results in Tab. 10, 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 10: **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.

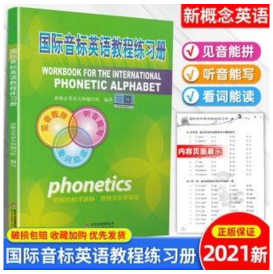
**Alt-text**: 台北家鴻燒鵝興安店：千人推薦4顆星港式燒臘店，吃肥美燒鵝飯秒殺便當、鮮蝦雲吞
**Classes**: plate,cup,twice-cooked pork,soy sauce,rice, salad,noodles,vegetable
**Synthetic Caption**: Plate of food that includes a serving of rice, noodles, and a piece of twice-cooked pork. There are also some vegetables, such as cucumbers and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

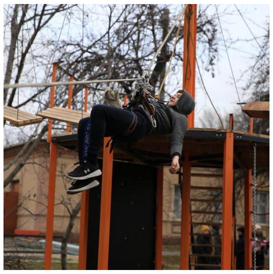
**Prompt**: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.



**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.

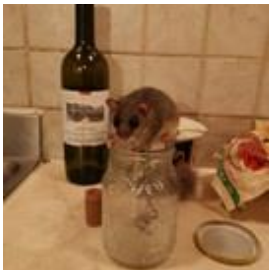


**Alt-text**: Веревочный парк «Активная страна» (Фото) - новости Днепра
**Classes**: woman, rope, playground, kid
**Synthetic Caption**: Kid hanging from a rope in a park called "Active Country" (Фото). The park is located in Dnepropetrovsk, as indicated by the text in the image.

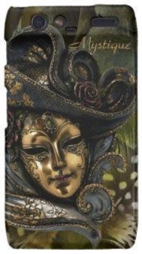
Figure 12: Some more examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

## Concept-Aware Image Caption

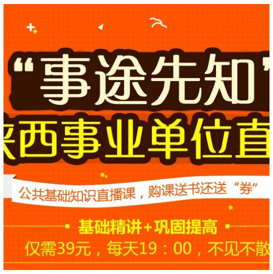
**Prompt**: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.



**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.`

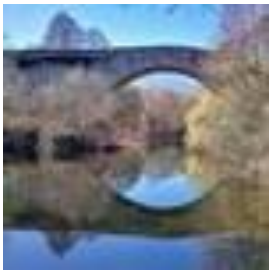


**Alt-text**: `M&#225;scara del carnaval - verde/caso de Droid RAZR Carcasa Para Motorola Droid RAZR`
**Classes**: `mask`,`phone`,`masquerader`
**Synthetic Caption**: `The image is of a phone case designed for a Motorola Droid RAZR. The case features a vibrant and intricate design that includes a green 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.`



“事途先知
陕西事业单位直
公共基础知识直播课，购课送书还送“券”
基础精讲+巩固提高
仅需39元，每天19：00，不见不散

**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.`

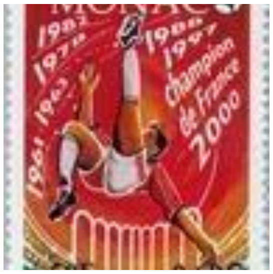


**Alt-text**: 的照片 `El Tiemblo y sus Puentes Medievales`
**Classes**: `water`
**Synthetic Caption**: `Picturesque scene of a stone bridge over a calm body of water, surrounded by trees and foliage. The bridge appears to be of medieval design, with an arched structure. The reflection of the bridge and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

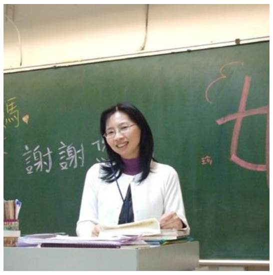
Figure 13: Some more examples of concept-aware synthetic captions. We observe good concept-adherence and multilingual understanding.

### Concept-Aware Image Caption

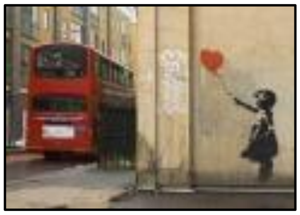
**Prompt**: Generate a brief and concise image caption using relevant details from alt-text and classes present in the image. Alt-Text: {alt-text}, classes: {classes}.



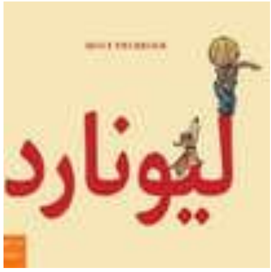
**Alt-text**: AS Monaco Championship
**Classes**: postage stamp, handstamp, football
**Synthetic Caption**: Postage stamp from Monaco featuring a football player in mid-air, celebrating a goal. The stamp commemorates the Monaco Football Club's championship victories in 1963, 1972, 1986, and 1998.



**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` (Ilharco et al., 2021) 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 (Gadre et al., 2023) to ensure that IID results are easily reproducible and that all the performance boosts occur due to CABS. Tab. 11 shows the general hyperparameters used for training as well as CABS-specific hyperparameters.

Table 11: 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 (Gadre et al., 2023) is critical to understand their zero-shot generalization properties. However, recent probes into the reliability of these benchmarks such as (Abbas et al., 2024b; Udandarao et al., 2025) 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 12: Datasets used in Zero-Shot Classification and Image-Text Retrieval Tasks

| **Task Type** | **Dataset** | **Test Set Size** | **Number of Classes** |
|---|---|---|---|
| Classification | Caltech-101 (Fei-Fei et al., 2004) | 6,085 | 102 |
| | Camelyon17 | 85,054 | 2 |
| | CIFAR-10 (Krizhevsky et al., 2009) | 10,000 | 10 |
| | CIFAR-100 (Krizhevsky et al., 2009) | 10,000 | 100 |
| | Country211 (Radford et al., 2021; Thomee et al., 2016) | 21,100 | 211 |
| | Dollar Street (Gaviria Rojas et al., 2022) | 3,503 | 58 |
| | DTD (Cimpoi et al., 2014) | 1,880 | 47 |
| | FGVC Aircraft (Maji et al., 2013) | 3,333 | 100 |
| | Food-101 (Bossard et al., 2014) | 25,250 | 101 |
| | FMoW (Christie et al., 2018; Koh et al., 2021) | 22,108 | 62 |
| | GeoDE (Ramaswamy et al., 2023) | 12,488 | 40 |
| | ImageNet (Deng et al., 2009) | 50,000 | 1,000 |
| | ImageNet-A (Hendrycks et al., 2021b) | 7,500 | 200 |
| | ImageNet-O (Hendrycks et al., 2021b) | 2,000 | 200 |
| | ImageNet-R (Hendrycks et al., 2021a) | 30,000 | 200 |
| | ImageNet-Sketch (Wang et al., 2019) | 50,889 | 1,000 |
| | ImageNet-V2 (Recht et al., 2019) | 10,000 | 1,000 |
| | Let-it-Wag! (Udandarao et al., 2024) | 130,000 | 290 |
| | ObjectNet (Barbu et al., 2019) | 18,574 | 113 |
| | Oxford Flowers-102 (Nilsback and Zisserman, 2008) | 6,149 | 102 |
| | Oxford-IIIT Pets (Parkhi et al., 2012; Zhai et al., 2019) | 3,669 | 37 |
| | Pascal VOC 2007 (Everingham, 2009) | 14,976 | 20 |
| | RESISCS45 (Cheng et al., 2017; Zhai et al., 2019) | 6,300 | 45 |
| | Stanford Cars Krause et al. (2013) | 8,041 | 196 |
| | STL-10 (Coates et al., 2011) | 8,000 | 10 |
| | SUN-397 (Xiao et al., 2016) | 108,754 | 397 |
| Retrieval | Flickr30k (Young et al., 2014) | 31,014 | N/A |
| | MSCOCO (Chen et al., 2015) | 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. 13 for CABS-DM performance and Tab. 14 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 13: **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 14: **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 (Roth et al., 2024), 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. 15 and 16. 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 (Feng et al., 2024; Blakeney et al., 2024). We can further draw a faint connection to data curriculums (Bengio et al., 2009; Zhang et al., 2025)—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 15: **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** | **Zero-shot Classification** | | | | **Let-it-Wag!** | **Avg (Clf)** |
|---|---|---|---|---|---|---|---|
| | | IN-Val | IN-shift | Obj | Scene | | |
| | | | | **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 16: **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.

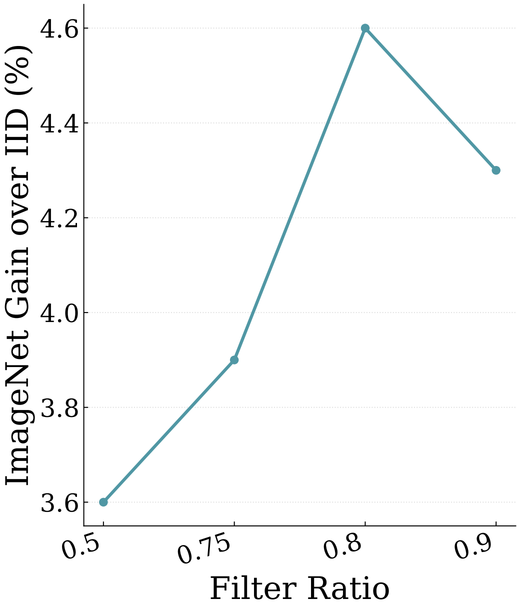


Figure 15: **CABS-DM filtering ratio ablation**. We choose $f = 0.8$ based on ImageNet validation performance. For simplicity, we maintain this filter ratio for CABS-FM as well, and still see strong performance gains on image-text retrieval benchmarks.

# G 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. (2025) for language models and autoregressive vision-language models but the principle may be applied to CLIP as well.

## G.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. 13 is shown below.

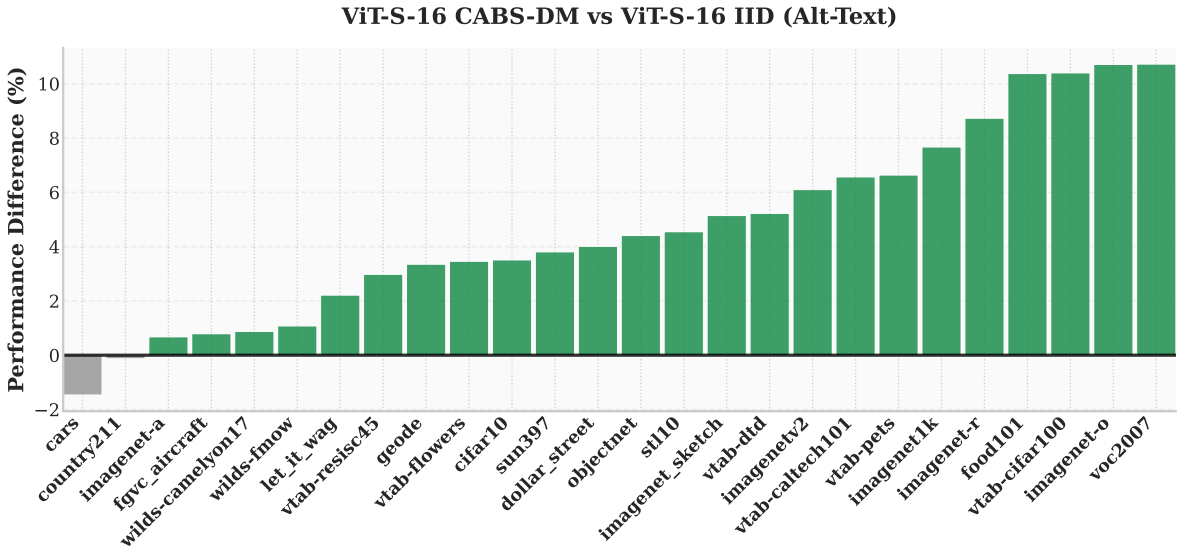


Figure 16: Dataset-wise comparisons for all benchmarks for CLIP ViT-S/16 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

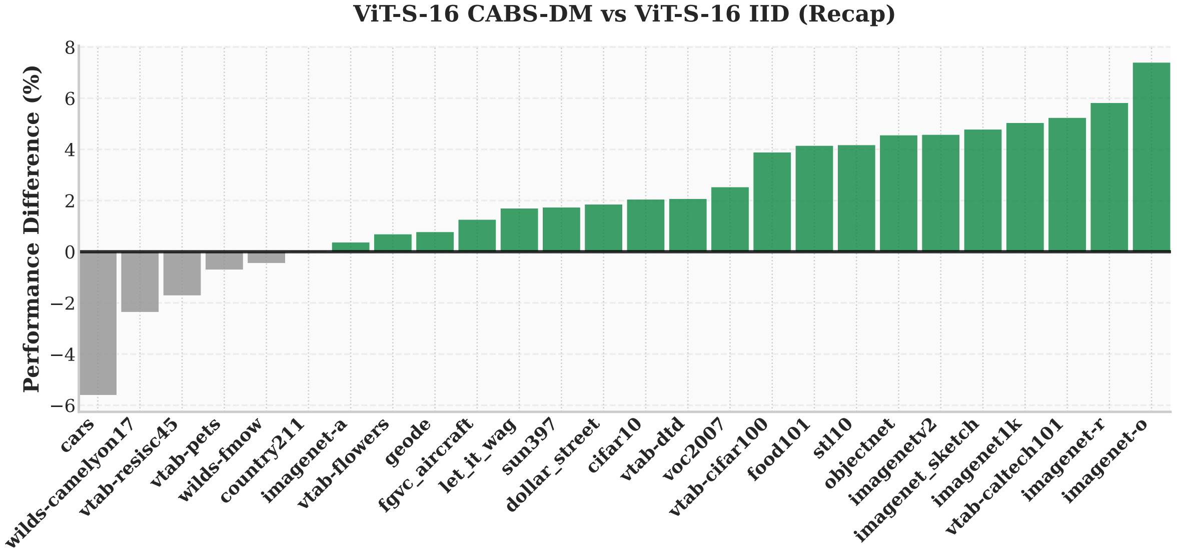


Figure 17: Dataset-wise comparisons for all benchmarks for CLIP ViT-S/16 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

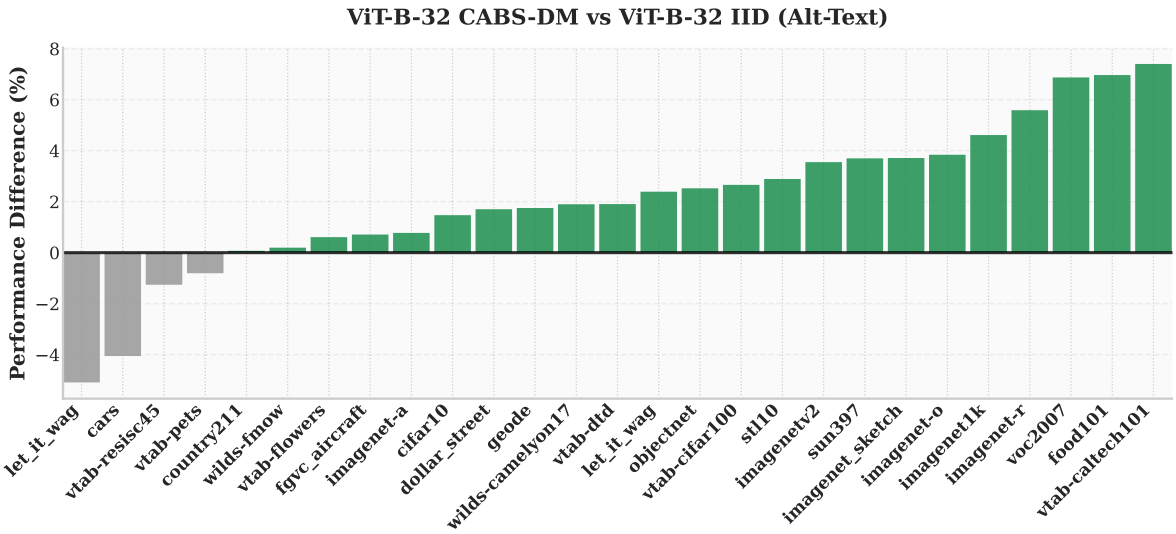


Figure 18: Dataset-wise comparisons for all benchmarks for CLIP ViT-B/32 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

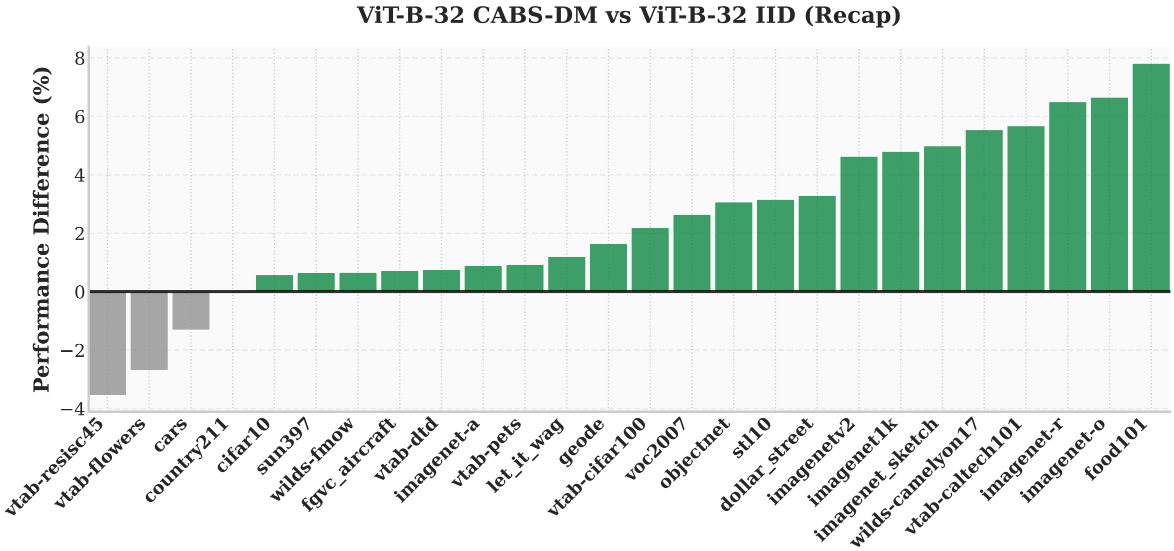


Figure 19: Dataset-wise comparisons for all benchmarks for CLIP ViT-B/32 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

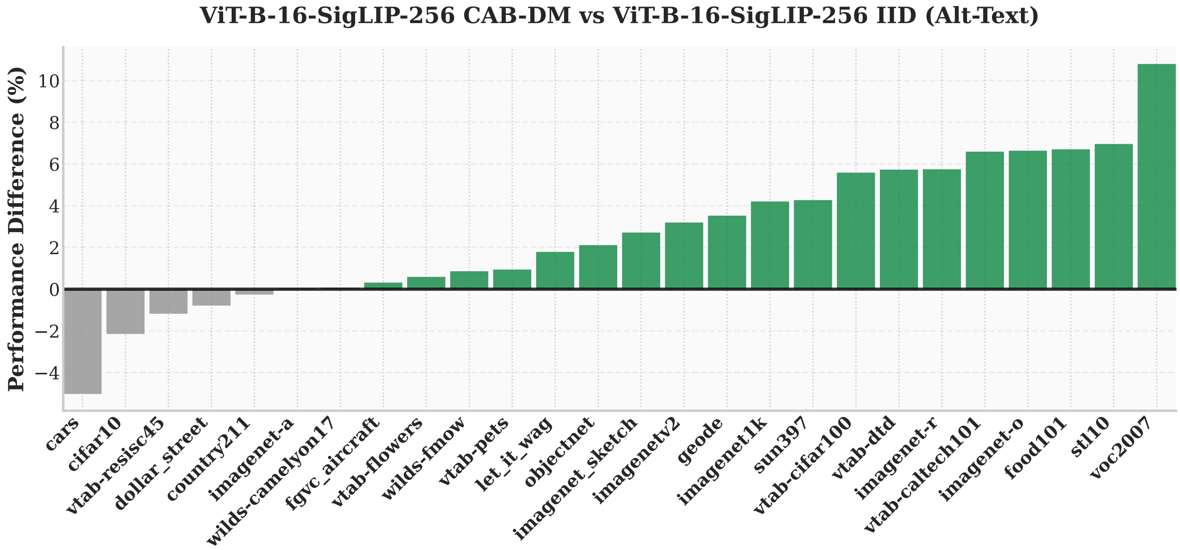


Figure 20: Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

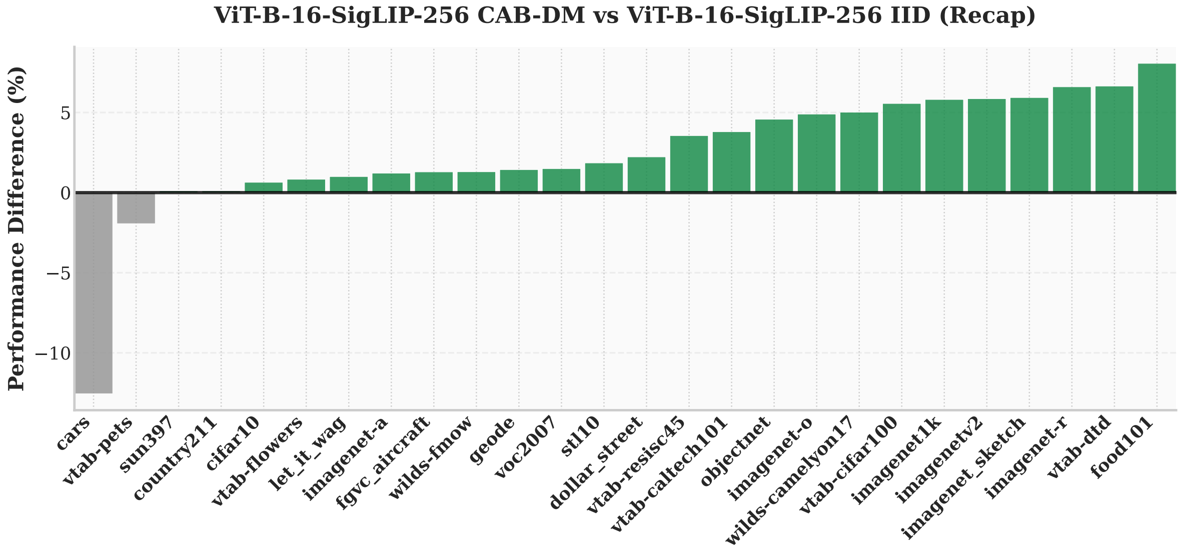


Figure 21: Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

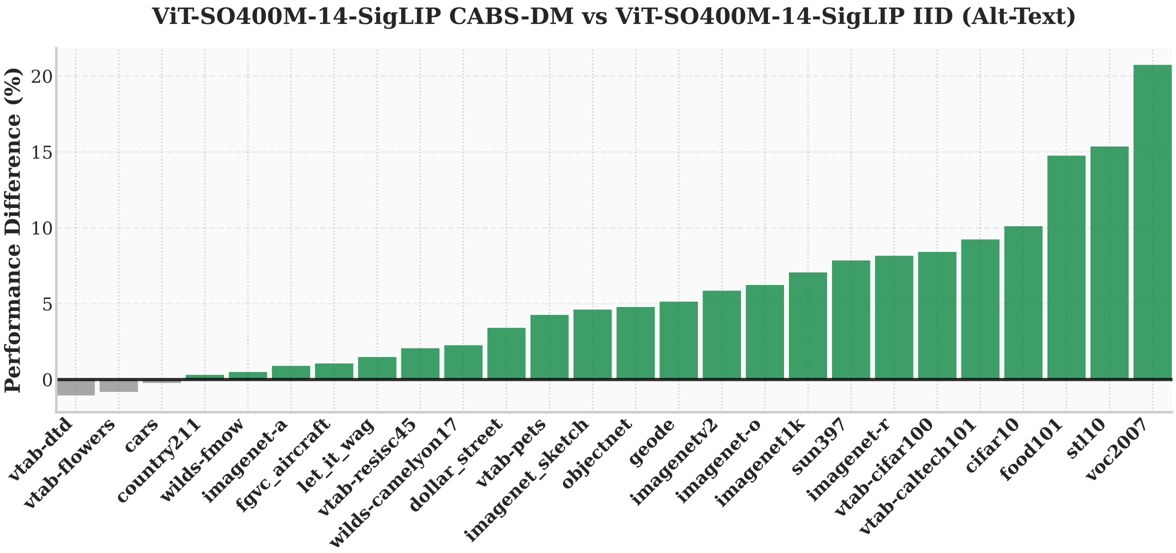


Figure 22: Dataset-wise comparisons for all benchmarks for SigLIP ViT-SO400M-14 between CABS-DM and IID sampling for alt-text. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

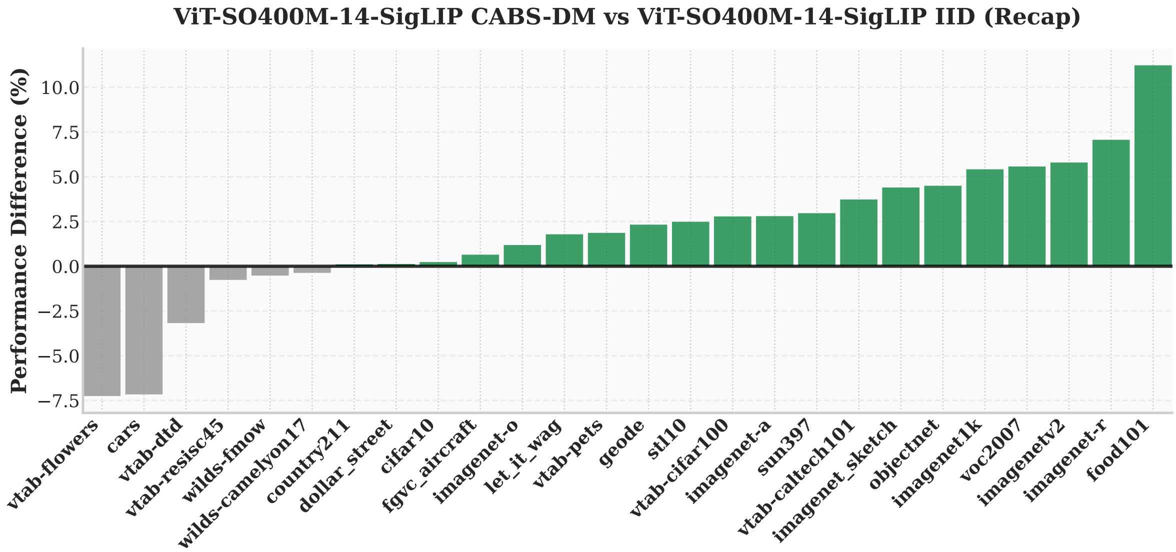


Figure 23: Dataset-wise comparisons for all benchmarks for SigLIP ViT-SO400M-14 between CABS-DM and IID sampling for synthetic recaptions. A positive performance difference indicates a benchmark where CABS-DM outperforms IID sampling.

## G.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 (Xu et al., 2024) 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 17: **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** |

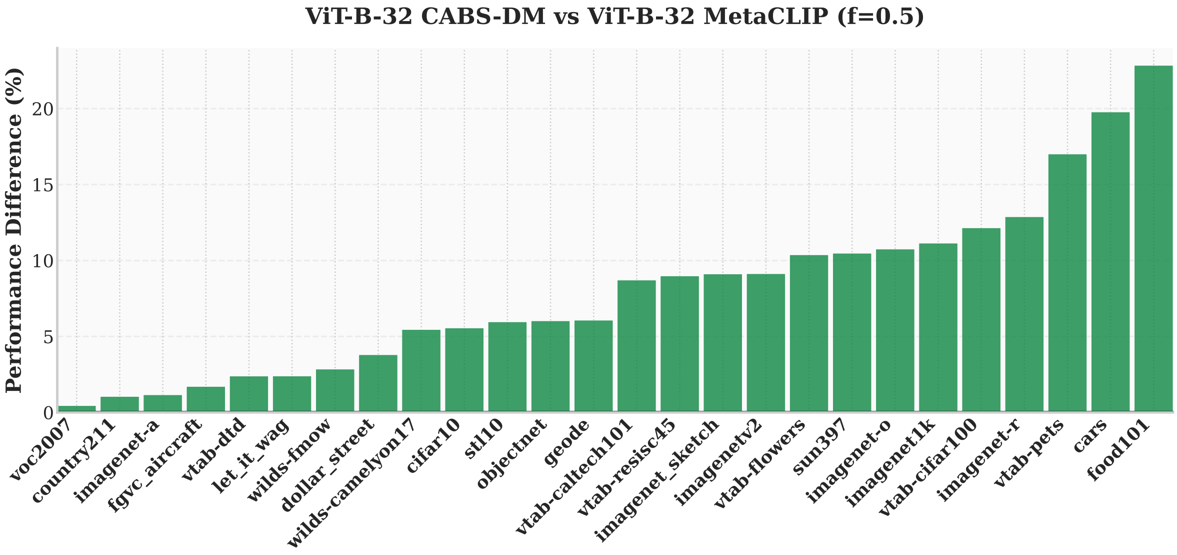


Figure 24: Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.5$) and MetaCLIP curation on alt-text.

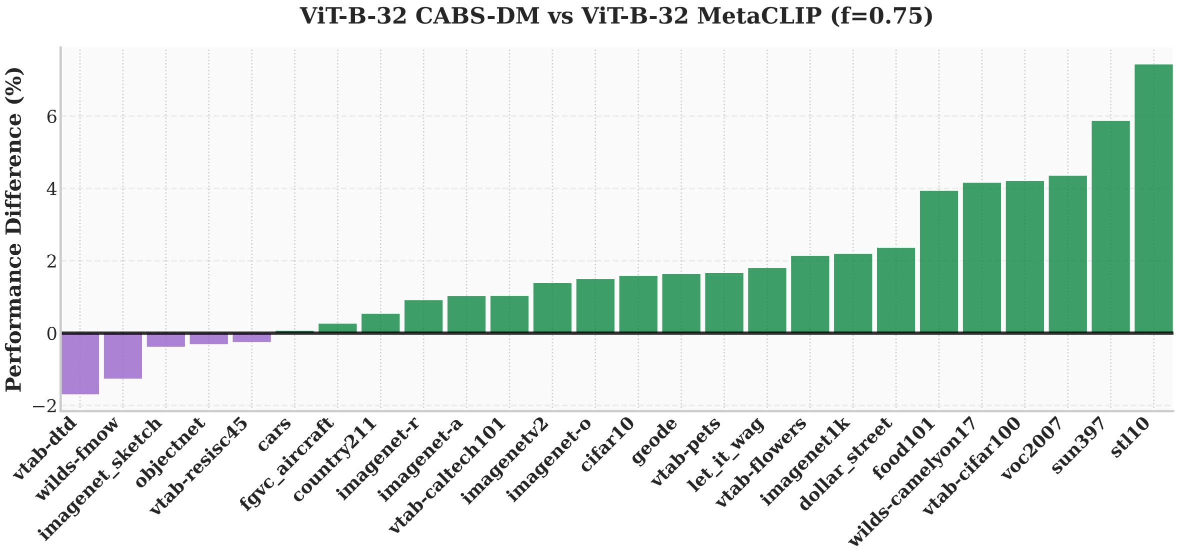


Figure 25: Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.75$) and MetaCLIP curation on alt-text.

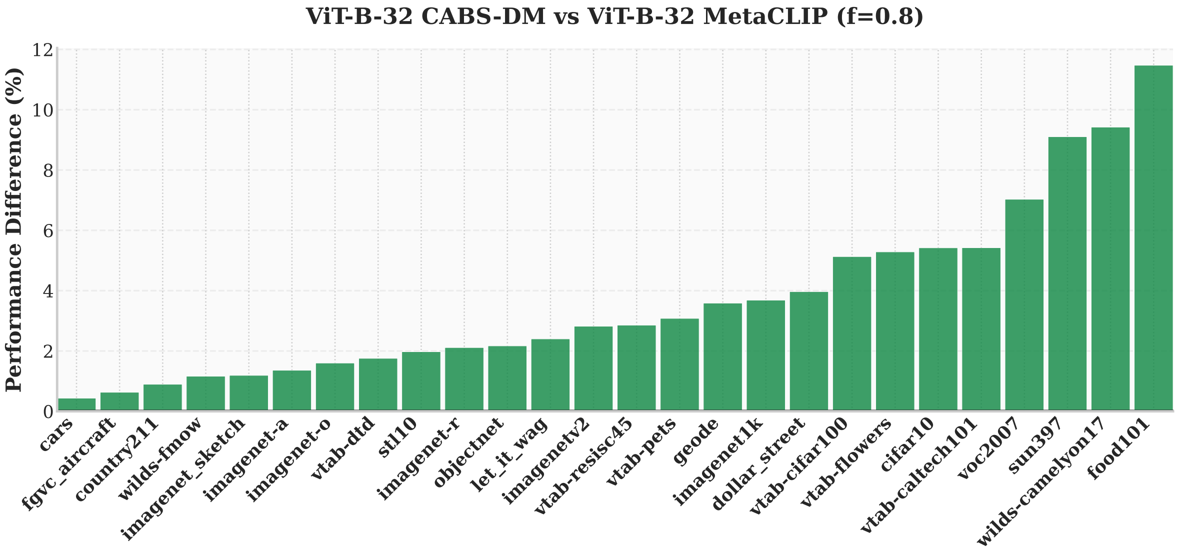


Figure 26: Dataset-wise comparisons for all benchmarks for CLIP ViT-B-32 between CABS-DM ($f = 0.8$) and MetaCLIP curation on alt-text.

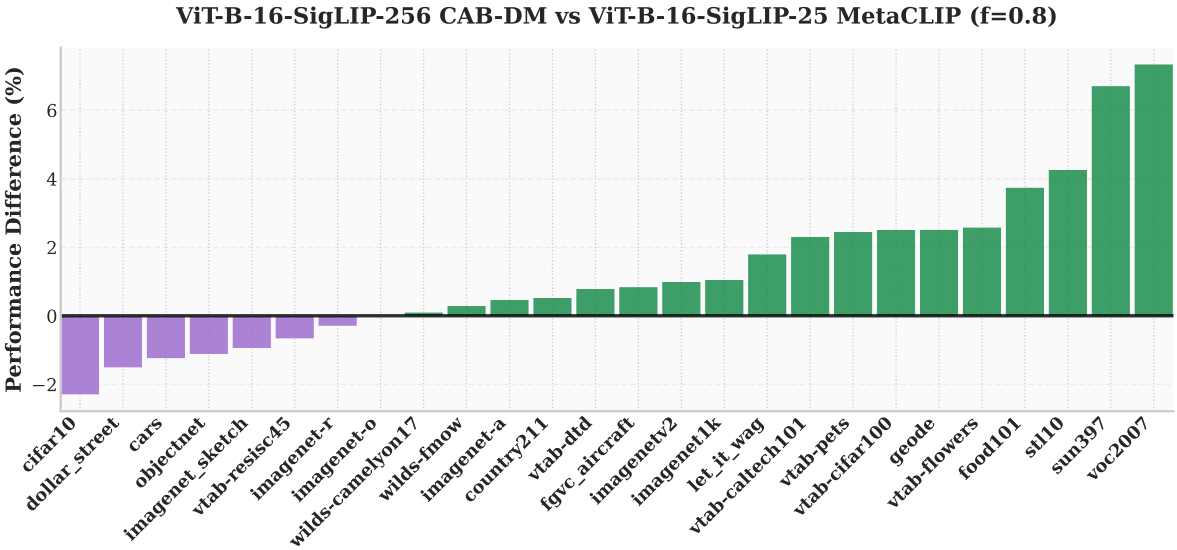


Figure 27: Dataset-wise comparisons for all benchmarks for SigLIP ViT-B-16 between CABS-DM ($f = 0.8$) and MetaCLIP curation on alt-text.