Pattern Recognition 171 (2026) 112236

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Precision at scale: Domain-specific datasets on-demand

Jesús M. Rodríguez-de-Vera [a,b,1,*], Imanol G. Estepa [a,b,1], Ignacio Sarasúa [c], Bhalaji Nagarajan [d], Petia Radeva [a,e]

[a] Department de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain
[b] Computer Vision Center, Cerdanyonla del Vallès, Barcelona, Spain
[c] NVIDIA Computing Spain, Madrid, Spain
[d] Barcelona Supercomputing Center (BSC), Barcelona, Spain
[e] Institut de Neurosciències, Universitat de Barcelona, Barcelona, Spain

ARTICLE INFO

Keywords:
Domain-specific datasets
Autonomous dataset creation
Data curation
Synthetic data
Pretraining
Dataset evaluation

ABSTRACT

Recent self-supervised learning methods rely on massive general-domain datasets for robust model pretraining. However, these datasets may lack specificity required in specialized domains. Collecting large, supervised datasets to compensate for this limitation is also cumbersome. This raises a key question: *Can automatically crafted domain-specific datasets serve as efficient and effective SSL pretrainers, performing comparable to—or even surpassing—much larger state-of-the-art general-domain datasets?* To address this challenge, we propose **Precision at Scale (PaS)**, a novel modular pipeline for automatic creation of domain-specific datasets on-demand. PaS leverages Large Language Models (LLMs) and Vision-Language Models (VLMs) through three distinct phases: Concept Generation, where LLMs identify relevant domain concepts; Image Collection, utilizing VLMs and Generative models to gather appropriate images; Data Curation, ensuring quality and relevance by eliminating unrelated or redundant images. We conduct extensive experiments across three complex domains — *food, insects, and birds* — proving that PaS datasets compete and often surpass existing domain-specific datasets in diversity, scale, and effectiveness as pretrainers. Models pretrained on PaS datasets outperform those trained on large-scale general-domain datasets (ImageNet-1K) by up to 21 % and surpass same-scale domain-specific datasets by 6.7 % across classification tasks. Notably, despite being an order of magnitude smaller, PaS datasets outperform ImageNet-21K pretraining, with improvements of 3.3 % in fine-tuning and 9.5 % in few-shot learning, and showing superior performance on specialized dense tasks. Furthermore, by efficiently fine-tuning pretrained VLMs like CLIP and SigLIP using low-rank methods, we achieve performance gains (+ 4.2 % over CLIP) in specialized domains with minimal overhead, demonstrating the versatility of PaS datasets.

1. Introduction

Self-Supervised Learning (SSL) models such as DINOv2 [1], trained on millions of images, obtain very high performance on most general discriminative tasks such as image classification and object detection [2,3]. However, most SSL methods focus on being as general as possible and require huge volumes of broad-domain images to ensure high performance in domain-specific tasks. These datasets, refered as pretrainer datasets (in short, “pretrainers”), are usually unsupervised or programmatically created.

Concurrently, various **domain-specific datasets** [4–7] are designed to train expert models in specific domains, trying to bridge the gap between existing computer vision solutions and real-world application in those domains. Nevertheless, these supervised datasets require expensive investments in human annotations. This leads to a small number of images compared to popular unsupervised datasets [1,8] (For example, Food-2K [4], the largest food benchmark has ≈ 600 K training images, while LAION [8] has 5.85B images), making them suboptimal for recent large models [9,10]. On the other hand, generic datasets like ImageNet lack comprehensive coverage for many specialized real-world domains. *Domain-specific datasets, in general, prove to be better in their expertise, however, lack the scalability* as the annotation process is not feasible at scale. With recent generative models, synthetic datasets [11,12] are gaining popularity. They show performance comparable to real ones and, importantly, provide scalability on demand. However, they fail to cover “under” represented (infrequent) classes. SynCLR [12], for

* Corresponding author.
E-mail addresses: j.molina.rdv@ub.edu (J.M. Rodríguez-de-Vera), estepa.gonzalez@ub.edu (I.G. Estepa), isarasua@nvidia.com (I. Sarasúa), bhalaji.nagarajan@bsc.es (B. Nagarajan), petia.ivanova@ub.edu (P. Radeva).
[1] Equal contribution.

https://doi.org/10.1016/j.patcog.2025.112236
Received 6 April 2025; Received in revised form 18 July 2025; Accepted 1 August 2025
Available online 5 August 2025

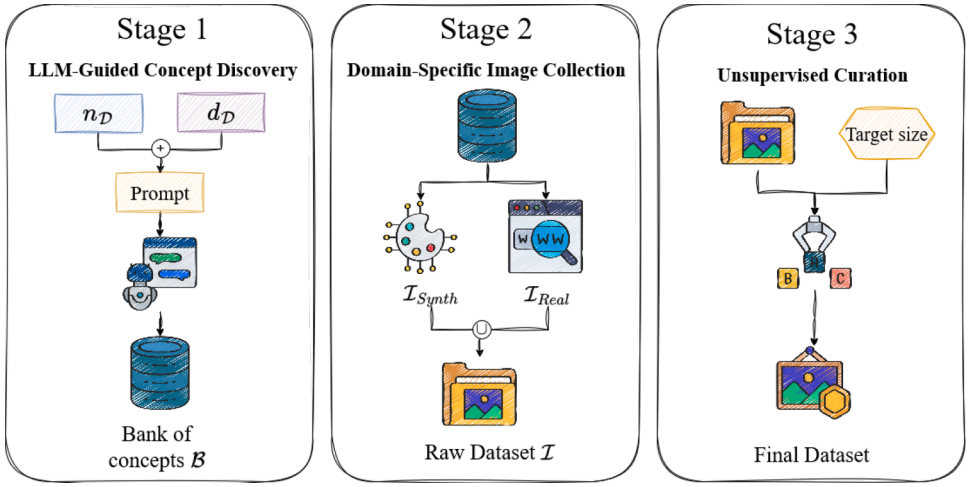


Fig. 1. PaS pipeline overview. With "only" two textual inputs describing a target domain (n_D and d_D), PaS autonomously creates SoTA domain-specific datasets in 3 stages: LLM-guided diverse concept bank creation (**Stage 1**); Web and synthetic images collection (**Stage 2**); Automatic removal of OOD images, reducing dataset size while preserving relevant information (**Stage 3**).

example, leverages an initial set of general captions from supervised datasets to create a completely synthetic dataset, making the dataset generation process tightly coupled with "external supervision". Ideally, a smart, well-balanced mix of unrestricted real and synthetic images could alleviate the problem.

Two natural research questions arise: *(1) Is it possible to design a framework that automatically crafts domain-specific datasets, outperforming existing SoTA datasets? (2) Can these datasets serve as more efficient and effective pretrainers than much larger general-domain datasets, when the target domain is known?* To address these questions, we introduce **Precision-at-Scale (PaS)** (illustrated in Fig. 1), a general and modular pipeline to autonomously generate high-quality, domain-specific datasets on demand ("PaS datasets"), without any dependence on external labels and human experts. PaS is elegant - it leverages the synergy between LLMs and VLMs to create scalable, well-curated domain-specific datasets through 3 key stages: (1) **LLM-guided Concept Discovery** to generate a comprehensive, rich bank of domain-specific concepts, (2) **Domain-Specific Image Collection** using diverse real and synthetic images, (3) **Autonomous Dataset Curation** using advanced unsupervised curation methods to ensure high quality and domain relevance. In addition, PaS offers an efficient task-agnostic analysis framework to assess the diversity of the generated PaS datasets (see Section 4).

PaS is a novel framework for **efficient domain-specific learning**. PaS datasets perform better at a much smaller size (one order smaller), leading to less computation and faster training. PaS is flexible, making no assumptions on specific components, allowing integration of any LLM, VLM, or Image generation models. This versatility ensures seamless incorporation of newer models as they are developed. PaS datasets match—and even surpass—the diversity of human-created domain-specific datasets [4,5,13,14], while providing more comprehensive coverage of real-world complex domains through a completely automated way. Distinct from knowledge distillation, our framework does not simply transfer knowledge, but instead refines and structures it into a new, hybrid data asset. We demonstrate that this generates emergent value, as our PaS datasets can be used to fine-tune and improve the original foundation models (see Section 5.3).

Extensive experiments on multiple domains —*food*, *birds* and *insects*— show SSL models pretrained on PaS datasets not only outperform existing domain-specific SoTA datasets at the same scale, but also show superior performance to models trained on much larger general-domain datasets like ImageNet-1K (IN-1K) and even ImageNet-21K (IN-21K) on specific domain evaluation. Beyond pretraining from scratch, PaS datasets offer a promising avenue for efficiently adapting existing VLMs to new domains. By leveraging automatically generated image-text pairs, we fine-tune a pretrained VLM on PaS datasets for a new domain, enabling cheap and on-demand adaptation. Remarkably, PaS datasets, through LoRA fine-tuning [15], efficiently enhance the capabilities of pretrained VLMs models like CLIP [16] and SigLIP [17] in their respective domains at a tiny fraction of the computational and data cost required to train them. The advantage of PaS datasets over traditional domain-specific datasets grows significantly when leveraging its scalability. In summary, **our contributions** are:

- We propose PaS, a novel **domain-specific dataset creation pipeline** (Section 3), that, given a domain, creates a hybrid dataset of web and synthetic images, at a given scale and computational budget.
- We define an **efficient low-resource task-agnostic diversity analysis framework** (Section 4) and emphasize that PaS datasets provide better diversity than current SoTA domain-specific supervised datasets.
- With **extensive validation** (Section 5) on three complex domains, we prove PaS datasets as much better pretrainers, showing consistent superior performance across domains, datasets and downstream tasks, including classification, and dense tasks such as semantic segmentation and keypoint detection. Compared to general-domain datasets (IN-1K), PaS demonstrate that on the same and even smaller scale, domain-specific datasets outperform general ones, proving that quality stands over quantity on model pretraining.
- We demonstrate that PaS datasets effectively improve the performance of large pretrained VLMs in specific domains leading to substantial improvements (an average gain of 4.2 % for CLIP) with minimal computational overhead.

The rest of the paper is as follows: Section 2 discusses the recent related works. Our proposed framework, Precision at Scale (PaS), is detailed in Section 3. We then present a comprehensive, diversity and domain alignment low-resource evaluation of PaS datasets in Section 4. The effectiveness of PaS datasets as pretrainers and for fine-tuning is demonstrated in Section 5, followed by the conclusions in Section 6.

2. Related works

The increasing **demand for large-scale data** in training models such as ConvNeXt [18] and ViT [19], makes dataset creation at scale very critical. DINOv2 [1] and Internet Explorer [20] use sophisticated automatic data curation pipelines to curate high-quality real images.

Generative models like Stable Diffusion [21] and MUSE [22] have propelled the creation of synthetic datasets [23]. SynCLR [12] uses labels of existing datasets, while SynthCLIP [11] creates large-scale synthetic image-text pairs using a predefined concept bank of MetaCLIP [24]. Despite having a vast collection of 500k concepts, the MetaCLIP concept repository fails to provide extensive coverage in certain specific domains (For example, MetaCLIP "only" contains only simple *paella*, a popular Mediterranean food as a single concept without any of its common variations). This shows that even large-scale concept banks may lack necessary domain specificity. These methods can generate large datasets, however they rely on curated information, like predefined class labels or captions, limiting their adaptability to new domains. Their focus on scale can sometimes compromise domain specificity.

Several works addressed **the need for domain-specific data.** Despite its noise, domain-specific web data proved more effective in fine-grained recognition [25]. InfoGrowth [26] and T-MARS [27] enabled dataset growth with maintained cleanliness and diversity. Large unsupervised datasets like LAION-5B warranted extraction of custom subsets tailored for specific use cases [20,28]. Alternatively, in linguistics, DoPAMine [29] mines domain-specific LLM training data. CRAFT [30] generates synthetic datasets by retrieving and augmenting corpora. Although focused on text, these methods underscore the importance of domain-specific datasets. Recent studies have extended this idea in vision. Performance of SSL models prove to increase "only" when data aligns closely with the target domain [31].

Self-Supervised Learning enables learning adaptable features from unlabeled data [32–34], reducing the reliance on extensive labeled datasets. Recent popular models like MoCoV3 [35], MAE [36], and CAE [37] have shown increased robustness and efficiency owing to better computational power, model complexity, and data scale [38]. VLMs like CLIP [16], and BASIC [39] focus learning joint text-image representations. Dual-encoder models [16,39] learn context-aware features in a shared latent space [40,41], enabling zero-shot image manipulations guided by textual prompts [28,42]. VLMs allow automatic data curation, while SSL potentiates learning from non-annotated data, forming the crux of our proposal.

Our proposed method stands out by autonomously generating high-quality, domain-specific datasets without external supervision or predefined data structures. Unlike other synthetic methods [11,12], PaS leverages zero-shot capabilities of LLMs and VLMs to discover domain-specific concepts and collect relevant images. This allows us to adapt to any domain without human intervention during the generation process. PaS focuses on generating domain-specific datasets that are precisely aligned with a target domain, favoring quality over quantity. Our datasets are smaller in scale and tailored to capture the nuances of the domain, leading to more efficient training of domain-specific models. This is highlighted by our experiments, where models pretrained on PaS datasets outperform those trained on larger, general-domain datasets. In summary, PaS contributes a novel, scalable, fully autonomous pipeline for domain-specific dataset creation, addressing the limitations of existing dataset generation methods and highlights the importance of domain-specific data in model pretraining and adaption.

3. Precision at scale

Precision at Scale (PaS) is a novel fully automated framework for generating on-demand, domain-specific datasets with minimal human intervention. Our modular pipeline (as shown in Fig. 1) comprises three stages: **Stage 1** (Section 3.1) utilizes LLMs to discover domain-specific concepts, establishing a foundational concept bank. **Stage 2** (Section 3.2) collects domain-specific images from web-scale databases and generates synthetic images using text-to-image models, enriching the dataset with a broader range of concepts. **Stage 3** (Section 3.3) refines the dataset by eliminating image redundancy and filtering irrelevant out-of-domain images through systematic curation techniques. PaS produces highly precise, scalable, datasets ("PaS datasets") suitable for training self-supervised visual models or image-text supervision. Its modular design allows integration with any LLMs, image generators, or image sources, ensuring flexibility across various domains.

3.1. In-domain LLM-guided concept discovery

Stage 1 involves building a comprehensive bank of textual concepts, $\mathcal{B}$, specific to a target domain, D. The process **begins with "only" *two inputs***: the domain name, n_D (e.g. "birds"), and a brief description, d_D, of the domain concepts (e.g. "bird species"). This is the only user input required. This stage (depicted in Fig. 2) consists of three main steps, each guided by LLMs: 1) generation, 2) expansion, 3) filtering.

Step 1: generation of initial concepts. We use an LLM, L_1, to generate an initial set of concepts, $\mathcal{B}_0$, by prompting it with n_D and d_D using the generation prompt template (as shown in Fig. 2). To maximize domain coverage, we sample multiple outputs from L_1 (with different random seeds) and aggregate them, until the addition of new concepts falls below a threshold λ_1 (empirically set hyperparameter). For instance, in the domain of birds, the initial concept set $\mathcal{B}_0$ might include species like "Canada Goose", "Crow", "Roseate Spoonbill", "Broad-billed Sandpiper", and "Imperial Eagle" (among others).

Step 2: expansion of concept bank. This step involves expanding and enriching the concept bank within the domain, $\mathcal{D}$. Each concept in $\mathcal{B}_0$ is expanded using another LLM, L_2 (can be same as L_1), which generates related concepts by using the expansion prompt template (see Fig. 2).

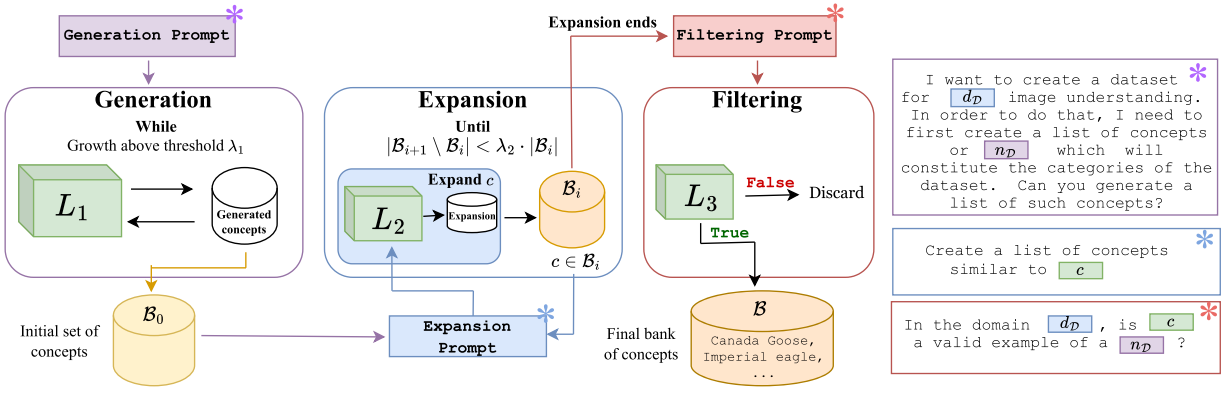


Fig. 2. Stage 1 Workflow: During the **generation** step, a prompt template is fed to an LLM, L_1, and outputs an initial set of concepts $\mathcal{B}_0$. In the **expansion** step, a template is used for each concept and prompted to the LLM, L_2, to find similar concepts. Upon maximizing the variety, we filter the generated concepts by prompting an auxiliary LLM, L_3 (**filtering** step). resulting in the final bank of concepts $\mathcal{B}$.

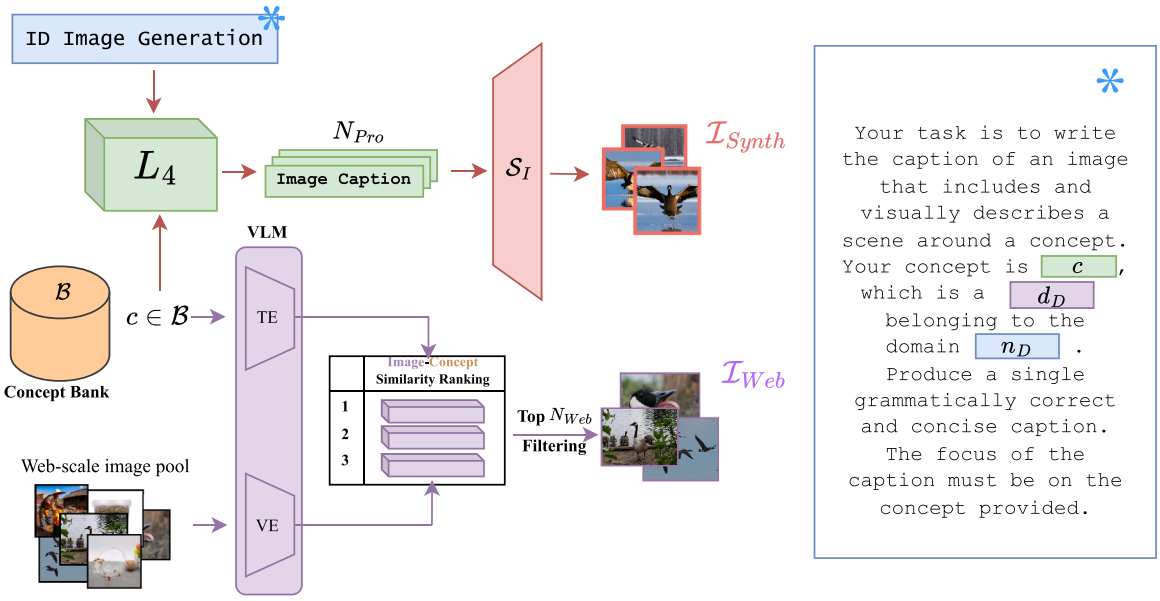


Fig. 3. Stage 2 Workflow: For each valid concept from Stage 1, we retrieve N_{Web} most similar images from a web-scale source ($\mathcal{I}_{\text{Web}}$). We prompt a text-to-image model S_I using LLM-created image captions to produce $N_{Pro} \times N_{Synth}$ synthetic images ($\mathcal{I}_{\text{Synth}}$). Together they form the unfiltered image dataset, $\mathcal{I}$.

This is done iteratively: for each concept c in the current set, $\mathcal{B}_i$, L_2 generates similar concepts which are aggregated to form the next version of the concept bank, $\mathcal{B}_{i+1}$. The process continues until the growth rate of new concepts falls below a threshold λ_2 (empirical). For example, when asked to expand the concept "Imperial Eagle", a potential answer of L_2 might include elements like "Bald Eagle", "Harpy Eagle", or "Golden Eagle". The final set of concepts at the end of this expansion is denoted as $\mathcal{B}_*$.

Step 3: concept filtering. To exclude irrelevant concepts and minimize computational costs, we use a separate LLM, L_3 ($L_3 \neq L_1, L_2$), to validate each concept in $\mathcal{B}_*$, mitigating potential hallucinations [43]. Only concepts confirmed to belong to D are kept in the final concept bank $\mathcal{B}$.

Stage outcome. The output of Stage 1 is the concept bank, $\mathcal{B}$, represented as a set of textual concepts relevant in the domain, D.

3.2. Collecting domain-specific images

Stage 2 (shown in Fig. 3) involves collecting high-quality, domain-specific images using the concept bank $\mathcal{B}$ built in Stage 1. It has two main components: web-data retrieval and synthetic image generation.

Web-data retrieval. Given a web-scale set of images, $\mathcal{I}_{pool}$, we aim to retrieve the images more aligned with $\mathcal{B}$. For each concept, we use a VLM to find the images from $\mathcal{I}_{pool}$ that match the concept. Given $c \in \mathcal{B}$, we generate a textual embedding, $\mathbf{t}_c = TE(c)$, using the text encoder $TE(\cdot)$ of a VLM. We use the visual encoder VE of the VLM to compute the image embedding of each image $I \in \mathcal{I}_{pool}$. Using cosine similarity, for each concept c, we retrieve the N_{Web} most similar images to t_c, provided the similarities are larger than a threshold λ_{Web} [8]. This results in $\mathcal{I}_{Web}$, a set of web images closely aligned with $\mathcal{B}$. For our experiments, I_{pool} was sourced from the Re-LAION-5B index. This dataset is a safety-revised version of LAION-5B [8] *(updated as of July 2024)*, to remove links to any known-illegal content.[1] Since these images are crawl from the web, they can vary from natural images to sketches. In addition to using this revised dataset, our own retrieval process programmatically respects `X-Robots-Tag` HTTP header directives, explicitly excluding any images marked with "noai", "noimageai", "noindex", or "noimageindex" to adhere to content owner preferences.

In-domain synthetic image generation. Instead of using plain concept names for image generation [23], we use an arbitrary LLM, L_4, to create detailed prompts that describe hypothetical images for each concept. This is achieved using a structured in-domain image generation template (see Fig. 3), which is designed to improve the descriptive quality and diversity of the synthetic images. For example, for the concept "Canada Goose", L_4 might generate "A majestic Canada Goose spreads its wings taking flight above the frozen lake". For each concept $c \in \mathcal{B}$, L_4 generates N_{Pro} prompts. These prompts are then used as prompts for a text-to-image model, S_I (e.g., Stable Diffusion [21]), which produces N_{Synth} images per prompt. Thus, the total number of synthetic images generated is $|\mathcal{B}| \times N_{Pro} \times N_{Synth}$, forming $\mathcal{I}_{Synth}$. By adjusting N_{Pro} and N_{Synth}, we can control the size and diversity of $\mathcal{I}_{Synth}$.

Stage outcome. The final unfiltered image dataset $\mathcal{I}$ is formed by combining web and synthetic images: $\mathcal{I} = \mathcal{I}_{\text{Web}} \cup \mathcal{I}_{\text{Synth}}$.

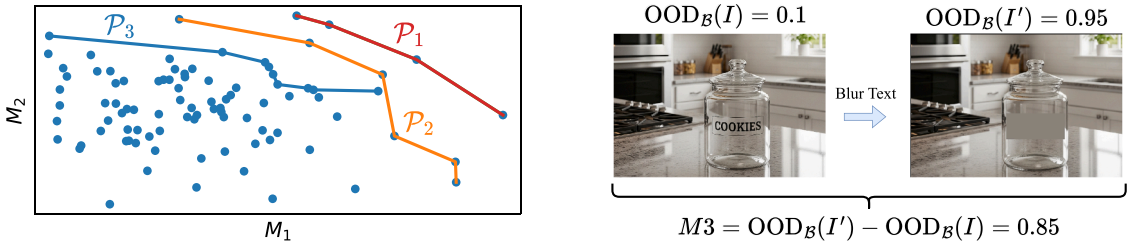
3.3. Autonomous dataset curation

Stage 3 focuses on refining the unfiltered dataset from Stage 2, $\mathcal{I}$, to enhance its relevance for D. By systematically eliminating redundant and out-of-distribution (OOD) images, we reduce training costs and minimize potential model performance drawbacks.

Self-supervised similarity-based deduplication. Duplicate and closely similar images increase image count (and resource consumption) without adding diversity. We use Self-Supervised Copy Detection (SSCD) [44] to identify and eliminate duplicates. This involves computing embeddings for each image using SSCD's visual encoder. We then build an approximate k-NN graph ($k = 64$) using the IVF-PQ algorithm [45] for efficient searching.[2] On this k-NN graph, we then apply a similarity threshold (of 0.6), keeping only the connections between images that exceed a certain cosine similarity. The connected components are then found on

[1] https://laion.ai/blog/relaion-5b/

[2] https://rapids.ai/



(a) Illustrative 2D Pareto-front example.

(b) Illustrative example of M_3 computation.

Fig. 4. Illustrative examples of the Pareto-based filtering.

this final, pruned graph. We retain one (random) representative image per group, resulting in the deduplicated dataset $\mathcal{I}_{\text{dedup}}$.

CLIP-based OOD assessment. Since $\mathcal{I}$ is partially sourced from uncurated sources and generated using unsupervised methods, it **may contain OOD images.** To assess and filter OOD images from $\mathcal{I}_{dedup}$, we leverage CLIP's zero-shot capabilities, further enhanced by CLIPN [41]. CLIPN utilizes learned dual prompts (*positive* prompts to assess the presence and *negative* prompts to assess the absence of a concept) to accurately determine the relevance of each image to $\mathcal{D}$. Specifically, we compute the similarity of each image I with the positive and negative prompt of the concept c. The probabilities $p_{c,I}, p^{no}_{c,I}$ are then derived by applying softmax to those two similarity scores. *We define a 3-dimensional characterization of how much OOD an image is using* M_1, M_2, *and* M_3:

- **M1:** $\text{OOD}_{\mathcal{B}}(I)$ calculates the OOD score of image I with respect to specific concept bank $\mathcal{B}$ and is computed as: $\text{OOD}_{\mathcal{B}}(I) = 1 - \sum_{c\in\mathcal{B}}(1 - p^{no}_{c,I}) \cdot p_{c,I}$, where $p_{c,I}$ and $p^{no}_{c,I}$ are the probabilities that image I contains and does not contain the concept c respectively.
- **M2: OOD**$_{\mathcal{B}'}(I)$ assesses the OOD score of image I with respect to general domain descriptors $\mathcal{B}' = \{n_D, d_D\}$. $\text{OOD}_{\mathcal{B}'}(I) = 1 - \sum_{c\in\mathcal{B}'}(1 - p^{no}_{c,I}) \cdot p_{c,I}$.
- **M3: Text Influence Mitigation** helps in reducing biases introduced by textual elements that may falsely align an image with the target domain based solely on text presence [27]. To mitigate this influence, we define M_3: $\text{OOD}_{\mathcal{B}}(I') - \text{OOD}_{\mathcal{B}}(I)$, where I' is obtained by using a text-detection algorithm to detect and blur text regions in I. A high M_3 score, indicating that the image is only relevant due to its text, heavily penalizes it during Pareto-front removal (as exemplified in Fig. 4(b)).

Pareto front-based removal. We employ a Pareto front-based multi-objective optimization approach which effectively ranks and eliminates OOD images based on the computed OOD metrics M_1, M_2, and M_3. Instead of prioritizing one single factor, we simultaneously consider all relevant factors to identify and rank the least relevant images. We show a simplified 2D Pareto fronts illustration in Fig. 4(a), where each point represents an image and the first three fronts ($\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$) are highlighted. The Pareto Front removal process is iterative and includes:

1. **Identify Pareto Front**: Determine the first Pareto front $\mathcal{P}_1$ within the deduplicated dataset $\mathcal{I}_{dedup}$. The Pareto front consists of images that are the most OOD across all metrics. An image is part of the Pareto front if there is no other image that simultaneously performs *worse* in all M_1, M_2, M_3. In other words, these images are among the worst in at least one metric and not better in any other metric.
2. **Remove Pareto Front**: Exclude $\mathcal{P}_1$ set from $\mathcal{I}_{dedup}$.
3. **Repeat**: Recompute next Pareto front $\mathcal{P}_2$ on the updated dataset and remove it. Continue this process iteratively (identifying & removing $\mathcal{P}_3$, $\mathcal{P}_4$, etc.) until the dataset meets desired size or quality threshold.

Stage outcome. Pareto-Front ensures that the most irrelevant OOD images are excluded first, maintaining the relevance and quality of the dataset, *leading to the final dataset,* $\mathcal{I}_{final}$.

3.4. Final curated dataset

The final output of PaS is a meticulously curated, domain-specific dataset, assembled autonomously without human oversight. PaS datasets are ideal for self-supervised pretraining of visual models, as well as fine-tuning existing VLMs, thanks to the image-text pairs obtained naturally from the PaS pipeline. These high-quality datasets enable the development of specialized models that excel in their respective domains. In the following sections, we assess the richness and suitability of PaS datasets (Section 4) and evaluate their effectiveness as SSL pretrainers and VLM "fine-tuners" (Section 5).

4. PaS dataset diversity & domain assessment

The primary goal of PaS is to create diverse and useful domain-specific datasets. Beyond curation, we include a low-resource task-agnostic technique for assessing the dataset diversity. In this section, we detail each component by evaluating our PaS datasets against well-established domain-specific supervised SoTA datasets - Food-2K [4], $\text{iNat}_{\text{Birds}}$ [6], AMI-GBIF [7] for *food*, *birds*, and *insects*, respectively. For each domain, we create a corresponding PaS dataset. (PaS-F for *food*, PaS-B for *birds*, PaS-I for *insects*).

4.1. Dataset overview.

PaS datasets, in general, can be created for any given size. For effective comparisons, we use IN-1K and SoTA domain datasets as size references for our datasets. Only for *food* and *birds*, we create mini-versions, $\text{PaS-F}_{\text{Mini}}$ and $\text{PaS-B}_{\text{Mini}}$ as both Food-2K and $\text{iNat}_{\text{Birds}}$ are much smaller than IN-1K. AMI-GBIF on the other hand is larger to IN-1K (PaS-I≈1.9M images). As shown in Table 1, the number of concepts in PaS datasets often exceeds that (classes) of domain-specific datasets (Food-2K has 2K, $\text{iNat}_{\text{Birds}}$ has 1486, AMI-GBIF has >5K classes), highlighting

Table 1
Statistics of the different PaS datasets evaluated in this paper. *Colors represent markers in Fig. 5.*

	Concepts	Raw Images	Final Images	Synth/Real%
PaS-F	3.3k	1.6M	1,177k	60/40
PaS-B	5.0k	1.5M	1,200k	46/54
PaS-I	16.6k	2.5M	1,904k	90/10
$\text{PaS-F}_{\text{Mini}}$	3.3k	1.6M	627k	86/14
$\text{PaS-B}_{\text{Mini}}$	5.0k	1.5M	450k	86/14
$\text{PaS-B}_{\text{Big}}$	5.0k	3.2M	2,421K	49/51

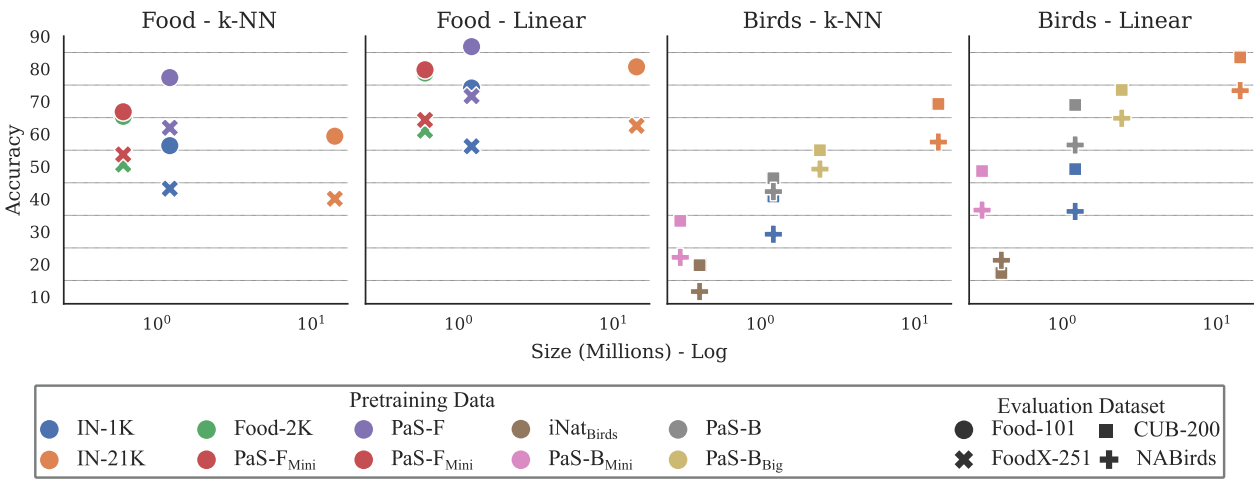


Fig. 5. K-NN Top-1 classification accuracy (in %) of different pretraining datasets against the pretraining dataset size (in log scale).

the diversity of our concept bank. As visualized in Fig. 5, PaS datasets show stronger performances as pretrainers on the same scale as domain-specific supervised datasets and also being way smaller than general-purpose datasets like IN-21K (one order smaller), highlighting the effectiveness of our curation pipeline. Detailed results can be found in Section 5.

4.2. Diversity and domain alignment.

We assess the suitability of PaS datasets by addressing the question: **Does a PaS dataset effectively represent the target domain?** To answer this, we propose a framework with three distinct perspectives: (1) distribution of lexical concepts, (2) distribution of image embeddings, (3) semantic richness of concept-level prototypes. We evaluate our autonomous PaS datasets under each perspective without any specific task. For each assessment, we compare the PaS datasets with SoTA-supervised datasets (Food-2K, $\text{iNat}_{\text{Birds}}$, AMI-GBIF) and other popular domain-specific benchmarks in each domain (Food-101 [5] and FoodX-251 [46] for *food*, CUB-200-2011 [13] and NABirds [14] for *birds*, and different official splits of AMI-GBIF for *insects*).

4.2.1. Distribution of lexical concepts.

LLM-generated concept bank determines the diversity of PaS datasets and should exclusively cover the target domain. To highlight the coverage of domains, we compute lexical embeddings of concepts and labels using OpenAI's CLIP ViT-L/14 text encoder [16] and compare the distributions using a reduced UMAP space [47]. The process involves the following steps:

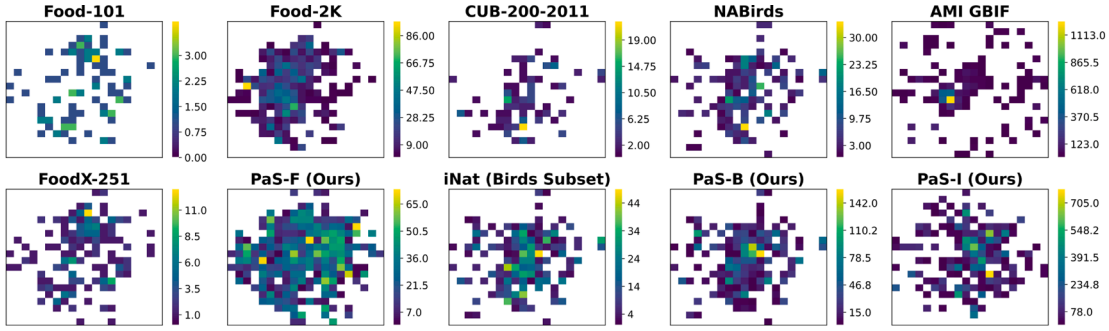
1. Encoding dataset concepts into latent representations using CLIP.
2. Reducing the dimensionality of these embeddings to 2D using UMAP.
3. Dividing the resulting 2D UMAP space into a uniform grid.
4. Creating density maps for each dataset by counting the number of concepts within each grid cell.

We show density maps of lexical concepts for all domains in Fig. 6(a). We opted for density maps over scatter plots to prevent densely populated regions from being obscured, thereby providing a clearer visualization of the concept distribution across different datasets. To prove that the enhanced coverage of PaS datasets is not merely an effect of size, we obtained the same plots after subsampling our concepts to match the scale of the supervised datasets (Fig. 6(b)). In *food*, the lexical distribution demonstrates that the concepts in PaS-F extensively cover the embedding space, effectively bridging the gaps between classes from different datasets. In *birds*, concepts of PaS-B exhibits a wide and varied distribution across the embedding space compared to CUB-200-2011 and NABirds, and it closely matches the granularity of $\text{iNat}_{\text{Birds}}$. This high overlap highlights that the generated concepts align well with the target domain, indicating an effective concept generation. In *insects*, we see less overlap. The fact that AMI-GBIF covers mostly moth species, a subset of all insects (target of PaS-I) could be a possible reason. Overall, the broad **coverage and significant alignment** across domains in both Fig. 6(a) and (b) underscore the ability of PaS-generated concepts to enrich dataset diversity and relevance to specific domains.

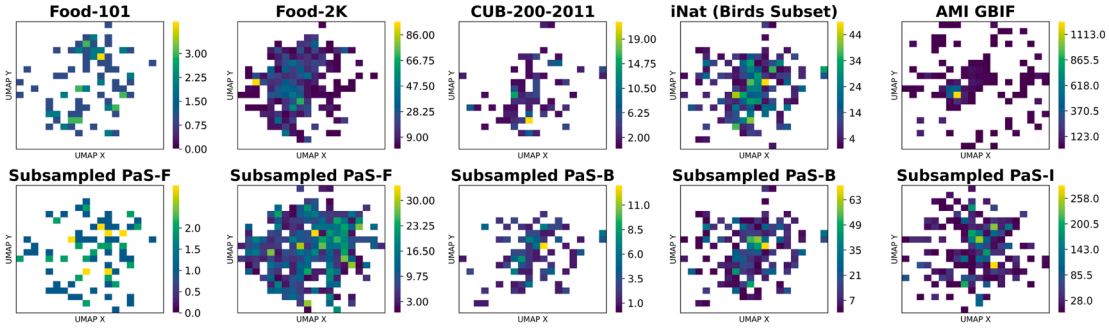
4.2.2. Distribution of image embeddings.

Similar to the analysis of lexical concepts, we also compare the image distributions of the different datasets. This process is completely analogous to the previous one. The main differences are that, in this case, we obtain an embedding per image (not per class) and that we use a ResNet-152 pretrained on IN-21K to compute embeddings. We then compare the latent distributions for each domain, which can be seen in Fig. 6(c).

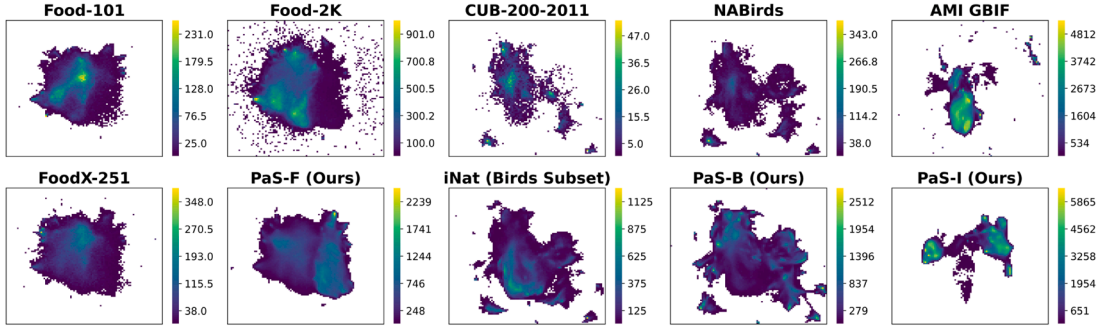
A good alignment and span in feature space would prove a comprehensive coverage of the target domain. In both *food* and *birds*, there is a notable alignment across all datasets. The impact of larger datasets, which exhibit fewer gaps in the visual space, is particularly evident in PaS-B. PaS-B achieves the most extensive coverage of the embedding space among the considered datasets. Upon analyzing the density distribution, we observe that while the CUB-200-2011 dataset has densely populated regions, PaS-B, along with others, display a more uniform distribution across the embedding space. This uniformity is attributed to two key factors: (1) A reduced number of images in CUB-200-2011, and, (2) A limited variety of bird species in CUB-200-2011 compared to other datasets. Similarly, in *food*, Food-2K spans a broader area but includes numerous outliers, potentially indicating OOD images (see Fig. 7 for some outliers examples). In contrast, PaS-F encompasses the embedding spaces of both Food-101 and Food-2K. In particular, PaS-F, exhibits a uniform distribution of embeddings that is well-balanced between densely covered and sparsely populated regions, which suggests a comprehensive representation of the *food* domain. Regarding *insects*, while the overall shape of the distribution aligns across datasets (PaS-I being slightly broader), the densities vary significantly. Sample-wise, underrepresentation of certain images and semantics in AMI-GBIF might cause potential differences. Specifically, most images in AMI-GBIF are concentrated in a particular area of the embedding space. A closer examination of the semantics of each dataset provides further insights into this behavior (see Section 4.2.3). Overall, these observations underscore the effectiveness of PaS in automatically creating large-scale, domain-specific datasets in terms of image coverage and alignment with existing datasets. This is not just an artifact of their larger size, as the trend



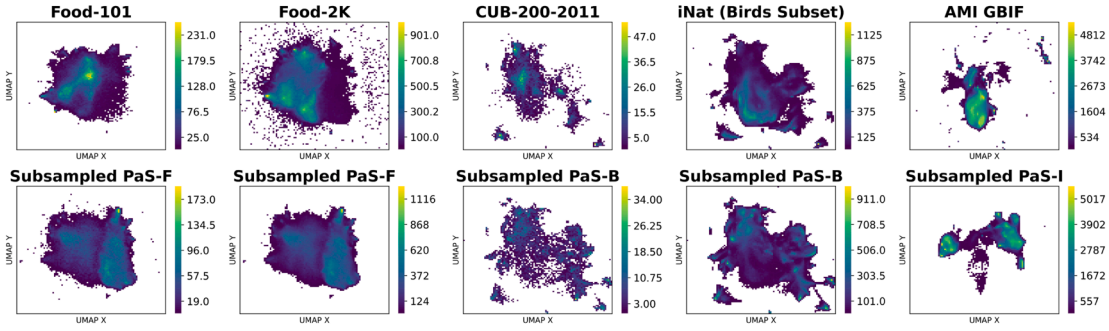
(a) Lexical space.



(b) Lexical space, comparison on the same scale. In each column, the number of concepts in the PaS dataset has been randomly subsampled to match the size of the vocabulary of the supervised dataset.



(c) Image embeddings.

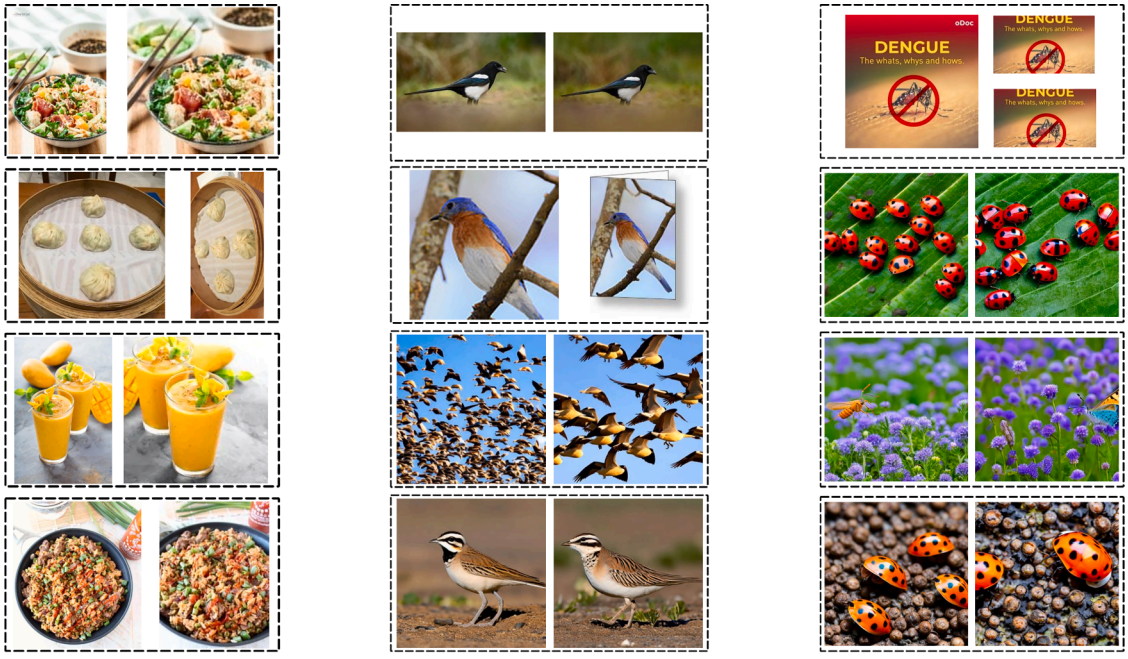


(d) Visual space, comparison on the same scale. In each column, the number of images in the PaS dataset has been randomly subsampled to match the size of the supervised dataset.

Fig. 6. Density maps of the UMAP representations for the lexical space of concepts and image embeddings across all domain-specific datasets. The scale indicates concept or image concentration in the latent space.

Fig. 7. Examples of OOD images in Food-2K dataset, identified in Fig. 6(c).

Fig. 8. Extracted exclusive prototypes examples. Each image belongs to an exclusive prototype.



(a) PaS-F (b) PaS-B (c) PaS-I

Fig. 9. Examples of duplicates found and removed in the creation of the PaS datasets.

holds when our datasets are subsampled to an equal number of images (Fig. 6(d)), highlighting a more diverse collection of visual features.

4.2.3. Semantic richness analysis

Semantic richness refers to the depth and diversity of concepts and information captured within a dataset, and evaluates how well a dataset captures the nuances, variations, and comprehensive coverage of concepts within a specific domain. A "prototype" [48] is a representative semantic that captures the most salient features of a concept, making them useful for understanding data diversity and exhaustiveness.

We show examples of the most relevant exclusive prototypes for each dataset in Fig. 8 and the number of unique prototypes with image count in Table 2. *Birds* has 8159 shared prototypes, indicating a substantial overlap between datasets. Comparing unique prototypes, PaS-B contains much greater diversity both in the prototypes and the population of those prototypes. *Food* has 8144 shared prototypes. Food-2K is human-labelled but still contains several repetitions that fill its exclusive prototypes. Exclusive prototypes of PaS-F highlights the diversity not included in Food-2K. In *insects*, there are 8188 shared prototypes. PaS-I contains all the semantics available in AMI-GBIF, evidenced by not identifying any AMI-exclusive prototype. While the number of samples per shared prototype might vary, we can confidently claim that PaS-I manages to match and improve the diversity provided by AMI-GBIF. Comparing the most significant shared prototype, we find that the most populated AMI-GBIF prototype represents a human-driven concept (quantity over diversity). The differences in the behaviour of distributions found in previous sections can be explained by the prevalence of this prototype in the AMI dataset. More details are provided in Appendix A.

Table 2
Exclusive prototypes for PaS and SoTA supervised datasets.

Dataset	Exclusive	Images
Food-2K	16	121
PaS-F	**32**	**310**
$iNat_{Birds}$	10	25
PaS-B	**23**	**358**
AMI	0	0
PaS-I	**4**	**650**

4.3. Illustrative examples of PaS dataset curation

In this section, we show sample images from PaS-F, PaS-B, and PaS-I datasets during different stages of PaS. The deduplication step has a significant impact, removing over 70,000, 41,000, and 27,000 near-duplicate groups for the food, birds, and insects domains, respectively. Fig. 9 illustrates groups of duplicate images identified by the PaS curation pipeline across different domains. Our pipeline detects three types of duplicates: (1) exact duplicates, (2) different crops of the same image, and (3) images containing very similar scenes. Having removed duplicate images, the next step in Stage 3 involves Pareto-based removal to further refine the dataset. Fig. 10 showcases examples from various Pareto fronts ($\mathcal{P}_1, \mathcal{P}_2, \ldots$) identified in $\mathcal{I}_{dedup}$ for each considered domain. The upper rows display images with the highest OOD scores as determined by PaS. Across all three domains, we observe a consistent pattern: there is a clear correlation between an image's Pareto position and its relevance to the target domain, underscoring the quality of the PaS datasets. Specifically, the initial images (top rows) removed through Pareto-front based filtering are generally not closely related to the target domain. In contrast, images retained in subsequent iterations (bottom rows) exhibit greater relevance. Some retrieved web images do not align well with the target domains, highlighting the necessity of a curation step. This is expected, as the underlying web-scale data pool (from Re-LAION-5B) includes varied content, from natural images to the sketches and signs visible in the figure's top rows. For instance, the second image in $\mathcal{P}_1$ of Fig. 10(c) demonstrates how misleading textual cues can affect the retrieval process, emphasizing the importance of the OOD metric M_3. While synthetic image generation is designed to remain within the target domain-ensuring most images are well-aligned-misalignments can occur due to captioning errors. An example is the presence of houses in the first Pareto front of Fig. 10(a). Additionally, synthetic images that are unrealistic, such as those in the last row of Fig. 10(c), are excluded from the final dataset. Finally, Fig. 11 contains samples from each of the final curated PaS datasets PaS-F, PaS-B, and PaS-I (9 real and 9 synthetic).

5. Validation of PaS datasets

This section validates the effectiveness of PaS datasets as pretrainers using common SSL downstream tasks. We employ a comprehensive evaluation, including feature evaluation via k-NN and linear probing, and adaptation through full and few-shot fine-tuning. We also explore fine-tuning large VLMs using PaS image-text pairs as a way to increase their competence in each domain. Additionally, we evaluate PaS datasets on real-world dense tasks in Appendix B. These experiments thoroughly evaluate the quality and transferability of representations learned from PaS across visual tasks.

5.1. Experimental setup

We compare models pretrained on PaS datasets with SoTA domain-specific datasets (Food-2K, $iNat_{Birds}$, and AMI-GBIF) and large-scale general domain datasets (IN-1K, IN-21K, and WIT (CLIP dataset) [16]). For all evaluations, we use a ViT-B/16 [10]. For a fair comparison, we pretrain using a common MoCo-v3 SSL setup, except for IN-21K and WIT, for which we use official *supervised* checkpoints (making them stronger discriminative baselines). To prevent data leakage, we also remove images from the PaS datasets that are similar to the evaluation test sets. The specific hyperparameters and detailed procedures are provided in Appendix C.

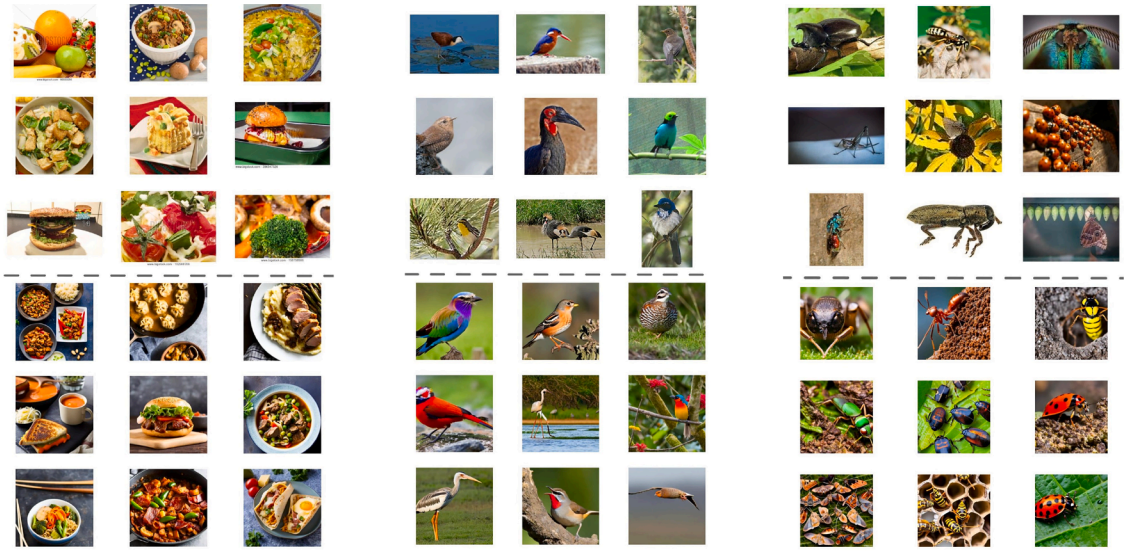
5.2. PaS Datasets as SSL pretrainers

5.2.1. Feature evaluation with k-NN & linear probing

To ensure a fair comparison and prevent potential bias, we exclude models overlapping with evaluation datasets ($iNat_{Birds}$, Food-2K, and AMI-GBIF). For linear probing, we use the setup detailed in Table C.10(b), however, only training the added linear layer (the backbone remains frozen). We report k-NN and Linear Probing performance evaluated on common benchmarks - Food-101 [5], FoodX-251 [46] for *food* (Table 3), CUB-200-2011 [13], and NABirds [14] for *birds* (Table 4) and k-NN performance on different AMI region subsets for

(a) PaS-F (b) PaS-B (c) PaS-I

Fig. 10. Examples of images removed during the Pareto-based removal. "P n" refers to images belonging to the nth Pareto front (removed in the nth iteration).



(a) PaS-F (b) PaS-B (c) PaS-I

Fig. 11. Examples of images in the final PaS datasets. For each domain, the first three rows display web images, followed by three rows of synthetic images.

Table 3
Classification Acc. (%) results for *food* domain. Best (bold), Second best (underlined) comparing only SSL models. † represents supervised pretraining.

PT Data	Size (M)	Food-101		FoodX251		Food-2K
		k-NN	Linear	k-NN	Linear	k-NN
IN-21K†	14	59.3	80.6	40.1	62.5	54.1
WIT†	400	81.9	90.8	61.5	73.3	61.3
IN-1K	1.2	56.4	74.1	43.2	56.2	57.2
Food-2K	0.6	65.4	78.7	50.6	61.0	–
PaS-F_{Mini}	0.6	66.8	79.7	53.7	64.3	59.2
PaS-F	1.2	**77.3**	**86.8**	**61.9**	**71.6**	**64.2**

Table 4
Classification Acc. (%) results for *birds* domain. Best (bold), Second best (underlined) comparing only SSL models. † represents supervised pretraining.

PT Data	Size (M)	CUB-200		NABirds		$iNat_{Birds}$
		k-NN	Linear	k-NN	Linear	k-NN
IN-21K†	14	69.2	83.5	57.5	73.2	32.4
WIT†	400	63.0	81.2	50.8	71.5	31.7
IN-1K	1.2	40.8	49.2	29.2	36.2	18.3
$iNat_{Birds}$	0.4	19.7	30.6	11.2	21.2	–
PaS-B_{Mini}	0.4	33.3	48.6	22.1	36.6	12.15
PaS-B	1.2	46.4	68.9	42.3	56.6	24.9
PaS-B_{Big}	2.4	**55.0**	**73.5**	**49.2**	**64.8**	**27.7**

insects (Table 5). Overall, the results highlight that models pretrained on PaS-generated datasets consistently outperform those pretrained on domain-specific datasets, with an avergae improvement of 6.7 % accross all considered tasks and domains. In particular, even the "Mini" versions achieve higher accuracies than supervised domain-specific datasets such as $iNat_{Birds}$ and Food-2K (same scale), highlighting the effectiveness of our pipeline in autonomously generating high-quality domain-specific datasets without manual labeling.

PaS-F datasets exhibit remarkable performance gains (Table 3). PaS-F (1.2M images) achieves a k-NN accuracy of 77.3 % on Food-101, significantly exceeding IN-1K (56.4 %) and even outperforming IN-21K (59.3 %), which is 12× larger. PaS-F competes closely with WIT (400M supervised dataset), outperforming it in k-NN for FoodX-251 and Food-2K. Similarly, PaS-B datasets (Table 4) consistently outperform the supervised $iNat_{Birds}$ on all evaluations. PaS-B significantly outperforms IN-1K with improvements ranging from 5.6 % to 20.4 %. Although IN-21K (14M images) achieves higher accuracy, PaS-B_{Big} (an enlarged version of PaS-B) narrows the gap from an average of 15.34 % to 9.12 % despite being nearly 6× smaller. This proves the ability of PaS to effectively scale the size of the dataset. PaS-I (Table 5) achieves the best results in the regional subsets (Central America, North America and Europe) of the AMI dataset at the family level, outperforming both IN-1K, IN-21K and WIT. The improvement is highest in CA, with a relative improvement of 20 % over the second-best pretrainer. Although IN-1K slightly outperforms PaS-I in the global task ("All"), PaS-I remains competitive with a difference below 0.4 %. In the "AMI Binary" classification task, PaS-I falls only behind the laboriously compiled AMI dataset (on the same scale), which was expected since the task is also part of the AMI benchmark. This alignment between the training (AMI) and test data is in accordance with Section 4.2.3, where we identified a dominant pattern in the AMI GBIF dataset (present in all the partitions).

Table 5
Classification Acc. (%) k-NN results for *insect* domain in different AMI partittions. Best (bold), Second best (underlined) comparing only SSL models. † represents supervised pretraining.

PT Data	Fine-grained (Regions)				Binary
	C-America	N-America	Europe	All	
IN-21K†	15.1	38.0	34.2	38.5	67.9
WIT†	15.1	26.7	23.5	26.6	60.6
IN-1K	16.0	38.2	34.3	**39.9**	62.4
AMI GBIF	–	–	–	–	**75.4**
PaS-I	**19.2**	**38.4**	**35.1**	39.5	68.9

5.2.2. *Full fine-tuning*

For full fine-tuning, we follow the setup in Table C.10(b). Given the number of experiments and the computational demands of this downstream task, the largest datasets are disregarded from this experiments. This includes Food-2K, $iNat_{Birds}$ and all the AMI partitions. To provide comprehensive results, the latter are included in the few-shot study (Following section). The results are displayed in Table 6. On the *food* domain, PaS datasets exhibit better performance than existing manually curated datasets (both general and domain-specific). Particularly, PaS-F and PaS-F_{Mini} attain the best and second-best results in both datasets, respectively. The larger version, PaS-F, beats the best non-PaS alterna-

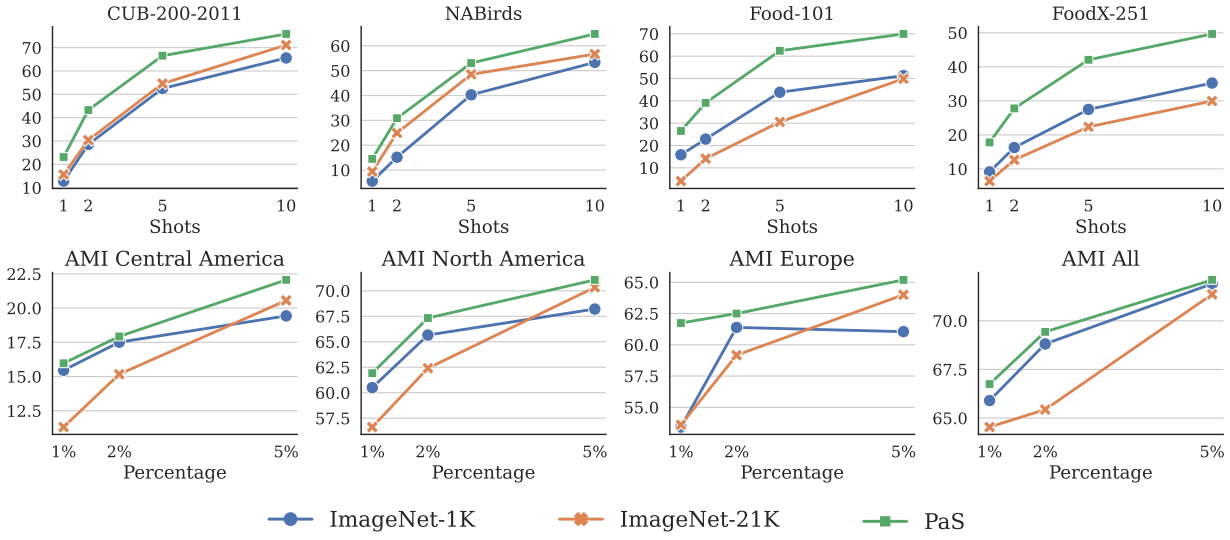


Fig. 12. Results on few-shot fine-tuning on ViT-B/16.

Table 6
Full fine-tuning on *food* and *birds*. In all the cases, the architecture used is ViT-B/16. † - supervised denotes backbones pretrained in a supervised way.

PT	Size	CUB	NABirds
IN-1K	1.2	78.5	73.2
IN-21K†	14.0	81.3	73.1
$iNat_{Birds}$	0.4	76.8	70.0
PaS-B_{Mini}	0.4	79.5	73.0
PaS-B	1.2	82.9	77.1
PaS-B_{Big}	2.4	**83.2**	**77.2**
PT	Size	F101	FX251
IN-1K	1.2	87.1	71.7
IN-21K†	14.0	86.3	70.1
Food-2K	0.6	87.9	72.5
PaS-F_{Mini}	0.6	88.3	73.6
PaS-F	1.2	**89.1**	**74.3**

tive by 2.08 % in Food-101 and 2.68 % in FoodX-251. PaS-B outperforms both domain-specific and general datasets in CUB-200 and NABirds, with an increase of 1.63 % and 3.94 % over the best non-PaS dataset, respectively. PaS-B_{Big}, further improves performance, reaching 83.2 % in CUB-200 and 77.2 % in NABirds. Furthermore, PaS-B_{Mini} beats $iNat_{Birds}$ (same scale) by at least 2.75 %.

5.2.3. Few-shot fine-tuning.

The ability to achieve high performance with limited, labeled data is crucial in many real-world applications, where acquiring large annotated datasets is expensive or impractical. Few-shot learning addresses this challenge by aiming to learn effectively from only a few examples per class. To evaluate the data efficiency of representations learned from PaS datasets, we perform fine-tuning with reduced portions of training datasets in the three domains (results displayed in Fig. 12). Backbones pretrained on PaS datasets outperform general-domain datasets in all considered scenarios, while being even an order of magnitude smaller. PaS-pretrained backbones achieve competitive performance (even surpass) other backbones when using fewer images for fine-tuning, highlighting, the potential of PaS datasets to be data-efficient and show better learnability of useful features for the desired domain. For this downstream task, we use the same setup as full fine-tuning, only reducing the number of samples per class.

5.3. Fine-tuning of VLMs

Multimodality is a core trait of our PaS pipeline. As a result, the final generated image dataset is a paired text associated with each image (the web caption in the case of web-retrieved images, and the generation prompt in the case of synthetic images). To assess the capacity and relevance of PaS in this regard, we leverage the text-image pairs to fine-tune pretrained VLMs, namely CLIP [16] and SigLIP [17], using LoRA [15]. In this task, the original weights of the VLM remain frozen, and only the LoRA matrices are updated. Specifically, LoRA is applied to the Q, K, and V linear layers of all transformers in the vision encoder. For both CLIP [16] and SigLIP [17] fine-tuning, we adopt the same hyperparameters as in Zanella and Ben Ayed [49], with a scheduler of 5 epochs. This approach allows for efficient fine-tuning by less than only 1 % of the model parameters using PaS' image-text pairs.

For evaluation, we take the visual encoder of the fine-tuned model and use it for k-NN and linear evaluation. We use ViT-B/16-based CLIP and SigLIP, pretrained on massive web-scale data: 400M and 4.8B images, respectively. Results in Table 7 show that PaS leverages the knowledge of CLIP to create datasets with enriched knowledge, yielding incremental improvements across domains with an average gain of 4.2 %. Regarding SigLIP, PaS datasets enable improvements in all domains while only representing a small fraction (0.03 %) of the original pretraining data of SigLIP. The results highlight the effectiveness of PaS in creating targeted fine-tuning datasets for complex domains, enabling improvements even when retraining foundational models is impractical due to continuous new data and domains.

5.4. Ablations

We perform ablations of key components of PaS (Table 8) using the *food* domain.

1. **Concept Discovery:** Our LLM-guided concept generation outperforms a PaS dataset curated directly from Food-101 concepts, highlighting greater generalization and diversity. Evaluating Stage 1, our LLM-based concept generation (**PaS** – $\mathbf{F}_{Mini}$) outperforms a manually curated set of concepts from Food-101 (**PaS** – $\mathbf{F}_{F101}$), indicating greater generalization and diversity.
2. **Data Distribution:** We ablate the automatic synthetic-to-real ratio of PaS by manually setting them. To achieve these precise ratios for the ablation, the curation methodology was modified: while deduplication is performed on the union of real and synthetic data to remove all near-duplicates globally, the subsequent Pareto-front

Table 7
PaS vs. CLIP and SigLIP using ViT-B/16 as visual encoder. We fine-tuned the VL models using LoRA with PaS datasets for 5 epochs.

	#Imgs	CUB		$iNat_B$	F101		F2K	AMI_{CA}	AMI_{All}
		kNN	Lin.	kNN	kNN	Lin.	kNN	kNN	kNN
CLIP (WIT)	400M	63.0	81.2	61.3	81.9	90.8	31.7	15.1	26.6
SigLIP	~ 4.8B	70.8	82.6	70.0	89.7	93.9	35.5	16.5	39.0
PaS	1.2M	46.4	68.9	64.2	77.3	86.8	24.9	**19.2**	**39.5**
&**CLIP**†	+1.2M	67.4	82.8	67.6	87.3	92.2	**37.5**	15.9	36.1
&**SigLIP**†	+1.2M	**71.3**	**82.9**	**70.7**	**90.1**	**94.2**	35.9	16.7	39.4

Table 8
Ablation on *food* using ViT-B/16 (k-NN and Linear probing). Columns: LLM-based concept discovery (LLM), Synth/Web proportions ('A' = automatic), deduplication (SF), and Pareto filtering (PF). Evaluation results are reported for Food-101, FoodX-251, and Food-2K on two tasks (k-NN and Linear).

PT Data	LLM	Synth	Web	SF	PF	Food-101		FoodX-251		F-2K
						k-NN	Linear	k-NN	Linear	k-NN
PaS-F_{F101}		A	A	✓	✓	67.03	81.12	52.94	64.12	59.22
PaS-F^{0-100}_{Mini}	✓	0	100	✓	✓	65.79	79.20	52.80	63.87	59.24
PaS-F^{25-75}_{Mini}	✓	25	75	✓	✓	65.84	79.11	52.83	63.84	59.30
PaS-F^{50-50}_{Mini}	✓	50	50	✓	✓	66.04	79.38	53.07	63.98	59.83
PaS-F^{75-25}_{Mini}	✓	75	25	✓	✓	66.79	79.93	53.82	64.38	60.67
PaS-F^{100-0}_{Mini}	✓	100	0	✓	✓	66.53	79.42	53.64	64.12	60.05
PaS-F_{Mini}	✓	A	A	✓	✓	66.75	79.68	53.68	64.29	60.46
midrule PaS_{NF}	✓	A	A			70.89	81.50	57.39	66.33	61.23
PaS_{SF}	✓	A	A	✓		73.11	83.48	59.44	68.15	62.45
PaS-F	✓	A	A	✓	✓	**77.31**	**86.84**	**61.86**	**71.61**	**64.23**

filtering is applied to the real and synthetic sets independently. Comparing manually set ratios ($\mathbf{PaS-F^{0-100}_{Mini}}$, $\mathbf{PaS-F^{25-75}_{Mini}}$, $\mathbf{PaS-F^{50-50}_{Mini}}$, $\mathbf{PaS-F^{75-25}_{Mini}}$, and $\mathbf{PaS-F^{100-0}_{Mini}}$) with the automatically distribution determined by PaS, $\mathbf{PaS\text{-}F_{Mini}}$ achieves competitive (a close second) performance without manual tuning, demonstrating an effective balance between both data sources. More importantly, while both pure synthetic and pure web datasets are effective, their hybrid combination consistently yields the best performance, reinforcing the value of our hybrid data collection approach.

3. **Data Curation:** We study the effectiveness of our dataset reduction by comparing unfiltered ($\mathbf{PaS}_{NF}$), deduplicated ($\mathbf{PaS}_{SF}$) and Pareto front-based OOD removal (**PaS – F**) of which the complete reduction yields the best performance, with fewer but more relevant images.

In summary, our ablation experiments highlight the effectiveness of each component of the PaS pipeline. The LLM-guided concept generation provides superior generalization, the automatic web-to-synthetic ratio achieves a balanced performance without manual tuning, and the data curation process effectively removes irrelevant images, leading to the best overall results.

5.5. Limitations

While PaS generates high-quality datasets and demonstrates promising pretraining results, it is crucial to identify its limitations.

1. While PaS leverages the knowledge of large pretrained models, such as LLMs and text-to-image generators, its effectiveness can be influenced by their general capabilities. Even if the design of PaS has been proven robust in several complex domains, limitations or biases in these foundational models for highly specialized domains might affect the generated dataset's applicability.
2. Evaluation pipeline is similarly sensitive, as the quality of the target domain can impact latent representations and lexical alignment analysis. However, PaS's modular architecture allows for easy replacement or upgrading of these components, effectively mitigating both limitations.
3. PaS leverages efficient large models, however, depending on model choice, computational demand may increase, potentially limiting the quality in low-resource settings. Using techniques such as model quantization can alleviate this limitation.

6. Conclusion

While being critical for real-world deployment, creating domain-specific datasets at scale is challenging due to its high costs. In this paper, we introduce PaS, an innovative framework for autonomously generating domain-specific datasets on-demand. PaS is modular, facilitating integration of various SoTA components like LLMs and VLMs, offers adaptability across diverse domains, and, allows future advancements. PaS incorporates efficient pruning techniques to provide high-performance relevant datasets with reduced size, tackling a key challenge in dataset curation. Our comprehensive task-agnostic analysis highlights our framework's ability to produce scalable datasets that surpass the richness and diversity of conventionally curated domain-specific SoTA datasets. When pretrained on PaS datasets, models achieve competitive or superior results compared to traditional supervised datasets of the same scale. Empirical results show that PaS datasets outperform IN-1K across all tested domains and even surpass IN-21K supervised setup in *food* and *insects* domain while being an order of magnitude smaller. Moreover, PaS multi-modality enables cost-effective adaption of existing large VLMs to new domains, easing their usage in new applications. PaS represents a paradigm shift: automatically generated domain-specific datasets can bridge the gap with large general-domain datasets, providing scalable, flexible solutions with enhanced performance in specialized applications. The efficiency of PaS makes it a promising approach for real-world deployment, including resource-constrained environments. Future directions include testing advanced components for complex domains, analysing data and distribution biases, and training cross-modal models.

CRediT authorship contribution statement

Jesús M. Rodríguez-de-Vera: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Imanol G. Estepa:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization; **Ignacio Sarasúa:** Writing – review & editing, Validation; **Bhalaji Nagarajan:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization; **Petia Radeva:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Co-author Petia Radeva is one of the Editors-in-Chief of the journal Pattern Recognition. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00) and IDEATE (AEI-MICINN, PID2022-141566NB-I00). J. M. Rodríguez-de-Vera and Imanol G. Estepa acknowledge the support of FPU Becas with code FPU22/03116 and FPU23/02822 respectively, Ministry of Universities, Spain. B. Nagarajan acknowledges AI4S fellowship within the "Generación D" initiative by Red.es, Ministerio para la Transformación Digital y de la Función Pública, for talent attraction (C005/24-ED CV1), funded by NextGenerationEU through PRTR. The authors thankfully acknowledge EuroHPC Joint Undertaking (EHPC-DEV-2023D12-059) for awarding us access to Leonardo at CINECA, Italy and Spanish Supercomputing Network (RES) (IM-2023-3-0019) for awarding us access to MareNostrum5 at BSC, Spain.

Appendix A. PaS dataset diversity & domain assessment - semantic richness analysis in the insects domain

As mentioned in Section 4.2.3, the most populated AMI-GBIF prototype represents a human-driven concept (quantity over diversity). PaS-I, manages to include similar concept images with much fewer samples, as it focuses on the *insects* domain without a human strategy that could bias the dataset. Specifically, this prototype appears in 38.94 % of AMI GBIF images, compared to only 0.89 % in PaS-I. Fig. A.13 showcases examples of images associated with this prototype. These images predominantly feature well-centered insects (mainly moths) captured on white surfaces. In contrast, PaS-I images-both web-sourced and synthetic-typically depict insects in natural environments. This fundamental difference accounts for the distribution discrepancies highlighted in the previous section. The prototypes across different domains demonstrate the richness of PaS datasets, highlighting that PaS datasets not only match but potentially exceed the semantic diversity of existing human-curated datasets.

Fig. A.13. Samples from the AMI GBIF dataset belonging to its most populated prototype.

Appendix B. Additional evaluations of PaS datasets as pretrainers

B.1. Domain-specific evaluation

In addition to the comprehensive experimentation in Section 5, we assess our pretrained models on two specialized real-world tasks: semantic segmentation for *food* using FoodSeg103 dataset [50] and keypoint detection for *birds* using Birdsnap dataset [51].

Semantic segmentation of food. Table B.9 contains the results for *food* semantic segmentation using *FoodSeg103* [50]. In all cases, we use default configurations provided by MMSegmentation[3] without any hyperparameter tuning. We use a scheduler of 80K training steps. The results indicate that, ViT-B backbone pretrained with PaS-F outperforms other backbones, including both ViT-B and Swin [9], trained on different datasets. Notably, significant improvements are observed in both mIoU and mAcc metrics. These findings highlight the effectiveness of specialized datasets like PaS-F in enhancing performance on dense prediction tasks such as semantic segmentation.

Table B.9
Results on the downstream task of semantic segmentation for the dataset *FoodSeg103* [50]. Results for *ImageNet-21K* pretraining have been taken from the *FoodSeg103* paper [50]. The other experiments follow the same setup.

PT Data	Method	Backbone	mIoU	mAcc
IN-21K	SETR [52]	ViT-B/16	41.3	52.7
	UperNet [53]	Swin-S	41.6	53.6
		Swin-B	41.2	53.9
Food-2K	UperNet [53]	ViT-B/16	38.1	51.1
PaS-F		ViT-B/16	**42.5**	**55.5**

Keypoint detection on birds. For keypoint detection using Birdsnap [51], we compare ViT-B/16 performance pretrained on IN-21K with that pretrained on PaS-B_{Big}, using ViTPose [54] with default hyperparameters. Despite PaS-B_{Big} being an order of magnitude smaller, it achieves superior results in mean mAP: 59.6 % compared to 56.3 % for IN-21K pretrained models.

Appendix C. Experimental setup

For all our experiments, we use ViT-B/16 as the visual encoder [10]. To ensure a fair comparison across datasets, we pretrain the encoders from scratch using a common SSL setup: MoCo-v3 [35] for 300 epochs. The only exceptions are IN-21K and WIT, for which we use the officially released, supervisedly trained weights [16,55], making them stronger discriminative baselines.

Table C.10
Detailed setups used for the different pretraining and fine-tuning.

Parameter	Value
(a) Moco v3 pretraining setup	
Backbone	VIT-B/16
Batch Size	4096
Learning Rate	2.4×10^{-3}
Scheduler	Cosine
Optimizer	AdamW
Epochs	300
Warmup Period	40
Weight Decay	0.1
Parameter	**Value**
(b) Fine-tuning setup	
Backbone	VIT-B/16
Batch Size	256
Learning Rate	5×10^{-4}
Scheduler	Cosine
Optimizer	AdamW
Epochs	100
Warmup Period	10
Weight Decay	0.05

[3] https://github.com/open-mmlab/mmsegmentation

To ensure that our evaluation is not biased by overlapping data, we meticulously remove images from the PaS datasets that are similar to those in the downstream evaluation test sets. For this, we use the state-of-the-art duplicate detection model, SSCD [44], following the procedure outlined in Oquab et al. [1]. This process is analogous to the deduplication step in our curation pipeline (see Section 3.3) but employs a stricter similarity threshold of 0.45 to aggressively remove any potential near-duplicates of the test images from our final pretraining datasets.

References

[1] M. Oquab, T. Darcet, T. Moutakanni, H.V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, DINOv2: learning robust visual features without supervision, Trans. Mach. Learn. Res. (2024).

[2] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, D. Tao, A survey on self-supervised learning: algorithms, applications, and future trends, IEEE Trans. Pattern Anal. Mach. Intell. 46 (12) (2024) 9052–9071.

[3] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, J. Tang, Self-supervised learning: generative or contrastive, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 857–876.

[4] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, S. Jiang, Large scale visual food recognition, IEEE Trans. Pattern Anal. Mach. Intell. 45 (8) (2023) 9932–9949.

[5] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: ECCV, 2014, pp. 446–461.

[6] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: CVPR, 2018, pp. 8769–8778.

[7] A. Jain, F. Cunha, M.J. Bunsen, J.S. Cañas, L. Pasi, N. Pinoy, F. Helsing, J. Russo, M.S. Botham, M. Sabourin, J. Fréchette, A. Anctil, Y. Lopez, E. Navarro, F. Pérez, A.C. Zamora, J.A. Ramirez-Silva, J. Gagnon, T.A. August, K. Bjerge, A.G. Segura, M. Belisle, Y. Basset, K.P. McFarland, D.B. Roy, T.T. Høye, M. Larrivee, D. Rolnick, Insect identification in the wild: the AMI dataset, in: ECCV, 2024.

[8] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al., Laion-5b: an open large-scale dataset for training next generation image-text models, Adv. Neural Inf. Process. Syst. 35 (2022) 25278–25294.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: ICCV, 2021, pp. 10012–10022.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16×16 words: transformers for image recognition at scale, ICLR (2021).

[11] H.A. A.K. Hammoud, H. Itani, F. Pizzati, P. Torr, A. Bibi, B. Ghanem, SynthCLIP: Are We Ready for a Fully Synthetic CLIP Training?, (2024). arXiv:2402.01832

[12] Y. Tian, L. Fan, K. Chen, D. Katabi, D. Krishnan, P. Isola, Learning vision from models rivals learning vision from data, in: CVPR, 2024, pp. 15887–15898.

[13] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset (2011).

[14] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, S. Belongie, Building a bird recognition app and large scale dataset with citizen scientists: the fine print in fine-grained dataset collection, in: CVPR, 2015.

[15] E.J. Hu, y. shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: low-rank adaptation of large language models, in: ICLR, 2022.

[16] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: ICML, PMLR, 2021, pp. 8748–8763. ISSN: 2640–3498.

[17] X. Zhai, B. Mustafa, A. Kolesnikov, L. Beyer, Sigmoid loss for language image pre-training, in: ICCV, 2023, pp. 11975–11986.

[18] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I.S. Kweon, S. Xie, Convnext v2: co-designing and scaling convnets with masked autoencoders, in: CVPR, 2023, pp. 16133–16142.

[19] X. Zhai, A. Kolesnikov, N. Houlsby, L. Beyer, Scaling vision transformers, in: CVPR, 2022, pp. 12104–12113.

[20] A.C. Li, E.L. Brown, A.A. Efros, D. Pathak, Internet explorer: targeted representation learning on the open web, in: ICML, 2023, pp. 19385–19406.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: CVPR, 2022, pp. 10684–10695.

[22] H. Chang, H. Zhang, J. Barber, A.J. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W.T. Freeman, M. Rubinstein, Y. Li, D. Krishnan, Muse: text-to-image generation via masked generative transformers, in: ICML, 2023.

[23] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, D.J. Fleet, Synthetic data from diffusion models improves ImageNet classification, Trans. Mach. Learn. Res. (2023).

[24] H. Xu, S. Xie, X. Tan, P.-Y. Huang, R. Howes, V. Sharma, S.-W. Li, G. Ghosh, L. Zettlemoyer, C. Feichtenhofer, Demystifying CLIP Data, in: ICLR, 2024.

[25] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, L. Fei-Fei, The unreasonable effectiveness of noisy data for fine-grained recognition, in: ECCV, Springer, 2016, pp. 301–320.

[26] Z. Qin, z. xu, Y. Zhou, K. Wang, Z. Zheng, Z. Cheng, H. Tang, L. Shang, B. Sun, R. Timofte, X. Peng, H. Yao, Y. You, Dataset growth, in: ECCV, 2024.

[27] P. Maini, S. Goyal, Z.C. Lipton, J.Z. Kolter, A. Raghunathan, T-MARS: improving visual representations by circumventing text feature learning, in: ICLR, 2024.

[28] Y. Zheng, H. Yang, T. Zhang, J. Bao, D. Chen, Y. Huang, L. Yuan, D. Chen, M. Zeng, F. Wen, General facial representation learning in a visual-linguistic manner, in: CVPR, 2022, pp. 18697–18709.

[29] V. Arannil, N. Narwal, S.S. Bhabesh, S.N. Thirandas, D.Y.-B. Wang, G. Horwood, A.A. Chirayath, G. Pandeshwar, DoPAMine: Domain-specific Pre-training Adaptation from seed-guided data Mining, (2024). arXiv:2410.00260

[30] I. Ziegler, A. Köksal, D. Elliott, H. Schütze, CRAFT Your Dataset: Task-Specific Synthetic Dataset Generation Through Corpus Retrieval and Augmentation, (2024). arXiv:2409.02098

[31] H.A. A.K. Hammoud, T. Das, F. Pizzati, P. Torr, A. Bibi, B. Ghanem, On pretraining data diversity for self-supervised learning, in: ECCV, 2024.

[32] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, PMLR, 2020, pp. 1597–1607.

[33] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: CVPR, 2020, pp. 9729–9738.

[34] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, With a little help from my friends: nearest-neighbor contrastive learning of visual representations, in: CVPR, 2021, pp. 9588–9597.

[35] X. Chen, S. Xie, K. He, An empirical study of training self-supervised vision transformers, in: ICCV, 2021, pp. 9640–9649.

[36] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: CVPR, 2022, pp. 16000–16009.

[37] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, J. Wang, Context autoencoder for self-supervised representation learning, Int. J. Comput. Vis. 132 (1) (2024) 208–223.

[38] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, J. Jitsev, Reproducible scaling laws for contrastive language-image learning, in: CVPR, 2023, pp. 2818–2829.

[39] H. Pham, Z. Dai, G. Ghiasi, K. Kawaguchi, H. Liu, A.W. Yu, J. Yu, Y.-T. Chen, M.-T. Luong, Y. Wu, et al., Combined scaling for zero-shot transfer learning, Neurocomputing 555 (2023) 126658.

[40] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, Unicoder-vl: a universal encoder for vision and language by cross-modal pre-training, in: Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020, pp. 11336–11344.

[41] S. Doveh, A. Arbelle, S. Harary, E. Schwartz, R. Herzig, R. Giryes, R. Feris, R. Panda, S. Ullman, L. Karlinsky, Teaching structured vision & language concepts to vision & language models, in: CVPR, 2023, pp. 2657–2668.

[42] Z. Yang, G. An, Z. Zheng, S. Cao, F. Wang, EPK-CLIP: external and priori knowledge CLIP for action recognition, Expert Syst. Appl. 252 (2024) 124183.

[43] N. Mündler, J. He, S. Jenko, M. Vechev, Self-contradictory hallucinations of large language models: evaluation, detection and mitigation, in: ICLR, 2024.

[44] E. Pizzi, S.D. Roy, S.N. Ravindra, P. Goyal, M. Douze, A self-supervised descriptor for image copy detection, in: CVPR, 2022, pp. 14532–14542.

[45] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 117–128.

[46] P. Kaur, K. Sikka, W. Wang, S. Belongie, A. Divakaran, Foodx-251: a dataset for fine-grained food classification, (2019). arXiv:1907.06167

[47] L. McInnes, J. Healy, N. Saul, L. Grossberger, UMAP: uniform manifold approximation and projection, J. Open Source Softw. 3 (29) (2018) 861.

[48] N. Van Noord, Prototype-based dataset comparison, in: ICCV, 2023, pp. 1944–1954.

[49] M. Zanella, I. Ben Ayed, Low-rank few-shot adaptation of vision-language models, in: CVPR, 2024, pp. 1593–1603.

[50] X. Wu, X. Fu, Y. Liu, E.-P. Lim, S.C.H. Hoi, Q. Sun, A large-scale benchmark for food image segmentation, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 506–515.

[51] T. Berg, J. Liu, S. Woo Lee, M.L. Alexander, D.W. Jacobs, P.N. Belhumeur, Birdsnap: large-scale fine-grained visual categorization of birds, in: CVPR, 2014, pp. 2011–2018.

[52] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P.H.S. Torr, L. Zhang, Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, in: CVPR, 2021.

[53] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in: ECCV, 2018, pp. 418–434.

[54] Y. Xu, J. Zhang, Q. Zhang, D. Tao, Vitpose: simple vision transformer baselines for human pose estimation, Adv. Neural Inf. Process. Syst. 35 (2022) 38571–38584.

[55] T. Ridnik, E. Ben-Baruch, A. Noy, L. Zelnik-Manor, ImageNet-21K pretraining for the masses, in: Proceedings of the Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track, 2021.