Reproducible Vision-Language Models Meet Concepts Out of Pre-Training

Ziliang Chen[1†], Xin Huang[2,3†], Xiaoxuan Fan[4], Keze Wang[2,1], Yuyu Zhou[4], Quanlong Guan[4], Liang Lin[1,2*]

[1]Research Institute of Multiple Agents and Embodied Intelligence, Peng Cheng Laboratory, [2]Sun Yat-sen University, [3]EPFL, [4]Jinan University

c.ziliang@yahoo.com, huangx353@mail2.sysu.edu.cn, kezewang@gmail.com, {zyy,Gql}@jnu.edu.cn, linliang@ieee.org

Abstract

Contrastive Language-Image Pre-training (CLIP) models as a milestone of modern multimodal intelligence, its generalization mechanism grasped massive research interests in the community. While existing studies limited in the scope of pre-training knowledge, hardly underpinned its generalization to countless open-world concepts absent from the pre-training regime. This paper dives into such Out-of-Pre-training (OOP) generalization problem from a holistic perspective. We propose LAION-Beyond benchmark to isolate the evaluation of OOP concepts from pre-training knowledge, with regards to OpenCLIP and its reproducible variants derived from LAION datasets. Empirical analysis evidences that despite image features of OOP concepts born with significant category margins, their zero-shot transfer significantly fails due to the poor image-text alignment. To this, we elaborate the "name-tuning" methodology with its theoretical merits in terms of OOP generalization, then propose few-shot name learning (FSNL) and zero-shot name learning (ZSNL) algorithms to achieve OOP generalization in a data-efficient manner. LAION-Beyond dataset and codes: http://m-huangx.github.io/laion_beyond/.

1. Introduction

In recent years, pre-training vision models with language supervision by contrastive learning schemes becomes a focal point of interest [12, 23, 25]. The so-called CLIPs and its variants, forging a synergy between images and natural language, are pre-trained by aligning visual and textual information with a vast trove of image-text pairs using coupled encoders. The success of image-text matching is quantified by classifying images into the classes with their names in a vocabulary, which are fed into the text encoder with its prompt template (*e.g.*, "a photo of a {*class*}") to generate their classification weights. Constructing Softmax classifier by the cosine similarity between normalized image features and weights, CLIP exhibits stellar few-shot / zero-shot learning on behave of its out-of-distribution (OOD) generalizability [10, 11, 41] .

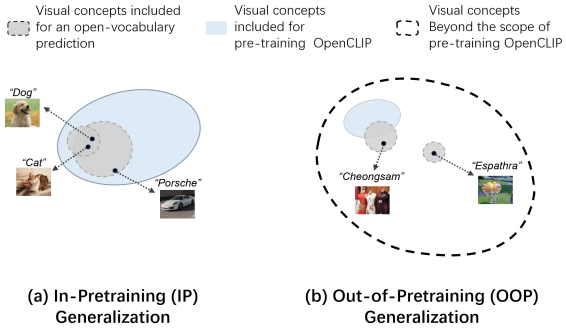


Figure 1. Comparison between IP and OOP generalization. The former evaluate OpenCLIP's generalization with visual concepts seen in pre-training phases, whereas the latter justifies its generalization through the concepts absent during pre-training.

Leading studies investigated the CLIP family from diverse aspects of OOD generalization, while visual concepts employed in their evaluations were almost expected to have been encountered during pre-training (Fig.1.a). In contrast of these In-Pre-training (IP) concepts, Out-of-Pre-training (OOP) concepts never shown previously yet might predominant in the majority of open-world cases, remain questionable (Fig.1.b). This limitation stems from the scarcity of available evaluation data: existing benchmarks mainly consist of visual concepts well-represented in large-scale pre-training datasets such as LAION [26]. This category overlap between evaluation and pre-training distributions potentially masks the CLIP's true OOD generalization evaluation beyond commonly encountered concepts.

To demystify the underlying capability of CLIP in OOP concepts, we propose ***LAION-Beyond*** benchmark with the counterpart classes absent in LAION datasets used to build OpenCLIP, the reproducible variants of CLIP. It derives LAION-Beyond's construction with 324 IP and 674 OOP classes under the rubric of 9 domains, which are tractably crawled from web to prevent the category-set contamination from the vocabulary of LAION. Combining LAION series and their OpenCLIP-derived models helps rigorously identify whether a concept joined contrastive pre-training.

*indicate corresponding author; †indicates the equal contribution.

It facilitates the fair comparison between the IP and OOP concepts given their images recognized by OpenCLIP.

Remarkable insights are found in the image features extracted from OOP concepts by the OpenCLIP encoder, illustrating significant clustering margins across arbitrary IP and OOP classes through visualization and clustering indexes. This phenomenon suggests that visual concepts are potentially categorized regardless of whether they were included in the pre-training dataset. However, in contrast with previous observations in IP classes, the zero-shot learning performance of OpenCLIP exhibits a tremendous deficit in OOP classes. It showcases the cross-modal alignment failure of the text encoder when OOP concepts appear in images.

In the realm of open-vocabulary classification in LAION -Beyond, the alignment failure can be solved via fine-tuning the name embeddings of OOP concepts. We analyze the OOP generalization from a principled view, verifying the merit of fine-tuning OOP name embeddings and the potential risk when we take prompt-tuning or adapter in this problem. Derived from this, we propose few-shot name learning (FSNL) algorithm, then extend it to suit the situations without image-text pairs available in OOP concepts, which leads to zero-shot name learning (ZSNL) algorithm. FSNL fine-tunes name embedding of OOP concepts by shuffling contexts across similar concept pairs; ZSNL combines the name-tuning with bipartite graph matching algorithms to align proper OOP-class image-cluster centers. They are evaluated via LAION-Beyond in the comparison with prevalent prompt-tuning and adapter baselines, which justify our theoretical claims and the superiority of our methodology.

2. Related Work

Vision-Language Models (VLMs). VisualBERT is a pioneer in pre-trained VLMs, adapting BERT for multimodal inputs that shed a light to the follow-up pre-training models and techniques [3, 13, 14, 27, 29]. CLIP [9, 24] and its variants [4, 28, 42] revolutionized multimodal learning by pre-training to align large-scale images and texts in a uniform embedding space. The paradigm shift owes the broad employment of their pre-trained image encoders in most existing visual large language models (vLLMs) [1, 2, 37], which capably fast adapts to new visual concepts without extensively re-training their image-encoding pipelines. It is hardly coordinated with the existing CLIP-based emprical studies without the concerns of clarifying the concepts in or beyond the scope of pre-training.

Generalization mechanism in VLM. Impressed by CLIP's capability, attentions were gradually paid to unveil its generalization mechanism behind. Some work are interested in its robustness [20, 30, 38] and its adaptation to long-tailed distribution bias [34, 49], while others examine train-test similarity [17] and compositionality understanding [40]. While Udandarao et al [31] reveal that CLIP requires exponentially more data for linear performance gains, these studies leave unexplored a critical question: *how does CLIP handle cross-modal alignment for concepts absent from pre-training?* Our work propose LAION-beyond benchmark to formally investigate this OOP generalization problem.

Data-Efficient Fine-tuning of VLM. Remarked by the extraordinary generalization, CLIP-based VLMs are primarily evaluated through open-vocabulary prediction with textual input as task-oriented prompt templates. Many studies aim for data-efficient fine-tuning CLIP to adapt downstream tasks. [8, 43] prefer a adapter layer inserted between CLIP's pre-training pipeline and the output, then fine-tuned to encourage the downstream-task adaptation. More recently, prompt-tuning approaches [15, 46–48] directly seek for optimizing the embedding parts behind task-specific context templates. The methodology is more flexible and broadly adopted in CLIP-based research. Distinct from conventional zero-shot learning settings [6, 7, 18], CLIP have observed large-scale vision-language pairs during pre-training so its zero-shot transfer inference heavily relies on the extensive supervision available in pretraining. *How do the concepts unseen in pre-training influcence CLIP's fine-tuning strategies?* It remains a mystery.

3. Preliminary

As a reproducible variant of CLIP, OpenCLIP [4] consists of an image encoder f and a text encoder g jointly pre-trained with massive image-text pairs $\{\boldsymbol{I}_i, \boldsymbol{T}_i\}_{i=1}^N$ drawn from open-source pre-training sets LAION [26], where normalized features $f(\boldsymbol{I}^{(j)})$, $g(\boldsymbol{T}^{(j)})$ are extracted to train the encoders via closing their consine similarity gap with InfoNCE [21]. Well-trained encoders serve the prompt-based open-vocabulary classification principle:

$$P_V^{(f,g)}(\boldsymbol{y}|\boldsymbol{I}) = \frac{\exp\left(\frac{\text{sim}(f(\boldsymbol{I}), g(\boldsymbol{T}(\boldsymbol{y})))}{\gamma}\right)}{\sum_{\boldsymbol{y}_i \in V} \exp\left(\frac{\text{sim}(f(\boldsymbol{I}), g(\boldsymbol{T}(\boldsymbol{y}_i)))}{\gamma}\right)}, \quad (1)$$

where $\boldsymbol{T}(\boldsymbol{y})$ indicates a text with the class name $\boldsymbol{y}$. It is also well-known as *prompt* if $\boldsymbol{T}(\cdot)$ takes a task-specific template, *e.g.*, $\boldsymbol{T}(\_)$ ="a photo of [_] " for few-shot/zero-shot learning paradigm. V denotes the vocabulary to identify the range of predicted classes.

Motivation. Given classes with respect to a vocabulary V, we may fine-tune a lightweight adaptation layer [8] or the context embedding of prompt [45] to classify their images. In this case, OOD generalization is measured via the performance balance between the classes in V or beyond. But no matter which, the success rises from vision-language alignment achieved during pre-training phase. Instead of constructing classifiers with pre-training words and phrases, the vocabulary V in this work also permits the class names beyond the textual scope of $\{\boldsymbol{T}_i\}_{i=1}^N$.

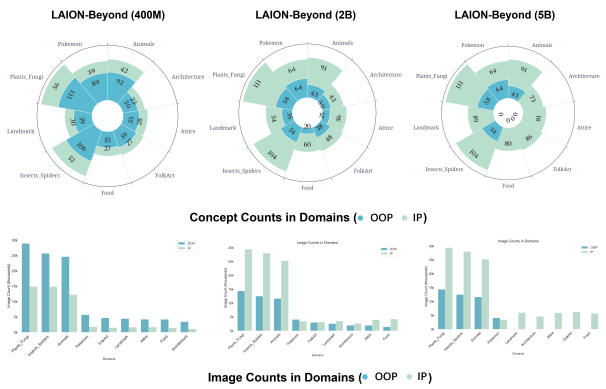


Figure 2. The statistics of OOP and IP concepts and their images in LAION-Beyond (400M),(2B), and (5B).

4. Unveiling OpenCLIP by Out-of-Pretraining (OOP) Concepts

As discussed, our study needs to examine the performances of OpenCLIP when evaluating visual concepts absent in the pre-training dataset $\{\boldsymbol{I}_i, \boldsymbol{T}_i\}_{i=1}^{N}$, whereas to the best of our knowledge, there is no proper identification of OOP concepts in existing benchmarks. Towards this end, we propose a evaluation benchmark LAION-Beyond.

4.1. LAION-Beyond Benchmark

Construction and statistic. LAION-Beyond consists of 106,052 images in 674 OOP concepts and 51,330 images in 324 IP concepts in terms of words and phrases in LAION-400M, distributed across 9 domains, *i.e.*, *Plants_Fungi*, *Insects_Spiders*, *Animals*, *Pokemon*, *FolkArt*, *Landmark*, *Attire*, *Food*, and *Architecture*. Each domain represented as a categorical ancient branches in a hierarchy, simultaneously contains OOP concepts and IP concepts under the rubic of its categorical branch. The IP-class names are found in LAION-400M while the OOP-class names stays beyond the lexical scope of LAION-400M vocabulary. Images with texts associated with the OOP concepts are divided into train, val, test sets to faciliate our OOP generalization experiments (Sec.6), instead, IP concepts solely contain their images to evaluate the generalization of OpenCLIP. Our construction promises both OOP and IP concepts visually perceptible and long-tail distributed, to ensure their fair comparison.

Scaling OOP concepts beyond LAION-2B and 5B. Despite LAION-Beyond drawn from concepts beyond LAION-400M, we further derive subsets to identify OOP concepts out of the scopes of LAION-2B and 5B, respectively. Specifically, we construct the word-and-phrase list of LAION-2B and 5B, as what was done previously, then screen the OOP concepts in LAION-Beyond (400M) to generate the image-text subsets drawn from OOP concepts excluded by LAION-2B and 5B. LAION-Beyond with 400M, 2B, and 5B splits, enable comprehensive evaluation of CLIP models pre-trained on their corresponding LAION datasets. This stratification encourages OOP-concept examination based upon the CLIP-based neural scaling law.

Details of construction and statistic of (400M), (2B), and (5B) versions in LAION-Beyond refer to our Appendix.A.

Table 1. The normalized clustering accuracy across features extracted from OOP-class test images and IP-class test images across 9 domains, respectively.

	Anim	*Arch*	*Atti*	*Folk*	*Food*	*Insect*	*Ladmk*	*Plant*	*Pokem*	Avg
IP classes	40.27	91.04	82.09	78.02	81.72	50.44	93.01	55.71	34.07	68.15
OOP classes	37.27	81.06	68.92	76.60	80.65	48.30	86.39	53.17	35.80	63.13
IP-OOP gap	3.00	10.02	13.83	1.42	1.07	2.14	6.62	2.54	1.73	5.02

(a). Image features of OOP concepts

(b). Image features of OOP and IP concepts

Figure 3. The t-SNE visualization for **(a)**.image features from 20 OOP classes drawn from *Plants_Fungi*; **(b)**.image features from 10 OOP classes and 10 IP classes. Colors indicate the image features extracted from corresponding OOP classes. The gray spots in **(b)** indicate the image features extracted from arbitrary IP classes.

4.2. Generalization of CLIP: IP *v.s.* OOP Concepts

Providing LAION-Beyond with concepts beyond LAION-400M, we evaluate OpenCLIP across the OOP and IP concepts, with the same categorical ancients for a fair comparison. It results in remarkable insights to better understand CLIP-derived systems in the realm of open-world concepts.

Finding 1: clustering margins of OOP concepts. We first investigate the OOP image features through their t-SNE embeddings [32]. The features in Fig.3.a are illustrated with significant clustering gaps across different OOP categories and what's more, these clusters would not be overlapped by category clusters with respect to IP concepts (Fig.3.b) (Full t-SNE-based observations across 9 domains can be found in Appendix.A and corroborate our finding). The report of clustering accuracy across different OOP and IP classes (Table.1) further confirm this finding. In particular, we observe both IP and OOP classes with high normalized clustering accuracy in 7 domains, and the performance gaps in 6 domains less than 3%. The digits coincide with the illustration in Fig.3, which jointly implies that ***image features extracted by the vision encoder of CLIP conceive the***

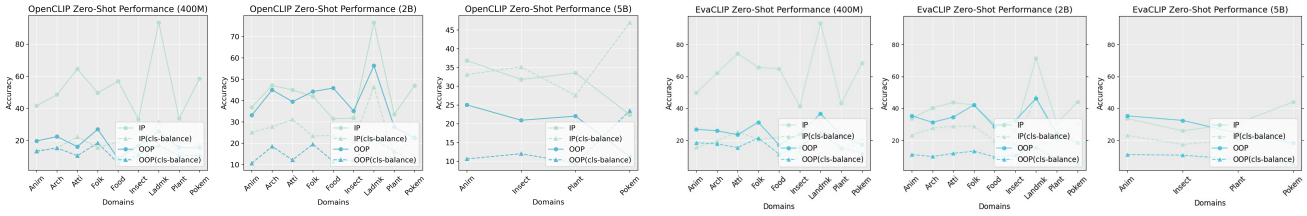


Figure 4. The zero-shot inference performances of OpenCLIP and EvaCLIP on OOP and IP classes in LAION-Beyond (400M), (2B), and (5B), respectively. IP (cls-balance) and OOP (cls-balance) indicate the class-balanced accuracy $ACC_{\text{cls-bal}}$: Suppose that N^c_{IP},N^c_{OOP} denote the number of IP, OOP classes, then $ACC_{\text{cls-bal}}(x) = \frac{N^c_{\text{IP}} ACC(x)}{N^c_{\text{IP}}+N^c_{\text{OOP}}}$ if $x \in D_{\text{IP}}$;$ACC_{\text{cls-bal}}(x) = \frac{N^c_{\text{OOP}} ACC(x)}{N^c_{\text{IP}}+N^c_{\text{OOP}}}$ if $x \in D_{\text{OOP}}$. The class-balanced accuracy debiases the performances with different numbers between IP and OOP classes for a fair evaluation.

discriminability across arbitrary classes, even if they have never shown the names in pre-training.

Finding 2: image-text alignment failure. Though OpenCLIP born with the promising discriminability in OOP concepts with its image encoder, the text encoder fails to achieve the cross-modal alignment. To justify this, we compare zero-shot learning performances on IP and OOP concepts with neural scaling law. As illustrated in Fig.4, compared with the accuracy in IP classes, OOP-class zero-shot learning results drop with huge gaps in LAION-Beyond (400M), while the gaps are narrower while considering the class-balance accuracy between the images in OOP and IP classes, which demonstrates the class imbalance also contributes the gap. In the cases in LAION-Beyond (2B), IP and OOP results are close, however, balancing the classes leads to a larger gap. To this, the image-text misalignment in LAION-Beyond (2B) become more serious. In LAION-Beyond (5B), no matter class balance or not, OOP-class zero-shot learning significantly outperforms IP-class zero-shot learning. Summarize them and we found, despite OpenCLIP pre-trained with more image-text pairs (LAION-400M *v.s.* LAION-5B), as long as the concepts absent from pre-training, ***there is no promise to solve the image-text misalignment by neural scaling law.***

Opposed with image features, the disaster performances across OOP domains in OpenCLIP may owe to the text encoder fails to generate OOP classification weights. It stems from the absence of alignment between prompts with OOP concepts and their corresponding visual features, because of token embeddings of OOP concepts not initialized with any image-text alignment in pre-training. Since OOP concepts typically refer to rare classes, ***optimizing their embeddings in an efficient manner is supposed to be the key that achieves OOP image-text alignment.***

5. Methodology

In this section, we first elaborate fine-tuning the name embeddings of OOP concepts along with its theoretical merits, then propose the derived algorithms in the open-vocabulary few-shot and zero-shot learning setups, respectively.

5.1. Fine-Tuning Names of OOP Concepts

Distinct from pretraining and prompt-tuning, name-tuning only optimizes the token embedding with regards to words and phrases of OOP concepts, while keeps the rest words' embeddings as usual. Suppose that we have OOP image-text pairs $\left\{\left\langle \hat{\boldsymbol{I}}(\boldsymbol{y}), \hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))\right\rangle\right\}$, where $\boldsymbol{y}$ indicates a word or a phrase with respect to a specific concept drawn from an OOP list $\boldsymbol{Y}_{\text{OOP}}$; $\left\langle \hat{\boldsymbol{I}}(\boldsymbol{y}), \hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))\right\rangle$ presents as a image-text pair with the concept $\boldsymbol{y}$ and its name tokenized into the embedding $e(\boldsymbol{y})$, is included by caption $\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))$. To this, tuning embeddings of OOP concepts is formulated

$$\min_{e(\boldsymbol{y})\in e(\boldsymbol{Y}_{\text{OOP}})} \hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}} = \mathcal{L}_{\text{InfoNCE}}\Big(\left\langle \hat{\boldsymbol{I}}(\boldsymbol{y}), \hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))\right\rangle\Big), \tag{2}$$

where $\mathcal{L}_{\text{InfoNCE}}$ denotes the optimization derived from InfoNCE whereas the image and text encoders are frozen to tune OOP word or phrase embeddings ahead of the text encoder input. $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$ denotes the empirical risk minimization (ERM) with respect to the list of OOP concepts.

OOP Generalization analysis. To show the necessity of name-tuning, we consider the population risk $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ that denotes the ideal pre-training with f, g and word embeddings in lists of IP and OOP concepts, and the practical pre-training ERM $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ only with IP concepts [1]. Obviously, minimizing $\mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ is perfect yet impossible since we are unable to go through the countless samples and the concepts newly up-coming, but combing $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ and $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$ leads to the generalization bound below

Proposition 1. *(Informal)Suppose that $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$ denotes the ERM of fine-tuning the name embeddings of OOP concepts on top of frozen $\hat{f}$,$\hat{g}$, which are pre-trained along with the*

[1]The definition of f,g in our analysis are apart from the word embedding of OOP concepts that follow the tokenization. This reconfiguration helps to isolate the OOP name-tuning from existing studies.

name embeddings of IP concepts. Suppose D_{IP}, D_{OOP} as the distributions with IP and OOP concepts, respectively; and N_{IP} and N_{OOP} denote the number of samples drawn from D_{IP}, D_{OOP} for pre-training and fine-tuning, respectively. $\forall\ \epsilon{>}0$, it holds the probability $1\text{-}\epsilon$ with

$$\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \leq \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}} + \hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}} + \frac{1}{2}\gamma(D_{\text{IP}}, D_{\text{OOP}}) + d_p(\hat{f}, f^*) + d_p(\hat{g}, g^*) + \mathfrak{R}_{D_{\text{IP}}}(\mathcal{F},\mathcal{G},\mathcal{E}_{\text{IP}}) + \mathfrak{R}_{D_{\text{OOP}}}(\mathcal{E}_{\text{OOP}}) + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{IP}}}} + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{OOP}}}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}}\left(\frac{1}{N_{\text{IP}}} + \frac{1}{N_{\text{OOP}}}\right), \tag{3}$$

where $\gamma(D_{\text{IP}}, D_{\text{OOP}})$ indicates the distribution gap, $d_p(\cdot,\cdot)$ indicates the approximation error via a p-norm (entry-wise) to measure the difference between functions, $\mathfrak{R}_{D_{\text{IP}}}$ and $\mathfrak{R}_{D_{\text{OOP}}}$ denote the Rademacher complexity for pre-training and fine-tuning, with $\mathcal{F}$,$\mathcal{G}$,$\mathcal{E}_{\text{IP}}$ and $\mathcal{E}_{\text{OOP}}$ denote the functional spaces of f, g, $e(\boldsymbol{Y}_{\text{IP}})$, $e(\boldsymbol{Y}_{\text{OOP}})$.

The insight of Eq.3 is that despite $\boldsymbol{Y}_{\text{OOP}}$ has never seen before, as long as some semantic connected between $\boldsymbol{Y}_{\text{IP}}$ and $\boldsymbol{Y}_{\text{OOP}}$ hold in $\gamma(D_{\text{IP}}, D_{\text{OOP}})$, tuning the name embeddings of OOP concepts after pre-training with $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ can control the ideal population risk $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$.

But why not prompt-tuning or adapter approaches? The following result unveils that the arbitrary initialization of OOP embeddings may lead to the risk of non-identification between IP and OOP concepts, regardless of what prompts or adapters used in open-vocabulary prediction in Eq.1:

Proposition 2. *(Informal) $\forall \boldsymbol{y}_1 \in \boldsymbol{Y}_{\text{IP}}$ and $\forall \boldsymbol{y}_2 \in \boldsymbol{Y}_{\text{OOP}}$, if the embeddings $e(\boldsymbol{Y}_{\text{OOP}})$ can be arbitrarily initialized, then given $\forall\epsilon > 0$ and an image $\boldsymbol{I}$ with label $\boldsymbol{y}_2$, for any prompt-tuning and adapter layer of CLIP behind f and g, it holds $d_p\big(\log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}), \log P_V^{(f,g)}(\boldsymbol{y}_2|\boldsymbol{I})\big) \leq \epsilon$.*

The theory is formalized in Appendix.B along with the further empirical study for better understanding. The aforementioned analysis motivates learning OOP name embedding in the data-efficient manner.

5.2. Few-Shot Name Learning

In the few-shot learning setup, we have the access of image-text pairs in OOP concepts, whereas the amount is very limited. Our first trick to address the training-data eager is to initialize OOP-class name embedding using language models. Our implementation inherits the tokenized input ahead of BERT [6] to initialize the OOP word embeddings. Given OOP phrases composed of old words, their embeddings are independently learnt by deploying the new tokens without the disturbance to IP word embeddings.

Augmentation by contexts of similar concepts. Given each OOP image-text pair $\left\langle \hat{\boldsymbol{I}}(\boldsymbol{y}), \hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y})) \right\rangle$, we further augment few-shot image-text matching by simply shuffling context $\hat{\boldsymbol{T}}$ with arbitrary contexts presented in the other pairs with the similar OOP concepts (*e.g.*, under the same categorical ancient). For instance, given a caption $\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))$ = "A women in a *cheongsam*", the augmentation may switch the context "A women in a [_]" by "People are dressed in [_]" rather than "This animal is [_]", etc. It derives to the formulation:

$$\min_{e(\boldsymbol{y}),\forall y\sim\boldsymbol{Y}_{\text{OOP}}} \sum_{\hat{\boldsymbol{T}}_j\sim P_{\text{CS}}} \mathcal{L}_{\text{InfoNCE}}\Big(\sum_{\hat{\boldsymbol{T}}_j\sim P_{\text{CS}}} \langle \hat{\boldsymbol{I}}(\boldsymbol{y}), \hat{\boldsymbol{T}}_j(\boldsymbol{e}(\boldsymbol{y})) \rangle \Big), \tag{4}$$

where P_{CS} denote the prompts generated with the shuffled contexts, then Eq.4 and Eq.2 are jointly optimized to fine-tune the OOP-name embedding. Similar with compositional prompts [19, 40] that we are inspired from, the augmentation may lead to noisy contexts with inevitable incorrect fine-grained information. While under the same umbrella of categorical ancient, our strategy promises the semantic is convincing in open-vocabulary prediction tasks.

5.3. Zero-Shot Name Learning

Few-shot learning permits an access of image-text pair for OOP concepts. To take a step further, we prescribe that the image set $\left\{\hat{\boldsymbol{I}}(\boldsymbol{y})\right\}$ does not have the text set $\left\{\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}))\right\}$ in OOP concepts to align with, which literally refers to a zero-shot learning setup since images in $\left\{\hat{\boldsymbol{I}}(\boldsymbol{y})\right\}$ were never labeled by any concept names. To classify them into $\boldsymbol{Y}_{\text{OOP}}$, we propose a bipartite-graph matching strategy to estimate the names of image in $\left\{\hat{\boldsymbol{I}}(\boldsymbol{y})\right\}$ with regards to $\boldsymbol{Y}_{\text{OOP}}$, then learning to classify them by name-tuning the label.

Cluster initialization by novel-class discovery. As discussed previously, despite images in OOP class not aligned with their names, those in the same OOP class tend to converge into a cluster. However, unsupervised clustering always suffer from the ambiguity in cluster granularity due to no explicit category information provided. To this, we follow the spirit of novel class discovery (NCD) [5], which guides the clustering of OOP-class images by the supervision from the images in IP classes, thus, obtained by low cost. In our methodology, we limit the training scope of the NCD model to the classification head while its backbone is frozen by the image encoder $f(\cdot)$. This strategy inject the OpenCLIP's pre-training knowledge to ensure the performance transfer from IP classes to OOP classes.

Image-text bipartite-graph matching. With category-aware image clusters obtained in OOP classes, we attempt to find the optimal bipartite-graph matching across the cluster centers and prompt embedding with the names in OOP classes. Specifically, we obtained the normalized average of each cluster to represent the OOP-class centers, then compute their cosine similarities with the prompt embedding representing each concept in OOP classes. Given this, we have a bipartite graph between the image-cluster centers and

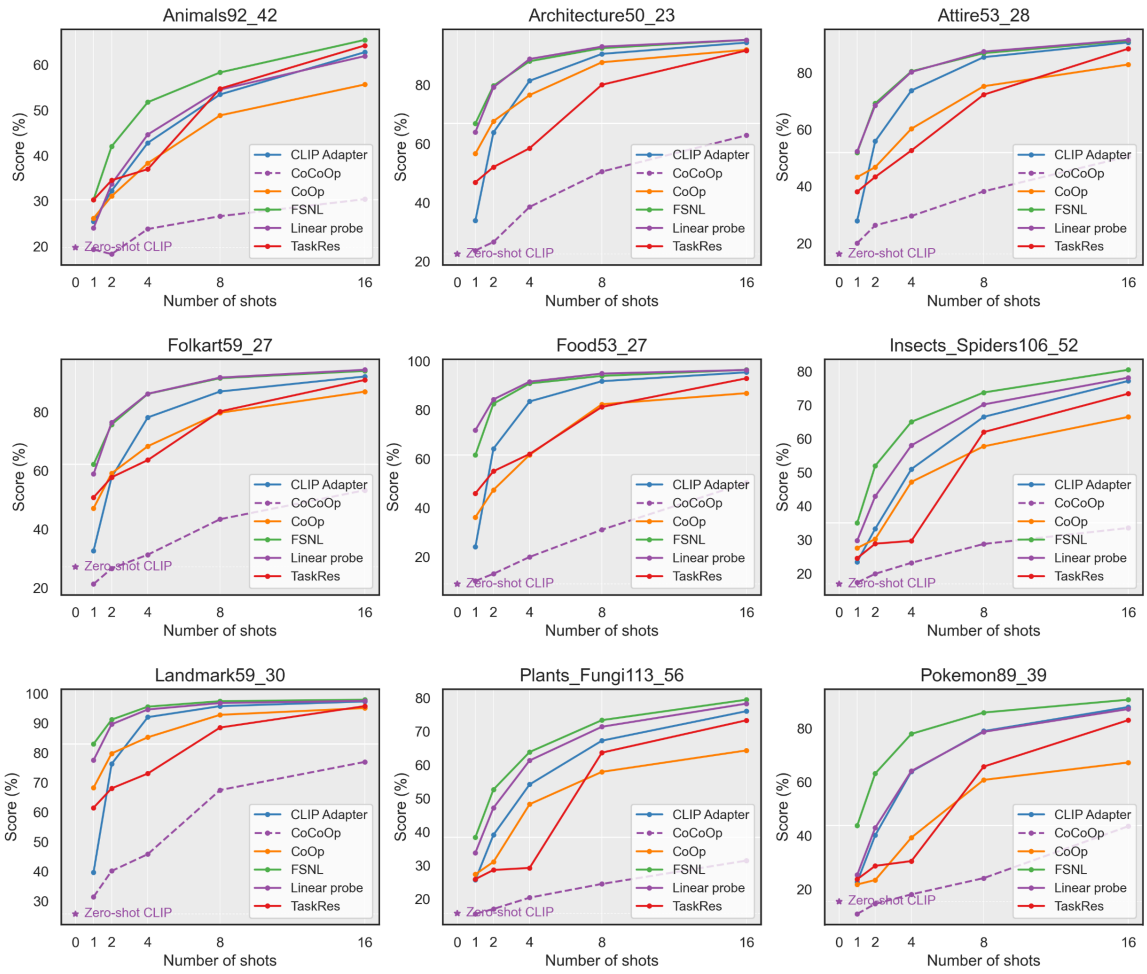


Figure 5. OOP few-shot learning performances (1,2,4,8,16 shots) of different baselines in the test sets of *Animals*, *Landmark*, and *Pokemon* across 9 domains in LAION-Beyond (400M).

the OOP-class prompt embeddings, then we assign pseudo OOP names to each cluster by executing the Hungarian algorithm to the bipartite graph. Fine-tune name embeddings by Eq.(2) with high-confidence samples drawn from OOP-class clusters, leading to our zero-shot learning algorithm with regards to OOP concepts for OpenCLIP.

The implementation details of FSNL and ZSNL, along with their codes, have been elaborated in SM.

6. Experiment

In this section, we provide extensive empirical evaluation for FSNL, ZSNL in OOP concepts, to justify their abilities to alleviate the image-text alignment failure. Codes and implementation details of our algorithms are found in SM.

6.1. Few-shot Learning for OOP Classes

Benchmark. We employed LAION-Beyond (400M) as our evaluation benchmark for few-shot classifying OOP concepts, since it contains the most concepts in our three splits of LAION-Beyond. In order to verify the open-vocabulary prediction on data with OOP concepts, we only employed OOP image-text pairs for training and validation, then incorporated the images drawn from OOP test set and the images drawn from IP classes to serve our evaluation.

Baselines To answer the first question in Sec.5.2, we compare FSNL with existing adapter-based and prompt-based tuning baselines, *i.e.*, *CoOp* [45], *CoCoOp* [44], *Clip-Adapter (Clip-Ada)* [8] , *TaskRes* [39], and *Linear Probe* [24]. Note that their original implementations solely promote fine-tuning with vanilla CLIP's backbone, which may violate the premise of OOP classes in this paper. Hence we reproduced their implementations by importing OpenCLIP as their backbone models, then further verified the results with their digits reported in their original papers. It promotes the reproducibility in LAION-Beyond

OOP-class few-shot learning setup. This task refers to the primary goal in this experiment. It requires the evaluated baselines to fine-tune their OpenCLIP backbones, in partic-

Table 2. OOP-to-IP open-vocabulary prediction. **OOP, IP**, and **H-mean** represent the accuracies of the models trained for OOP few-shot learning (4-shot), their accuracies in IP image dataset, and their compound Harmonic mean score [35], respectively (best viewed in color).

Baslines	Metric	Domains										Extra
		Animals	*Architecture*	*Attire*	*FolkArt*	*Food*	*Insects_Spider*	*Landmark*	*Plants_Fungi*	*Pokemon*	**Avg**	**subnet**
OpenCLIP	**OOP**	19.70	22.49	16.18	27.07	8.88	16.86	25.65	15.71	15.47	18.67	
	IP	**41.66**	**48.60**	**64.57**	**49.67**	**56.93**	**33.28**	**93.41**	**33.71**	**58.58**	**53.38**	No
	H-mean	26.75	30.75	25.88	35.04	15.36	22.38	40.25	21.43	24.48	26.92	
CoOp	**OOP**	38.31	76.54	60.28	68.35	61.67	47.18	85.24	48.25	39.24	58.34	
	IP	26.56	46.43	43.29	42.04	32.48	17.69	86.54	16.67	32.45	38.24	No
	H-mean	31.37	57.8	50.39	52.06	42.55	25.73	85.89	24.78	35.52	45.12	
CoCoOp	**OOP**	23.82	38.53	29.58	31.29	19.94	23.16	45.81	20.35	18.10	27.84	
	IP	36.82	41.30	39.40	33.19	36.15	30.76	85.27	28.29	37.22	40.93	Yes
	H-mean	28.93	39.87	33.79	32.21	25.7	26.42	59.6	23.67	24.36	32.73	
CLIP-Adapter	**OOP**	42.80	81.40	73.72	78.20	83.48	51.00	92.03	54.17	63.82	68.69	
	IP	35.78	46.60	57.43	44.01	52.31	23.86	89.64	22.68	48.30	46.73	Yes
	H-mean	38.98	59.27	64.56	56.32	64.32	32.51	90.82	31.97	54.99	54.86	
Learning-to-Name	**OOP**	26.18	41.64	43.24	54.11	44.18	31.21	50.13	24.76	37.99	39.27	
	IP	32.86	40.16	51.24	39.89	42.17	28.90	88.21	26.05	45.11	43.84	Yes
	H-mean	29.14	40.89	46.90	45.92	43.15	30.01	63.93	25.39	41.25	40.73	
FSNL(ours)	**OOP**	**51.77**	**88.03**	**80.47**	**86.23**	**90.85**	**65.03**	**95.57**	**63.85**	**77.88**	**77.74**	
	IP	**41.66**	**48.60**	**64.57**	**49.67**	**56.93**	**33.28**	**93.41**	**33.71**	**58.58**	**53.38**	No
	H-mean	**46.17**	**62.63**	**71.65**	**63.03**	**70.0**	**44.03**	**94.48**	**44.12**	**68.87**	**62.55**	

ular, using the random shots drawn from the training set that contains OOP image-text pairs. The number of shots is ranged from {1,2,4,8,16}, which follows the efficient-tuning setup broadly employed to evaluate IP generalization [45]. Then well-trained models would be evaluated to classify the OOP test-set images into their OOP classes.

Table 3. The open-world transfer results (ACC across all OOP-class test images and IP-class images) across 9 domains. *Linear Probe* and *TaskRes* have been excluded due to their failure to transfer the training vocabulary. Δ indicates the absolute ratio that FSNL exceeds the second best.

	Anim	*Arch*	*Atti*	*Folk*	*Food*	*Insect*	*Ladmk*	*Plant*	*Pokem*	Avg
OpenCLIP	19.40	25.21	25.23	27.75	18.86	16.43	46.10	16.47	26.89	24.7
CoOp	23.99	57.93	43.85	52.33	32.63	26.37	80.07	27.21	22.78	40.8
CoCoOp	18.86	29.40	23.78	22.54	17.09	17.78	50.14	16.93	18.38	23.88
CLIP-Adap	29.51	64.60	58.92	59.98	64.14	29.23	85.05	32.89	54.47	53.2
L2Name	21.78	33.02	25.26	27.09	25.14	21.13	67.05	22.89	26.47	29.98
FSNL(ours)	**36.35**	**71.00**	**63.75**	**69.27**	**68.09**	**39.70**	**92.54**	**40.53**	**65.29**	**60.72**
Δ	**+6.84**	**+6.40**	**+5.42**	**+9.29**	**+3.95**	**+10.47**	**+7.49**	**+7.64**	**+10.82**	**+7.52**

Results. The main results across baselines with 3 random seeds for their fine-tuning, have been comprehensively illustrated in Appendix.C (some scenarios shown in Fig.5). Their findings yield crucial insights into the OOP generalization issue. Firstly, FSNL excels as the state of the arts across 8 domains except *Food*, achieving the optimal image-text alignment with the upperbound curve ranged from 1 to 16 shots. Second, perhaps surprising, the basic linear probe without the text-encoder assistance, rivals FSNL's performance in 4 domains. Its superiority verifies our claim to the image-feature distribution in Sec.4.3. Finally, despite surpassing OpenCLIP's zero-shot results, neither prompt-tuning nor structural adaptation are even close to linear probes'. This suggests that these CLIP-adaptive algorithms basicaly fail to align their textual modalities with image features. It justifies the wisdom of name-tuning.

While the linear probe and TaskRes improve OpenCLIP, their class prediction are limited to a fixed vocabulary. Such design renders them unavailable in the open world.

OOP *v.s.* IP few-shot learning. Open-vocabulary prediction is defined to switch their word and phrase list to facilitate the categorization to images with class name beyond the existing vocabulary. In this regards, it is significant to justify whether FSNL maintains the open-vocabulary learning behind OpenCLIP. We consider the few-shot learning trade-off between OOP and IP classes, the evaluation setup inspired from existing base-to-new setup for prompt-tuning approaches [44]. Given each domain, we simultaneously evaluated a model's few-shot learning in the domain-specific OOP test set and its IP-class prediction in the domain-specific IP counterpart dataset, then take their Harmonic means to judge the model's performance of balancing OOP and IP class predictions. All models are fine-tuned with 4-shot samples repeated by 3 random seeds.

Table 4. Zero-shot learning ACC (%) in OOP classes drawn from domains in LAION-Beyond (400M) and (5B). Δ indicates the absolute ratio that ZSNL exceeds the second best.

	Splits	*Anim*	*Arch*	*Att*	*Folk*	*Food*	*Insect*	*Ladmk*	*Plant*	*Pokem*
OpenCLIP		19.7	22.5	16.2	27.1	8.9	16.9	25.7	15.7	15.5
TransCLIP	(400M)	21.8	25.2	19.1	**29.3**	10.1	17.3	29.5	16.8	19.6
ZSNL(ours)		**25.7**	**39.9**	**34.9**	27.6	**29.4**	**24.2**	**43.9**	**26.4**	**62.7**
Δ		**+3.9**	**+14.7**	**+15.8**	**-1.7**	**+19.3**	**+6.9**	**+14.4**	**+9.6**	**+43.1**
OpenCLIP		25.9	-	-	-	-	32.2	-	34.3	21.7
TransCLIP	(5B)	31.1	-	-	-	-	35.1	-	35.1	28.1
ZSNL(ours)		45.6	-	-	-	-	59.8	-	70.7	72.5
Δ		**+14.5**	-	-	-		**+24.7**		**+35.6**	**+42.4**

Results. As the task requries models to change their vocabulary to adapt IP classes, linear probe and TaskRes are unavailable. To this, we compare our FSNL algorithm with OpenCLIP, CoOp, CoCoOp, CLIP-adapter, and *Learning-to-Name* (NTL) [22], another name-learning baseline that depends on an extra subnet to model the diverse semantic behind fixed name embeddings. As observed in Tab.2, we found that despite OpenCLIP's failure in OOP-class prediction, it remains the state-of-the-art in IP classes compared with prompt-tuning and CLIP-adapters. It implies CLIP's generalizability underestimated in many existing work. Besides, NTL fails in OOP classes because its extra subnet depends on the original OOP embedding, which is unreliable in cross-modal alignment. Thanks to tuning the embedding instead of the subnet, FSNL optimizes OOP-class embeddings without altering their IP embeddings in OpenCLIP. The merit makes FSNL to reap the best of both worlds.

Open-world vocabulary transfer. In the open world, the boundary between OOP and IP classes is rather obscure in the sense of general AI. In term of this concern, we propose the task *open-world vocabulary transfer* to judge whether the OOP few-shot learning models could classify arbitrary images drawn from OOP and IP classes. It is distinct from OOP-to-IP since the test vocabulary should simultaneously include OOP and IP classes for each domain. So we only need to evaluate in the average across images.

Results. *TaskRes* and *Linear Probe* are not avaiable in this task either therefore we report the performances of the rest in Tab.3. FSNL still significantly outperform the other baselines with the least $+7.52$ leap to the second-best. We even found that CoCoOp underperform OpenCLIP.

6.2. Zero-shot Learning for OOP Classes

As previously discussed about Fig.4.b, zero-shot learning OOP classes is significantly challenging due to no available OOP concepts with their names to align image features during fine-tuning. As a alternative, we allow the usage of image-text training pairs in IP concepts, and the number of classes can be observed. It aims to justify the motivation behind ZSNL algorithm, *i.e.*, whether the IP-concept knowledge helps zero-shot learning OOP concepts.

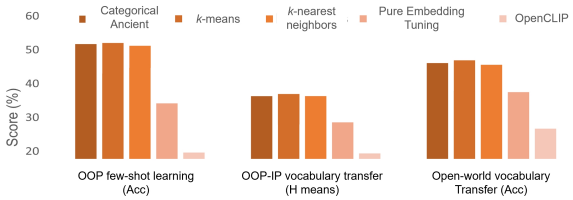


Figure 6. The FSNL ablation in *Animals* (viewed in colors). .

Benchmark. LAION-Beyond (400M),(5B) are employed as our training-evaluation sets, which contains 16-shot samples for training in each OOP class yet their labels and names in OOP concepts are masked, and the rest images in OOP classes are evaluated. It leads to 13 train-test domain splits to thoroughly justify the zero-shot results. We compare our algorithm with OpenCLIP and TransCLIP [16], a strong baseline derived from CLIP's IP ability.

Results. OOP-class zero-shot learning results across 13 train-test splits are illustrated in Table.4. We observe that despite TransCLIP with positive transfer influence to OpenCLIP, the trivial zero-shot benefit in OOP concepts implies that it is not able to address the image-text misalignment. It is probably due to its methodology regardless of unreliable name embeddings in OOP concepts. In contrast, our ZSNL algorithm encourages the name-embedding tuning strategy based on image-text bipartite graph, whose matching also incorporates IP knowledge via NCD to increase the reliability of clustering centers. Across 13 domains, ZSNL significantly reaps the improvement to OpenCLIP with huge gaps in OOP-concept zero-shot learning. Some even outperform the 1-shot learning results by our FSNL algorithm.

6.3. Ablation

We ablate FSNL by different concept similarity strategies (categorical ancient branching, k-means, and nearest neighbors, their details in Appendix.C) for context augmentation, along with the pure embedding fine-tuning in Eq.(2) and the original OpenCLIP. Fig.6 demonstrates that FSNL performs robustly across concept similarity strategies, and purely learning OOP concept embeddings indeed improves the text alignment with OOP image features, whereas the benefit is relatively marginal due to the shortage of training data augmentation. Context augmentation remedies the issue to yield the state of the art in OOP-class generalization.

7. Conclusion

This research propose LAION-Beyond, not only illuminating the abilities and limitations of vision-language models in OOP concepts but also enlightens few-shot and zero-shot learning strategies to OOP generalization, contributing to the advancement of more adaptable multimodal systems.

Reproducible Vision-Language Models Meet Concepts Out of Pre-Training

Supplementary Material

Appendix.A: LAION-Beyond

Construction Details

Building LAION-Beyond consists of three phases:

Collecting words and phrases with respect to OOP visual concepts. We started with 9 categorical branches derived from the LAION-400M dataset to define the basic domains in LAION-Beyond. The supercategories are manually screened from the candidates generated by GPT-4 with prompt engineering to suggest visually realized concepts that contains sufficient long-tail classes. With 9 concept branches obtained, the term checking system derive the OOP classes for each branch via, **first**, feed IP classes in this branch into GPT4 to suggest the other M classes in this branch; **second**, ensure the suggested classes is OOP by checking the terms out of the LAION vocabulary list, then the selected terms join back into the first phase to repeat the loop until sufficient OOP classes obtained. After checking the full word-and-phrase list of LAION-400M by GPT-4 API, we suggested novel words and phrases visually realizable under the same categorical branches whereas not included in LAION-400M's vocabulary by our term-checking prompt system via the GPT-4 core. It is noteworthy that since the text encoder converts all text to lowercase for processing, we performed our term checking in lowercase to ensure all variants of each term not presented in LAION-400M.

In terms of domains *Plants_Fungi*, *Insects_Spiders*, and *Animals*, we found that they are mostly included in iNaturalist. Given this, we directly analyzed all class names in iNaturalist by our GPT-4 term-checking system to suggest the novel visual concepts unseen in LAION-400M. Note that, data from iNaturalist usually have both a "name" (scientific name) and a "common name." We conducted checks for both names to generate the OOP concept list, ultimately using the "common name" as the species name in LAION-Beyond. To generate the terms belonging to *FolkArt*, *Landmark*, *Attire*, *Food*, and *Architecture*, we also resorted to the GPT-4 API to suggest OOP terms in the same manner. We realize that, even if LAION-400M is predominantly an English-based image-text pair dataset, it remains a significant number of non-English terms with respect to national or cultural visual concepts. We follow the LAION-400M's convention to suggest some national or cultural terms in *FolkArt*, *Landmark*, *Attire*, *Food*, and *Architecture*, which are presented in their original language. Besides, we further verify their names both in their original (non-English) and English expressions [2], in order to promise the basic premise behind LAION-Beyond. As for the final domain *Pokemon*, we directly downloaded the official pokemon list from NO.0001 - NO1010 with their offical names in English, then select those never shown in the LAION-400M.

Collecting images with OOP and IP concepts. Images belonging to OOP and IP concepts from the *Plants_Fungi*, *Insects_Spiders*, and *Animals* were sourced from iNaturalist. Their IP concepts were selected through the overlap concepts between the LAION-400M vocabulary and the class list of iNaturalist, then seperated into *Plants_Fungi*, *Insects_Spiders*, and *Animals* according to their biological specification. Then we directly employed their images in iNaturalist to construct the image sets of *Plants_Fungi*, *Insects_Spiders*, and *Animals*. As for *FolkArt*, *Landmark*, *Attire*, *Food*, *Architecture*, and *Pokemon*, the names of their IP classes were selected in LAION-400M in the long-tail statistic, and we utilized the Bing Image API and Google Image browser to search the images with the keywords as their OOP and IP concepts. Human verification promises that each web-crawled image contains the visual information corresponding to its OOP or IP visual concept.

Captioning the images with OOP concepts. We employed Alibaba's vLLM API (github.com/QwenLM/Qwen-VL) to generate the caption for each OOP image. Human labor further verified all OOP-concept names shown in the corresponding caption and the visual side information correctly descirbed with the OOP concept.

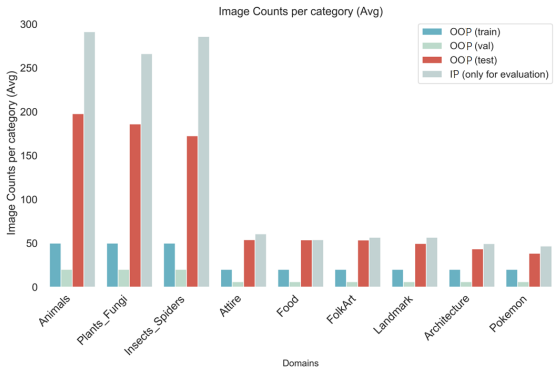


Figure 7. The statistic of LAION-Beyond images per category.

We illustrate the average number of train, val, test splits of OOP images, and the average number of IP images (only

[2] We prompt GPT-4 to find their domestic English presentation. If the presentation only refers to a general description, *e.g.*, a hat in Arabian style, our check will abandon the general English description to promise the term specifying the visual concept without ambiguity. This process is verfied by human labor.

for evaluation) per category in Fig.7. Since few-shot learning only takes training samples less than 16, we control the size of training pools from 26 to 50 across domains to generate the random seeds for training. It results in huge OOP test sets and IP generalization evaluation sets across domains, therefore, facilitate more trustful results to evaluate OOP and IP generalization performances of different baselines.

Discussion of Domain-specific Modal Shift

The 9 domains in LAION-Beyond present significantly different types of modal shifts. More specifically, IP and OOP concepts in *Pokemon* are typically separated by generations and designers, resulting their visual appearance with stylistic variations. Additionally, for each specific pokemon, it is shown with diverse presentations *e.g.*, game screenshots, trading cards, figurines, and illustrations. For the domains *Plants_Fungi*, *Insects_Spiders*, and *Animals*, most images refer to photographs collected from the wild. Despite featured in a relatively uniform style, most of the concepts are close in their breeds, which can be only separated with their fine-grained visual charateristic. The images from the domains of *FolkArt*, *Landmark*, *Attire*, *Food*, and *Architecture* were web-crawled in different languages, where the modal shift is naturally mixed with language bias.

OOP Word-and-Phrase List with Data Instances

We presented the specific list of Out-of-Pre-training (OOP) and In-Pre-training (IP) visual concepts for each domain in LAION-Beyond (Fig.28, 29, 30, 31, 32, 33, 34, 35, 36). We also illustrated some instances of OOP image-caption pair and IP image for each domain (Fig.10, 11, 12, 13, 14, 15, 16, 17, 18). The comprehensive benchmark data, along with the associated code, are going to be released with open access to facilitate further research in the community.

Appendix.B: OOP Generalization Analysis

We provide the theoretical analysis derived from Proposition.1,2.

7.1. OOP Generalization of Name-Tuning

To prove Proposition.1, we need to elaborate the definitions of $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}$, $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\rm IP}}$, and $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\rm OOP}}$ through the InfoNCE loss $\mathcal{L}_{\rm InfoNCE}\Big(\big\{\langle\hat{\boldsymbol{I}}(\boldsymbol{y}_i),\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}_i))\rangle\big\}_{i=1}^{M}\Big)$ using an image query as the anchor[3]. Specifically, given a visual concept $\boldsymbol{y}_i$ that belongs to either IP ($\boldsymbol{y}_i\in\boldsymbol{Y}_{\rm IP}$) or OOP concepts ($\boldsymbol{y}_i\in\boldsymbol{Y}_{\rm OOP}$), then for each image $\hat{\boldsymbol{I}}(\boldsymbol{y}_i)$ with respect to $\boldsymbol{y}_i$, the InfoNCE loss holds the image-text matching between $\hat{\boldsymbol{I}}(\boldsymbol{y}_i)$ and a text $\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}_i))$ incorporating the concept $\boldsymbol{y}_i$:

$$\begin{aligned}&\mathcal{L}_{\rm InfoNCE}\Big(\big\{\langle\hat{\boldsymbol{I}}(\boldsymbol{y}_i),\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}_i))\rangle\big\}_{i=1}^{M}\Big)\\ =&\frac{1}{M}\sum_{i=1}^{M}-\log\frac{\exp\big(\mathrm{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)),g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_i))))/\gamma\big)}{\sum_{j=1}^{M}\exp\big(\mathrm{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)),g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_j))))/\gamma\big)},\end{aligned}\tag{5}$$

where M denotes the number of image-text pairs per training batch. Eq.5 is regarded as self-supervised learning objective due to noisy image-text training pairs while CLIP was endowed with remarkable ability to classify images via $P_V^{(f,g)}(\boldsymbol{y}|\boldsymbol{I})$. The phenomenon could be explained if the majority of training batches holds the concept disambiguity assumption as follows

Assumption 3. (**Batch-level Disambiguity across Concepts**) A training batch consists of M image-text pairs and $\forall i\in[M]$, for the i-th pair $\langle\hat{\boldsymbol{I}}(\boldsymbol{y}_i),\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}_i))\rangle$, $\boldsymbol{y}_i$ denotes its modality-shared concept. Then $\forall i,j\in[M]$, $\boldsymbol{y}_i$,$\boldsymbol{y}_j\in\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}$ and $\{\boldsymbol{y}_i\}\cap\{\boldsymbol{y}_j\}=\emptyset$.

The assumption promises the concept across the image and text can be identified in a training batch with M samples. To this, an image $\hat{\boldsymbol{I}}(\boldsymbol{y}_i)$ with its class represented by the concept embedding $\boldsymbol{e}(\boldsymbol{y}_i)$, in terms of its extracted feature, is supposed to approach the texts that contains $\boldsymbol{y}_i$ while go far from the others. In particular, if the contexts across $\{\hat{\boldsymbol{T}}(\boldsymbol{e}(\boldsymbol{y}_i))\}_{i=1}^{M}$ are consistent, Eq.5 typically refers to the softmax loss function with class-specific prompts. Given this, Assumption.3 bridges self-supervised pre-training and population / empirical risk to encourage OOP generalization analysis. Specifically, we consider the population risk $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}$ with respect to the encoders f^*,g^* pre-trained with the empirical risk minimization (ERM) $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}$, *i.e.*,

$$\begin{aligned}&f^*,g^*,\boldsymbol{e}^*=\arg\min_{f,g,\boldsymbol{e}}\ \hat{\mathcal{R}}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}\\ &\text{s.t. }\hat{\mathcal{R}}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}=\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\rm InfoNCE}\big(B_k(\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP})\big)\end{aligned}\tag{6}$$

where $\boldsymbol{e}^*$ indicates the optimal concept embedding with respect to IP and OOP concepts, pre-trained along with the encoders f and g[4]; $B_k(\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP})$ indicates the k-th training batch at the size M, incorporating image-text pairs constructed by concepts in $\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}$. Accordingly, the population risk $\mathcal{R}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}$ can be seen as the extension of $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\rm IP}\cup\boldsymbol{Y}_{\rm OOP}}$ to cover all instances drawn from the universal

[3]The InfoNCE loss in this analysis does not include the part with text queries, *i.e.*, $\frac{1}{M}\sum_{i=1}^{M}-\log\frac{\exp\left(\mathrm{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)),g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_i))))/\gamma\right)}{\sum_{j=1}^{M}\exp\left(\mathrm{sim}(f(\boldsymbol{I}(\boldsymbol{y}_j)),g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_i))))/\gamma\right)}$, since its image-query part is better aligned with the image classification objective. The pre-training process balances their optimization so that the minimization of image-query part always indicates the sufficiently small value of the original InfoNCE loss.

[4]Note that, in existing research, the parameter of concept embedding $\boldsymbol{e}$ was interpreted as a part of the text encoder g while isolated from g in this paper. It signify the difference between existing approaches and name-tuning.

image distribution $P_{\mathsf{img}}(\boldsymbol{Y}_{\text{IP}} \cup \boldsymbol{Y}_{\text{OOP}})$ that simultaneously includes IP and OOP concepts:

$$\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} = \mathbb{E}_{\boldsymbol{I}(\boldsymbol{y}_i)\sim D_{\mathsf{img}}(\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}})}\Big[- \log\frac{\exp\big(\text{sim}(f^*(\boldsymbol{I}(\boldsymbol{y}_i)), g^*(\boldsymbol{T}(\boldsymbol{e}^*(\boldsymbol{y_i})))/\gamma\big)}{\sum_{j=1}^{M}\exp\big(\text{sim}(f^*(\boldsymbol{I}(\boldsymbol{y}_i)), g^*(\boldsymbol{T}(\boldsymbol{e}^*(\boldsymbol{y_j})))/\gamma\big)}\Big]. \tag{7}$$

The generalization of CLIP can be achieved via deriving the insightful upper bound of $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$.

Most existing studies that investigated the generalization bound via the relation between population risk and its corresponding empirical risk. It also works in IP generalization but can not be transferred to OOP generalization, due to the impossibility to bound $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ via the pre-training ERM $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ that contains OOP concepts. Instead, we only have the access to IP concepts to derive the following empirical pre-training objective, *i.e.*,

$$\begin{aligned}&\hat{f},\hat{g},\hat{\boldsymbol{e}} = \arg\min_{f,g,\boldsymbol{e}}\ \hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{IP}}}\\ &\text{s.t. } \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{InfoNCE}}\big(B_k(\boldsymbol{Y}_{\text{IP}})\big).\end{aligned} \tag{8}$$

Then image-text pairs drawn from $\boldsymbol{Y}_{\text{OOP}}$ join the fine-tuning on top of the pre-trained models $\hat{f},\hat{g},\hat{\boldsymbol{e}}$ to achieve OOP generalization.

Our analysis focus on learning the concept embedding of $\boldsymbol{Y}_{\text{OOP}}$ given the pre-trained models $\hat{f},\hat{g},\hat{\boldsymbol{e}}$, thus,

$$\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}} = \frac{1}{\hat{K}}\sum_{i=1}^{\hat{K}}\mathcal{L}_{\text{InfoNCE}}\big(B_k(\boldsymbol{Y}_{\text{OOP}})\big). \tag{9}$$

Here we demonstrate that $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{IP}}}$ in Eq.8 and $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$ in Eq.9 jointly derive the upper bound of $\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$. It leads to the formal version of Proposition.1

Proposition 4. *Consider the generalization bound on image distribution $D_{\mathsf{img}}(\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}})$ including IP and OOP concepts. Suppose that D_{IP}, D_{OOP} indicates the distributions with IP and OOP concepts, respectively; and N_{IP} and N_{OOP} denote the number of samples drawn from D_{IP}, D_{OOP} for pre-training and fine-tuning, respectively. If the training samples and batches for pre-training and post-training are drawn and constructed independently and identically, and Assumption.3 is fulfilled, then $\forall\ \epsilon > 0$, it holds the probability $1\text{-}\epsilon$ with*

$$\begin{aligned}&\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \le \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}} + \hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}} + \frac{1}{2}\gamma(D_{\text{IP}}, D_{\text{OOP}})\\ &+d_p(\hat{f},f^*)+d_p(\hat{g},g^*) + \mathfrak{R}_{D_{\text{IP}}}(\mathcal{F},\mathcal{G},\mathcal{E}_{\text{IP}}) + \mathfrak{R}_{D_{\text{OOP}}}(\mathcal{E}_{\text{OOP}})\\ &+\frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{IP}}}} + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{OOP}}}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}}\left(\frac{1}{N_{\text{IP}}}+\frac{1}{N_{\text{OOP}}}\right),\end{aligned} \tag{10}$$

where $\gamma(D_{\text{IP}}, D_{\text{OOP}})$ indicates the distribution gap, $d_p(\cdot,\cdot)$ indicates the approximation error via a p-norm (entry-wise) to measure the difference between functions, $\mathfrak{R}_{D_{\text{IP}}}$ and $\mathfrak{R}_{D_{\text{OOP}}}$ denote the Rademacher complexity for pre-training and fine-tuning, with $\mathcal{F}$,$\mathcal{G}$,$\mathcal{E}_{\text{IP}}$ and $\mathcal{E}_{\text{OOP}}$ denote the functional spaces of f, g, $e(\boldsymbol{Y}_{\text{IP}})$, $e(\boldsymbol{Y}_{\text{OOP}})$.

Proof. Let consider the decomposition

$$\begin{aligned}\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \le\ & ||\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}||\\ &+||\mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}|| + \mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}\end{aligned} \tag{11}$$

where the feature outputs of encoders $\hat{f}$, $\hat{g}$, f^*, g^* are normalized so that they hold the output bounded by 1. In terms of the connection between $\mathcal{R}^{(f,g)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ and Softmax loss, we can derive the similar upper bounds of $||\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}||$ and $||\mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}||$, *i.e.*,

$$\begin{aligned}&||\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}||\\ \le&\mathbb{E}_{\boldsymbol{x}\sim P_{\mathsf{img}}(\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}})}||f^*(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})|| = d_p(\hat{f},f^*)\end{aligned} \tag{12}$$

and

$$\begin{aligned}&||\mathcal{R}^{(f^*,g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} - \mathcal{R}^{(\hat{f},g^*)}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}||\\ \le&\mathbb{E}_{\boldsymbol{x}\sim P_{\mathsf{img}}(\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}})}||g^*(\boldsymbol{T}(\boldsymbol{y}|\boldsymbol{x})) - \hat{g}(\boldsymbol{T}(\boldsymbol{y}|\boldsymbol{x}))||\\ =&d_p(\hat{g},g^*)\end{aligned} \tag{13}$$

where $\boldsymbol{T}(\boldsymbol{y}|\boldsymbol{x})$ indicates the text contains the name of concept $\boldsymbol{y}$ with respect to the image $\boldsymbol{x}$, and $d_p(\hat{f},f^*)$, $d_p(\hat{g},g^*)$ denote the the approximation error via a p-norm to measure the difference between $\hat{f}$,$\hat{g}$ and f^*, g^* [33].

Given this, we only need to prove the inequality

$$\begin{aligned}\mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \le\ & \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}} + \hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}} + \frac{1}{2}\gamma(D_{\text{IP}}, D_{\text{OOP}})\\ &+ \mathfrak{R}_{D_{\text{IP}}}(\mathcal{F},\mathcal{G},\mathcal{E}_{\text{IP}}) + \mathfrak{R}_{D_{\text{OOP}}}(\mathcal{E}_{\text{OOP}})\\ +\frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{IP}}}} &+ \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{OOP}}}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}}\left(\frac{1}{N_{\text{IP}}}+\frac{1}{N_{\text{OOP}}}\right).\end{aligned} \tag{14}$$

Let consider the ERM $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$ derived from the population risk $\mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}}$, and its relation with $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ and $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$. It is obvious that

$$\begin{aligned}&\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{InfoNCE}}(B_k(\boldsymbol{Y}_{\text{IP}}))\\ &= \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{i=1}^{M} - \log\frac{\exp\big(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_{i,k})), g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}})))/\gamma\big)}{\sum_{j=1}^{M}\exp\big(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_{i,k})), g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_{\boldsymbol{j},\boldsymbol{k}})))/\gamma\big)}\\ &= \frac{1}{MK}\sum_{i=1}^{MK} - \log\frac{\exp\big(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)), g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y_i})))/\gamma\big)}{\sum_{\boldsymbol{I}(\boldsymbol{y}_i)\in B,\hat{\boldsymbol{y}}\sim\boldsymbol{Y}_{\text{IP}}(B)}\exp\big(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)), g(\boldsymbol{T}(\boldsymbol{e}(\hat{\boldsymbol{y}})))/\gamma\big)}\end{aligned} \tag{15}$$

where $\boldsymbol{Y}_{\text{IP}}(B)$ indicates the IP category set included by the batch B, which the current image $\boldsymbol{I}(\boldsymbol{y}_i)$ is drawn from. Notice that the size of $\boldsymbol{Y}_{\text{IP}}(B)$ is M and due to Assumption.3,

$$-\log \frac{\exp\left(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)),g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_i)))/\gamma\right)}{\sum_{\boldsymbol{I}(\boldsymbol{y}_i)\in B,\hat{\boldsymbol{y}}\sim\boldsymbol{Y}_{\text{IP}}(B)}\exp\left(\text{sim}(f(\boldsymbol{I}(\boldsymbol{y}_i)),g(\boldsymbol{T}(\boldsymbol{e}(\hat{\boldsymbol{y}})))/\gamma\right)}$$

could be interpreted as a M-class Softmax loss. Since the samples and batches are drawn and constructed independently and identically, the minimization of $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ exactly represents ERM with regards to $\hat{f}$,$\hat{g}$, $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{IP}})$ to classify images in $\boldsymbol{Y}_{\text{IP}}$.

Resemble the same derivation, we confirm $\hat{\mathcal{R}}^{\hat{e}(\boldsymbol{Y}_{\text{OOP}})}_{\boldsymbol{Y}_{\text{OOP}}}$ as ERM with regards to the OOP name-embedding parameters $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{OOP}})$ to classify images in $\boldsymbol{Y}_{\text{OOP}}$. They further lead to two observations:

- The minimization of $\hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}}$ leads to $\hat{f}$,$\hat{g}$, $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{IP}})$ while pre-training. It only measures the empirical risk to classify the samples with respect to IP concepts, and does not cause any change of $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{OOP}})$.
- The minimization of $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$ focuses on the optimization of $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{OOP}})$ on the shoulder of pre-trained $\hat{f}$,$\hat{g}$, $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{IP}})$. It only measures the empirical risk to classify the samples with respect to OOP concepts, and does not fine-tune any pre-trained parameters of $\hat{f}$,$\hat{g}$, $\hat{\boldsymbol{e}}(\boldsymbol{Y}_{\text{IP}})$.

From this observations, it holds

$$\begin{aligned}\hat{f},\hat{g},\hat{e}(\boldsymbol{Y}_{\text{IP}}),\hat{e}(\boldsymbol{Y}_{\text{OOP}}) &= \arg\min_{f,g,\boldsymbol{e}} \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \\ &= \arg\min_{f,g,\boldsymbol{e}} \hat{\mathcal{R}}^{(\hat{f},\hat{g},\hat{e}(\boldsymbol{Y}_{\text{IP}}))}_{\boldsymbol{Y}_{\text{IP}}} + \hat{\mathcal{R}}^{\hat{e}(\boldsymbol{Y}_{\text{OOP}})}_{\boldsymbol{Y}_{\text{OOP}}}.\end{aligned} \tag{16}$$

Derived from the theoretic results in [36], we have the inequality

$$\begin{aligned}\mathcal{R}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} \leq \hat{\mathcal{R}}^{(\hat{f},\hat{g})}_{\boldsymbol{Y}_{\text{IP}}\cup\boldsymbol{Y}_{\text{OOP}}} &+ \frac{1}{2}\gamma(D_{\text{IP}},D_{\text{OOP}}) \\ &+\mathfrak{R}_{D_{\text{IP}}}(\mathcal{F},\mathcal{G},\mathcal{E}_{\text{IP}}) + \mathfrak{R}_{D_{\text{OOP}}}(\mathcal{E}_{\text{OOP}}) \\ &+\frac{3}{2}\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2MK}} + \frac{3}{2}\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2MK'}} \\ &+\frac{1}{2}\sqrt{\frac{(b-a)^2\ln(4/\epsilon)}{2}\left(\frac{1}{MK}+\frac{1}{MK'}\right)}.\end{aligned} \tag{17}$$

The proposition can be proved by replacing MK,MK' by N_{IP}, N_{OOP}; then combining Eq.12,13,16,and 17 together. □

Insights

Proposition.4 is interpreted from three parts:

1. Optimizing the OOP-name embedding parameters with fixed pre-trained CLIP models, *i.e.*, $\hat{\mathcal{R}}^{\hat{e}(\boldsymbol{Y}_{\text{OOP}})}_{\boldsymbol{Y}_{\text{OOP}}}$, can lead to the upper bound to achieve the universal generalization across IP and OOP concepts.
2. $\frac{1}{2}\gamma(D_{\text{IP}},D_{\text{OOP}})$ reflects that, the OOP generalization relies on the connection between pre-training knowledge and OOP concepts. If an OOP concept is hardly correlated with the pre-training knowledge, the upper bound would be very loose even if the OOP-name embedding is well-trained by $\hat{\mathcal{R}}_{\boldsymbol{Y}_{\text{OOP}}}$.
3. The number of training instances used for OOP-name learning is quiet important, which can efficiently control the upper bound via $\frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{IP}}}} + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_{\text{OOP}}}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}}\left(\frac{1}{N_{\text{IP}}}+\frac{1}{N_{\text{OOP}}}\right)$ because the number of pre-training instances N_{IP} is sufficiently large. This finding supports our training pair augmentation strategy in OOP concepts in Sec.5.2.

7.2. Pitfall of Prompt-tuning and Adapter in OOP Generalization

Despite the advantage of name learning in OOP concepts, it would be important to know whether OOP generalization can be achieved using existing fine-tuning algorithms for CLIP. These algorithms are basically categorized into two branches: prompt-tuning, which fine-tunes the pre-trained context embedding given a prompt; adapters, which is inserted as a lightweight layer ahead of the image feature normalization. A flood of diverse fine-tuning algorithms derived from CLIP can be viewed as their variants, while they can be concluded into two principles:

- Fine-tune a part of encoders f,g or the word embedding observed during pre-training;
- Ignore the specification of name embedding about OOP concepts.

Here we elaborate that the second principle may lead to the disaster results in OOP generalization.

Proposition 5. *$\forall\boldsymbol{y}_1\in\boldsymbol{Y}_{\text{IP}}$ and $\forall\boldsymbol{y}_2\in\boldsymbol{Y}_{\text{OOP}}$, if the embeddings $e(\boldsymbol{Y}_{\text{OOP}})$ can be arbitrarily initialized, then given $\forall\epsilon>0$ and an image $\boldsymbol{I}$ labeled by the OOP concept $\boldsymbol{y}_2$, if the text encoder g hold L-Lipschitz to its word embeddings, it holds $d_p\big(\log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}),\log P_V^{(f,g)}(\boldsymbol{y}_2|\boldsymbol{I})\big)\leq\epsilon$.*

Proof. Let $\boldsymbol{e}(\boldsymbol{y}_1)$ denotes the embedding of the IP class $\boldsymbol{y}_1$ and $\boldsymbol{e}(\boldsymbol{y}_2)$ denotes the embedding of the IP class $\boldsymbol{y}_2$. Here we consider a vector $\mathbf{v}$ with the size consistent with word embedding and its norm holds $||\mathbf{v}||=1$. Since $e(\boldsymbol{y}_2)$ can be arbitrarily initialized, given the pre-trained embedding $\boldsymbol{e}(\boldsymbol{y}_1)$ and $\forall\epsilon>0$, we initialize $e(\boldsymbol{y}_2)=e(\boldsymbol{y}_1)+\frac{\epsilon}{L}\mathbf{v}$.

Here we consider to classify the image $\boldsymbol{I}$ via Eq.1:

$$\begin{aligned}P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}) &= \frac{\exp\left(\frac{\text{sim}(f(\boldsymbol{I}),g(\boldsymbol{T}(\boldsymbol{y})))}{\gamma}\right)}{\sum_{\boldsymbol{y}_i\in V}\exp\left(\frac{\text{sim}(f(\boldsymbol{I}),g(\boldsymbol{T}(\boldsymbol{y}_i)))}{\gamma}\right)}; \\ P_V^{(f,g)}(\boldsymbol{y}_2|\boldsymbol{I}) &= \frac{\exp\left(\frac{\text{sim}(f(\boldsymbol{I}),g(\boldsymbol{T}(\boldsymbol{y})))}{\gamma}\right)}{\sum_{\boldsymbol{y}_i\in V}\exp\left(\frac{\text{sim}(f(\boldsymbol{I}),g(\boldsymbol{T}(\boldsymbol{y}_i)))}{\gamma}\right)},\ \text{s.t. } \boldsymbol{y}_1,\boldsymbol{y}_2\in V.\end{aligned} \tag{18}$$

Here we consider their logarithmic comparison:

$$
\begin{aligned}
& d_p\big(\log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}), \log P_V^{(f,g)}(\boldsymbol{y}_2|\boldsymbol{I})\big) \\
=& ||\log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}) - \log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I})|| \\
=& ||\mathrm{sim}(f(\boldsymbol{I}), g(\boldsymbol{T}(\boldsymbol{y}_1)) - \mathrm{sim}(f(\boldsymbol{I}), g(\boldsymbol{T}(\boldsymbol{y}_2))|| \\
=& \left|\left| f(\boldsymbol{I})^\top \left(g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_1))) - g(\boldsymbol{T}(e(\boldsymbol{y}_1) + \frac{\epsilon}{L}\mathbf{v})) \right) \right|\right| \\
\leq & ||f(\boldsymbol{I})|| \left|\left| \left(g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_1))) - g(\boldsymbol{T}(e(\boldsymbol{y}_1) + \frac{\epsilon}{L}\mathbf{v})) \right) \right|\right|
\end{aligned}
\tag{19}
$$

Since the encoders provide normalized output $||f(\boldsymbol{I})|| = 1$, combine the given Lipschitz condition of $g(\boldsymbol{T}())$ then we have

$$
\begin{aligned}
& ||\log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I}) - \log P_V^{(f,g)}(\boldsymbol{y}_1|\boldsymbol{I})|| \\
\leq & ||f(\boldsymbol{I})|| \left|\left| \left(g(\boldsymbol{T}(\boldsymbol{e}(\boldsymbol{y}_1))) - g(\boldsymbol{T}(e(\boldsymbol{y}_1) + \frac{\epsilon}{L}\mathbf{v})) \right) \right|\right| \\
\leq & L \left|\left| \left(\boldsymbol{e}(\boldsymbol{y}_1) - e(\boldsymbol{y}_1) + \frac{\epsilon}{L}\mathbf{v}) \right) \right|\right| \\
\leq & L\frac{\epsilon}{L} = \epsilon.
\end{aligned}
\tag{20}
$$

□

Proposition.5 demonstrate that if we do not provide the optimization to OOP embeddings, their initialization may lead to a poor classification results when their word embedding close to the IP embedding optimized via pre-training.

Appendix.C: Implementation

We elaborate the specific implementation of the Few-Shot Name Learning (FSNL) along with various baseline models. Unless specified, all algorithms were implemented with the OpenCLIP ViT-B-16, 224x224, laion400m_e32 version as their backbones in Sec.6.2.1, 6.2.2, 6.2.3, and the ablation study. In the empirical study about reproducible scaling law in Sec.6.3, we consider 6 different backbones implemented by the original CLIP proposed by OpenAI (*i.e.*, OpenAI ViT-B-16), and 5 CLIP variants derived from the pre-training with LAION at different scales of dataset and model size (*i.e.*, versions OpenCLIP ViT-B-16; OpenCLIP ViT-L-14; OpenCLIP ViT-g-14; EVA01-CLIP-g-14; and EVA02-CLIP-E-14+). The architecture, model size, pre-training dataset and implementation refer to [28]*. Note that, some CLIP variants pre-trained with WIT-400M or LAION-2B may observe OOP concepts defined by LAION-Beyond. Even so, CLIP, OpenCLIP, and its variants consistently underperform their FSNL counterparts in the OOP class prediction (see our empirical study "consistency with neural scaling law", where the evaluation is based upon *Pokemon* with 4-shot tuning with our FSNL).

For a fair comparison, all the few-shot tuning and evaluation across baselines should be implemented by the same device. So we consistently took a single Nvidia A100 (80-G) for different models' training, in order to incorporate the largest model (EVA-CLIP02-E-14+).

FSNL. FSNL's batch size and learning rate vary across different domains and shot counts. Specifically, we assume an optimal learning rate of 2e-4 for the cap of batch size as 128, and the actual learning rate is scaled linearly according to the set batch size. For 1, 2, and 4 shots, the implementation batch size is set to the maximum number of training samples (i.e., num_shots * num_classes) to cover all samples in a batch. For 8 and 16 shots, we use 4 * num_classes as the implementation batch size. It is made to tolerate the memory consumption and model performance. FSNL is trained for 200 epochs in all scenarios, using SGD as the optimizer and a cosine annealing rule for learning rate decay. We took the 4-shot experimental results in Sec.6.2.2,6.2.3, and 6.3.

CoOp. Compared with the official implementation, we set the batch size and the max epoch counts consistent with FSNL for a fair comparison. The rest remains the same.

CoCoOp. Due to the extremely slow training speed behind the CoCoOp algorithm, we raise the batch size of the original implementation from 1 to 16. Our reconfiguration accelerates the CoCoOp's training process with the similar time consumption in FSNL. It results in their fair comparison and moreover, improves the CoCoOp's performance using a single image per batch applied in the original paper. The rest setup remains the same.

CLIP-Adapter. Following the variant recommended in the official paper, we used the version that finetunes the image feature while freezing the classifier weight. The residual ratio α was fixed at 0.2, as per the official code, across all experiments. The prompt template for different domains in LAION-Beyond was consistent with FSNL. Notably, we increased the learning rate for CLIP-Adapter, as the recommended 1e-5 proved almost ineffective for open-vocabulary tasks in our experiments, with negligible changes to its performance. Through further empirical exploration, we found 0.01 as the optimal learning rate for CLIP-Adapter in open-vocabulary tasks, resulting a stronger baseline.

Linear Probe. The implementation was carried forward from the CoOp project.

TaskRes. We followed TaskRes's strategy of using different epoch number for different shot numbers and fixed the scaling factor at the frequently used 0.5 in the original paper to accurately reflect its performance. The rest hyper-parameters were kept consistent with the official code.

ZSNL. We assume the setup with IP image-text pairs $\boldsymbol{D}_{\mathsf{ip}}^{<I,Y>}$(16 shots drawn from IP labeled images), OOP images $\boldsymbol{D}_{\mathsf{oop}}^{I}$ and OOP texts $\boldsymbol{D}_{\mathsf{oop}}^{T}$ (**zero-shot implies unpaired between** $\boldsymbol{D}_{\mathsf{oop}}^{I}$, $\boldsymbol{D}_{\mathsf{oop}}^{T}$). The first phase is formulated

$$
\min_{\mathbf{w}} \ \lambda\mathcal{L}_{\mathsf{SC}}(\mathbf{w}\circ f; \boldsymbol{D}_{\mathsf{ip}}^{<I,Y>}) + (1-\lambda)\mathcal{L}_{\mathsf{MSC}}(\mathbf{w}\circ f; \boldsymbol{D}_{\mathsf{oop}}^{I}) \tag{21}
$$

where $\mathbf{w}$ is a learnable linear head inserted ahead of visual feature normalization; $\mathcal{L}_{\mathsf{SC}}$ and $\mathcal{L}_{\mathsf{MSC}}$ denote supervised contrastive loss and unsupervised constrastive clustering loss ($\lambda = \frac{1}{2}$), then $\mathbf{w}$ trained to assign $\boldsymbol{D}^I_{\mathsf{oop}}$ with cluster labels. Given $\boldsymbol{D}^I_{\mathsf{oop}}$ features with cluster labels, their cluster centers are computed then Huggarian algorithm run to find the optimal matching between the centers and $\boldsymbol{D}^T_{\mathsf{oop}}$. Finally, given $D_1 \subset \boldsymbol{D}^I_{\mathsf{oop}}$ and $D_2 \subset \boldsymbol{D}^T_{\mathsf{oop}}$ share the same cluster, we pair them in shuffle to train $e(\boldsymbol{y}_{\mathsf{oop}})$ via Eq.4 in the paper.

Prompt templates. FSNL used the best prompt-based probing results through general-task prompts. The general-task prompt templates are derived from all hand-craft templates verified in [45]*, and their combination with specific domain's name. Specifically, if the base template refers to "a photo of {}", here are the specific extension with the domain names, respectively.

Pokemon: a photo of {}, a type of pokemon.

Animals: a photo of {}, a type of animal.

Architecture: a photo of {}, a type of architecture.

Attire: a photo of {}, a type of attire.

FolkArt: a photo of {}, a type of folk art.

Food: a photo of {}, a type of food.

Insects_Spiders: a photo of {}, a type of insect or spider.

Landmark: a photo of {}, a type of landmark.

Plants_Fungi: a photo of {}, a type of plant or fungus.

CoOp, CoCoOp, OpenCLIP, CLIP-Adapter, and TaskRes also took the aforementioned prompts in testing for a fair comparison. Their results were produced through the best prediction between their original prompts and the prompts derived with the domain names. Our reproduced CoOp, CoCoOp, CLIP-Adapter, and TaskRes were empirically compared with their original implementation to ensure their performances significantly better than the previous versions.

Appendix.D: More Empirical Results

Visualizing image features across OOP concepts in LAION-Beyond

We extend our t-SNE visualiztion of OOP-concept image features in Sec.4.3 from *Plants_Fungi* to all 9 domains of LAION-Beyond. In the vast majority of domains, we observed that the features are distributed with significant clustering gaps between different OOP categories (Fig.19.a, 20.a, 21.a, 22.a, 23.a, 24.a, 25.a, 26.a) and more importantly, these OOP-named clusters have no overlaps with In-Pre-training (IP) image features (Fig.19.b, 20.b, 21.b, 22.b, 23.b, 24.b, 25.b, 26.b). This observation is consistent with our finding in Sec.4.3. Thus, it illustrates the discriminability of OOP-image features even if their corresponding OOP concepts never shown in language during pre-training.

Interestingly, in *Pokemon*'s visualization (Fig.27.a-b), it was noted that despite clusters being sparser within this domain compared to others, they still exhibited clear clustering characteristics. This phenomenon is likely attributable to the more modality shift in the Pokemon domain, spanning game screenshots, trading cards, Pokémon figurines, and illustrations. This variety in representation styles contributes to the sparsity within clusters yet maintains their distinctiveness, offering insights into the robustness of the vision encoder in handling diverse visual inputs and recognizing underlying patterns.

Ablation details of FSNL in OOP-concept few-shot learning

We devise the ablation based on different similarity definitions based on ancient branching (what we use in the main experiment), along with k nearest neigbors and k-means clustering. For k-nn, for each image feature and its text embedding, we search the top-8 text embeddings with the nearest distances to construct the similarity; for k-means, we directly apply the class number as the cluster number, then execute the k-mean clustering over all text embeddings to generate the similarity (the same cluster shared the texts).

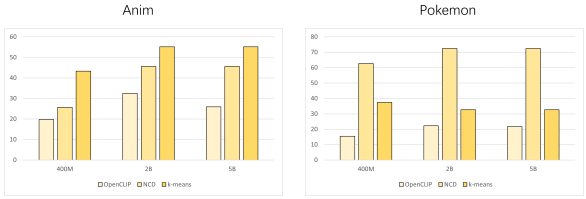


Figure 8. The ablation of ZSNL for cluster initialization based on the results in LAION-Beyond (400M),(2B),(5B).

Ablation of ZSNL in OOP-concept zero-shot learning

Our ablation for ZNSL focuses on verifying the cluster initialization. Derived from the empirical studies based on LAION-400M, 2B, and 5B, We compare our NCD-based strategy along with OpenCLIP and the initialization based on k-mean clustering. The results can be found in Fig.8. As we observed, k-mean clustering perhaps the more suitable cluster initialization approach for ZSNL when the OOP concepts share more semantic with existing IP concepts, *e.g.*, in *Animal*; however, NCD would be more superior when OOP concepts involved with large gap in their appearances to their IP concepts, *e.g.*, in *Pokemon*.

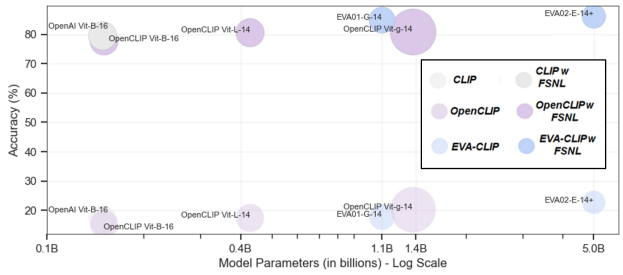


Figure 9. The strong model adaptability with OpenCLIP variants and Their performances with FSNL.

OOP image caption:

The **Yellowfin Leatherjacket** is hiding in a crevice on a coral reef.

IP image name:

Blue Dacnis

Figure 10. The Visualization of OOP-class and IP-class instances in Animals domain.

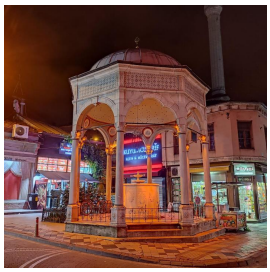
Adaptability of OpenCLIP-model familiy.

Leading study recently justify the CLIP's performance can be consistently improved by scaling the size of architecture and pre-training set [4]. To prove its violation in OOP-class generalization, we reported zero-shot learning with 6 CLIP systems [28] scaling with their model sizes and pre-training sets, then compare their counterparts with OOP word-and-phrase embeddings fine-tuned by FSNL. As observed in Fig.9, we found that despite using OpenCLIP varaints with larger architectures and pre-training sets, their performances are invisibly improved. It implies the neural scaling laws can not solve OOP generalization, consistent with our finding in Sec.4.2. In contrast, FSNL hugely improves these CLIP-based variant models, whatever the architecture or the size of pre-training data.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint*



OOP image caption:

Familok stands tall amidst the charming European-style architecture. The cobblestone street and parked cars add to the warm and cozy atmosphere.

IP image name:

Şadırvan

Figure 11. The Visualization of OOP-class and IP-class instances in Architecture domain.

OOP image caption:

Тюбетейка is worn by two men in the image, one with a beard and the other with a mustache.

IP image name:

Chopines

Figure 12. The Visualization of OOP-class and IP-class instances in Attire domain.

OOP image caption:

Bonecas de Estremoz are displayed on a black table, accompanied by decorative items, creating a warm and cozy atmosphere.

IP image name:

Kuckucksuhr

Figure 13. The Visualization of OOP-class and IP-class instances in Folkart domain.

OOP image caption:

きんぴらごぼう is showcased on a white wooden table, accompanied by chopsticks and another bowl in the background.

IP image name:

Spätzle

Figure 14. The Visualization of OOP-class and IP-class instances in Food domain.

OOP image caption:

In the image, the **Fox-colored Stingless Bee** is in a yellow flower.

IP image name:

Dark-barred Twin-spot Carpet

Figure 15. The Visualization of OOP-class and IP-class instances in Insects_Spiders domain.

OOP image caption:

N 서울타워 stands proudly on Namsan Mountain in the heart of Seoul. The iconic landmark is set against a picturesque backdrop of blue and white clouds.

IP image name:

Duomo di Milano

Figure 16. The Visualization of OOP-class and IP-class instances in Landmark domain.

OOP image caption:

Hangehange thrives on a tree in a lush forest ecosystem surrounded by vibrant plants and trees.

IP image name:

Pink Earth Lichen

Figure 17. The Visualization of OOP-class and IP-class instances in Plants_Fungi domain.

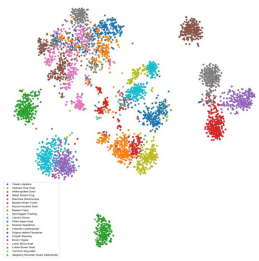
OOP image caption:

Chien-Pao, a type of pokemon, is standing on a beach with a stone block in the background.

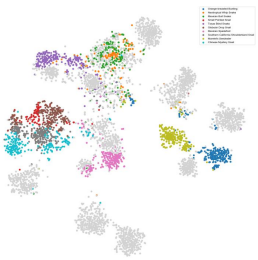
IP image name:

Bulbasaur

Figure 18. The Visualization of OOP-class and IP-class instances in Pokemon domain.



(a). Image features of OOP concepts



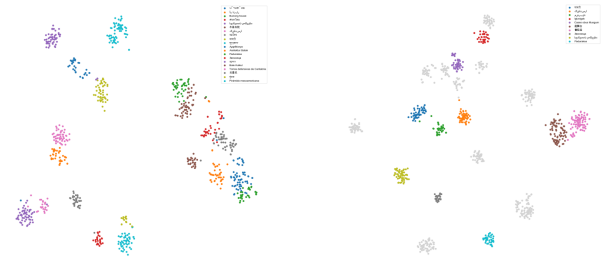
(b). Image features of OOP and IP concepts

Figure 19. The t-SNE visualization results for Animals domain.

arXiv:2309.16609, 2023. 2

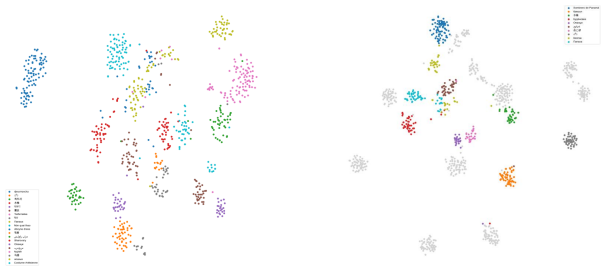
[3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 2

[4] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scal-



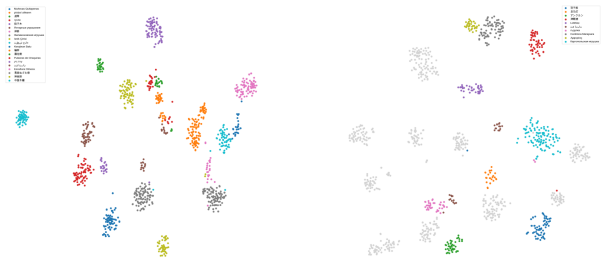
(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 20. The t-SNE visualization results for Architecture domain.



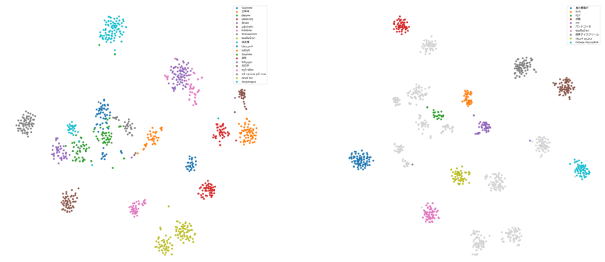
(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 21. The t-SNE visualization results for Attire domain.



(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

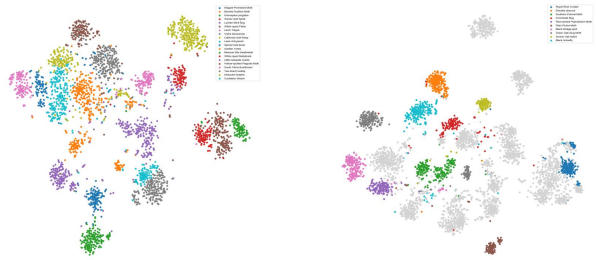
Figure 22. The t-SNE visualization results for Folkart domain.



(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

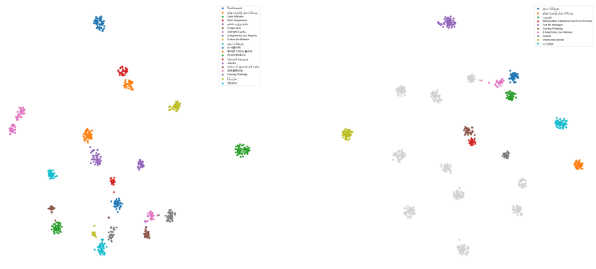
Figure 23. The t-SNE visualization results for Food domain.

ing laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision*



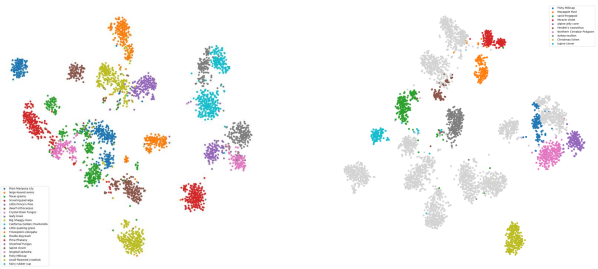
(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 24. The t-SNE visualization results for Insects_Spiders domain.



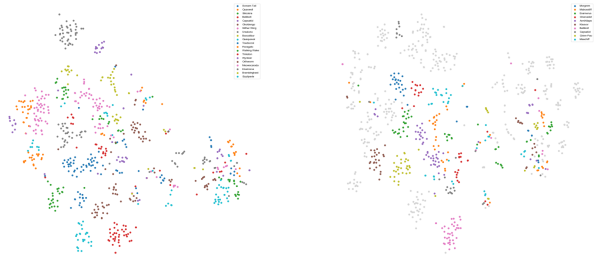
(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 25. The t-SNE visualization results for Landmark domain.



(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 26. The t-SNE visualization results for Plants_Fungi domain.



(a). Image features of OOP concepts (b). Image features of OOP and IP concepts

Figure 27. The t-SNE visualization results for Pokemon domain.

Animals92_42

OOP:

Rio Grande Chirping Frog, Rosenberg's Gladiator Frog, Northeastern China Hynobiid Salamander, Moore's Frog, Garden Slender Salamander, Allegheny Mountain Dusky Salamander, Schlegel's Frog, Swinhoe's frog, Italian Stream Frog, Guenther's Frog, Robust Kajika Frog, Temple Tree Frog, Spot-legged Treefrog, Mexican Spadefoot, Orange-breasted Bunting, Scrub Euphonia, Grayish Baywing, Variable Oriole, Common Chlorospingus, Morelet's Seedeater, Masked Tityra, Masked Water-Tyrant, Sulphur-bellied Flycatcher, Swinhoe's White-eye, Yucatán Squirrel, Gould Beanclam, Giant Floater Mussel, Pacific Littleneck Clam, Chinese Mystery Snail, Globular Drop Snail, Volcano Keyhole Limpet, Lewis' Moon Snail, Round-mouthed Snail, Ostrich Foot Snail, Scaled Worm Snail, shag-rug nudibranch, Modest Cadlina, Black-margined Nudibranch, White-spotted Dorid, Branched Dendronotus, Colorful Dirona, Spotted Leopard Dorid, Pancake Aphelodoris, Black-tipped Spiny Doris, Orange-spike Polycera, Button's Banana Slug, Arboreal Snail, Slippery Moss-snail, Rounded Snail, Small Pointed Snail, California Lancetooth Snail, Robust Lancetooth Snail, Chocolate-band Snail, White Italian Snail, Yellow Garden Slug, Jet Slug, Draparnaud's Glass-snail, Changeable Mantleslug, Chinese Slug, Southern Flatcoil, Southern California Shoulderband Snail, Redwood Sideband, Cuban Brown Snail, Black-velvet Leatherleaf, Tropical Leatherleaf Slug, Brown Tegula, California Spiny Chiton, Yellowfin Leatherjacket, African Redhead Agama, Taiwan Japalure, Italian Slow Worm, Transvolcanic Alligator Lizard, Northern Three-lined Boa, Coast Night Snake, Western Milksnake, Neotropical Whip Snake, Western Leaf-nosed Snake, Mexican Bull Snake, Clouded Anole, Common Slug-eater, Texas Blind Snake, Eastern Spiny Lizard, Longtail Mabuya, Many-lined Sun-skink, Pale-flecked Garden Sunskink, Common New Zealand Skink, Gilbert's Skink, Rainbow Mabuya, Common Spotted Whiptail, Gray-checkered Whiptail, Middle American Ameiva, Khorat Blind Snake

IP:

Florida Fighting Conch, Boomslang, Florida Green Watersnake, Channeled Applesnail, Black Tegula, Eastern Water Skink, Clown Doris, Central American Boa, Heath's Dorid, Blue Dacnis, Green Tree Frog, Blanchard's Cricket Frog, Gumboot Chiton, Plantain Squirrel, Broadhead Skink, Alder Flycatcher, Hawfinch, Common Five-lined Skink, Common Tree Frog, Blue Grosbeak, Greater Earless Lizard, Common Bluetongue, Blue Dragon, Eastern Glass Lizard, Garden Snail, African Striped Skink, Greenhouse Frog, Altamira Oriole, Giant Keyhole Limpet, Guineafowl Puffer, Green Falsejingle, Orchard Oriole, Common Jingle, Foothill Yellow-legged Frog, Monterey Dorid, Linnet, Copse Snail, Hilton's Aeolid, Indian White-eye, Common Periwinkle, Central American Indigo Snake, Eastern Red-backed Salamander

Figure 28. Visual Concept List for Animals doamin.

and Pattern Recognition, pages 2818–2829, 2023. 2, 7

[5] Sua Choi, Dahyun Kang, and Minsu Cho. Contrastive mean-shift learning for generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23094–23104, 2024. 5

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 5

[7] Henghui Ding, Scott Cohen, Brian Price, and Xudong Jiang. Phraseclick: toward achieving flexible interactive segmentation by phrase and click. In *Computer Vision–ECCV 2020:*

Architecture50_23

OOP:

Cases obus Musgum, نوبي البيت, Arsitektur Batak, Bale Kulkul, Bunong house, Nhà Rông, Paduraksa, Rangkiang, Rumah Bubungan Tinggi, Uma Sumbanese, Юрта, गुंबज, गुम्बा, गुरुद्वारा, बावड़ी, ชุมชนแพ, ศาลาไทย, หอไตร, ပုဂံ, ບ້ານຫ້ອຍ, ພິບທາມ, ប្រាសាទខ្មែរ, 각루, 庙宇, 木造寺院, 能舞台, 書院造, 窑洞, 伝統的な町家, Bündnerhaus, Familok, Molinos de La Mancha, Rietdak, Torres defensivas de Cantabria, Αμφιθέατρο, Κυκλαδίτικη αρχιτεκτονική, Звонница, სვანეთის კოშკები, 트룰로, Pirámide mesoamericana, بית עלמין, מקווה, آب انبار, دار الحجر, كُتّم, كاروانسرا, مشربية, مناره, Igluvigak, קרוואן

IP:

Bridge tender's house, Château, Cleit, Crannog, Iglesia, Medina Haram Piazza Shading Umbrellas, Mudhif, Nissen hut, Palloza, Polar Research Station, Rorbu, Sassi di Matera, Shabono, Shepherd's hut, Stavekirke, Tata Somba, Şadırvan, கோபுரம், ტაძარი, 五重塔, 合掌造, 牌坊, 福建土楼

Figure 29. Visual Concept List for Architecture domain.

Attire53_28

OOP:

Aboyne dress, Babban riga, Costume Arlésienne, Feileadh, Váy đầm dạ hội, Φουστανέλα, Дээл, Камзол, носија, ношња, ფერაჯა, برنوس, جلابة, لباس محلى, घागरा, शेरवानी, చీర, ᓄᓕᐊᖅ, 马褂, 毛褂, 旗袍裙, 水袖, 蓑衣, 中山装, 치마저고리, Het Gymreig, kupiah, Nón quai thao, Perak Headdress, Sombrero de Panamá, Будёновка, Калпак, кокошник, Папаха, Тюбетейка, כיפה, عمامة, غترة, مفتن, फेटा, កំណត់, 鉢巻, 족두리, Sharovary, شلوار پاره, botas picudas mexicanas, Опанци, おこぼ, कोल्हापुरी चप्पल, मोज़री, ᑖᕕᒃ, 虎头鞋, 足袋ブーツ

IP:

Anorak, Avarcas, Bascinet, Biretta, Breeches, Béret, Chopines, Chullo, Gele, Ghillie, Klompen, Lederhose, Mantilla, Puletasi, Surcot, Tiara, Trews, Tutu de Ballet, áo bà ba, уша́нка, 汉服, 羽織, 화관, Sombrero, Barong Tagalog, dashiki, Dirndl, Nón lá

Figure 30. Visual Concept List for Attire domain.

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 417–435. Springer, 2020. 2

[8] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023. 2, 6

[9] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom

Folkart59_27

OOP:

Cerâmica Marajoara, Isnik Çinisi, Αμφορέας, 广彩, Khatam-kari, Pūhoro, Птица счастья, ხაფანგური, עין הרע, خط اسلامي, نقوش حناء, 津軽塗, 蓮花燈, 青森ねぶた祭, 神轿, 油纸伞, 羽子板, 走马灯, Mosaïque Byzantine, Bonecas de Estremoz, Muñecas Quitapenas, Poupée Akuaba, Каргопольская игрушка, Филимоновская игрушка, कथपुतली, 三春駒, Ni‘ihau Shell Lei, Pulseras de Chaquiras, Κομπολόι, Янтарные украшения, Кубачинское серебро, подстака́нник, Flaută pană, kulintang a kayo, píobaí uilleann, Pūrerehua, гусле, гъдулка, домбыра, кобза, морин хуур, アンクルン, ジェンベ, ムックリ, نی انبان, کمانچه, सारंगी, 编钟, 釣り太鼓, 胡琴, 拍子木, 神楽鈴, 月琴, Escultura Olmeca, Kerajinan Batu, Lületaşı, अशोक स्तम्भ, 敔, 中国木雕

IP:

Alebrije, Ankh, Arpillera, Chapman Stick, Dhokra, Kachina, Kalaga, Kanun, Kuckucksuhr, Matryoshka, Maultrommel, Renaissance sculptures, slenthem, Troll Cross, Wampum, yidaki, Árbol de la vida, سنتور, こけし, 中国结, 兵马俑, 招き猫, 水墨画, 笙, 糖人, 赤べこ, 青花瓷

Figure 31. Visual Concept List for Folkart domain.

Food53_27

OOP:

Spatlo, Štrukli, კაჭაპური, 法棍, Сельдь под шубой, パンナコッタ, מקרונים, ઢોકળી, 抹茶アイスクリーム, سمبوسة, पानीपुरी, ভটুরা, Будаг, سمك مشوي, רוסטביף, तंदूरी चिकन, หมูสะเต๊ะ, အသားဆီးကြော်, □ □, 鳥の唐揚げ, ขนมจีนน้ำยา, Sooparagua, Σαγανάκι, Σπανακόπιτα, Деруни, Сырники, கோதுமை தோசை, 肉夹馍, Туршия, きんぴらごぼう, □ □ □, Паэлья, المنسف, كشري, 什锦炒饭, Таратор, خورش فسنجان, रस मलाई, मगा, แกงเห็ด, ຜອຍທອງ, 寿喜烧, □ □ □ □, □ □ □, Гуляш, Amok trei, Bánh chưng, Koldūnai, Poutine râpée, Котлеты по-киевски, Фаршированная капуста, ხინკალი, كوسة محشي

IP:

Baklava, Bhapa pitha, Croissant, Empada, Panettone, לחמניה, Chlodnik, Gazpacho, Pozole, خوراک لوبیا, Causa, Stroopwafel, プリン, Babi panggang, 北京烤鸭, Carbonara, Pho, Spätzle, うどん, 牛肉麵, تبولة, سلطة, बिरयानी, カレーライス, Jamaican patty, ギョーザ, वड़ा

Figure 32. Visual Concept List for Food domain.

Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[10] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1

[11] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv preprint arXiv:2306.09316*, 2023. 1

[12] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1

[13] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020. 2

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[15] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 2

[16] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed. Transductive zero-shot and few-shot clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28816–28826, 2024. 8

[17] Prasanna Mayilvahanan, Thaddäus Wiedemer, Evgenia Rusak, Matthias Bethge, and Wieland Brendel. Does clip's generalization performance mainly stem from high train-test similarity? *arXiv preprint arXiv:2310.09562*, 2023. 2

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 2

[19] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023. 5

[20] Balamurali Murugesan, Julio Silva-Rodrguez, Ismail Ben Ayed, and Jose Dolz. Robust calibration of large vision-language adapters. In *European Conference on Computer Vision*, pages 147–165. Springer, 2024. 2

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[22] Sarah Parisot, Yongxin Yang, and Steven McDonagh. Learning to name classes for vision and language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23477–23486, 2023. 8

[23] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555: 126658, 2023. 1

[24] 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, 6

[25] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[26] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1, 2

[27] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[28] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 2, 5, 7

[29] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2

[30] Weijie Tu, Weijian Deng, and Tom Gedeon. A closer look at the robustness of contrastive language-image pre-training (clip). *Advances in Neural Information Processing Systems*, 36, 2024. 2

[31] 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

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 3

[33] Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. Universality and limitations of prompt tuning. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[34] Xin Wen, Bingchen Zhao, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. What makes clip more robust to long-tailed pre-training data? a controlled study for transferable insights. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2

[35] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591, 2017. 7

[36] Shuo Yang, Songhua Wu, Tongliang Liu, and Min Xu. Bridging the gap between few-shot and many-shot learning via distribution calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9830–9843, 2021. 4

[37] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023. 2

[38] Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Mark A Hasegawa-Johnson, Yingzhen Li, and Chang D Yoo. C-tpt: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. In *The Twelfth International Conference on Learning Representations*. 2

[39] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 6

[40] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 2, 5

[41] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021. 1

[42] 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. 2

[43] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2

[44] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 6, 7

[45] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 6, 7

[46] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2

[47] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[48] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 2

[49] Beier Zhu, Kaihua Tang, Qianru Sun, and Hanwang Zhang. Generalized logit adjustment: Calibrating fine-tuned models by removing label bias in foundation models. *Advances in Neural Information Processing Systems*, 36, 2024. 2

Insects_Spiders106_52

OOP:

Asian Spinybacked Orbweaver, Black-headed Jumping Spider, Trapezoid Crab Spider, Saddleback Harvestman, S-banded Tiger Beetle, Banded Graphisurus, Spined Oak Borer, Ivory-marked Borer, Clytra laeviuscula, Eight-spotted Flea Beetle, Epilachna mexicana, Cocklebur Weevil, Texas Eyed Click Beetle, Dark Flower Scarab, Delta Flower Scarab, Eleodes obscura, Poecilanthrax lucifer, Common Eastern Physocephala, Eastern Phantom Crane Fly, Dusky-winged Hover Fly, Maize Calligrapher Fly, Black Horsefly, Common Picture-winged Fly, Hackberry Petiole Gall Psyllid, Sumac Gall Aphid, Excultanus excultus, Little mesquite cicada, Thasus gigas, Ormenoides venusta, Charcoal Seed Bug, Lime seed bug, Conchuela Bug, Four-spurred Assassin Bug, Western Box Elder Bug, Lychee Stink Bug, Wilke's Mining Bee, Fox-colored Stingless Bee, Modest Masked Bee, American Sand Wasp, California Gall Wasp, Karoo Balbyter Sugar Ant, Enicospilus purgatus, Dasymutilla aureola, Noble Scoliid Wasp, Steel-blue Cricket-hunter Wasp, Willow Apple Gall Sawfly, Widow Yellowjacket, Two-lined Hooktip, Delicate Cycnia Moth, Deduced Graphic, Brunia antica, Dark Marathyssa Moth, Red-necked Peanutworm Moth, Hollow-spotted Plagodis Moth, Insigillated Pug, Horned Spanworm Moth, Common Aspen Leaf Miner, Common Bush Hopper, Guava Skipper, Speckled Lactura, Larch Tolype, Green Oak-Slug Moth, Tailless Line Blue, Southern Flannel Moth, American Dun-bar Moth, Lawn Armyworm, Black Wedge-spot, Doubleday's Baileya Moth, Elegant Prominent Moth, Destolmia lineata, Georgian Prominent Moth, Dingy Purplewing, White-rayed Patch, South China Bushbrown, Suzuki's Promalactis Moth, Mexican Kite Swallowtail, Eastern Dotted Border, Common Bagworm Moth, Plain Plume Moth, Boxwood Leaftier Moth, Broad-banded Eulogia, Rosy Tabby, White-rayed Metalmark, Banded Scythris Moth, Great Ash Sphinx Moth, Mournful Thyris, Texas Grass Tubeworm Moth, White Triangle Tortrix, Garden Tortrix, Bidens Borer Moth, Urania Swallowtail Moth, Zygaena transalpina, Grass-like Mantid, Spatterdock Darner, Formosan Jewelwing, Pseudagrion pilidorsum, Australian Emerald Dragonfly, Slender Ringtail, Pygmy Percher, Asiatic Blood Tail, Royal River Cruiser, Oriental Longheaded Locust, Painted Meadow Grasshopper, Restless Bush Cricket, Conocephalus melaenus, Gray silverfish

IP:

Leconte's Haploa, Dark-barred Twin-spot Carpet, Grapeleaf Skeletonizer, European Field Cricket, Common Green Darner, Forest Semilooper, Goldenrod Crab Spider, Cotton Tipworm Moth, Green-underside Blue, California Prionus, Common Mestra, Golden Paper Wasp, Little Metalmark, Agrypnus murinus, Idiodes apicata, High Brown Fritillary, Brown House Moth, Hieroglyphic Moth, Galeruca tanaceti, Black Corsair, Azalea Sphinx Moth, European Harvestman, Evergreen Bagworm, Broad-headed Sharpshooter, Bird Hover Fly, Common Silverfish, Dorantes Longtail, Fiery Searcher Beetle, Florida Predatory Stink Bug, Brown Scoopwing, Apple Looper, Horse-chestnut Leafminer, Flea Jumper, Common Eastern Velvet Ant, Common Darter, Large Milkweed Bug, Copper Demoiselle, European Rhinoceros Beetle, Double-banded Scoliid Wasp, Common Gluphisia, Dark Arches, Dark-branded Bushbrown, Leptoglossus zonatus, California Oak Moth, Black Giant Ichneumon Wasp, Small Honey Ant, Milkweed Aphid, Garden Locust, Green Silver-lines, Organ-pipe Mud-dauber Wasp, Aurora Bluetail, Forest Tent Caterpillar Moth

Figure 33. Visual Concept List for Insects_Spiders domain.

Landmark59_30

OOP:

Cueva de Altamira, Ipoġew ta' Ħal Saflieni, Centro Histórico de Salvador, Ἔφεσος, Fotevikens Museum, ἡ Ἀκρόπολις τῶν Ἀθηνῶν, تيمقاد ,أَلْبَتْراء, Lake Wānaka, معبد الأقصر, N 서울타워, Öngtupqa, ปราสาทหินพิมาย, Parc National de l'Andringitra, Stare Miasto w Krakowie, Tallinna vanalinn, Tsé Bii’ Ndzisgaii, Vestnorske fjordar, Carraig Phádraig, Catedral de Notre-Dame de Chartres, Heydər Əliyev Mərkəzi, Мост Багратион, Мост Крымский, Мост через Золотой Рог, Juayúa, Metropolitan Cathedral Church of St David, Стари мост, Хонгорын элс, Państwowe Muzeum Auschwitz-Birkenau, Кижский погост, Петропавловская крепость, Регистон, Մատենադարան, Սանահին վանք, Տաթևի վանք, ანანურის ციხე, სვეტიცხოვლის ტაძარი, المتحف المصري, חומות ירושלים, پل خواجو, برج خليفه, الهرم الأكبر, المسجد الأقصى, جَامِع الشَّيْخ زَايِد الْكَبِير, قصر الحمراء, قلعه بم, مسجد سلطان أحمد, مینارِ پاکستان, میدان نقش جهان, نِزْوَى, बौद्धनाथ, மீனாட்சி அம்மன் கோயில், फ़तेहपुर सीकरी, อนุสาวรีย์ชัยสมรภูมิ, ရွှေတိဂုံဘုရား, 伏见稻荷大社, 红河哈尼梯田, 동대문 디자인 플라자, 天坛大佛

IP:

Borobudur, Brooklyn Bridge, Choquequirao, Château de Versailles, Cinque Terre, Clifton Suspension Bridge, CN Tower, Cristo Redentor, Duomo di Milano, Empire State Building, Golden Gate Bridge, hrad Karlštejn, Hội An, Lincoln Memorial, Líneas de Nazca, Machu Picchu, Millau Viaduct, Murchison Falls, Niagara Falls, Palenque, Ponte di Rialto, Ponte Dom Luís I, Stonehenge, Tour Eiffel, Vasco da Gama Bridge, Μετέωρα, Σαντορίνη, Рилски манастир, ים המלח, 大阪城

Figure 34. Visual Concept List for Landmark domain.

Plants_Fungi113_56

OOP:

Questionable Stropharia, Silverleaf Fungus, totally tedious tubaria, Christmas lichen, Crystal Brain Fungus, Chicken Fat Mushroom, Hoary Rosette Lichen, California Golden Chanterelle, alpine jelly cone, Lemon discos, Orange Moss Agaric, many-forked cladonia, porpidia lichen, Ochre Jelly Club, Common Script Lichen, Smooth Lungwort, Candy Lichen, hairy rubber cup, white basket fungus, Smoky polypore, Hairy Bracket, Northern Cinnabar Polypore, Mayapple Rust, Fishy Milkcap, jellied false coral, slender orange-bush, leafy brain, Cramp Balls, Northern Water Plantain, Broad-leaved Chervil, slender celery, Patē, Rosy sandcrocus, Ponerorchis cucullata, Pale Yucca, false yellowhead, Canada wild lettuce, common elephant's-foot, fourspike heliotrope, Poodle-dog bush, sand fringepod, hairypod pepperweed, Leucolepis Umbrella Moss, Drosera aberrans, Redstem Springbeauty, red sand-verbena, Fen Grass of Parnassus, Streambank Stickleaf, coastal manroot, twinberry honeysuckle, longleaf ephedra, smooth horsetail, giant woollystar, Mexican False Calico, Little Prince's Pine, Lindheimer's Senna, lupine clover, Common Flat-pea, Muller's oak, Bonfire moss, Lindheimer's silktassel, Star Milkvine, Lesser Centaury, Hangehange, woolly cranesbill, Scouring-pad alga, false staghorn fern, Hedwig's fringeleaf moss, veined bristle-fern, Big Shaggy-moss, greater whipwort, smooth ruellia, Slender Hedeoma, dwarf orthocarpus, Beilschmiedia tawa, Plain Mariposa Lily, hanging clubmoss, Whiteywood, Miracle Violet, turkey-mullein, Newberry's velvet-mallow, Southern Checkerbloom, California asterella, panicled willowherb, Hartweg's Sundrops, scarlet beeblossom, leathery grapefern, Crepe fern, Manyleaf Sorrel, mountain toatoa, Little quaking-grass, rosy sedge, Texas grama, Blechnum procerum, narrow-leaved glade fern, common pig fern, Flat-leaved Scalewort, Common Pin Spiderhead, Tmesipteris elongata, small-flowered crowfoot, Shrub Yellowroot, Curveseed Butterwort, large-leaved avens, Stansbury's cliff rose, Fendler's ceanothus, Cape Sumach, Western Soapberry, Arizona chalk dudleya, Bigelow's spike moss, Carolina ponysfoot, sorrelvine, Crêpe ginger, Pima Rhatany

IP:

Common Powderhorn, Burnet-saxifrage, Ocean spray, Silvery Bryum, Desert Blue Bells, Malabar Melastome, anemone stinkhorn fungus, Green Wood Cup, oak mistletoe, Pencil Milkbush, alder buckthorn, Northern wolf's-bane, Goldilocks Buttercup, spearleaf stonecrop, Texas madrone, Pearl Milkweed, Canyon larkspur, black crowberry, ashy sunflower, Hooded Sunburst Lichen, jade plant, Black Witches' Butter, Sea Spurge, European Searocket, Krauss's clubmoss, Carolina sweetshrub, Swiss Cheese Plant, billygoat weed, Hollyhock Rust, Silky Phacelia, Long-spurred violet, Late-flowering Yellow Rattle, Golden Chanterelle, Coast silk tassel, Emory oak, Salad Burnet, Pine Bracket, Pacific poison oak, Fat Jack, Hoof Fungus, Pigeonwood, Indian-shot, Golden Dock, Irish moss, Island Mallow, Maryland meadowbeauty, Fir-Cone Mushroom, climbing rata, Candlesnuff Fungus, Pink Earth Lichen, New Zealand common broom, Pale pink-sorrel, Blue Eryngo, Golden Sunshinebush, Seabeach Groundsel, California Maidenhair Fern

Figure 35. Visual Concept List for Plants_Fungi domain.

Pokemon89_39

OOP:

Barbaracle, Skwovet, Corvisquire, Rolycoly, Silicobra, Sizzlipede, Morgrem, Perrserker, Runerigus, Stonjourner, Arctozolt, Wyrdeer, Kleavor, Ursaluna, Basculegion, Sneasler, Overqwil, Enamorus, Sprigatito, Floragato, Meowscarada, Fuecoco, Skeledirge, Quaxly, Quaxwell, Quaquaval, Oinkologne, Tarountula, Spidops, Tandemaus, Maushold, Fidough, Dachsbun, Arboliva, Squawkabilly, Naclstack, Garganacl, Charcadet, Armarouge, Ceruledge, Tadbulb, Bellibolt, Kilowattrel, Maschiff, Mabosstiff, Shroodle, Grafaiai, Brambleghast, Toedscool, Toedscruel, Capsakid, Scovillain, Espathra, Tinkatink, Tinkatuff, Tinkaton, Wugtrio, Bombirdier, Palafin, Revavroom, Cyclizar, Orthworm, Glimmet, Glimmora, Greavard, Dondozo, Tatsugiri, Annihilape, Clodsire, Farigiraf, Dudunsparce, Scream Tail, Brute Bonnet, Flutter Mane, Slither Wing, Sandy Shocks, Iron Jugulis, Iron Thorns, Frigibax, Arctibax, Baxcalibur, Gimmighoul, Gholdengo, Wo-Chien, Chien-Pao, Iron Valiant, Koraidon, Miraidon, Walking Wake

IP:

Bulbasaur, Mr. Mime, Probopass, Froslass, Rotom, Dialga, Palkia, Heatran, Giratina, Victini, Pignite, Emboar, Pidove, Tranquill, Unfezant, Woobat, Swoobat, Excadrill, Palpitoad, Seismitoad, Sewaddle, Swadloon, Leavanny, Venipede, Whirlipede, Scolipede, Cottonee, Whimsicott, Sandile, Krokorok, Krookodile, Dwebble, Crustle, Scraggy, Scrafty, Sigilyph, Tirtouga, Carracosta, Archen, Zekrom

Figure 36. Visual Concept List for Pokemon domain.