

# ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models

Chunyuan Li<sup>\*1</sup>♣, Haotian Liu<sup>\*2</sup>, Liunian Harold Li<sup>3</sup>, Pengchuan Zhang<sup>1</sup>, Jyoti Aneja<sup>1</sup>  
 Jianwei Yang<sup>1</sup>, Ping Jin<sup>1</sup>, Houdong Hu<sup>1</sup>, Zicheng Liu<sup>1</sup>, Yong Jae Lee<sup>2</sup>, Jianfeng Gao<sup>1</sup>

<sup>1</sup>Microsoft    <sup>2</sup>University of Wisconsin–Madison    <sup>3</sup>UCLA

## Abstract

Learning visual representations from natural language supervision has recently shown great promise in a number of pioneering works. In general, these language-augmented visual models demonstrate strong transferability to a variety of datasets and tasks. However, it remains challenging to evaluate the transferability of these models due to the lack of easy-to-use evaluation toolkits and public benchmarks. To tackle this, we build ELEVATER<sup>1</sup>, the first benchmark and toolkit for evaluating (pre-trained) language-augmented visual models. ELEVATER is composed of three components. (*i*) Datasets. As downstream evaluation suites, it consists of 20 image classification datasets and 35 object detection datasets, each of which is augmented with external knowledge. (*ii*) Toolkit. An automatic hyper-parameter tuning toolkit is developed to facilitate model evaluation on downstream tasks. (*iii*) Metrics. A variety of evaluation metrics are used to measure sample-efficiency (zero-shot and few-shot) and parameter-efficiency (linear probing and full model fine-tuning). ELEVATER is a platform for *Computer Vision in the Wild* (CVinW), and is publicly released at <https://computer-vision-in-the-wild.github.io/ELEVATER/>.

## 1 Introduction

Visual recognition has become ubiquitous in our society [76], with applications in geolocation [66], action recognition [73], street number transcription [58], satellite remote sensing [31], medical imaging [80], self-driving cars [26], *etc.* Core to these applications are visual recognition tasks such as image classification (IC) and object detection (OD). It is of high value to develop transferable visual models that perform well on a wide range of downstream applications. By leveraging large web crawled image-text corpora, recent advances in language-augmented visual models such as CLIP [66] and ALIGN [35] have demonstrated strong transfer performance, making this direction one of the most practical visual learning approaches. The reason is twofold: (*i*) open-set recognition is made possible by reformulating classification tasks as retrieval; (*ii*) model generalization is improved as language supervision significantly increases the coverage of visual concepts for model training.

The success has immediately inspired many studies of large-scale model pre-training [88, 92, 89, 49, 57, 27, 47, 95]. However, these studies use their own evaluation settings based on customized sets of downstream tasks where the detailed process of adapting the models to these tasks is typically not accessible to the public. Thus, it is extremely difficult for researchers to fairly compare models and develop new models based on other people’s works. To fill this gap, we develop an open-source benchmark and toolkit, ELEVATER, to make the research results (*e.g.*, model’s task-level transferability) more rigorous, and reproducible. ELEVATER is composed of three components.

<sup>\*</sup>Equal Technical Contribution    ♣Project Lead

<sup>1</sup>Evaluation of Language-augmented Visual Task-level Transfer

- **Benchmark (Datasets and Knowledge).** We build the first publicly available benchmark to evaluate the *large-scale task-level transferability* of language-augmented visual models. The benchmark consists of two challenges: *Image Classification in the Wild (ICinW)* with 20 IC datasets and *Object Detection in the Wild (ODinW)* with 35 OD datasets. A collected external knowledge base for each dataset which could be used for language data augmentation.
- **Comprehensive Metrics.** To measure the cost of deploying models for real-world applications, we measure a model’s sample-efficiency in the zero-shot, few-shot and full-shot settings and parameter-efficiency in the linear probing and full model fine-tuning settings.
- **Reproducible Toolkit & Language-augmented Adaptation Methods.** We develop an open-source software toolkit to support model adaptation and evaluation. Automatic hyper-parameter tuning is employed to avoid human-in-the-loop tuning, thus reducing human labor and ensuring a fair comparison among different model checkpoints. We also present a set of new model adaptation methods for pre-trained language-augmented visual models. Our methods significantly outperform the traditional vision-only adaptation methods. These methods serve as baselines for the development of more advanced adaptation methods.

In addition, our empirical study leads to interesting findings. (i) Leveraging both text and vision in these models consistently yields better performance than vision-only in few-shot settings; In contrast, random initialization of the linear head in language-augmented visual models is sub-optimal. We also find that few-shot results are always better than zero-shot results, which is different from the results reported in [66]. (ii) For language-augmented visual models, linear probing performs better than full model fine-tuning in the few-shot settings. As the task-specific training data increases, fine-tuning outperforms linear probing. (iii) Our study shows that the use of external knowledge, including *explicit* knowledge of human-compiled thesaurus/dictionaries/documents and *implicit* knowledge stored in GPT3 [6], can improve zero-shot and few-shot learning performance. We summarize the pipeline to use ELEVATER in Figure 1, and organize the paper to focus on the benchmark and toolkit.

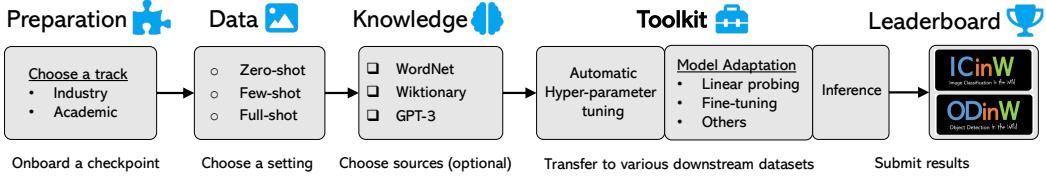


Figure 1: The illustrative pipeline to use ELEVATER to evaluate a model checkpoint.

## 2 Related Work: From Class-level to Task-level Transfer

Zero-shot learning in computer vision has been studied for decades. The research topic has witnessed two distinctive notions of zero-shot: the traditional *class-level zero-shot* that usually refers to the study of generalizing to unseen object categories [42], and the recently popular *task-level zero-shot* that refers to the study of generalizing to unseen datasets/tasks [44, 66]. In Table 1, we compare our benchmark against existing benchmarks. Existing zero-shot learning benchmarks are developed for the class-level zero-shot setting. They are usually in a single domain, with a manual split of categories to produce disjoint training and test categories, *e.g.*, Animal with Attributes (AwA) [43], Caltech-UCSD Birds-200 (CUB) [82], SUN [61], aPY [21], and ZS-ImageNet [67, 24]. For the OD problem, COCO [50] and LVIS [29] represent two well established datasets to compare various OD methods in a single domain, while UODB [86] is a multi-domain OD benchmark.

In contrast, our benchmark focuses on task-level transfer across domains, *i.e.*, it aims to evaluate the transferability of models, by pre-training from their own large corpus, then evaluating zero-shot performance on a diverse set of downstream datasets. This setting has been recently studied [44, 66, 45, 92], and is arguably more practical for real-world applications, as it brings the convenience towards the spirit of one-model-for-all. The well-known ImageNet-1K dataset [14] was originally proposed as a large dataset for model training and testing. It has also recently been considered as one downstream task to study zero-shot transfer [66, 45, 92]. Our work presents the first *public benchmark* to standardize the zero-shot task-level transfer setting. Note that visual task transfer has been previously explored in VTAB [94], which measures good visual representations as those that adapt to diverse, unseen tasks with an emphasis on few training examples. The pre-trained models and task adaptation in VTAB are considered for vision backbones only, and no language model/modality is involved. Our benchmark shares a similar spirit of task-level transfer to VTAB, but

Problem	Benchmark Statistics					Evaluation		
	#Datasets	#Image	#Concepts	Knowledge Source	Zero	Few	Full	
Image Classification (IC)	AwA [43]	1	30337 / 6985	40 / 10	Attributes	✓		
	CUB [82]	1	8855 / 2933	150 / 50	Attributes	✓		
	SUN [61]	1	12900 / 1440	645 / 72	Attributes	✓		
	aPY [21]	1	12695 / 2644	20 / 12	Attributes	✓		
	ZS-ImageNet [67]	1	1.2M / 54K	1K / 360	WordNet	✓		
	ImageNet-1K [14]	1	1.2M / 50K	1K	WordNet	✓		✓
	VTAB [94]	19	2.2M / -	940	-		✓	✓
Object Detection (OD)	ELEVATER (ICinW)	20	638K / 193K	1151 $\diamond$	WordNet, Wiki, GPT-3	✓	✓	✓
	COCO [50]	1	83K / 41K	80	-			✓
	LVIS [29]	1	120K / 40K	1723	WordNet		✓	
	UODB [86]	11	113K / 40K	109	-		✓	
ELEVATER (ODinW)	ELEVATER (ODinW)	35	132K / 20K	314 $\diamond$	WordNet, Wiki, GPT-3	✓	✓	✓

Table 1: Comparison of dataset statistics and evaluation settings. For existing zero-shot datasets in IC, the number of images and concepts are reported for development / evaluation stages separately.  $\diamond$  represents the total number of concepts in the benchmark to evaluate task-level transfer, and there is no train-evaluation category split as in class-level transfer.

strives to analyze the vital role of language and knowledge in visual transfer. All of them are usually evaluated in full-shot settings, without considering task-level transfer. We have further made several novel contributions to consolidate the benchmark: (i) We add external knowledge for each dataset to cultivate new research directions in knowledge-augmented visual models, inspired by the success of knowledge in traditional class-level transfer. (ii) We consider the full spectrum in measuring the sample-efficiency of task adaptation, including zero-shot, few-shot, and full-shot.

In this paper, we develop ELEVATER as a platform for “**computer vision in the wild**”, whose ultimate goal is to develop a transferable foundation model/system that can *effortlessly* adapt to *a large range of visual tasks in the wild*. It consists of two key factors: (i) **The task transfer cost is low**, which is formally defined in Section 3.3, where our evaluation metrics is designed with efficiency considerations. (ii) **The task transfer scenarios are broad**. We illustrate and compare CVinW with other settings using a 2D chart in Figure 2, where the space is constructed with two orthogonal dimensions: input image and output concept. The 2D chart is divided into four quadrants, based on how the model evaluation stage is different from model development stage. Both training and evaluation distributions are consistent in both dimensions for the traditional close-set recognition. Open-set recognition allows new concepts in evaluation, while typically remains the same visual domain [87, 93]; Domain shift allows new visual domain in evaluation, while typically remains the same concept pool [63, 28]. CVinW allows the flexibility in both dimensions, where any new tasks/datasets in the wild essentially fall into.

### 3 Benchmarks

#### 3.1 A Suite of Datasets with Language/Knowledge Augmentations

As a proxy for performing unseen tasks in the wild, we collect a diverse set of public datasets from various domains in computer vision, as the basis of our benchmark. Specifically, we consider 20 datasets for IC and 35 datasets for OD. We exhibit the dataset names in Figure 3 (a), and the detailed statistics of each dataset in Table 5 and Table 6 in Appendix. It is recommended in [66] that studying task-level zero-shot transfer is a way of measuring the task learning capabilities of machine learning systems. The task definition of each downstream recognition dataset is typically specified using category names. Adding user specification/note is a natural way to clarify the task definition *e.g.*, the attribute or explanation of a visual concept. Importantly, a similar spirit has been implemented

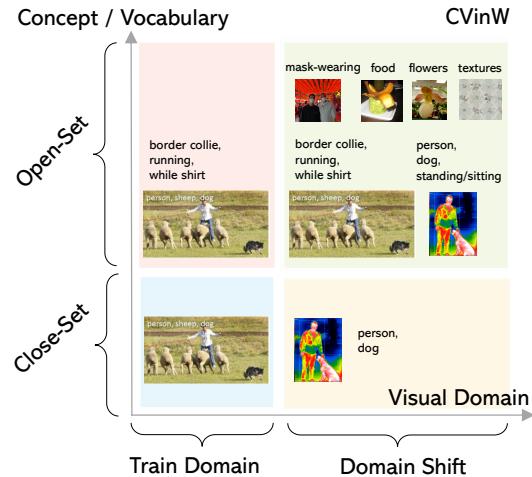
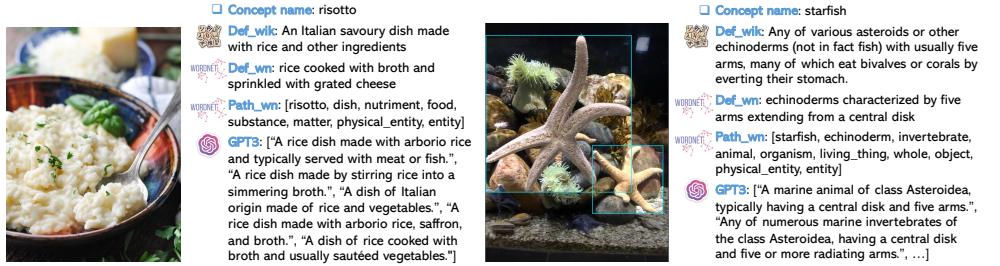


Figure 2: Illustration of CVinW in comparison with close-set, open-set and domain shift.

As a proxy for performing unseen tasks in the wild, we collect a diverse set of public datasets from various domains in computer vision, as the basis of our benchmark. Specifically, we consider 20 datasets for IC and 35 datasets for OD. We exhibit the dataset names in Figure 3 (a), and the detailed statistics of each dataset in Table 5 and Table 6 in Appendix. It is recommended in [66] that studying task-level zero-shot transfer is a way of measuring the task learning capabilities of machine learning systems. The task definition of each downstream recognition dataset is typically specified using category names. Adding user specification/note is a natural way to clarify the task definition *e.g.*, the attribute or explanation of a visual concept. Importantly, a similar spirit has been implemented



(a) Dataset names. The font size is proportional to the number of concepts in each dataset.



(b) Examples of collected external knowledge.

Figure 3: Illustration of our benchmark. Left: Image classification, Right: Object detection.

in traditional class-level zero-shot by adding individual domain-specific knowledge (see Table 1), and demonstrated promising zero-shot performance gains. In this paper, we generalize the notion of “zero-shot” to task-level, collecting external knowledge from general sources for our benchmark.

- *WordNet Hierarchy* (`def_path`). The words along the traversal path from the query node in WordNet [56] to the highest parent node is recorded as the hierarchy knowledge.
- *WordNet Definition* (`def_wn`). The definition in WordNet synsets [56] is used to explain the query.
- *Wiktionary Definition* (`def_wik`). The definition of a query in Wiktionary [55] is used.
- *GPT3 Knowledge* (`gpt3`). For the above three knowledge sources, it is not always feasible to retrieve valid knowledge for any query. To enable full knowledge coverage, we propose to use GPT3 [6] to generate “pseudo” knowledge using in-context-learning, where prompts are constructed with multiple pairs of class names and their Wiktionary definitions. We generate five GPT3 knowledge sequences for each class name, by constructing different context prompts with randomly sampled pairs. See details in Section C.3.

In Fig. 3 (b), we show examples to illustrate the knowledge sources. In practice, there is a trade-off between the knowledge quality and its coverage. For example, WordNet has relatively rich and precise knowledge, but the coverage is low; GPT3 knowledge has the full coverage (as it is generated via prompting a pre-trained neural language model), but it is hard to assess its quality. In the experiment section, we provide baseline results to demonstrate the benefits of external knowledge, and encourage the community to design advanced prompting techniques to leverage these knowledge sources.

### 3.2 Pre-trained Models for Transfer Learning

**Industry Track and Academic Track.** Our benchmark is an evaluation platform for pre-trained models, whose performance largely depends on the scale of the pre-training corpus. Larger corpus typically yields higher performance, but unfortunately results in a barrier to many participants, especially a majority of researchers from university labs. To increase inclusivity, we create two tracks with restrictions on the pre-training data scale: (i) *Academic track* is a setting that limits the data in established public large datasets (*i.e.*, ImageNet-21K [14], GCC3M [70] & 12M [8], YFCC15M [78]). This track is more academic-friendly, aiming to encourage the exploration in data-efficient pre-training methodologies. (ii) *Industry track* has no limit on pre-training data scale, except that images in our benchmark are not allowed in pre-training when reporting zero-/few-shot performance. This track aims to explore the scaling limit. We encourage participants to report the pre-training datasets to enable reproducible research.

**Pre-trained Models.** To establish baseline results on ELEVATER, we evaluate the pre-trained model checkpoints in Table 2. More details of the checkpoints are described in Appendix. Most existing visual models are language-free, where no text is used in model training. Till recently, visual

Checkpoints	Taxonomy		Pre-training Settings		Network Architecture		
	Language Knowledge		Training Objective	Dataset	Vision	Language	Others
<b>Image Classification</b>							
MoCo-v3 [10]	X	X	Self-Supervised	ImageNet-1K (1.2M)	ViT-B	-	-
MAE [30]	X	X	Self-Supervised	ImageNet-1K (1.2M)	ViT-B	-	-
DeiT [79]	X	X	Supervised	ImageNet-1K (1.2M)	ViT-B	-	-
ViT [18]	X	X	Supervised	ImageNet-22K (14M)	ViT-B	-	-
CLIP [66]	✓	X	Image-Text Contrast	WebImageText (400M)	ViT-B	T-B	-
UniCL [88]	✓	X	Image-Text Contrast	ImageNet-21K (13M)	Swin-T	T-B	-
K-LITE [71]	✓	✓	Image-Text Contrast	ImageNet-21K (13M)	Swin-T	T-B	-
<b>Object Detection</b>							
DyHead [13]	X	X	Supervised	Object365	Swin-T	-	-
GLIP [47]	✓	X	Supervised	Object365 & Grounding	Swin-T	Bert-B	Fusion
GLIP-A [47]	✓	X	Supervised	Object365	Swin-T	Bert-B	-
K-LITE [71]	✓	✓	Supervised	Object365	Swin-T	Bert-B	-

Table 2: The pre-trained models evaluated in ELEVATER as baselines. In terms of taxonomy, ✓ indicates the model checkpoint is pre-trained with the use of language / knowledge, while X indicates language- / knowledge-free. For image classification, the number of images in pre-training is reported. T-B indicate a Base-size Transformer architecture, using a 63M-parameter 12-layer 512-wide model with 8 attention heads. Swin-T is a Tiny-size Swin Transformer [52], Bert-B is a Base-size Bert [16], and Fusion indicates a cross-attention module to fuse the image-text features [47].

models are trained in a language-augmented and/or knowledge-augmented manner using a language model [66, 88, 71, 47], among which CLIP [66] represents a strong baseline in the industry track. Please see the detailed taxonomy in Appendix Section F.1.

### 3.3 Evaluation Settings: Efficiency Considerations

One major advantage of pre-trained models is the promise that they can transfer to downstream tasks *effortlessly*. The cost is considered in two orthogonal dimensions: sample-efficiency and parameter-efficiency, as illustrated in Figure 4. The bottom-left corner and top-right corner is the most inexpensive and expensive adaptation strategy, respectively. One may interpolate and make combinations in the 2D space, to get different model adaptation methods with different cost.

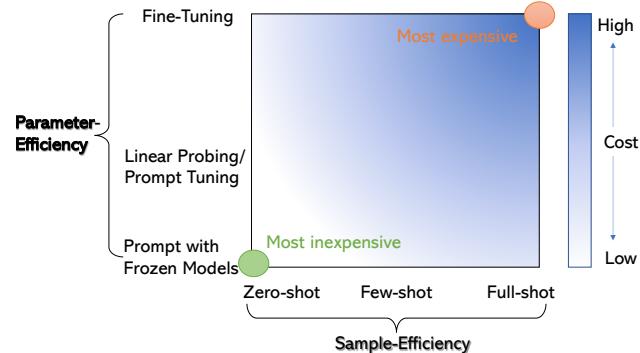


Figure 4: The model adaptation cost chart.

**Sample-efficiency: Zero-, Few-, and Full-shot.** Due to the high cost of annotating data, it is often desired to provide a small number of labeled image-label pairs in downstream datasets. Transferable models should be able to reach high performance in this data-limited scenario. To assess this ability, we vary the number of training set size  $N$  per category in the downstream dataset. For IC,  $N = 0, 5, 20, 50$ . For OD<sup>2</sup>,  $N = 0, 1, 3, 5, 10$ . Three random seeds are chosen, each of which identifies a subset of samples from the full dataset in a deterministic manner. Once the random seed is given, the indices of training samples in few-shot settings are fixed to encourage reproducible research. We also consider the full-shot setting, where all samples of a given dataset are used.

**Parameter-efficiency: Linear Probing vs Full Model Fine-tuning.** Maintaining a small number of dataset-specific model parameters is often favored for model maintenance, as it can be expensive to maintain a unique copy of large model checkpoints for each of the thousands of downstream applications. In IC, linear probing provides a simple strategy for training a dataset-specific linear embedding matrix, while keeping the pre-trained visual backbone frozen. It arguably represents the minimum cost solution for parameter-efficiency. In contrast, fine-tuning often updates the entire weights in backbone and linear head, representing the most expensive solution to model adaptation.

<sup>2</sup>For OD,  $N$ -shot means providing at least  $N$  images per category[85, 47]

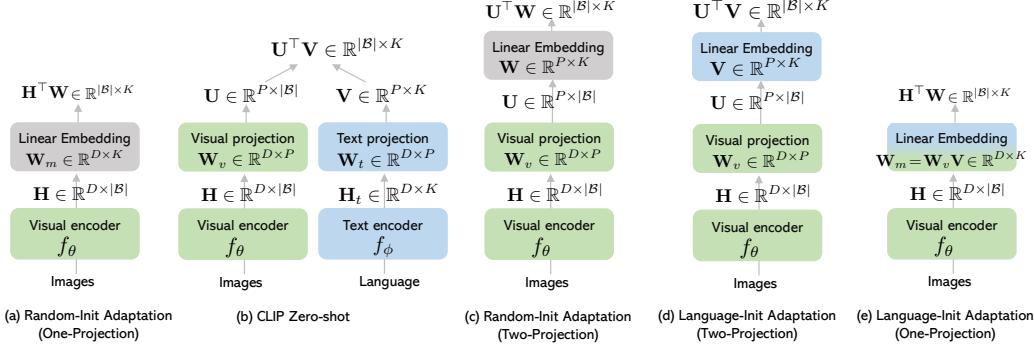


Figure 5: Illustrative comparison of different model evaluation and adaption methods.

In OD [85], linear probing means updating the linear heads for classification and localization tasks only, while fine-tuning means updating all model weights including the backbone and the detectors.

## 4 Toolkits

To ease the process to onboard new checkpoints for evaluation, we provide a software toolkit, including (*i*) automatic hyper-parameter tuning and (*ii*) various strategies for model adaptation to downstream tasks. First, automatic hyper-parameter tuning pipeline is developed to avoid human-in-the-loop tuning, thus reducing human labor and ensuring fair comparisons of different model checkpoints. We follow CLIP [66] to implement a simple grid-search style tuning pipeline, and leave more sophisticated methods like BOHB [20] and DEHB [3] as future work. Details are provided in Appendix. Second, we provide several model adaptation methods as strong baselines, which allow effective transfer learning of pre-trained visual models. The ideas are illustrated in Figure 5. For a downstream dataset, we first represent it in a triplet-wise data format  $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{t}_n, y_n)\}_{n=1}^N$ , where  $\mathbf{x} \in \mathcal{X}$  is the image,  $\mathbf{t} \in \mathcal{T}$  is its corresponding language description, and  $y \in \mathcal{Y}$  is a label indicating the index of the unique language description in the dataset.  $|\mathcal{B}|$  is batch size. In IC, the number of labels  $|\mathcal{Y}| = K$ , *i.e.*, the number of category names.

**Language-free Visual Models.** Most existing visual models are language-free, as language is often not considered in training, *e.g.*, supervised and self-supervised methods. Such models can not be directly used for zero-shot transfer, and model adaptation is often enabled by adding additional weights. For each image  $\mathbf{x}$ , an image encoder  $f_\theta$  parameterized by  $\theta$  first represents  $\mathbf{x}$  as a visual feature vector  $\mathbf{h} \in \mathbb{R}^{D \times 1}$ :  $\mathbf{h} = f_\theta(\mathbf{x})$ . One randomly initialized linear projection layer with  $\mathbf{W}_m \in \mathbb{R}^{D \times K}$  (we absorb the bias  $b$  in  $\mathbf{W}_m$  for simplicity) is used as the classifier; see Figure 5 (a).

**Language-augmented Visual Models.** Recent works [66, 35] that learn visual models with language supervision often employ a two-encoder architecture. Besides the image encoder model  $f_\theta$ , a text encoder  $f_\phi(\mathbf{t})$  parameterized by  $\phi$  is also used to encode text  $t$ . Additional projection layers  $\mathbf{W}_v$  and  $\mathbf{W}_t$  are introduced for image and language features, embedding them into a joint space with dimension  $P$ , with projected features as  $\mathbf{u}$  and  $\mathbf{v}$  respectively. Note that lowercase  $\mathbf{u}$  and  $\mathbf{v}$  are single feature vectors while  $\mathbf{U}$  and  $\mathbf{V}$  are a batch with multiple feature vectors. As in Figure 5 (b), zero-shot learning can be directly performed in this space: the mean text feature  $\mathbf{v}$  is first obtained for each category, by averaging text features of the category name in different language prompts. The image is predicted as the category yielding the highest similarity  $\mathbf{u}^\top \mathbf{v}$ .

- *Random initialized Adaptation.* In the original CLIP paper [66], one randomly initialized linear projection layer  $\mathbf{W} \in \mathbb{R}^{P \times K}$  (similarly, we absorb the bias  $b$  in  $\mathbf{W}$  for simplicity) is added on the pre-retained visual projection, which is shown as the two-projection method in Figure 5 (c).
- *Language-initialized Adaptation.* We argue that the full capacity of language-augmented visual models is not leveraged in [66]. The power of pre-trained language encoder and text inputs must play a vital role in model adaptation. Hence, we propose two language-initialized adaption methods, each of which is ensured as a fair comparison variant for language-augmented and language-free models, respectively. (*i*) *Two-Projection*. For the linear head  $\mathbf{W} \in \mathbb{R}^{P \times K}$  added on the projection space, we initialize  $\mathbf{W}$  with  $\mathbf{V}$  (bias terms are initialized as zeros), as shown in Figure 5 (d). In this way, the visual and text heads are separated. Note that the language-initialized Two-Projection

scheme is also basically equivalent to Figure 5 (b) in zero-shot settings. Please see Appendix Section E for discussions. (ii) *One-Projection*. To fairly compare with language-free model adaptation in Figure 5 (a), one linear projection head should directly be added on the backbone (before the visual projection) to ensure that the same number of trainable parameters are updated. Therefore, we propose to initialize  $\mathbf{W}_m \in \mathbb{R}^{D \times K}$  in this case with the multiplication result of two linear matrices  $\mathbf{W}_v \mathbf{V}$ , as shown in Figure 5 (e).

**Discussion.** We highly recommend the proposed language-initialized methods as the standard to adapt language-augmented visual models for two reasons: (i) This simple method yields surprisingly superior empirical performance, as demonstrated in our experiments. (ii) It provides an effective mechanism to leverage the external knowledge that is collected for a downstream task in our benchmark. Specifically, the knowledge can be concatenated with the original language prompt (with a simple “;” in our experiments), then encoded into contextualized text features. When multiple knowledge items exist (*e.g.*, the case of GPT-3) for each concept, we concatenate one of its prompts and one of its knowledge items, and get the encoded text embedding of the concatenated sequence via the language encoder. This is performed for all the combinations between all prompts and knowledge items for this concept, then the averaged embedding is computed to represent the concept. In contrast, random initialization would ignore this knowledge source. The language-initialized method can serve as a strong baseline to encourage more effective knowledge-augmented adaptation methods.

In OD, GLIP is a language-augmented detector, whose overall architecture can be simply considered as adding a cross-modal module over the CLIP-like dual-encoder. In GLIP [47], its linear probing has been implemented via updating  $\mathbf{W}_v$  and  $\mathbf{W}_t$ . A prompt-tuning strategy was proposed, by initializing the language input of the cross-modal module as  $\mathbf{V}$ , and only updating  $\mathbf{V}$  during adaptation. This is similar to our language-initialized strategy.

## 5 Empirical Results and Findings

We present the experimental results with our benchmark to illustrate two points. Q1: The importance of language in visual model transfer in the adaptation stage. Q2: We present three playgrounds that our benchmark can help to cultivate research in, including sample-efficiency, parameter-efficiency and external knowledge for visual transfer. We also present novel empirical findings.

### 5.1 The Role of Language for Vision

**Effectiveness of Language-initialized Adaptation Methods.** In Table 3, we compare the effectiveness of the proposed language-initialization methods with the checkpoint CLIP ViT-B32. The one-projection scheme is consistently better than two-projection scheme in all settings (though the gain is minor). This is because the former often has less parameters than the latter, as  $D = 768 > P = 512$ . To ensure fair comparisons with the random initialization of linear head in CLIP [66] (*i.e.*, # trainable parameters is the same), in the ensuing experiments, we consider the two-projection language-initialization scheme as the default, unless the one-projection scheme is specified.

As shown in Fig. 6, under both linear probe (LP) and fine-tuning (FT) settings, language-based initialization significantly outperforms random initialization. Notably, we show that even with very few shots (*e.g.*, 2-shots), both our LP and FT is able to outperform the zero-shot CLIP. This is contradictory to the finding in the original CLIP paper [66], where zero-shot outperforms linear probing in the fewer shot (less than 4) settings. With the proposed language-init method, one can ensure that few-shot performance is always better than zero-shot, as we essentially reduce to zero-shot when zero iteration is updated in our language-init method. Moreover, we also find that with random initialization, FT performs significantly worse than LP under few-shot settings. However, with language-init, FT starts to outperform LP with more than 20 shots. Both findings demonstrate the proposed language-based initialization is consistently effective, suggesting that it is an important

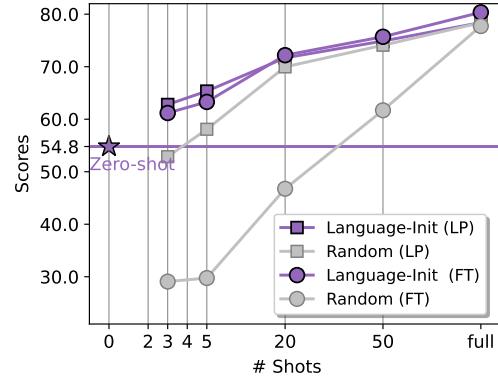


Figure 6: Comparison of random- and language-initialized adaptation.

Checkpoint	Settings		20 IC datasets		
	Adaptation	Initialization	Zero-shot <sup>†</sup>	Few-shot (5, 20, 50)	Full
<b>Industry Track (No pre-train data scale limit)</b>					
CLIP (ViT-B32)	LP	Random-2P		58.09 ± 2.80, 69.97 ± 1.30, 74.09 ± 0.69	78.38
	LP	Language-2P		65.35 ± 1.24, 71.69 ± 0.93, 74.89 ± 0.79	78.40
	LP	Language-1P	56.64	65.88 ± 0.79, 72.05 ± 0.85, 75.08 ± 0.73	78.96
	FT	Random-2P		29.75 ± 6.64, 46.76 ± 11.9, 61.70 ± 9.97	77.77
	FT	Language-2P		63.29 ± 3.18, 72.19 ± 1.31, 75.70 ± 1.14	80.35
Supervised (ViT-B32)	LP	Random-1P	-	56.00 ± 2.67, 67.23 ± 1.66, 71.35 ± 1.17	75.29
	FT	Random-1P		58.55 ± 2.58, 71.27 ± 1.25, 75.36 ± 1.42	80.39
<b>Academic Track (Pre-trained on large established public datasets)</b>					
UniCL (Swin-Tiny)	LP	Language-2P	27.15	54.31 ± 4.15, 66.42 ± 2.08, 70.49 ± 1.01	74.75
	FT	Language-2P		44.75 ± 5.42, 56.53 ± 5.37, 67.90 ± 5.31	78.48
K-LITE (Swin-Tiny)	LP	Language-2P	33.44	55.06 ± 2.36, 66.26 ± 1.56, 70.16 ± 1.09	74.47
	FT	Language-2P		48.41 ± 2.84, 58.06 ± 4.30, 71.66 ± 2.02	78.05

Table 3: Averaged results on 20 IC datasets using linear probing (LP) and fine-tuning (FT). Random-1P, Random-2P, Language-1P and Language-2P indicates the initialization method in Figure 5 (a), (c), (e) and (d), respectively. <sup>†</sup> Note that one zero-shot result is reported for each model checkpoint using the method in Figure 5 (b), which is independent from adaptation/initialization methods.

Checkpoint	Adaptation	35 OD datasets		
		Zero-shot	Few-shot (1, 3, 5, 10)	Full
GLIP (Swin-Tiny)	Prompt		29.7 ± 0.4, 36.5 ± 0.6, 39.0 ± 1.1, 41.8 ± 1.2	54.4
	LP	19.7	22.2 ± 0.1, 24.4 ± 0.2, 25.1 ± 0.2, 25.6 ± 0.6	35.2
	FT		32.2 ± 0.7, 39.2 ± 0.3, 42.5 ± 0.9, 49.1 ± 0.6	63.2
DyHead (Swin-Tiny)	LP	-	15.2 ± 0.6, 19.2 ± 0.9, 19.8 ± 1.0, 20.6 ± 1.1	31.4
	FT		25.6 ± 0.4, 37.1 ± 0.5, 40.1 ± 1.5, 44.6 ± 0.7	63.9

Table 4: Averaged results on 35 OD datasets.

technique, and should be the standard adaptation method for language-augmented visual models like CLIP. Further, the correct adaptation methods for language-augmented visual models should leverage both the pre-trained visual and text encoder. It is not sufficient to solely transfer from the visual encoder, pre-trained language encoder plays an important role in task transfer.

**The Competition of Pre-trained Models: Language-free vs Language-augmented.** We summarize the transfer performance of pre-trained models for IC in Table 3 and OD in Table 4. For IC, we also compare CLIP against other language-free visual models including MoCov3, MAE, ViT, DeiT in Appendix. We see that the language-augmented model (CLIP) outperforms language-free model (Supervised ViT) in the limited data settings. The gap is closed when more training examples are observed (*e.g.*, , 50-shot and full-shot). This is probably because the pre-training power is gradually dominated by larger-scale downstream training. Further, language-augmented models are able to perform zero-shot task transfer, while traditional language-free models cannot. Similar conclusions can be drawn for OD in Table 4. Hence, we recommend the use of language-augmented visual models for task-level transfer.

## 5.2 Playground I: Sample Efficiency

We explore sample efficiency in Fig. 7 (a) for IC. First, we find that CLIP consistently outperforms supervised ViT (Sup-ViT), yielding a significant 5~10% gain in the 5-shot settings. This suggests that CLIP is more sample-efficient than supervised ViT . Furthermore, we find that fine-tuning CLIP yields better performance than linear probing in >20-shot settings, while being worse in the 5-shot setting. This is a bit surprising, as it is contradictory to the common convention that fine-tuning is always better than linear probing. We hypothesize this is because fine-tuning tends to over-fit in the scenarios with a large number of trainable parameters and a small number of training samples. Overall, it suggests that fine-tuning CLIP potentially has a better sample efficiency than linear probing, and a better adaptation strategy on fewer-shot settings can be explored in the future. For supervised ViT, FT is always better than LP, the performance gap becomes larger when more samples are used. In 5-shot settings, the gap is minor, which is similar to observations made for supervised CNNs [91, 37, 38]. To compare pre-trained models, we suggest to report the evaluation results on the entire spectrum of sample-efficiency to fully study the behaviors of a pre-trained model. If compute resource is limited, zero-shot or few-shot evaluation can be used as a quick assessment.

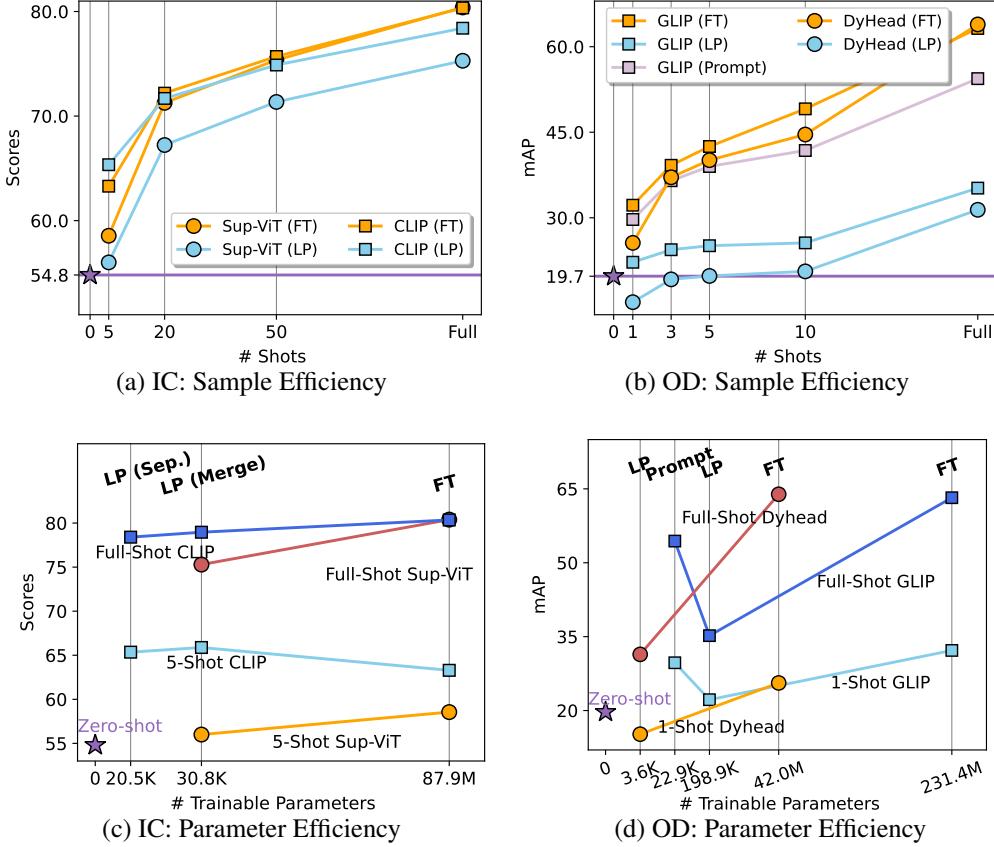


Figure 7: Adaptation efficiency considerations. For IC, comparison of adaptation efficiency between CLIP and supervised ViT (Sup-ViT). For OD, comparison of adaptation efficiency between GLIP and DyHead.

We explore sample efficiency for OD in Fig. 7 (b). The conclusion is similar to IC in that the language-augmented visual model (GLIP) is more sample-efficient than the language-free visual model (DyHead), when the models are adapted using either LP or FT settings. The performance gap is large in the fewer-shot settings and is small in the full-shot settings. The difference is that FT consistently outperforms LP in all settings for OD. This is probably because there are a lot of boxes (training instances) per image in OD, which makes OD less likely to over-fit compared to IC.

### 5.3 Playground II: Parameter Efficiency

We study the parameter efficiency in Fig. 7 (c) for IC. For CLIP, we experiment with two different settings of linear probing on whether to merge the last two linear projection layers ( $\mathbf{W}_v$  and  $\mathbf{V}$  in Fig. 5). Merging these two layers in CLIP allows  $1.5\times$  trainable parameters in the linear probe classifier as keeping them separated. First, it shows a trend that a larger number of trainable parameters leads to better performance, as demonstrated by three curves/scenarios: full-shot CLIP, full-shot Sup-ViT and 5-shot Sup-ViT. This also verifies that LP and FT provide the lower bound and upper bound, respectively, in terms of both #parameter and performance. Most existing parameter-efficient adaptation methods play a trade-off game. However, in the scenario of 5-shot CLIP, we do notice a slight drop in performance when we further increase the number of trainable parameters to full-model fine-tuning. It suggests that the scenario of adapting language-augmented visual models for data-limited settings is a more meaningful playground to explore the line of research in parameter-efficient adaptation methods, as the best performance may require an optimal number of trainable parameters, which has been less explored.

We study the parameter efficiency in Fig. 7 (d) for OD. The overall trend is similar in that better performance comes with more parameters. It turns out that prompt tuning an language-augmented OD model is an effective parameter-efficient approach. For example, prompting is better than linear

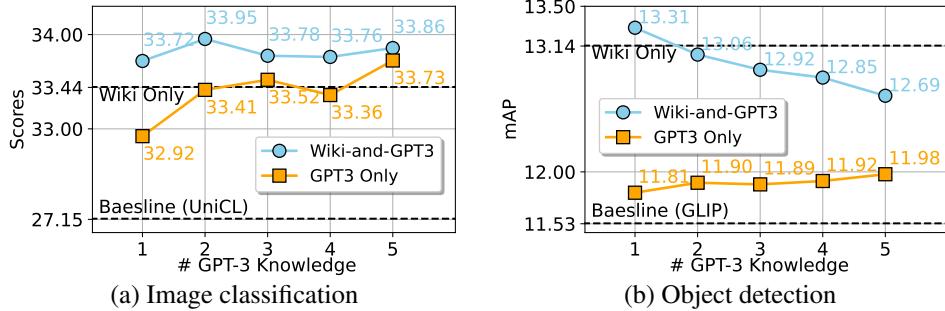


Figure 8: Zero-shot task transfer with various external knowledge sources in the evaluation stage. In (a) and (b), a varying number of generated GPT-3 knowledge sequences is utilized for inference, and “Wiki-and-GPT3” indicates both Wiktionary and GPT3 knowledge are used simultaneously. The bottom table summarizes the prediction result for each knowledge source.

probing in GLIP. Further, prompt tuning GLIP outperforms fine-tuning DyHead in the 1-shot setting, where the former has less than 0.1% parameters of the latter.

#### 5.4 Playground III: The Benefit of External Knowledge for Vision

We investigate the effectiveness of external knowledge in Fig. 8, measured by zero-shot task transfer performance. The model K-LITE is evaluated, as its pre-training is knowledge-augmented. We find that leveraging external knowledge improves upon the knowledge-free pre-training counterparts (UniCL and GLIP). For example, UniCL is improved from 27.15 to 29.92~33.93, and GLIP-A is improved from 11.53 to 11.70~13.30. Further, for GPT3 knowledge, a larger number of generated knowledge items often leads to higher performance. When combining GPT3 knowledge with Wiktionary knowledge, we see a further performance boost. With an increasing number of GPT3 knowledge items, the gain is consistently improved for IC, but not for OD. In Table 3, we study the role of knowledge for task transfer in model adaptation. Initializing the linear head using features encoded with knowledge is an effective way to leverage the collected knowledge sources, especially for the fewer-shot settings.

One may wonder if the collected knowledge in ELEVATER benchmark can also benefit knowledge-free pre-trained models such as CLIP during model adaptation? We confirm its effectiveness in Appendix. In zero-shot transfer, external knowledge improves the baseline on four datasets. In few- and full-shot transfer, one may selectively choose whether to update the model using external knowledge, by observing the best performance in the auto-tuning stage. This selective strategy with the availability of external knowledge demonstrates a consistent improvement (or tie) on 15+ out of 20 datasets.

## 6 Conclusions

We have presented ELEVATER, a platform to evaluate the recently emerging language-augmented visual models for task-level transfer. It consists of 20 image classification datasets and 35 object detection datasets. All of them are collected from public domains, and are enriched with various external knowledge sources to enhance the language modality. We have developed open-source toolkits with an auto hyper-parameter pipeline and novel language-initialized adaptation methods to ensure easy utilization and fair comparisons. Strong baseline results are produced from the toolkit to cultivate research in a variety of topics, *e.g.*, more transferable language-augmented visual models, advanced model adaption methods (sample-efficiency and parameter-efficiency), and external knowledge for task-level transfer. The question of how to design general-purpose task-level transferable visual models remains largely unanswered. Given benchmarks and toolkits we have developed from the perspective of language-augmented visual models, we believe that ELEVATER can provide fertile soil for addressing this challenge.

## Acknowledgments

The authors gratefully acknowledge Haotian Zhang for building the ODinW leaderboard on Eval AI, Pengcheng He for helpful discussions to have a separate track dedicated for users from academia, Baolin Peng and Zhengyuan Yang for the inspirations of GPT3 to generate knowledge for dialogue and OK-VQA tasks, Bo Li for insights on the topic of domain generalization, Zhuowen Tu for the inspirations to make benchmark scope wider to measure all pre-trained vision models, Ce Liu for encouragement and suggestions to compare the benchmark with well-established vision datasets such as ImageNet and COCO. The benchmark depends on publicly available datasets; we acknowledge all the original authors who made their datasets public. Please follow the original license of each dataset and keep this benchmark for academic purposes. This work was supported in part by NSF CAREER IIS-2150012, ETRI, and the Wisconsin Alumni Research Foundation.

## References

- [1] FER 2013: Kaggle challenges in representation learning facial expression recognition. <https://www.kaggle.com/>.
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015.
- [3] Noor Awad, Neeratyoy Mallik, and Frank Hutter. Dehb: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. *arXiv preprint arXiv:2105.09821*, 2021.
- [4] Sven Bambach, Stefan Lee, David Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *ICCV*, 2015.
- [5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.
- [7] Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. Iglue: A benchmark for transfer learning across modalities, tasks, and languages. *arXiv preprint arXiv:2201.11732*, 2022.
- [8] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.
- [9] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.
- [10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *ICCV*, 2021.
- [11] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.
- [12] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.
- [13] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, 2021.
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [15] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE signal processing magazine*, 2012.
- [16] 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.

- [17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.
- [18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [19] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010.
- [20] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018.
- [21] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.
- [22] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *CVPR workshop*, 2004.
- [23] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *ITSC*. IEEE, 2013.
- [24] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016.
- [25] 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. *arXiv preprint arXiv:2110.04544*, 2021.
- [26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.
- [27] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.
- [28] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.
- [29] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.
- [30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.
- [31] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.
- [32] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.
- [33] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021.
- [34] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [35] Chao Jia, Yafei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021.
- [36] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.
- [37] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013.

- [38] Nikolaos Karianakis, Zicheng Liu, Yinpeng Chen, and Stefano Soatto. Reinforced temporal attention and split-rate transfer for depth-based person re-identification. In *ECCV*, 2018.
- [39] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *NeurIPS*, 2020.
- [40] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.
- [41] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [42] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.
- [43] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 2013.
- [44] Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Learning visual N-grams from web data. In *ICCV*, 2017.
- [45] Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *arXiv preprint arXiv:2107.07651*, 2021.
- [46] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. Value: A multi-task benchmark for video-and-language understanding evaluation. *arXiv preprint arXiv:2106.04632*, 2021.
- [47] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *CVPR*, 2022.
- [48] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
- [49] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.
- [50] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.
- [51] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.
- [52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021.
- [53] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.
- [54] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *CVPR*, 2019.
- [55] Christian M Meyer and Iryna Gurevych. *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. 2012.
- [56] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.
- [57] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets language-image pre-training. *arXiv preprint arXiv:2112.12750*, 2021.
- [58] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [59] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008.
- [60] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

- [61] Genevieve Patterson and James Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.
- [62] Malte Pedersen, Joakim Bruslund Haurum, Rikke Gade, Thomas B. Moeslund, and Niels Madsen. Detection of marine animals in a new underwater dataset with varying visibility. In *CVPR Workshops*, 2019.
- [63] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.
- [64] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. KILT: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*, 2020.
- [65] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.
- [66] 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 *ICML*, 2021.
- [67] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.
- [68] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.
- [69] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.
- [70] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.
- [71] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, Kurt Keutzer, Trevor Darrell, and Jianfeng Gao. K-LITE: Learning transferable visual models with external knowledge. *arXiv preprint*, 2022.
- [72] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. PlantDoc: A dataset for visual plant disease detection, 2019.
- [73] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [74] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011.
- [75] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-adapter: Parameter-efficient transfer learning for vision-and-language tasks. *CVPR*, 2022.
- [76] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [77] Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. *arXiv preprint arXiv:2010.06775*, 2020.
- [78] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016.
- [79] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.
- [80] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018.

- [81] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *MICCAI*, 2018.
- [82] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. 2011.
- [83] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *NeurIPS*, 2019.
- [84] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- [85] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.
- [86] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7289–7298, 2019.
- [87] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *PAMI*, 2018.
- [88] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Lu Yuan, Ce Liu, and Jianfeng Gao. UniCL: unified contrastive learning in image-text-label space. *CVPR*, 2022.
- [89] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.
- [90] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. CPT: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021.
- [91] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *NIPS*, 2014.
- [92] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.
- [93] 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.
- [94] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.
- [95] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. RegionCLIP: Region-based language-image pretraining. *CVPR*, 2022.
- [96] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. *arXiv preprint arXiv:2203.05557*, 2022.
- [97] Wangchunshu Zhou, Yan Zeng, Shizhe Diao, and Xinsong Zhang. Vlue: A multi-task multi-dimension benchmark for evaluating vision-language pre-training. In *International Conference on Machine Learning*, pages 27395–27411. PMLR, 2022.

## Checklist

1. For all authors...
  - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[Yes]**
  - (b) Did you describe the limitations of your work? **[Yes]**
  - (c) Did you discuss any potential negative societal impacts of your work? **[Yes]** See Appendix Section A
  - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[Yes]**
2. If you are including theoretical results...
  - (a) Did you state the full set of assumptions of all theoretical results? **[N/A]**
  - (b) Did you include complete proofs of all theoretical results? **[N/A]**
3. If you ran experiments (e.g. for benchmarks)...
  - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[Yes]**
  - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[Yes]**
  - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[Yes]**
  - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[Yes]**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
  - (a) If your work uses existing assets, did you cite the creators? **[Yes]**
  - (b) Did you mention the license of the assets? **[Yes]**, in Appendix Section C
  - (c) Did you include any new assets either in the supplemental material or as a URL? **[Yes]**
  - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[Yes]**, in Appendix Section C
  - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[Yes]**, in Appendix Section C
5. If you used crowdsourcing or conducted research with human subjects...
  - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[N/A]**
  - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[N/A]**
  - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[N/A]**

---

## Supplementary Material for “ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models”

---

This appendix is organized as follows.

- In Section A (referred by CheckList), we discuss the societal impact.
- In Section B.2 (referred by Section 2), we discuss the related work in pre-trained language models in NLP.
- In Section C (referred by Section 3.1), we summarize the datasets statistics and license used in our benchmark suite. We also describe how to obtain external knowledge from GPT-3, and construct language prompts.
- In Section D.1 (referred by Section 4), we introduce more details of our toolkits, including the automatic hyper-parameter tuning pipeline and implementation details.
- In Section E (referred by Section 4), we discuss the gap between language-image model pre-training and adaptation.
- In Section F (referred by Section 5.1), we provide performance of comparison different vision pre-trained models.
- In Section G (referred by Section 5.4), we provide empirical evidence that external knowledge improves CLIP adaptation.

## A Societal Impact

We do not anticipate a specific negative impact, but, as with any Machine Learning method, we recommend to exercise caution. The existing knowledge bases such as Word-Net and Wiktionary are the results of crowd-sourcing various human knowledge or commonsense into a centered place. ELEVATER provides evidence to leverage such knowledge bases for AI research. It encourages the community to contribute more to improve the coverage and quality of knowledge items, which will further benefit AI research. We also leverage GPT3 to generate knowledge, which is stored as a part of benchmark for public academic use. The related societal impact on the usage of AI-generated content may apply to our work.

## B Our Position

### B.1 Computer Vision in the Wild

In this paper, we advocate our perspective on “**Computer Vision in the Wild (CVinW)**”, whose ultimate goal is to develop a transferable foundation model/system that can *effortlessly* adapt to a *large range of visual tasks in the wild*. We further illustrate two key factors as follows.

**Factor I: The Task Transfer Cost is Low.** One major advantage of pre-trained/foundation models is the promise that they can transfer to downstream tasks *effortlessly* (or in an inexpensive manner). It means that model adaptation efficiency is an important factor to measure the performance of the pre-trained models. To concretely illustrate the notion of inexpensive adaptation, we provide a 2D chart on the model adaptation cost in Figure 4. The cost is considered in two orthogonal dimensions: sample-efficiency and parameter-efficiency. One may interpolate and make combinations in the 2D space, to get different model adaptation methods with different cost. This is design philosophy behind our comprehensive evaluation metrics. Two playgrounds with different efficiency considerations presented in the main paper are simplified settings to study model performance. As a north star, one foundation could with fixed weights should zero-shot transfer well on many downstream tasks, the most inexpensive regime in the bottom-left corner of Figure 4.

**Factor II: The Task Transfer Scenarios are Broad.** We illustrate and compare the settings of CVinW using a 2D chart in Figure 2. It consists of two dimensions: the input visual content and output concept prediction. For the example provided in the standard setting, the natural image with concept “person, sheep, dog” is presented. We divide the the 2D chart into four quadrants

1. **The Standard Close-Set Setting.** The bottom-left quadrant is the standard setting, where most existing visual recognition lie in, training and evaluation are consistent in both their visual input distributions and output category sets. For example, only natural images with concept “person, sheep, dog” are presented in training and evaluation.
2. **Open-Set/Vocabulary/World Setting.** In the top-left quadrant, the recognition of new concepts is enabled, while the visual input distributions of training and evaluation are in the same domain. This research problem is usually tackled by traditional class-level zero-shot transfer, or some experimental settings in the open-set recognition. For example, natural images with concepts “person, sheep, dog” are presented in training, but natural images with concepts “border collie, running, white shirt” are presented in evaluation. Though the testing concepts are closely to training concepts, but they have not been observed by the models in training.
3. **The Domain Shift Setting.** In the bottom-right quadrant, the input image distributions are shifted between training and evaluation sets, while the output category sets are the same. This research problem is often tackled in the area of domain adaptation and out-of-distribution. For example, natural images with concepts “person, sheep, dog” are presented in training, but thermal images are presented in evaluation, though the concepts have been observed in training.
4. **Computer Vision in the Wild Setting.** In the top-right quadrant, the strong generalization ability to both new concepts and new visual distributions is required. Therefore, the model can perform well on new tasks of any customized set of concepts in any visual domains. This is a setting we advocate for computer vision in the wild, where any new downstream tasks can appear in this quadrant, and it requires models with a strong task-level visual transfer ability.

## B.2 Related Works in NLP: Benchmarks, Adaptation, and Knowledge

With a focused scope, our benchmark evaluates language-image models on two core CV problems: IC and OD. Though language-image models can also be deployed and evaluated in other scenarios, including joint visual-text evaluation [97, 7] (e.g., visual question answering [2, 54], video-and-language understanding [46]) and the scenario of improving language encoders with vision [77]. Our benchmark is complementary to them in its focus on evaluating vision encoders.

Our work takes major inspiration from the development of pre-trained language models in natural language processing (NLP) in several aspects: *(i) Benchmarks.* Platforms with a suite of small datasets such as GLUE [84]/SuperGLUE [83] have been extensively used to evaluate the general language understanding ability of pre-trained models [16]. Recently, there is a trend in NLP to develop task-agnostic models such as the GPT family [6] that demonstrate task-level transfer learning ability, enabling zero-shot and few-shot transfer to downstream datasets. The success in NLP encourages us to build a generic benchmark to measure the similar transferability for visual models. *(ii) Efficient adaptation.* The democratization of large pre-trained models for efficient adaptation in downstream applications is an important topic in practice. Many algorithms have been developed for various efficiency considerations, including adapters [32] and prompt tuning [48, 51]. In particular, natural language prompting is the method of reformatting NLP tasks in the format of a natural language response to natural language input, has attracted attentions in zero-shot and few-shot learning in NLP [68]. It has inspired a few recent works for language-augmented visual models [96, 75, 90, 25]. Our benchmark can serve as a comprehensive playground to quantify the progress in the emerging field of visual model adaptation. We also propose to use external knowledge for prompt engineering, and a novel language/knowledge-initialized model adaptation method as a strong baseline. *(iii) Knowledge.* Knowledge-intensive tasks [54, 64] — those where a human can only be expected to perform the task with access to a knowledge source such as Wikipedia — are challenging for even cutting edge NLP and vision-and-language models, as it is infeasible to train large models to memorize everything. KILT [64] is a benchmark that contains a suite of tasks/datasets for evaluating and analyzing knowledge-intensive NLP models. Similarly, we also add various external knowledge sources in each downstream dataset for our vision benchmark.

## C Benchmark Suite

### C.1 Detailed Dataset Statistics

In Table 5 and Table 6, we list the basic statistics of 20 image classification datasets and 35 object detection datasets in the benchmark.

The benchmark may inherit data biases from the public datasets we have considered, both in the images and the annotations. Such biases might be reflected in the predictions of the systems trained on these data. Users should not completely rely on such systems for making real-world decisions.

Dataset	#Concepts	Train size	Test size	Evaluation metric	Source link
Hateful Memes [39]	2	8,500	500	ROC AUC	Facebook
PatchCamelyon [81]	2	262,144	32,768	Accuracy	Tensorflow
Rendered-SST2 [66]	2	6,920	1,821	Accuracy	OpenAI
KITTI Distance [23]	4	6,347	711	Accuracy	KITTI website
FER 2013 [1]	7	28,709	3,589	Accuracy	Kaggle fer2013
CIFAR-10 [41]	10	50,000	10,000	Accuracy	Tensorflow
EuroSAT [31]	10	5,000	5,000	Accuracy	Tensorflow
MNIST [15]	10	60,000	10,000	Accuracy	Tensorflow
VOC 2007 Classification [19]	20	2,501	4,952	11-point mAP	VOC 2007
Oxford-IIIT Pets [60]	37	3,680	3,669	Mean-per-class	Tensorflow
GTSRB [74]	43	26,640	12,630	Accuracy	GTSRB website
Resisc-45 [11]	45	3,150	25,200	Accuracy	Tensorflow
Describable Textures [12]	47	1,880	1,880	Accuracy	Tensorflow
CIFAR-100 [41]	100	50,000	10,000	Accuracy	Tensorflow
FGVC Aircraft (variants) [53]	100	3,334	3,333	Mean-per-class	FGVC website
Food-101 [5]	101	75,750	25,250	Accuracy	Tensorflow
Caltech-101 [22]	102	3,060	6,084	Mean-per-class	Tensorflow
Oxford Flowers 102 [59]	102	1,020	6,149	Mean-per-class	Tensorflow
Stanford Cars [40]	196	8,144	8,041	Accuracy	Tensorflow
Country-211 [66]	211	31,650	21,100	Accuracy	OpenAI
Total	1151	638429	192677	-	-

Table 5: Statistics of 20 datasets used in image classification.

### C.2 License

As per the original authors, the licenses of each dataset include CC BY-NC-SA 3.0<sup>3</sup>, CC BY-NC-SA 4.0<sup>4</sup>, CC BY 4.0<sup>5</sup>, ODbL v1.0<sup>6</sup>, MIT<sup>7</sup>, CC0 1.0<sup>8</sup>. Some datasets have published dedicated usage agreements: Hateful Memes<sup>9</sup>. All datasets allow the usage for research purposes. The images used in the datasets are from Internet, on non-offensive topics. The annotations in the datasets do not contain personally identifiable information.

For external knowledge collected on ELEVATER, we suggest the users to follow the corresponding licenses: WordNet<sup>10</sup>, Wiktionary<sup>11</sup>, GPT-3<sup>12</sup>. For the GPT-3 generated knowledge, we have the approval from OpenAI to release it as a part of ELEVATER to encourage future research.

### C.3 Generating GPT-3 Knowledge with In-Context-Learning

Wiktionary and WordNet do not provide a 100% coverage for all downstream concepts. As shown in [71], an incomplete knowledge coverage can lead to deteriorated model performance. In this paper,

<sup>3</sup><https://creativecommons.org/licenses/by-nc-sa/3.0/>

<sup>4</sup><https://creativecommons.org/licenses/by-nc-sa/4.0/>

<sup>5</sup><https://creativecommons.org/licenses/by/4.0/>

<sup>6</sup><https://opendatacommons.org/licenses/odbl/1-0/>

<sup>7</sup><https://choosealicense.com/licenses/mit/>

<sup>8</sup><https://creativecommons.org/publicdomain/zero/1.0/>

<sup>9</sup><https://www.drivendata.org/competitions/64/hateful-memes/page/214/>

<sup>10</sup><https://wordnet.princeton.edu/license-and-commercial-use>

<sup>11</sup>[https://en.wiktionary.org/wiki/Wiktionary:Main\\_Page](https://en.wiktionary.org/wiki/Wiktionary:Main_Page)

<sup>12</sup><https://openai.com/api/policies/sharing-publication/>

Dataset	#Concepts	#Image		#Annotated Regions		Source link
		Train	Test	Train	Test	
CottontailRabbits	1	1980	10	2070	11	Roboflow
EgoHands(generic) [4]	1	3840	480	12015	1514	Roboflow
MountainDewCommercial	1	17	1	453	32	Roboflow
Packages	1	19	3	31	5	Roboflow
Raccoon	1	150	17	164	20	Roboflow
WildfireSmoke	1	516	74	516	74	Roboflow
Pistols	1	2377	297	2728	358	Roboflow
Pothole	1	465	67	1256	154	Roboflow
MaskWearing	2	105	15	696	96	Roboflow
NorthAmericaMushrooms	2	41	5	67	9	Roboflow
OxfordPets(species) [60]	2	2523	358	2527	358	Roboflow
PKLot640	2	8691	1242	497856	70684	Roboflow
ThermalCheetah	2	90	14	152	31	Roboflow
ThermalDogsAndPeople	2	142	20	181	27	Roboflow
BCCD	3	255	36	3450	471	Roboflow
HardHatWorkers	3	5069	1766	19455	6808	Roboflow
ShellfishOpenImages	3	407	58	859	116	Roboflow
EgoHands(specific)	4	3840	480	12015	1514	Roboflow
AerialMaritimeDrone(large)	5	52	7	873	78	Roboflow
AerialMaritimeDrone(tiled)	5	371	32	1237	98	Roboflow
VehiclesOpenImages	5	878	126	1676	258	Roboflow
BrackishUnderwater [62]	6	11739	1468	28518	3466	Roboflow
Dice	6	576	71	1439	225	Roboflow
Aquarium	7	448	63	3324	584	Roboflow
DroneControl	8	32688	4675	32734	4694	Roboflow
WebsiteScreenshots	8	1688	242	76820	10656	Roboflow
SelfDrivingCar	11	24000	3000	156730	19598	Roboflow
ChessPieces	13	202	29	2108	376	Roboflow
UnoCards	15	6295	899	18885	2697	Roboflow
PascalVOC [19]	20	13690	3422	31356	7835	Roboflow
AmericanSignLanguageLetters	26	1512	72	1512	72	Roboflow
Plantdoc [72]	30	2128	239	7629	454	Roboflow
BoggleBoards	36	285	35	5727	647	Roboflow
OxfordPets(breed)	37	2437	345	2441	345	Roboflow
OpenPoetryVision	43	2798	402	8392	1198	Roboflow
Total	314	132314	20070	937892	135563	-

Table 6: Statistics of 35 datasets used in object detection. Box mAP is used as the evaluation metric. Datasets are downloaded from Roboflow. For the datasets without a citation, we refer to Roboflow links for the original sources.

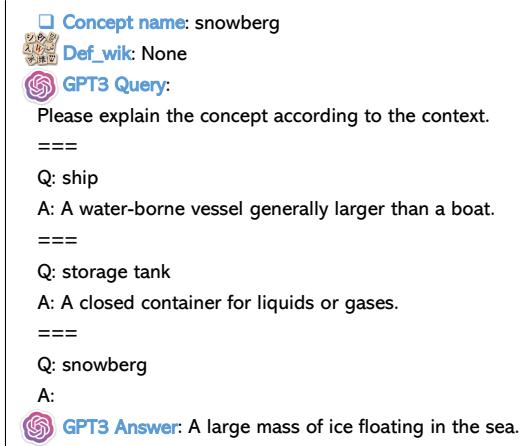


Figure 9: Example of generating external knowledge with GPT3 using in-context learning even when Wiktionary knowledge is missing.

we show that GPT3 can be used for generating additional external knowledge and providing a full coverage for downstream concepts.

We use in-context-learning to prompt GPT-3. As an input to GPT3, we start by asking “Please explain the concept according to the context”. In addition, we provide multiple concept-explaining Q (concept)-A (explanation) pairs. Each pair of the concept and explanation are sampled from the

<b>Dataset name</b>	Oxford Flowers 102
<b>Category names</b>	['pink primrose', ...]
<b>Templates</b>	[ 'a photo of a {}, a type of flower.', ]
<b>Knowledge</b>	[{"classname": "pink primrose", "def_wiki": "A flowering plant of the genus Primula.", "path_wn": "", "def_wn": "", "gpt3": [" A plant of the genus Primula, having a pink flower.", " Primula vulgaris, a plant of the primrose family, with pink flowers.", " A flowering plant of the genus Primula.", " A primrose, Primula × polyantha, with pink flowers.", " A plant of the genus Primula, of the family Primulaceae, having showy flowers of various colors."], ... ]
<b>Prompt</b>	• 'a photo of a pink primrose, a type of flower.'
<b>Prompt + Knowledge</b>	<ul style="list-style-type: none"> <li>• 'a photo of a pink primrose, a type of flower ; A flowering plant of the genus Primula.'</li> <li>• 'a photo of a pink primrose, a type of flower ; A plant of the genus Primula, having a pink flower.'</li> <li>• 'a photo of a pink primrose, a type of flower ; Primula vulgaris, a plant of the primrose family, with pink flowers.'</li> <li>• 'a photo of a pink primrose, a type of flower ; A flowering plant of the genus Primula.'</li> <li>• 'a photo of a pink primrose, a type of flower ; A primrose, Primula × polyantha, with pink flowers.'</li> <li>• 'a photo of a pink primrose, a type of flower ; A plant of the genus Primula, of the family Primulaceae, having showy flowers of various colors.'</li> </ul>

Table 7: Examples of prompt construction with and without external knowledge for the concept ‘pink primrose’ on dataset ‘Oxford Flowers 102’.

concepts that have the Wiktionary knowledge available. Finally, we send a different concept to GPT3, and ask for the explanation. In this way, GPT3 is able to generate explanatory descriptions for the concepts even when its Wiktionary knowledge is missing. For example, as shown in Fig. 9, there is no Wiktionary knowledge available for “snowberg”, while “ship” and “storage tank” have their corresponding Wiktionary explanations. By providing the concept-explanation pairs of “ship” and “storage tank”, GPT3 recognizes this as a concept explaining task, and when a new concept “snowberg” is given, it explains the concept *without* the need for its external knowledge. By randomly sampling different Q-A groups from the concepts *with* Wiktionary knowledge, we are able to generate a diverse set of GPT3 responses.

#### C.4 Prompting and Knowledge

For each visual recognition dataset, there comes naturally with a set of category names. A specific set of natural language templates are created for each dataset, following [66]. In our toolkit (`vision_benchmark/datasets/prompts.py`), we maintain the mappings from a dataset to its specific category names and template sets, respectively. External knowledge for each dataset is maintained at the folder `vision_benchmark/resources/knowledge`. To construct the language prompt, we suggest the following steps:

1. For a given dataset, choose one category from a set of its category names
2. Choose one template from a set of pre-defined dataset-specific language templates.
3. Fill in the category name into the template, which yields the constructed language prompt for this category.
4. (Optional) If external knowledge is preferred to add into the prompt construction, please select a knowledge source with non-empty value, and concatenate the knowledge sequence after the text sequence in Step 3, separated by “;”.

In Table 7, we provide examples to construct prompts with and without external knowledge, by following the above procedure.

## D Toolkit

Our code is under MIT license.

## D.1 Automatic Hyper-parameter Tuning

**Image Classification.** For a given dataset, we split its training set into training and validation with a ratio 80% vs 20%. At least one training sample per class is ensured for training and validation. Grid search is applied over learning rate  $\eta$  and weight decay  $\alpha$ . In the hyper-parameter search stage, the model is trained with a given configuration  $(\eta, \alpha)$  for 10 epochs, the best hyper-parameter configuration is chosen as the one with the best validation performance along the entire process. After that, a final run is performed for 50 epochs to report the performance on the testing set.

**Object Detection.** A validation set is chosen in the hyper-parameter search stage. We consider validation set size  $(1, 1, 1, 3, full)$  for  $N = 1, 3, 5, 10$ , respectively. For each type of checkpoints (DyHead, GLIP) and each adaption method, we have a set of pre-selected hyper-parameters, *i.e.*, batch size  $|\mathcal{B}|$ , initial learning rate  $\eta_0$  and weight decay  $\alpha$ , as shown in Table 8 in Appendix. They are determined by either empirical rules or simple hyper-parameter tuning. For each setting and each train/val split, we evaluate on the val split after every training epoch to decrease the learning rate in a step-wise manner. More specifically, we use the PyTorch *ReduceLROnPlateau* with patience 3 and factor 0.1 to decrease the learning rate when there is no improvement on val. We terminate the fine-tuning process if we do not see improvements for continuously 9 epochs, return the checkpoint with the best score on val, and report its score on the test split. For each few-shot setting, we random sample the train/val split 3 times, and report the average score and standard deviation on the test split. For each type of checkpoints (DyHead, GLIP) and each adaption method, we have a set of pre-selected hyper-parameters, *i.e.*, batch size  $|\mathcal{B}|$ , initial learning rate  $\eta_0$  and weight decay  $\alpha$ , as shown in Table 8. They are determined by either empirical rules or simple hyper-parameter tuning.

Checkpoint	Settings	35 OD datasets		
		$ \mathcal{B} $	$\eta_0$	$\alpha$
GLIP (Swin-Tiny)	Prompt	4	0.05	0.25
	Linear Probing		0.0001	0.05
	Fine-tuning		0.0001	0.05
DyHead (Swin-Tiny)	Linear Probing	4	0.0001	0.05
	Fine-tuning		0.0001	0.05

Table 8: Pre-selected hyperparameters for OD datasets.

## D.2 Implementation details

**Image Classification.** To make a fair comparison between different methods in image classification, we conduct experiments with FP32 precision. Our preliminary experiments show that on average FP16 and FP32 yields similar zero-shot performance, while FP32 models outperform FP16 ones on 16 out of 20 datasets.

**Object Detection.** For OD, one image could contain multiple classes. We run an algorithm to go over the images in the full training set one by one, and add the image to the  $N$ -shot training set if the image contains some classes that do not have  $N$  images yet. We stop if all classes have at least  $N$  images or we have exhausted the full training set. Thus, the total number of images in the dataset could be between  $N \sim N * K$ , where  $K$  is the number of categories. We will release all the  $N$ -shot samples we used for experiments [47]. For OD full fine-tuning, the common practice is to freeze the bottom two layers of the backbone<sup>13</sup>.

## E Close the Gap between Pre-training and Adaption for CLIP

In Section 4, we have proposed language-initialized adaptation strategy, which consistently improves the linear probing and fine-tuning performance of language-image pre-trained models like CLIP. By initializing the linear head of CLIP model with the embeddings from the language encoder, it allows the model update and prediction of CLIP in few- / full-shot adaptation settings behaving in a similar way as in the zero-shot setting. This, in other words, narrows the gap between the pre-training

<sup>13</sup>[shorturl.at/A0Z13](https://shorturl.at/A0Z13)

CLIP objective and the downstream image classification objective (cross-entropy). In this section, we explore other factors that differs in the pre-training CLIP and downstream CLIP adaptations.

				Caltech101	CIFAR10	CIFAR100	Country2/11	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPots	PatchCamelyon	SST2	REVISCS45	StanfordCars	VOC2007		
BN	$\ell_2$	$\exp(\tau)$	🔥	mean	std																				
✓	✗	1.0	✗	63.3	3.2	88.8	91.3	73.0	16.6	51.8	79.3	52.2	23.1	84.0	60.4	55.8	44.3	60.5	67.3	86.9	61.8	59.2	70.8	56.3	82.4
✗	✓	1.0	✗	61.5	2.0	88.9	90.2	72.2	17.3	48.5	79.5	53.5	21.1	84.2	36.5	55.8	42.0	54.4	67.4	87.7	65.3	56.9	67.1	59.6	82.3
✓	✓	1.0	✗	60.8	2.4	86.0	90.4	70.3	16.6	45.6	71.7	53.9	19.6	83.6	35.9	55.8	41.8	66.5	64.9	85.6	65.6	58.9	65.8	54.6	82.5
✗	✗	1.0	✗	62.7	3.1	88.8	91.2	72.7	17.3	49.9	73.5	53.2	21.9	84.4	37.0	55.8	52.8	52.0	80.7	87.8	64.9	59.3	68.9	59.9	82.0
✓	✗	1.0	✓	65.0	2.8	90.0	91.1	71.4	16.9	57.8	80.0	52.7	26.5	83.4	69.2	55.8	41.6	61.4	79.5	87.3	64.2	59.1	76.3	54.2	82.7
✓	✗	100	✓	58.5	3.9	86.8	90.5	45.4	7.8	47.2	71.8	42.3	20.0	79.4	59.1	54.2	40.1	61.5	58.9	86.0	62.8	59.6	69.0	48.4	79.1

Table 9: Effect of the normalization and temperature with 5-shot finetuning CLIP (ViT/B-32). The linear head is initialized with the proposed language-initialization adaptation strategy. 🔥Trainable  $\tau$ .

## E.1 Visual Feature Normalization

There are two difference in the normalization strategy between the CLIP pre-training and fine-tuning. In CLIP, visual features  $\mathbf{U}$  are normalized per-instance using  $\ell_2$ -norm [66]; while in downstream adaptation, usually a batch normalization (BN) [34] without the learnable affine transformation is used for feature normalization [17, 30]. We compare between these two normalization strategies as well as the setting without feature normalization.

As shown in Table 9 (Row 1-4), using the channel BN yields the best performance. In addition, adding instance-wise  $\ell_2$  normalization does not help improve the performance. This suggests that it is not always beneficial to adopt the objectives / tricks from CLIP, as there are still differences in the training objectives between CLIP and downstream classification, which we discuss in Sec. E.2.

## E.2 Training objective

Although the training objective is aligned between the pre-training and downstream adaptation already with the proposed language-initialization adaptation strategy, there are several factors that may cause a difference in the gradient flow between pre-training and downstream adaptation, which can potentially hurdle the model training.

**The size of Softmax:  $|\mathcal{B}|$  vs  $K$ .** In CLIP, a scaled pairwise cosine similarity is first computed between all image-text pairs, and the bidirectional cross entropy loss is then applied to the computed similarity score. Although the loss function of pre-training CLIP and downstream adaptation can be reduced to the same objective, one key difference is the size of the similarity matrix. For each image, the similarity is computed with *all* text embeddings. In CLIP, it is the number of all text samples in a *large* batch (*e.g.*,  $|\mathcal{B}|=32,768$ ); while in downstream, it is the number of text embeddings of all classes  $K$  (which is typically less than 200). Such disparity can cause a significant change in the pattern of the gradient flow.

**Temperature.** In CLIP, a trainable log-parameterized temperature  $\tau$  controls the range of the logits in the Softmax, which is typically not used in downstream adaptation. Although the temperature parameter does not alter the ranking of its predictions, it modifies the scale of the gradients when backward propagation is performed in downstream adaptation.

**Experiment/Analysis.** Based upon the above analysis, we design experiments to explore the effect of these factors on the gradient flow and the downstream adaptations.

We compare the initialization of the temperature  $\tau$  and whether to keep it frozen during the adaptation in Table 9 (Row 1,5-7). First, setting it to trainable has minimal effect to the training process; as there are now only  $K$  classes, it might not be as important in CLIP to have a learnable  $\tau$ . Second, initializing it with the pretrained checkpoint (after training with CLIP,  $\exp(\tau) = 100$ ) yields a significant performance drop. We attribute this performance drop to the change in the size of Softmax from  $|\mathcal{B}_{\text{CLIP}}|$  to  $K$ , where  $|\mathcal{B}_{\text{CLIP}}| \gg K$ . Having a large temperature coefficient like  $\exp(\tau) = 100$

dramatically increases the sharpness in the pattern of Softmax and its gradient flow, which is inappropriate for training.

### E.3 Conclusion

The language-augmented initialization is the most critical component in aligning the training behavior of CLIP models (30%+ mean score improvement for 5-shot finetuning), without which the pre-trained capacity in the language encoder would be completely lost. Other factors like visual feature normalization, batch size, temperature, etc. have a much smaller effect to the training procedure. We choose to use the parameter-free batch normalization, keep the traditional batch size, and not bring in additional parameters like temperatures, for trading off between the performance and the simplicity of the model.

## F Empirical Comparisons of Existing Pre-trained Vision Models

### F.1 A Taxonomy of Pre-trained Vision Models

We provide the taxonomy for pre-trained vision models from the perspective whether language and/or is employed in pre-training, as shown in Table 10. The taxonomy is a two-level hierarchy.

1. In the 1st level hierarchy, given a visual recognition problem (IC or OD), the models are first categorized into language-augmented or language-free, depending on whether language is used or not in pre-training.
2. In the 2nd level hierarchy, the language-augmented models are further categorized into knowledge-augmented or knowledge-free, depending on whether the textual external knowledge is used or not in pre-training.

Note that our taxonomy is only related to pre-training, which is independent from how the model is adapted to a downstream task.

For knowledge-augmented pre-trained models such as K-LITE [71], the model is pre-trained with both natural language supervision and external knowledge supervision. The external knowledge is employed in the following manner: (1) For image-text pairs, query is identified using entity extraction on the text, (2) The relevant “knowledge text” of the query is retrieved from knowledge bases; (3) The retrieved “knowledge text” is appended to the original text. In the downstream adaptation stage, it follows the same prompting process with other pre-trained models, as described in Section C.4.

Model Taxonomy Hierarchy		Taxonomy		Checkpoints
		Language Knowledge		
Image Classification	Language-free	x	x	MoCo-v3 [9]
	Language-free	x	x	MAE [28]
	Language-free	x	x	DeiT [74]
	Language-augmented	✓	x	ViT [42]
Object Detection	Language-free	✓	x	CLIP [62]
	Language-free	✓	x	UniCL [82]
	Language-augmented	✓	✓	K-LITE [63]
	Language-augmented	x	x	DyHead [42]
Object Detection	Language-free	✓	x	GLIP [43]
	Language-free	✓	x	GLIP-A [43]
	Language-augmented	✓	✓	K-LITE [63]

Table 10: A Taxonomy of Vision Pre-trained Models

### F.2 Baseline with Vision Pre-trained Models

**Image Classification** We consider seven checkpoints to produce baseline results for IC. In the main paper, we report the following four checkpoints.

- *Supervised ViT* [18] represents a checkpoint for the traditional language-free visual models, where model training is performed on ImageNet-22K with cross-entropy loss.
- *CLIP ViT* [66] represents a checkpoint for the family of the language-augmented visual models, trained with 400M image-text pairs.
- *UniCL Swin* [88] represents knowledge-free language-augmented visual models with Swin [52] as the visual backbone, trained in the academic setting with ImageNet-21K, which excludes ImageNet-1K categories from ImageNet-22K.
- *Knowledge UniCL Swin* [71] represents knowledge-enriched language-augmented visual models. Its pre-training setting is the same as UniCL Swin, but external knowledge such as Wiktionary is leveraged in model pre-training.

We also consider three popular language-free visual models in Appendix:

- *DeiT* [79] represents a checkpoint for the supervised visual backbone, where model training is performed on ImageNet-1K with cross-entropy loss and advanced data augmentation and training schedule.
- *MoCo* [10] represents a checkpoint for the family of augmented-view-based methods for image self-supervised learning, trained with images only in ImageNet-1K.
- *MAE* [30] represents a checkpoint for the family of recent masked region (visual token) modeling based methods for image self-supervised learning, trained with images only in ImageNet-1K.
- *CAE* [9] represents a checkpoint that benefits the separation of the representation learning (encoding) role and the pretext task completion role, trained with images only in ImageNet-1K.

**Object Detection** We consider four checkpoints to produce baseline results for OD. They are for the academic track, as they are pre-trained on public datasets. All of them employ Swin-Tiny backbone [52].

- *DyHead* [13] represents a checkpoint for the traditional language-free object detector, where model is pre-trained on Object365 [69] without leveraging the category name information.
- *GLIP* [47] represents a checkpoint for the family of the language-augmented object detector, trained with Object365 and Flickr phrase grounding data [65].
- *GLIP-A* [47] represents knowledge-free language-augmented object detector, where model is trained on Object365 and the semantics of category names is leveraged.
- *Knowledge GLIP-A* [71] represents knowledge-enriched language-augmented object detector. Its training setting is the same as GLIP-A, except that Wiktionary knowledge is leveraged in model pre-training.

In summary, among four checkpoints for each problem, the first two are used to compare the state-of-the-art in language-free and language-augmented models, and latter two are used to compare the knowledge-free and knowledge-augmented models (both belongs to language-augmented models, as knowledge is presented as a structured form of language).

### F.3 Experimental Results of Different Model Checkpoints

In Table 11, we report IC performance with ViT-B16 pre-trained with representative methods, using different objectives and datasets. We present its breakdown experimental results in Table 12. Note that all of the models are adapted to downstream datasets, using the same automatic hyper-parameter tuning process in our toolkit, and no model- / dataset-specific tuning is employed. This ensures fairness in model adaptation process, but may not represent the best transfer performance of each pre-trained model, if more careful tuning efforts are paid. Nevertheless, we believe the results represent the model transferability with affordable efforts, and use them as baseline results for ELEVATER benchmark.

We found that the overall ranking of the models in the descending order: CLIP, ViT, DeiT, MoCo-v3, MAE. Surprisingly, we found that MAE performs worse than MoCo, and both of them are worse than supervised method DeiT, though all three of them are pre-trained on the same ImageNet-1K

Pre-training Settings			20 Image Classification Datasets			
Checkpoint	Method	Dataset	5-shot	20-shot	50-shot	Full-shot
<b>Linear Probing</b>						
CLIP <sup>‡</sup>	Image-Text Contrast	WebImageText (400M)	68.27 ± 0.97	74.76 ± 1.11	77.75 ± 0.81	81.17
ViT <sup>†</sup>	Supervised	ImageNet-22K (14M)	57.61 ± 3.62	69.93 ± 0.71	73.74 ± 0.79	77.60
DeiT	Supervised	ImageNet-1K (1.2M)	54.06 ± 3.02	68.57 ± 3.43	75.53 ± 0.72	79.56
MAE	Self-Supervised	ImageNet-1K (1.2M)	33.37 ± 1.98	48.03 ± 2.70	58.26 ± 0.84	68.70
MoCo-v3	Self-Supervised	ImageNet-1K (1.2M)	50.17 ± 3.43	61.99 ± 2.51	69.71 ± 1.03	74.92
Random	-	-	19.64 ± 1.68	23.89 ± 1.47	26.86 ± 0.69	31.64
<b>Fine-tuning</b>						
CLIP <sup>‡</sup>	Image-Text Contrast	WebImageText (400M)	69.12 ± 1.66	74.76 ± 2.34	78.21 ± 2.04	83.63
ViT <sup>†</sup>	Supervised	ImageNet-22K (14M)	57.18 ± 2.02	72.45 ± 2.85	78.53 ± 0.69	82.02
DeiT	Supervised	ImageNet-1K (1.2M)	54.06 ± 3.02	68.53 ± 3.47	75.57 ± 0.68	79.55
MAE	Self-Supervised	ImageNet-1K (1.2M)	36.10 ± 3.25	54.13 ± 3.86	65.86 ± 2.42	74.43
MoCo-v3	Self-Supervised	ImageNet-1K (1.2M)	39.30 ± 3.84	58.75 ± 5.55	70.33 ± 1.64	77.71
Random	-	-	20.85 ± 1.59	26.29 ± 1.21	30.88 ± 1.68	43.73

Table 11: Averaged scores on 20 IC datasets with the **ViT-B16** network architecture. <sup>‡</sup> CLIP is adapted using the proposed language-augmented initialization. <sup>†</sup> ViT checkpoint is pre-trained on ImageNet-22K, then fine-tuned on ImageNet-1K. The zero-shot performance of CLIP is 59.96%.

dataset. We note that an similar observation is made in [36], when evaluated these checkpoints on a large range of downstream datasets. This is perhaps because the region-based pre-training tasks in MAE is can better capture region-level dependency (thus benefits dense prediction tasks such as object detection), while view-based pre-training tasks in MoCo can better capture image-level dependency (thus benefits image classification). ViT outperforms DeiT probably due to the larger pre-training dataset. CLIP performs the best. To the best of our knowledge, language-augmented visual models such as CLIP enjoy the best scaling performance; In contrast, the scaling performance of language-free visual models are either less studied or less successful so far.

In Table 14, we presented the comparisons of random and language-augmented initialization for language-image model adaptation with more checkpoints under 5-shot settings. This includes ViT-Base and ViT-Large models of DeCLIP [49], OpenCLIP [33] and CLIP [66].

In Table 15, we presented zero-shot results of more model checkpoints for both Industry and Academic Tracks. For Academic Tracks, we consider CLIP [66], DeCLIP [49], FILIP [89], SLIP [57], with network ViT-Base32 pre-trained on YFCC (15M). For Industry Tracks, we consdier DeCLIP, OpenCLIP and CLIP, with models ranging from ViT-Base to ViT-Large, and training data ranging from 88M to 400M image-text pairs.

#### F.4 Breakdown Experimental Results on CLIP

We show the individual linear probing and finetuning scores for comparing the random and language-augmented initialization in Table 13. Language initialization consistently outperforms random initialization across different domains: sample efficiency, parameter efficiency, and different datasets. See Sec. 4 for more discussions on the design and the effectiveness of the language-augmented initializations.

## G Benefits of External Knowledge in Model Adaptation

We also explore the benefits of the external knowledge to models that are pre-trained without the external knowledge (*e.g.*, CLIP). On CLIP, we compare the effect of adding different combinations of external knowledge (Wiktionary, the nubmer of GPT3 knowledge items). The results are summarized in 16, and detailed in Table 17.

In zero-shot settings, we find that when the external knowledge is available, CLIP demonstrates consistent improvement on four datasets and considerable gains on the other three datasets. This suggests that the knowledge can benefit language-image models (though varying between datasets) as a new language prompting technique for some datasets, even if the pre-trained model is trained without the external knowledge.

Ckpt.	Shot	Score	Linear Probing																			
			Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	REISC45	StanfordCars	VOC2007
CLIP	5	68.3	91.3	91.4	71.1	21.7	61.6	76.7	53.6	36.0	89.7	55.9	58.0	44.8	76.7	94.2	90.5	54.3	62.0	78.3	73.6	84.2
	20	74.8	94.3	93.0	75.4	25.2	73.7	86.6	54.7	48.1	90.6	75.7	58.5	50.3	90.5	96.8	92.3	68.0	63.8	87.5	83.9	86.3
	50	77.8	94.4	93.8	78.0	27.7	76.3	90.0	57.5	53.5	91.3	81.6	60.0	61.4	95.9	96.8	93.8	69.9	68.7	90.1	87.2	87.1
	full	81.2	94.4	95.8	82.2	31.2	77.4	94.5	68.5	52.8	92.8	88.6	65.1	67.7	98.9	96.5	94.0	83.5	74.5	90.8	87.2	87.0
MAE	5	33.4	59.0	34.0	21.2	2.8	35.0	64.4	21.3	7.0	7.7	17.5	51.4	46.1	63.4	50.9	17.2	54.9	50.1	38.9	6.3	18.3
	20	48.0	85.5	44.9	43.5	4.4	58.3	74.1	23.5	29.9	30.4	41.1	51.7	49.8	52.9	71.9	60.0	52.7	53.2	67.4	25.5	39.9
	50	58.3	88.7	67.3	53.3	6.9	66.0	86.4	27.1	39.2	48.2	57.0	50.8	54.0	81.5	71.9	76.5	69.4	51.6	78.6	36.7	59.2
	full	68.7	87.7	88.2	68.3	10.1	66.3	94.8	56.0	39.1	65.1	76.3	56.2	78.8	99.3	72.0	81.6	86.0	58.4	81.2	37.2	71.4
MoCo-v3	5	50.2	80.8	78.5	60.5	4.8	57.1	77.1	20.5	11.8	36.6	31.4	50.7	46.7	64.1	79.5	76.2	54.7	50.0	61.1	13.4	47.9
	20	62.0	91.3	67.7	75.5	7.6	66.3	84.8	30.9	38.2	59.3	53.9	53.5	48.5	81.8	89.5	86.4	52.1	51.6	77.3	49.5	74.2
	50	69.7	92.1	93.6	79.0	10.3	73.4	92.3	40.2	48.0	66.8	66.7	50.3	60.5	88.3	89.5	90.2	75.1	51.3	84.1	63.1	79.2
	full	74.9	92.1	96.9	85.3	13.7	73.1	95.9	60.1	48.0	78.0	78.7	53.7	68.8	98.4	89.5	91.4	86.7	57.1	86.3	63.0	81.7
DeiT	5	54.1	86.2	70.1	61.5	4.4	52.9	62.5	14.5	24.1	41.9	46.7	51.1	47.6	83.8	82.7	87.8	51.5	50.1	63.4	27.6	70.9
	20	68.6	93.9	91.2	73.7	6.2	68.7	90.7	35.2	34.1	61.5	86.7	50.8	52.4	90.7	92.7	91.9	66.7	51.7	82.7	68.8	81.1
	50	75.5	94.7	94.2	82.0	8.8	73.9	94.4	40.8	60.6	73.2	96.5	53.4	69.7	98.1	92.7	93.4	77.4	52.2	89.4	82.9	82.3
	full	79.6	94.9	98.2	89.6	14.1	72.8	98.2	69.3	59.3	84.5	98.8	44.3	82.0	99.6	92.4	93.9	89.9	52.6	90.8	83.0	83.1
ViT	5	57.6	93.2	88.2	75.4	6.8	63.9	70.0	25.2	22.7	59.0	29.9	48.5	46.5	68.3	99.2	89.6	61.3	49.9	57.9	27.6	69.2
	20	69.9	95.6	94.8	84.0	11.5	75.7	86.5	45.4	40.5	81.7	51.1	53.5	57.1	87.7	99.2	92.6	72.0	52.4	79.7	53.9	83.7
	50	73.7	96.0	96.4	86.8	15.2	78.8	91.5	50.0	48.5	85.1	62.1	51.0	60.1	91.7	99.2	93.9	77.7	51.5	85.4	67.3	86.6
	full	77.6	95.9	98.2	89.8	16.6	78.9	96.0	64.5	47.8	89.6	76.5	55.1	69.3	98.2	99.2	94.8	85.5	54.6	86.6	67.5	87.3
Random	5	19.6	9.0	17.6	5.8	1.2	8.2	41.0	15.4	3.0	2.7	7.9	49.6	40.9	26.7	17.8	4.1	52.7	51.5	18.6	1.5	17.5
	20	23.9	13.0	25.1	9.8	1.9	12.4	46.3	20.4	3.6	4.7	9.4	54.4	42.1	40.8	22.4	7.0	64.8	52.2	25.5	2.3	19.9
	50	26.9	15.9	27.3	12.1	2.2	14.2	60.4	20.2	4.1	6.0	11.1	54.1	40.8	56.0	22.4	8.7	73.7	53.1	30.6	2.6	21.6
	full	31.6	16.5	43.0	18.7	3.1	13.8	69.0	30.4	4.4	10.8	15.3	56.6	45.1	85.0	21.7	9.5	77.6	55.0	31.0	2.7	23.3
Fine-tuning																						
CLIP	5	69.1	91.2	92.1	73.2	22.2	53.8	79.0	55.9	33.5	87.5	84.3	55.3	41.9	84.9	87.1	91.7	59.4	59.8	80.1	66.0	83.5
	20	74.8	93.7	93.9	79.7	21.8	70.6	94.1	59.0	52.2	89.0	91.9	54.3	52.7	70.0	93.8	93.0	71.9	62.6	87.0	80.0	84.2
	50	78.2	94.5	94.7	82.8	21.9	75.0	95.7	61.0	61.5	89.3	91.3	54.5	65.1	85.4	93.8	93.7	75.4	64.9	91.0	86.5	86.3
	full	83.6	94.9	98.6	89.4	23.6	74.7	98.4	72.0	60.8	91.8	99.0	65.9	84.1	99.6	94.1	94.2	89.7	76.5	91.9	86.9	86.5
MAE	5	36.1	70.8	34.4	13.1	2.1	41.4	64.1	20.8	8.2	13.3	14.8	49.6	38.0	46.8	68.8	37.8	53.3	50.9	50.4	6.0	37.4
	20	54.1	91.0	50.1	40.4	3.6	59.7	79.5	22.6	32.5	22.4	62.2	54.8	46.0	90.9	81.6	78.0	67.7	51.7	65.5	21.6	60.8
	50	65.9	92.9	71.5	54.7	4.9	66.2	87.8	34.1	42.8	51.9	95.6	51.3	50.1	96.1	81.6	84.8	77.3	52.4	85.2	68.0	68.0
	full	74.4	92.8	97.7	85.5	9.3	66.2	97.5	68.5	46.2	84.6	99.1	55.2	82.8	99.6	75.3	89.8	76.0	56.7	87.0	47.0	71.7
MoCo-v3	5	39.3	73.7	70.3	17.4	2.3	45.6	60.0	13.5	7.2	27.6	16.5	50.8	43.5	18.1	65.7	77.1	50.9	50.7	58.2	11.2	25.7
	20	58.8	91.9	58.4	59.2	5.0	63.4	69.7	19.8	47.4	55.5	86.7	53.5	48.5	53.4	85.8	87.4	51.5	51.4	78.5	49.2	59.2
	50	70.3	92.8	89.1	77.5	6.9	71.3	92.6	31.0	53.4	63.2	96.5	50.9	57.3	94.3	85.8	90.2	74.2	50.4	87.3	66.2	75.7
	full	77.7	93.3	98.1	88.7	11.7	71.3	97.3	68.3	51.9	84.1	98.8	54.5	80.5	99.6	87.1	90.9	91.4	52.5	88.6	67.9	77.6
DeiT	5	54.1	86.2	70.1	61.5	4.4	52.9	62.5	14.5	24.1	41.9	46.7	51.1	47.6	83.8	82.7	87.8	51.5	50.1	63.4	27.6	70.9
	20	68.5	93.9	91.2	73.7	6.2	68.7	90.7	34.4	34.1	61.5	86.7	50.8	52.4	90.7	92.7	91.9	66.7	51.7	82.7	68.8	81.1
	50	75.6	94.7	94.2	82.0	9.6	73.9	94.4	40.8	60.6	73.2	96.5	53.4	69.7	98.0	92.7	93.4	77.4	52.2	89.4	82.9	82.3
	full	79.5	94.9	98.2	89.6	14.1	72.8	98.2	69.3	59.2	84.5	98.8	44.3	82.0	99.6	92.4	93.9	89.9	52.6	90.8	83.0	83.1
ViT	5	57.2	90.8	82.7	67.6	4.0	56.0	75.2	24.5	21.4	58.0	51.5	47.6	38.4	82.6	99.0	83.8	53.8	51.0	61.5	21.0	73.2
	20	72.5	96.1	93.6	86.7	8.4	74.2	91.7	43.6	51.6	68.2	92.8	51.9	57.8	95.8	99.4	92.1	71.6	51.8	84.8	65.5	71.4
	50	78.5	96.3	97.3	89.9	11.8	79.1	95.0	52.1	63.6	83.0	97.5	54.7	68.9	97.5	99.5	93.3	80.5	52.3	90.1	83.0	85.2
	full	82.0	96.6	99.0	93.4	16.8	79.4	98.3	72.6	61.9	90.7	99.1	53.4	84.5	99.7	99.5	94.0	91.1	50.1	91.5	83.2	85.6
Random	5	20.9	12.4	16.2	6.6	1.3	9.4	38.3	19.9	3.2	3.2	8.6	52.4	41.7	18.6	25.4	4.7	62.3	51.1	21.7	1.8	18.3
	20	26.3	24.5	25.3	13.1	2.2	16.4	55.3	20.2	5.1	5.9	16.6	52.8	35.2	38.0	38.3	8.2	65.7	50.6	29.1	3.4	20.1
	50	30.9	27.5	31.1	19.6	3.0	19.9	67.3	21.9	6.5	8.5	28.5	56.0	42.2	42.7	38.3	11.6	73.5	52.3	41.2	3.7	22.2
	full	43.7	28.3	65.9	41.4	4.0	21.8	85.3	41.0	7.2	34.0	83.0	55.7	50.5	95.6	35.1	15.3	81.8	55.6	44.5	2.6	26.1

Table 12: Breakdown results of **ViT-B16** checkpoints in Table 11.

In few- / full-shot settings, we argue that the pre-trained model can *selectively* incorporate different knowledge sources to achieve the best adaptation performance. One simple strategy is to train the model with different knowledge sources, compare the split validation accuracy of checkpoints with different knowledge sources, and use the best one for testing. We called it as **knowledge-augmented adaptation**, in contrast to the baseline method **knowledge-free adaptation**, where no collected external knowledge is employed at all. We find such simple strategy is already effective for linear probing and fine-tuning CLIP. As shown in Table 16, knowledge-based adaptation of CLIP consistently improves over knowledge-free adaptation both in terms of accuracy and the number of wins. Notably, by selectively incorporating the external knowledge, it shows a significant 1.8

Shot	Lang-Init	Score	Fine-tuning																				
			Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007	
5	✗	29.8	40.8	19.6	15.5	0.9	25.2	55.8	21.1	13.4	14.7	30.6	46.1	41.5	52.2	31.8	44.5	52.5	51.2	16.5	3.7	17.5	
	✓	63.3	88.8	91.3	73.0	16.6	51.8	79.3	52.2	23.1	84.0	60.4	55.8	44.3	60.5	67.3	86.9	61.8	59.2	70.8	56.3	82.4	
20	✗	46.8	82.6	63.2	26.5	1.9	57.9	81.6	27.4	33.2	36.6	60.9	53.1	41.7	35.6	34.6	54.2	74.9	51.7	43.8	32.9	41.0	
	✓	72.2	93.3	91.9	76.0	17.2	60.0	90.4	57.9	42.7	84.2	92.0	53.9	46.4	93.1	86.4	90.8	72.4	59.4	82.9	69.9	82.9	
50	✗	61.7	91.4	88.7	42.7	2.7	68.2	85.8	42.7	50.3	72.7	77.3	52.6	52.0	71.9	34.6	84.3	78.0	52.7	88.0	51.4	46.0	
	✓	75.7	94.0	93.3	79.1	17.5	71.7	94.9	58.7	51.6	85.1	95.2	55.0	59.1	89.7	86.4	91.1	78.6	62.0	88.4	76.8	85.7	
full	✗	77.7	88.9	97.4	85.8	14.6	70.8	97.7	69.8	46.3	85.4	97.9	60.5	78.9	98.9	81.8	89.5	88.7	55.3	89.4	76.1	81.1	
	✓	80.3	94.0	97.8	87.0	19.1	70.0	98.1	68.8	50.7	87.7	98.5	61.9	81.0	99.5	88.5	91.6	91.0	70.6	89.4	75.8	85.7	
<b>Linear Probing</b>																							
5	✗	58.1	88.1	87.0	56.1	10.1	58.1	73.8	33.9	28.2	70.0	52.8	51.0	40.9	77.5	89.5	66.5	57.0	49.4	75.3	53.1	43.3	
	✓	65.3	89.8	90.0	67.4	17.5	59.6	73.2	47.4	28.4	84.2	52.5	56.0	44.9	71.1	90.5	88.0	63.2	57.5	76.6	65.0	84.0	
20	✗	70.0	92.2	91.0	69.2	16.6	71.0	81.2	48.6	39.8	81.3	73.1	51.3	51.3	92.4	93.8	83.7	65.4	58.0	84.4	73.0	82.1	
	✓	71.7	92.9	90.8	71.5	19.6	71.3	83.0	52.2	40.2	85.3	74.1	57.1	50.8	92.5	94.2	88.5	63.2	58.9	84.4	77.9	85.5	
50	✗	74.1	92.8	92.1	73.7	21.1	74.5	88.1	53.6	44.3	84.0	80.5	51.3	58.7	95.1	93.8	88.2	75.2	62.3	87.0	81.0	84.2	
	✓	74.9	93.1	91.6	74.9	22.9	74.8	88.2	53.6	44.6	86.1	80.7	57.7	60.9	95.1	94.2	89.7	72.3	62.1	87.3	82.0	86.0	
full	✗	78.4	92.7	94.5	79.6	25.2	74.0	93.4	67.8	44.3	88.1	86.9	64.0	65.8	98.8	93.9	89.9	83.2	71.4	88.1	80.8	85.0	
	✓	78.4	86.0	95.1	79.8	25.9	75.3	93.8	67.8	44.7	88.6	86.9	63.1	65.8	98.8	94.5	91.0	83.2	71.6	88.1	82.1	86.0	

Table 13: Comparison of random and language-augmented initialization on CLIP (ViT-B32).

Backbone	Pretrain	Language-Init	Fine-tuning																				
			Average Score	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
B32	DeCLIP	✗	58.8	88.2	78.0	59.4	6.0	58.3	73.8	20.1	26.4	61.4	66.0	51.5	30.2	77.0	98.1	74.9	53.1	52.7	68.2	57.6	75.5
B32	DeCLIP	✓	64.5	92.9	91.6	77.5	11.7	55.2	77.1	38.4	20.0	76.2	69.7	54.5	43.6	73.1	95.5	85.6	61.4	52.0	71.7	60.2	82.3
B32	OpenCLIP	✗	34.6	28.7	73.2	13.6	1.1	36.2	76.3	29.1	9.1	9.2	7.8	50.4	31.2	66.2	40.5	32.1	63.8	51.5	39.1	1.2	31.7
B32	OpenCLIP	✓	64.8	91.6	91.6	74.1	10.0	54.3	72.2	46.6	23.0	79.0	82.3	54.4	33.8	85.9	83.8	86.1	62.8	53.1	76.0	53.7	82.2
B16	OpenCLIP	✗	27.6	19.6	32.8	6.4	1.2	28.7	76.1	15.6	3.5	6.0	7.1	48.2	46.1	60.8	33.2	6.0	57.1	51.6	30.9	1.7	20.1
B16	OpenCLIP	✓	66.2	86.8	91.5	74.6	17.0	60.6	79.8	45.2	15.4	84.2	60.6	54.1	34.1	85.8	86.0	88.2	67.4	55.0	72.5	82.9	83.1
<b>Linear Probing</b>																							
B32	DeCLIP	✗	57.2	88.4	86.8	60.6	7.9	58.6	70.4	29.8	23.9	63.9	29.2	50.5	31.5	68.3	98.3	74.8	60.8	49.5	67.3	59.6	64.9
B32	DeCLIP	✓	62.5	93.0	92.0	73.3	12.8	62.1	72.2	36.3	23.9	76.2	29.2	54.7	45.7	68.3	98.6	84.9	61.0	52.6	66.4	65.2	80.2
B32	OpenCLIP	✗	61.9	89.7	88.1	64.2	8.1	53.3	78.8	33.2	28.8	68.6	64.4	50.6	36.6	80.7	92.2	72.7	61.1	52.5	73.3	74.3	67.2
B32	OpenCLIP	✓	68.6	91.2	91.2	72.0	14.4	68.9	76.9	45.9	31.2	81.4	65.1	52.9	46.7	86.0	94.0	87.9	65.9	54.7	79.1	83.0	84.5
B16	OpenCLIP	✗	62.9	90.2	85.5	63.9	9.2	65.6	78.3	24.3	33.0	74.8	62.3	51.9	30.8	88.1	94.0	74.5	52.0	50.2	78.7	77.2	73.1
B16	OpenCLIP	✓	69.7	93.2	91.6	72.7	18.1	69.4	79.7	46.3	34.3	84.0	64.1	53.4	38.7	92.8	95.0	88.2	66.3	57.4	78.5	86.0	85.0
L14	OpenCLIP	✗	66.5	91.7	92.1	70.0	12.1	66.3	80.1	36.4	37.2	81.0	72.0	51.9	27.1	87.6	96.0	81.0	53.2	52.2	81.4	84.1	76.6
L14	OpenCLIP	✓	72.5	92.9	94.1	78.8	22.6	72.0	86.0	52.9	40.1	89.2	74.2	54.7	41.1	86.4	97.2	91.5	60.2	58.8	83.3	89.3	84.9
L14 <sup>†</sup>	CLIP	✗	68.3	93.3	92.1	70.2	19.6	65.0	85.1	42.5	46.0	88.0	72.7	51.3	45.0	80.9	96.6	83.8	60.3	56.5	80.9	79.5	57.8
L14 <sup>†</sup>	CLIP	✓	75.2	94.5	95.3	79.3	34.2	70.0	87.0	58.4	50.1	93.8	74.2	59.8	35.0	83.0	98.0	94.2	65.8	71.3	87.8	85.7	86.5

Table 14: Comparisons of random and language-augmented initialization for language-image model adaptation with more checkpoints under 5-shot settings. <sup>†</sup> Input image size 336×336 .

improvement for 5-shot CLIP fine-tuning. Note that such gain comes for *free*, even when the base CLIP model is *not* pre-trained with the external knowledge. We believe more sophisticated knowledge adaptation strategy can yield even better performance and we leave that to future work.

These experiments show that the collected external knowledge on ELEVATER is a useful resource for improving the adaptation of language-augmented visual models.

Backbone	Pretrain Method	Pretrain Dataset	Average Score	Academic Track																			
				Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RENSC45	StanfordCars	VOC2007
B32	CLIP	YFCC (15M)	32.0	55.9	70.2	33.7	5.1	15.6	29.9	23.3	2.5	32.1	5.6	53.5	39.9	14.3	48.7	19.1	50.0	49.0	17.3	2.3	71.6
B32	DeCLIP	YFCC (15M)	37.9	69.1	85.3	55.5	8.8	26.3	27.5	29.8	2.9	48.6	10.4	51.7	28.4	11.1	59.8	34.9	50.6	49.9	25.0	4.0	77.5
B32	FILIP	YFCC (15M)	34.5	65.1	83.6	50.8	7.5	23.2	23.4	23.3	3.0	40.8	7.4	50.8	24.2	7.9	49.5	22.5	51.8	49.9	25.9	3.1	77.1
B32	SLIP	YFCC (15M)	31.2	58.8	69.5	39.0	5.1	14.0	19.5	22.8	1.3	32.8	6.7	52.9	29.0	10.3	45.9	24.4	50.0	49.9	17.5	2.2	71.6
Industry Track																							
B32	CLIP	WebImageText (400M)	56.8	87.4	89.8	65.2	17.2	44.1	46.0	42.0	19.5	84.0	32.7	56.0	29.0	48.4	66.5	87.2	60.7	58.8	60.0	59.6	82.6
B32	DeCLIP	DeCLIP (88M)	51.0	89.2	90.9	66.8	12.0	44.9	39.9	23.3	9.0	75.0	11.4	53.9	39.7	13.6	83.0	83.7	55.3	50.1	47.6	49.7	80.6
B32	OpenCLIP	LAION (400M)	57.5	90.1	90.8	70.6	14.8	54.5	51.7	42.4	16.6	80.8	42.0	52.8	31.6	37.6	65.9	86.5	50.1	52.3	57.5	79.3	82.1
B16	CLIP	WebImageText (400M)	60.0	88.9	90.8	68.2	22.8	44.8	54.7	48.5	24.3	88.7	43.5	58.1	27.0	52.0	69.4	89.0	54.0	60.9	65.6	64.8	83.7
B16	OpenCLIP	LAION (400M)	59.1	90.3	90.2	70.0	17.4	48.7	48.6	44.9	15.3	83.2	38.6	53.4	23.9	71.1	63.7	87.6	51.0	57.2	63.6	81.6	82.2
L14	CLIP	WebImageText (400M)	65.9	92.6	95.6	78.2	31.8	55.4	64.1	50.0	31.9	93.1	50.5	59.3	13.5	76.2	79.1	93.5	51.2	68.9	71.0	77.9	83.9
L14	OpenCLIP	LAION (400M)	62.5	92.9	93.5	76.2	21.2	56.4	53.7	50.3	20.8	89.1	45.6	55.3	28.8	63.9	70.9	89.7	50.5	57.0	64.1	87.4	82.3
L14 †	CLIP	WebImageText (400M)	66.8	92.4	94.9	77.0	34.5	56.0	63.0	48.3	33.3	93.9	52.3	60.0	11.5	79.0	78.5	93.8	62.3	70.6	71.3	79.3	84.0

Table 15: Zero-shot results of more checkpoints in Academic and Industry Tracks. † Input image size 336×336.

Adaptation Methods	5-shot		Full-shot	
	LP	FT	LP	FT
Knowledge-free adaptation	65.35 ± 1.24	63.29 ± 3.18	78.40	79.97
Knowledge-augmented adaptation	<b>65.83</b> ± 1.50	<b>65.10</b> ± 2.08	<b>78.75</b>	<b>80.32</b>
Gain	+0.48	+1.81	+0.35	+0.35
# win / tie / lose	<b>7 / 8 / 5</b>	<b>8 / 8 / 4</b>	<b>12 / 4 / 4</b>	<b>10 / 5 / 5</b>

Table 16: Benefits of adapting CLIP with external knowledge.

Wiki #GPT3	mAcc	Zero-Shot																			
		Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RENSC45	StanfordCars	VOC2007
✓ –	56.8	87.4	89.8	65.1	17.2	44.4	45.5	42.3	19.6	84.0	32.5	56.0	29.0	48.2	66.5	87.2	60.6	58.6	60.0	59.7	82.6
✓ –	52.1	83.6	85.4	56.1	13.2	44.4	40.3	39.6	18.4	79.8	28.9	55.5	27.3	10.6	66.2	81.0	52.4	<b>62.2</b>	57.8	59.7	80.1
✓ 1	53.3	86.8	88.4	57.6	14.9	<b>47.0</b>	36.6	42.0	18.4	81.8	<b>34.0</b>	55.5	28.3	19.1	<b>67.6</b>	85.3	56.6	<b>61.9</b>	58.1	45.4	81.3
✓ 5	54.2	87.3	88.8	63.9	16.0	<b>50.1</b>	41.1	<b>43.4</b>	18.5	82.3	<b>36.4</b>	55.5	<b>32.2</b>	11.9	<b>69.5</b>	87.0	52.9	<b>62.0</b>	58.6	45.1	82.0
✓ 1	53.2	86.1	87.6	61.5	14.7	43.6	<b>51.2</b>	33.3	18.5	80.0	31.4	55.5	<b>33.2</b>	23.1	<b>66.6</b>	84.5	51.5	<b>61.2</b>	53.7	45.4	81.1
✓ 5	54.5	87.0	88.7	63.9	16.1	<b>49.5</b>	<b>50.9</b>	<b>44.0</b>	18.6	81.9	<b>35.5</b>	55.5	<b>31.2</b>	14.7	<b>69.4</b>	<b>87.3</b>	49.9	<b>62.2</b>	57.5	45.1	82.0
5-Shot Linear Probing																					
✓ –	65.3	89.8	90.0	67.4	17.5	59.6	73.2	47.4	28.4	84.2	52.5	56.0	44.9	71.1	90.5	88.0	63.2	57.5	76.6	65.0	84.0
✓ 5	65.2	89.5	89.3	67.4	17.5	<b>61.9</b>	<b>72.0</b>	<b>48.4</b>	<b>28.5</b>	84.2	52.2	55.5	39.3	<b>76.2</b>	<b>91.1</b>	<b>88.1</b>	<b>63.5</b>	<b>58.4</b>	72.7	65.0	83.5
✓ –	<b>65.6</b>	89.2	<b>90.7</b>	67.4	17.5	<b>61.0</b>	<b>74.1</b>	45.8	28.4	84.2	52.5	55.5	37.5	<b>76.4</b>	<b>91.0</b>	88.0	<b>67.8</b>	<b>59.7</b>	76.6	65.0	83.6
✓ 5	<b>65.8</b>	89.3	89.2	66.5	17.5	<b>61.4</b>	<b>73.5</b>	<b>48.4</b>	<b>28.5</b>	84.2	<b>53.2</b>	55.5	<b>45.5</b>	<b>76.2</b>	<b>91.1</b>	<b>88.6</b>	<b>63.3</b>	<b>59.9</b>	76.6	65.0	83.2
5-Shot Fine-tuning																					
✓ –	63.3	88.8	91.3	73.0	16.6	51.8	79.3	52.2	23.1	84.0	60.4	55.8	44.3	60.5	67.3	86.9	61.8	59.2	70.8	56.3	82.4
✓ 5	62.0	88.0	90.3	73.0	16.6	<b>56.4</b>	<b>81.3</b>	52.2	22.3	83.9	60.4	55.4	<b>47.0</b>	60.5	<b>79.9</b>	<b>87.5</b>	<b>65.0</b>	<b>62.7</b>	19.8	56.3	81.8
✓ –	<b>65.1</b>	88.8	88.9	73.0	16.6	44.9	79.3	52.2	<b>24.4</b>	<b>84.1</b>	66.7	55.4	<b>44.5</b>	<b>82.4</b>	<b>79.4</b>	86.9	60.6	<b>63.3</b>	<b>71.8</b>	56.3	82.4
✓ 5	<b>64.2</b>	88.5	89.8	73.0	16.6	<b>53.1</b>	<b>82.0</b>	52.2	21.6	83.9	60.4	55.5	34.9	<b>81.4</b>	<b>77.7</b>	<b>87.9</b>	54.2	<b>62.6</b>	70.8	56.3	81.8
Full-Shot Linear Probing																					
✓ –	78.4	86.0	95.1	79.8	25.9	75.3	93.8	67.8	44.7	88.6	86.9	63.1	65.8	98.8	94.5	91.0	83.2	71.6	88.1	82.1	86.0
✓ 5	<b>78.7</b>	<b>92.8</b>	94.9	79.8	25.7	75.1	93.3	67.7	<b>44.9</b>	88.6	<b>87.0</b>	<b>64.1</b>	<b>66.8</b>	<b>94.9</b>	90.8	<b>83.7</b>	70.3	87.0	81.9	85.8	
✓ –	<b>78.8</b>	<b>93.2</b>	<b>95.2</b>	<b>79.9</b>	25.7	73.5	93.3	<b>67.8</b>	<b>44.8</b>	88.2	<b>86.9</b>	<b>64.1</b>	65.8	98.8	<b>94.9</b>	<b>91.1</b>	<b>83.7</b>	71.6	<b>88.3</b>	<b>82.2</b>	86.0
✓ 5	<b>78.6</b>	<b>93.1</b>	94.9	79.8	25.7	74.7	93.4	65.4	44.6	88.6	<b>87.0</b>	<b>64.1</b>	<b>66.7</b>	<b>98.9</b>	<b>94.9</b>	90.8	<b>83.7</b>	70.3	<b>88.3</b>	81.9	85.7
Full-Shot Fine-tuning																					
✓ –	80.0	93.1	97.5	87.3	19.2	70.9	98.0	70.2	47.7	88.0	98.5	61.1	81.9	99.5	87.3	90.7	90.6	66.7	89.4	76.1	85.5
✓ 5	<b>80.0</b>	<b>93.5</b>	<b>97.5</b>	84.4	<b>19.4</b>	<b>72.5</b>	97.9	69.5	47.7	87.8	<b>98.5</b>	61.1	81.3	<b>99.5</b>	<b>89.5</b>	<b>92.5</b>	90.1	66.7	<b>90.0</b>	76.1	85.4
✓ –	<b>80.1</b>	93.0	97.4	87.3	19.2	<b>73.0</b>	98.0	70.0	47.7	87.6	<b>98.5</b>	61.1	80.9	99.5	<b>89.6</b>	<b>92.2</b>	89.2	66.7	<b>89.5</b>	<b>76.2</b>	85.5
✓ 5	<b>80.3</b>	<b>93.6</b>	<b>97.6</b>	<b>87.4</b>	19.2	70.9	<b>98.1</b>	<b>71.7</b>	47.7	87.8	98.4	61.1	81.6	99.3	<b>89.3</b>	<b>92.5</b>	87.3	<b>70.9</b>	<b>90.0</b>	76.1	<b>85.9</b>

Table 17: Benefit of external knowledge for CLIP. For adaptation with linear probing and fine-tuning, we make use of the external knowledge when it has a higher validation accuracy.