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 E.1 (referred by Section 4), we introduce more details of our toolkits, including the automatic hyper-parameter tuning pipeline and implementation details.
- In Section F (referred by Section 4), we discuss the gap between language-image model pre-training and adaptation.
- In Section G (referred by Section 5.1), we provide performance of comparison different vision pre-trained models.
- In Section H (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, while 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.

For the readers who are interested in the literature on Computer Vision in the Wild, we create an up-to-date CVinW reading list at `https://github.com/Computer-Vision-in-the-Wild/CVinW_Readings`.

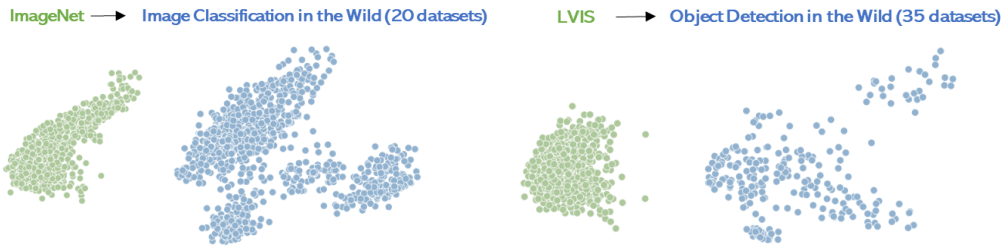
B.2 Related Works in NLP: Benchmarks, Adaptation, and Knowledge

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

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.



(a) Image Classification (b) Object Detection

Figure 9: Semantic space comparison with 2D PCA. For IC or OD, the CLIP text feature of category names in each benchmark are projected together with PCA, and visualized separately.

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.

C.2 Visualization Comparison with Established Vision Datasets

We also compare our benchmark with well established datasets in computer vision: ImageNet-1K for IC and COCO/LVIS for OD. Note that LVIS is much diverse than COCO in terms of concept coverage. The visualization of concept semantic space is Figure 9. The semantics is computed by extracting the CLIP text features from the category names. To quantitatively measure the diversity of different benchmarks, we compute the standard derivation (STD) over text features. The STD of ImageNet1-K and ICinW is 0.610 and 0.680, respectively. The STD of LVIS and ODinW is 0.533 and 0.619, respectively.

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.

C.3 License

As per the original authors, the licenses of each dataset include CC BY-NC-SA 3.0[3], CC BY-NC-SA 4.0[4], CC BY 4.0[5], ODbL v1.0[6], MIT[7], CC0 1.0[8]. Some datasets have published dedicated usage aggrements: Hateful Memes[9]. All datsets 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[10], Wiktionary[11], GPT-3[12]. For the GPT-3 generated knowledge, we have the approval from OpenAI to release it as a part of ELEVATER to encourage future research.

[3] https://creativecommons.org/licenses/by-nc-sa/3.0/

[4] https://creativecommons.org/licenses/by-nc-sa/4.0/

[5] https://creativecommons.org/licenses/by/4.0/

[6] https://opendatacommons.org/licenses/odbl/1-0/

[7] https://choosealicense.com/licenses/mit/

[8] https://creativecommons.org/publicdomain/zero/1.0/

[9] https://www.drivendata.org/competitions/64/hateful-memes/page/214/

[10] https://wordnet.princeton.edu/license-and-commercial-use

[11] https://en.wiktionary.org/wiki/Wiktionary:Main_Page

[12] https://openai.com/api/policies/sharing-publication/

Dataset name	Oxford Flowers 102
Category names	['pink primrose', · · ·]
Templates	['a photo of a {}, a type of flower.',]
Knowledge	["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."], · · ·]
Prompt	• 'a photo of a pink primrose, a type of flower.'
Prompt + Knowledge	• 'a photo of a pink primrose, a type of flower ; A flowering plant of the genus Primula.' • 'a photo of a pink primrose, a type of flower ; A plant of the genus Primula, having a pink flower.' • 'a photo of a pink primrose, a type of flower ; Primula vulgaris, a plant of the primrose family, with pink flowers.' • 'a photo of a pink primrose, a type of flower ; A flowering plant of the genus Primula.' • 'a photo of a pink primrose, a type of flower ; A primrose, Primula × polyantha, with pink flowers.' • '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.'

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

C.4 Generating GPT-3 Knowledge with In-Context-Learning

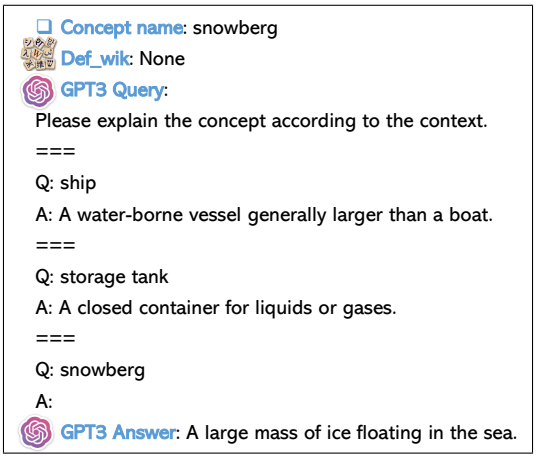


Figure 10: Example of generating external knowledge with GPT3 using in-context learning even when Wiktionary knowledge is missing.

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, 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 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. 10, 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.5 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 Evaluation

D.1 Leaderboards

As demonstrated in Section 3.3, we advocate an evaluation setting with efficiency considerations, which decomposes the adaptation cost into two orthogonal dimensions: sample-efficiency and parameter-efficiency. To encourage future users compare their models with efficiency considerations, We build the public leaderboards on EvalAI:

- Image Classification in the Wild (ICinW)
  `https://eval.ai/web/challenges/challenge-page/1832/overview`
- Object Detection in the Wild (ODinW)
  `https://eval.ai/web/challenges/challenge-page/1839/overview`

D.2 A new metric with performance-efficiency trade-off

For parameter-efficiency track, to compare different methods with a single number that considers both prediction accuracy and parameter-efficiency, we define the performance-efficiency (PE) metric:

$$\text{PE} = \texttt{score} * \exp(\log_{10}(\texttt{\# trainable-parameters}/M_0 + 1)) \tag{1}$$

where `score` measures the prediction accuracy, while `# trainable-parameters` is the number of updated parameters in the model adaptation stage, and M_0 is the normalization constant. We set $M_0 = 10^8$ because most existing vision backbone model size are designed in this magnitude, for example, ViT-Base (80M parameters) and ViT-Large (300M parameters). With larger models designed in the future, one may increase M_0 for sensible measurement.

E Toolkit

Our code is under MIT license.

E.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 η and weight decay α. In the hyper-parameter search

stage, the model is trained with a given configuration (η, α) 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 η_0 and weight decay α, 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 η_0 and weight decay α, as shown in Table 8. They are determined by either empirical rules or simple hyper-parameter tuning.

Settings		35 OD datasets		
Checkpoint	Adaptation	$\|\mathcal{B}\|$	η_0	α
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.

E.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[13].

F 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 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.

[13] `shorturl.at/AOZ13`

BN	ℓ_2	exp(τ)	🔥	mean	std	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
✓	✗	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 τ.

F.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 ℓ_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 ℓ_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. F.2.

F.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 τ 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 τ 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 τ. 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.

F.3 Conclusion

The language-augmented initialization is the most critical componenet 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.

G Empirical Comparisons of Existing Pre-trained Vision Models

G.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.5.

Model Taxonomy Hierarchy			Taxonomy: Language	Taxonomy: Knowledge	Checkpoints
Image Classification	Language-free		✗	✗	MoCo-v3 [9]
			✗	✗	MAE [28]
			✗	✗	DeiT [74]
			✗	✗	ViT [17]
	Language-augmented	Knowledge-free	✓	✗	CLIP [63]
			✓	✗	UniCL [84]
		Knowledge-augmented	✓	✓	K-LITE [68]
Object Detection	Language-free		✗	✗	DyHead [12]
	Language-augmented	Knowledge-free	✓	✗	GLIP [43]
			✓	✗	GLIP-A [43]
		Knowledge-augmented	✓	✓	K-LITE [68]

Table 10: A Taxonomy of Vision Pre-trained Models

G.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.

Checkpoint	Pre-training Settings: Method	Pre-training Settings: Dataset	20 Image Classification Datasets: 5-shot	20-shot	50-shot	Full-shot
Linear Probing						
CLIP‡	Image-Text Contrast	WebImageText (400M)	68.27 ± 0.97	74.76 ± 1.11	77.75 ± 0.81	81.17
ViT†	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
CAE	Self-Supervised	ImageNet-1K (1.2M)	44.15 ± 0.31	57.93 ± 0.19	64.37 ± 0.23	70.56
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
Fine-tuning						
CLIP‡	Image-Text Contrast	WebImageText (400M)	69.12 ± 1.66	74.76 ± 2.34	78.21 ± 2.04	83.63
ViT†	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
CAE	Self-Supervised	ImageNet-1K (1.2M)	37.87 ± 1.03	58.04 ± 2.07	71.39 ± 0.79	77.79
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. ‡ CLIP is adapted using the proposed language-augmented initialization. † ViT checkpoint is pre-trained on ImageNet-22K, then fine-tuned on ImageNet-1K. The zero-shot performance of CLIP is 59.96%.

- *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.
- *KLITE, 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 Flicker 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.

Ckpt.	Shot	Score	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
Linear Probing																						
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	42.8	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
CAE	5	43.8	74.7	61.6	38.3	3.5	43.7	76.7	24.5	14.3	18.6	33.8	47.9	42.3	57.8	70.3	37.3	63.2	52.1	54.4	8.7	51.3
	20	57.9	87.3	76.4	55.1	5.5	62.0	89.0	32.5	32.6	35.7	54.3	51.6	57.3	88.9	81.2	63.3	69.9	52.2	72.1	27.5	64.4
	50	71.4	93.9	90.6	78.3	6.8	69.4	93.2	43.2	56.1	59.4	93.5	53.0	61.6	96.2	85.9	90.0	81.1	52.3	88.0	69.5	65.8
	full	70.6	90.0	93.9	78.9	11.4	66.3	96.7	57.9	40.8	67.4	78.9	55.6	75.7	99.0	81.2	79.8	85.9	58.8	82.7	40.4	70.0
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
CAE	5	39.2	74.4	54.2	30.3	1.8	47.5	68.2	18.9	5.8	18.1	10.9	48.4	35.3	22.6	73.3	51.9	50.0	52.7	58.2	3.7	57.7
	20	58.0	89.3	30.6	62.1	4.9	62.3	73.6	24.9	36.3	30.9	84.9	49.7	56.0	66.1	86.0	84.0	75.8	52.4	75.3	50.4	65.2
	50	71.4	93.9	90.6	78.3	6.8	69.4	93.2	43.2	56.1	59.4	93.5	53.0	61.6	96.2	85.9	90.0	81.1	52.3	88.0	69.5	65.8
	full	77.8	93.2	98.6	89.1	12.8	68.0	98.1	68.8	44.3	87.3	99.2	57.1	84.3	99.8	87.6	92.1	91.8	56.7	89.9	61.6	75.4
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.

- *KLITE, 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).

G.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 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.

G.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.

H 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.

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

Shot	Lang-Init	Score	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
Fine-tuning																						
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
Linear Probing																						
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	Average Score	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
Fine-tuning																							
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
Linear Probing																							
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 †	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 †	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. † Input image size 336×336 .

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 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	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
Academic Track																							
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 LP	5-shot FT	Full-shot LP	Full-shot FT
Knowledge-free adaptation	65.35 ± 1.24	63.29 ± 3.18	78.40	79.97
Knowledge-augmented adaptation	**65.83** ± 1.50	**65.10** ± 2.08	**78.75**	**80.32**
Gain	+0.48	+1.81	+0.35	+0.35
# **win** / **tie** / **lose**	**7** / **8** / **5**	**8** / **8** / **4**	**12** / **4** / **4**	**10** / **5** / **5**

Table 16: Benefits of adapting CLIP with external knowledge.

Wiki	#GPT3	mAcc	Caltech101	CIFAR10	CIFAR100	Country211	DTD	EuroSat	FER2013	FGVCAircraft	Food101	GTSRB	HatefulMemes	KittiDistance	MNIST	Flowers102	OxfordPets	PatchCamelyon	SST2	RESISC45	StanfordCars	VOC2007
Zero-Shot																						
	–	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	**62.2**	57.8	59.7	80.1
✓	1	53.3	86.8	88.4	57.6	14.9	**47.0**	36.6	42.0	18.4	81.8	**34.0**	55.5	28.3	19.1	**67.6**	85.3	56.6	**61.9**	58.1	45.4	81.3
✓	5	54.2	87.3	88.8	63.9	16.0	**50.1**	41.1	**43.4**	18.5	82.3	**36.4**	55.5	**32.2**	11.9	**69.5**	87.0	52.9	**62.0**	58.6	45.1	82.0
	1	53.2	86.1	87.6	61.5	14.7	43.6	**51.2**	33.3	18.5	80.0	31.4	55.5	**33.2**	23.1	**66.6**	84.5	51.5	**61.2**	53.7	45.4	81.1
	5	54.5	87.0	88.7	63.9	16.1	**49.5**	**50.9**	**44.0**	18.6	81.9	**35.5**	55.5	**31.2**	14.7	**69.4**	**87.3**	49.9	**62.2**	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	**61.9**	72.0	**48.4**	**28.5**	84.2	52.2	55.5	39.3	**76.2**	**91.1**	**88.1**	**63.5**	**58.4**	72.7	65.0	83.5
✓	–	**65.6**	89.2	**90.7**	67.4	17.5	**61.0**	**74.1**	45.8	28.4	84.2	52.5	55.5	37.5	**76.4**	**91.0**	88.0	**67.8**	**59.7**	76.6	65.0	83.6
	5	**65.8**	89.3	89.2	66.5	17.5	**61.4**	**73.5**	**48.4**	**28.5**	84.2	**53.2**	55.5	**45.5**	**76.2**	**91.1**	**88.6**	**63.3**	**59.9**	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	**56.4**	**81.3**	52.2	22.3	83.9	60.4	55.4	**47.0**	60.5	**79.9**	**87.5**	**65.0**	**62.7**	19.8	56.3	81.8
✓	–	**65.1**	88.8	88.9	73.0	16.6	44.9	79.3	52.2	**24.4**	**84.1**	**66.7**	55.4	**44.5**	**82.4**	**79.4**	86.9	60.6	**63.3**	**71.8**	56.3	82.4
	5	**64.2**	88.5	89.8	73.0	16.6	**53.1**	**82.0**	52.2	21.6	83.9	60.4	55.5	34.9	**81.4**	**77.7**	**87.9**	54.2	**62.6**	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	**78.7**	**92.8**	94.9	79.8	25.7	75.1	93.3	67.7	**44.9**	88.6	**87.0**	**64.1**	**66.8**	**98.9**	**94.9**	90.8	**83.7**	70.3	87.0	81.9	85.8
✓	–	**78.8**	**93.2**	**95.2**	**79.9**	25.7	73.5	93.3	**67.8**	**44.8**	88.2	**86.9**	**64.1**	65.8	98.8	**94.9**	**91.1**	**83.7**	71.6	**88.3**	**82.2**	86.0
	5	**78.6**	**93.1**	94.9	79.8	25.7	74.7	93.4	65.4	44.6	88.6	**87.0**	**64.1**	**66.7**	**98.9**	**94.9**	90.8	**83.7**	70.3	**88.3**	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	**80.0**	**93.5**	**97.5**	84.4	**19.4**	**72.5**	97.9	69.5	47.7	87.8	**98.5**	61.1	81.3	**99.5**	**89.5**	**92.5**	90.1	66.7	**90.0**	76.1	85.4
✓	–	**80.1**	93.0	97.4	87.3	19.2	**73.0**	98.0	70.0	47.7	87.6	**98.5**	61.1	80.9	99.5	**89.6**	**92.2**	89.2	66.7	**89.5**	**76.2**	85.5
	5	**80.3**	**93.6**	**97.6**	**87.4**	19.2	70.9	**98.1**	**71.7**	47.7	87.8	98.4	61.1	81.6	99.3	**89.3**	**92.5**	87.3	**70.9**	**90.0**	76.1	**85.9**

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.