ReCLIP: Refine Contrastive Language Image Pre-Training with Source Free Domain Adaptation (Supplementary Material)

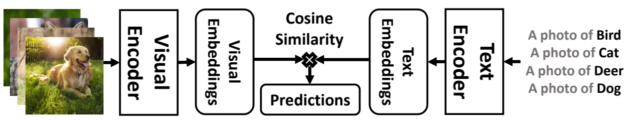

Figure 1. CLIP performs classification on target classes by comparing visual embeddings with the text embeddings generated from class names.

Appendix I: Background on CLIP

CLIP performs contrastive learning over 400 millions web-retrieved pairs of images and captions by pulling the visual and text representation near if they are from the same pair and away if they are not. At inference stage, CLIP makes classification prediction by matching the visual embeddings of query images with the text embeddings of categories names (wrapped in template text such as "`a photo of {}`", or a list of templates and uses the averaged embedding, as discussed in the main paper), and selects the category with the highest cosine similarity as prediction, as shown in Figure 1. CLIP is capable of performing classification over novel tasks without any training example, as long as the category names are provided. CLIP has demonstrated outstanding zero-shot classification accuracy, e.g. 76.3% top-1 accuracy on ImageNet without seeing any examples from the dataset. [25].

Appendix II: Algorithms

As described in Section 3.3 of the main paper, ReCLIP is composed of two parallel components that are designed for visual and text encoder fine-tuning, namely ReCLIP-V and ReCLIP-T. On top of ReCLIP-T and ReCLIP-V, we integrate the pseudo labels by filtering the commonly-agreed ones to produce high-confidence training signals for both sides. In this Section, we present the detailed description of ReCLIP-T and ReCLIP-V in Algorithm 1, and the pseudo label sharing in Algorithm 2.

Appendix III: Evaluation Benchmarks

For the main result from the paper, we have evaluated our model as well as the baseline methods on the validation or test splits from 22 image classification benchmarks, according to the setup as stated from Radford, et al [25]. The 22 benchmarks is composed of the one ablation datasets AID [32] that we used for hyper-parameter selection, and the 21 benchmarks (Caltech101 [21], CIFAR10 [20], CIFAR100 [20], ImageNet [9], SUN397 [33], Birdsnap [1], Country211 [25], DTD [7], EuroSAT [15], FER2013 [35], FGVC [22], Flowers [23], Food101 [2], GTSRB [27], MNIST [10], Oxford Pet [24], PCam [30], SST2 [25], RESISC45 [6], Cars [19], STL10 [8]) from the 27 benchmarks CLIP reported in Radford, et al [25], except: i) KITTI [13], UCF101 [26], VOC2007 [12], Kinetics700 [3] that are object detection or video classification benchmarks that are out of the scope of our discussion; ii) HatefulMemes [18] and CLEVR [17], where CLIP uses custom splits that are not released at the time of this submission. The detailed statistics on the number of images and the number of classes are reported in Table 1.

For comparison with POUF published score, we reported our scores on the Office-Home datasets. Office-Home contains 65 categories and 15588 images from four different domains: 2427 Art images, 4365 Clipart images, 4439 Product images and 4357 Real-World Images.

Appendix IV: Implementation Details

As mentioned in the main paper, we use AID to choose the best hyper-parameters for each baselines and evaluate them with the same hyper-parameters across the 22 datasets for SFDA evaluation.

For ReCLIP, we use learning rate of 10^{-3}, weight decay of 10^{-4}, momentum of 0.9, batch size of 64, maximum length of $\min\{5000 \text{ iterations}, 50 \text{ epochs}\}$ and SGD optimization on both visual and text encoders. For Birdsnap, Country211, SUN397 and ImageNet which have more than 200 classes, we use a batch size of 32 due to large memory occupation from text inputs to fit the training on a single V100 GPU. For Label Propagation, we use propagation strength $\alpha = 0.99$ and neighbor size $k = 20$. For datasets with more than 500 classes (Birdsnap, ImageNet), we notice the accuracy of pseudo labels generated by label propagation becomes unstable, and it requires additional hyper-parameter tuning to achieve good performance. To maintain

Algorithm 1 Visual and Text Encoder Self-Training: ReCLIP-V and ReCLIP-T

Require: Vision Language Pre-trained Model $M = \{M_v, M_t\}$
Require: Unlabeled Images $X = \{x_1, ..., x_n\}$
Require: Class Names $C = \{c_1, ..., c_m\}$
Require: Mode = ReCLIP-V or ReCLIP-T ▷ ReCLIP-V updates M_v with M_t frozen
▷ ReCLIP-T updates M_t with M_v frozen

for epoch $\leftarrow 1$ to Max Epoch **do**
  $\{t_1, ..., t_m\} \leftarrow M_t(\{c_1, ..., c_m\})$
  $\{v_1, ..., v_n\} \leftarrow M_v(\{x_1, ..., x_n\})$ ▷ Calculate Visual and Text Embeddings
  $U, S, V \leftarrow svd([t_1, ..., t_m])$, where $U = [e_1, ..., e_m]$
  $P_2 \leftarrow [e_2, ..., e_m][e_2, ..., e_m]^\top$ ▷ Prepare Projection Matrix with Singular Value Decomposition
  $\hat{t_i} \leftarrow \frac{t_i P_2}{\|t_i P_2\|}$
  $\hat{v_j} \leftarrow \frac{v_j P_2}{\|v_j P_2\|}$ ▷ Align Visual and Text Embeddings in Projection Space
  $L \leftarrow \{\hat{t_1}, ..., \hat{t_m}, \hat{v_1}, ..., \hat{v_n}\}$
  $\tilde{Y} \leftarrow$ Label_Propagation(L) ▷ Generate Pseudo Label through Label Propagation
  if Mode=ReCLIP-T **then**
    $\hat{Y} \leftarrow [\hat{v_1}, ..., \hat{v_n}]^\top [\hat{t_1}, ..., \hat{t_m}]$ ▷ Generate Predictions through Cosine-Similarity
    Loss$^T \leftarrow$ Cross-Entropy$(\hat{Y}, \tilde{Y})$
    Back-Propagation over M_t
  else if Mode=ReCLIP-V **then**
    $w_i \leftarrow \left(\sum_{\tilde{Y}_j = i} v_j\right) / \left(\sum_{\tilde{Y}_j = i} 1\right)$, for $i \in \{1, 2, ..., m\}$
    $\hat{w_i} \leftarrow \frac{w_i}{\|w_i\|}$ for $i \in \{1, 2, ..., m\}$ ▷ Calculate the average embeddings for each class i
    $\hat{Y} \leftarrow [\hat{v_1}, ..., \hat{v_n}]^\top [\hat{w_1}, ..., \hat{w_m}]$ ▷ Generate Predictions through Cosine-Similarity
    Loss$^V \leftarrow$ Cross-Entropy$(\hat{Y}, \tilde{Y})$
    Back-Propagation over M_v
  end if
end for

Algorithm 2 ReCLIP with Pseudo Label Sharing

Require: Component 1 $M^1 = \{M_v^1, M_t^1\}$ (for ReCLIP-V),
Require: Component 2 $M^2 = \{M_v^2, M_t^2\}$ (for ReCLIP-T)
Require: Unlabeled Images $X = \{x_1, ..., x_n\}$
Require: Class Names $C = \{c_1, ..., c_m\}$
Self-Training Adaptation Stage:
for epoch $\leftarrow 1$ to Max Epoch **do**
  $\hat{Y}^1, \tilde{Y}^1 \leftarrow$ ReCLIP-V(M^1, X, C)
  $\hat{Y}^2, \tilde{Y}^2 \leftarrow$ ReCLIP-T(M^2, X, C) ▷ ReCLIP-V/T generate predictions $\hat{Y}^1, \hat{Y}^2$ and pseudo labels $\tilde{Y}^1, \tilde{Y}^2$.
  Commonly Agreed Index Map $Q \leftarrow (\tilde{Y}_1 = \tilde{Y}_2)$ ▷ Boolean Index with $True$ indicates $\tilde{Y}^1$ agrees with $\tilde{Y}^2$.
  Loss$^V \leftarrow$ Cross-Entropy$(\hat{Y}^1[Q], \tilde{Y}^1[Q])$
  Loss$^T \leftarrow$ Cross-Entropy$(\hat{Y}^2[Q], \tilde{Y}^2[Q])$ ▷ Only calculate loss on entries where Q is True ($\tilde{Y}^1$ agrees with $\tilde{Y}^2$).
  Back-Propagate M_v^1 with Loss^V
  Back-Propagate M_t^2 with Loss^T
end for

Inference Stage:
$\hat{Y}^1 \leftarrow$ ReCLIP-V(M^1, X, C) ▷ Generate inference predictions from ReCLIP-T/V
$\hat{Y}^2 \leftarrow$ ReCLIP-T(M^2, X, C) ▷ At inference time, ReCLIP-T/V skip the pseudo label generation.
$\hat{Y} \leftarrow \frac{1}{2}(\hat{Y}^1 + \hat{Y}^2)$ ▷ Aggregate prediction logits from both ReCLIP-T/V for prediction.
return $\arg\max_i \hat{y}_{ji}$ as prediction for image x_j ▷ $Y = \{\hat{y}_{ji}\}$, where $\hat{y}_{ji}$ is probability of image x_j on class i.

	Average	AID [32]	Birdsnap [1]	Caltech101 [21]	CIFAR10 [20]	CIFAR100 [20]	Country211 [25]	DTD [7]	EuroSAT [15]	FER2013 [35]	FGVC [22]	Flowers [23]	Food101 [2]	GTSRB [27]	ImageNet [9]	MNIST [10]	Oxford Pet [24]	PCam [30]	SST2 [25]	RESISC45 [6]	Stanford Cars [19]	STL10 [8]	SUN397 [33]
Image Number		1500	2,149	9,146	10,000	10,000	21,100	1,880	5000	3,574	3,333	6,149	25,250	12,630	50,000	10,000	3,669	32,768	1,821	25,200	8,041	8,000	19,850
Class Number		30	500	102	10	100	211	47	10	8	100	102	102	43	1,000	10	37	2	2	45	196	10	397
AaD (h)	1.19	0.49	0.56	0.98	1.26	1.26	1.30	0.42	4.39	0.71	0.71	1.24	1.24	1.29	1.29	1.27	0.77	1.31	0.38	1.34	1.26	1.30	1.32
POUF (h)	6.18	4.51	7.07	5.61	5.80	5.71	7.30	5.50	5.60	3.73	5.02	5.82	6.38	6.41	13.58	5.74	4.13	6.79	4.91	6.33	5.97	5.92	8.19
ReCLIP (h)	2.35	0.68	0.97	2.94	1.62	2.68	1.58	1.08	1.82	0.90	1.24	2.73	5.66	3.82	3.23	2.19	0.95	2.99	0.61	3.12	4.17	2.18	4.63

Table 1. Metadata and Runtime comparison of AaD, POUF and ReCLIP of the 22 Evaluation Benchmarks. Time reported in the unit of hour (h).

stable performance, we turn off label propagation and simply use model predictions as pseudo labels on datasets with over 500 categories (Birdsnap, ImageNet). For all other datasets, we follow the exact process as described in Algorithm 1 and 2.

For both AaD and POUF, we have tested different hyper-parameters and report the the best performing setting, with learning rate of 10^{-3}, weight decay of 10^{-3}, momentum of 0.9, SGD optimization on AaD, and learning rate of 10^{-2}, weight decay of 10^{-3}, momentum of 0.9, SGD optimization on POUF. For both AaD and POUF, we extended their default training length to match our training length of ReCLIP, with batch size of $64 \times \min\{5000 \text{ iterations}, 50 \text{ epochs}\}$ steps on AaD, and batch size of $32 \times \min\{10000 \text{ iterations}, 100 \text{ epochs}\}$ steps on POUF.

For ReCLIP on Office-Home, we use the Real-World (Rw) domain to choose the hyper-parameter. We use SGD optimizer with learning rate of 10^{-2} on the visual encoder and 10^{-3} on the text encoder, batch size of 64 and 5000 iteration as maximum step across all domains. For label propagation, we use $k = 10$ due to the smaller dataset size.

Appendix V: Additional Ablation Results

Choice on Learnable Modules

In Table 2, we evaluate different learnable modules by comparing their fully-supervised fine-tuned performance. As suggested in [31], fine-tuning the normalization weights is shown to be efficient and stable, compared to fine-tuning the entire weights in self-training of ReCLIP.

Recent research [16] as well as POUF [28] also suggests that learnable prompts can also be effective in providing stable and fast performance improvement during the fine-tuning of Transformer [11, 29] based models. In Table 2, we perform Visual Prompt tuning following [16], and our own designed Text Prompt. Please refer to Appendix VII for more details.

As shown in Table 2, fine-tuning Layer-Norm weights from Visual Encoder has the best fully supervised accuracy on both CIFAR10 and CIFAR100, while fine-tuning Layer-Norm weights from Text Encoder has the best fully supervised accuracy on AID. As described in Section 2 from the Main Paper, on some datasets (including AID), the performance of CLIP is mainly limited by the poor quality text embeddings from inaccurate class names. In this case, fine-tuning the text encoder will achieve better performance as we observed. Table 2 results suggest the necessity of fine-tuning CLIP from both the visual and text side to handle different scenarios.

	CIFAR10	CIFAR100	AID	SUN397
Vanilla CLIP	95.54	76.48	64.87	67.25
Learnable Text Prompts	97.50	82.18	93.73	75.27
Learnable Visual Prompts [16]	96.70	80.68	74.27	68.09
Text Encoder Layer-Norm	97.32	83.30	**94.8**	**78.47**
Visual Encoder Layer-Norm	**97.8**	**85.16**	69.40	68.30

Table 2. Fully supervised fine-tuning accuracy of CLIP with different learnable modules on ablation datasets. On AID, fine-tuning weights from Text Encoder Layer-Norm is shown to be most effective; On CIFAR10 and CIFAR100, fine-tuning weights from Visual Encoder Layer-Norm is shown to be most effective.

	CIFAR10	CIFAR100	AID	SUN397
CLIP	95.60	78.22	68.73	67.97
ReCLIP (Transductive)	97.04	83.42	79.27	71.25
ReCLIP (Inductive)	96.92	82.30	79.87	74.53

Table 3. Inductive and Transductive performance comparison of ReCLIP on ablation datasets.

Inductive Results

We perform the SFDA evaluation in Table 1 from the main paper, to follow the protocols of AaD [34] and POUF [28] and to fully utilize the test examples. However, ReCLIP can also be applied in the inductive manner, so that the adaptation only has to be performed once for the target domain, and the adapted model will be effective on new and unseen examples of the target domain. In Table 3 we run ReCLIP in an inductive setting, where ReCLIP performs self-training on the training split of a dataset (0.5 to 5 GPU-Hour), and inference on the test split (similar to CLIP inference time). ReCLIP achieves similar improvements in the inductive setting as in the transductive setting.

Pseudo Label Quality

In Table 4 we report the pseudo label accuracy of ReCLIP. We report the pseudo label accuracy from ReCLIP on the first epoch, before the self-training algorithm updates the model weights. From Table 4 we observe that the label propagation over projected visual and text embeddings has obtained ReCLIP pseudo labels with consistent improved accuracy over CLIP, only except Birdsnap and ImageNet which have more than 500 categories, as we discussed in Appendix IV. The results from Table 4 demonstrate the effectiveness of our version of the label propagation method in generating reliable pseudo labels for vision-language models. More discussion on pseudo label generation is also covered in Section 4.3.2 of the main paper.

Appendix VI: Time Analysis

We present the runtime required by SFDA methods, namely AaD, POUF and ReCLIP, in Table 1. We matched all methods to be at the same training steps for fair comparison. As shown by the result, AaD takes an average of 1.19 hours to adapt, ReCLIP takes 2.35 hours and POUF takes 6.18 hours. ReCLIP is not much slower than AaD although ReCLIP trains two sets of encoders at the same time, except on datasets with more categories due to the time required for the Label Propagation process. However, POUF is much slower than both AaD and ReCLIP, due to its less efficient implementation. However, all three algorithms are very efficient as the adaptation only has to be applied once for each new target domain.

Appendix VII: Details on the design of learnable Language Prompt

What is Language Prompts

During the large-scale contrastive pre-training, CLIP [25] was trained to match visual-text embedding between training images with their caption sentences such as ``A Golden Retriever dog sitting on grass''. However, during inference time, category descriptions are usually provided in the form of phrases such as ``Golden Retriever'' or just ``Dog'' instead of captions in complete sentences. To mitigate this gap, CLIP has proposed to use templates to wrap the category description phrase into complete sentences to generate better text embeddings.

For optimal performance, CLIP [25] further claims that specific templates which provide contexts to the category names might help generate better text embeddings for classification. For example, CLIP finds the template prompt ``A photo of {category name}, a type of pet'' works the best for OxfordIII-Pet [24]. CLIP has designed different lists of template prompts for all datasets it was evaluated on. The details can be found on their official GitHub repository https://github.com/openai/CLIP/blob/main/data/prompts.md.

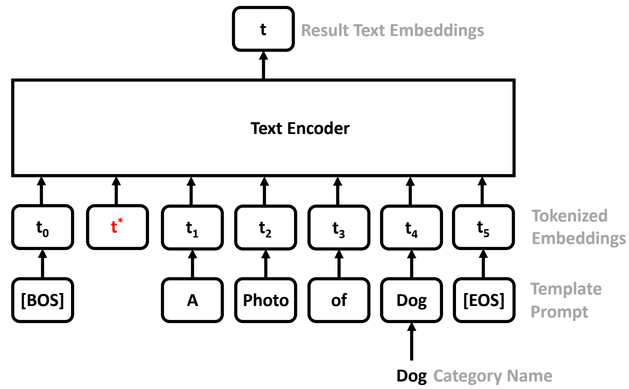

Figure 2. Demonstration of the design of Learnable Prompt. t^* represents a learnable token embedding that is inserted at the beginning of the sequence of inputs to the transformer-based text encoder. "BOS" and "EOS" stands for "beginning of sentence" and "end of sentence" and they serve as the special tokens for the text encoder to identify the beginning and end of the input sentence.

Learnable Language Prompts

As demonstrated by CLIP [25], the wisely chosen template prompts might play a vital role in generating accurate text embeddings. However, this process largely depends on the heuristic design. Our goal for the learnable language prompt design is to make the prompt learnable and to avoid having different template prompts for different datasets. Additionally, this can also be an efficient and stable way to fine-tune the performance of CLIP.

We start from the default template prompt ``A photo of {category name}'', and insert an additional learnable token embedding $t^*$ at the beginning of the sentence, right after the Begin-Of-Sentence (BOS) token, as shown in Figure 2. $t^*$ is initialized with the same embedding value of word ``is'' for reasonable performance before it is fine-tuned. During the fine-tuning process, token t^* is made to be learnable while token embeddings for all other words are fixed.

Appendix VIII: Limitation and Future Work

As mentioned in the Implementation Details section, we have observed that on datasets with more than 500 classes (Birdsnap, ImageNet), the accuracy of pseudo labels generated by label propagation becomes unstable, and it requires additional hyperparameter tuning to achieve good perfor-

	Average	AID [32]	Birdsnap [1]	Caltech101 [21]	CIFAR10 [20]	CIFAR100 [20]	Country211 [25]	DTD [7]	EuroSAT [15]	FER2013 [35]	FGVC [22]	Flowers [23]	Food101 [2]	GTSRB [27]	ImageNet [9]	MNIST [10]	Oxford Pet [24]	PCam [30]	SST2 [25]	RESISC45 [6]	Stanford Cars [19]	STL10 [8]	SUN397 [33]
CLIP *repro*	69.83	68.73	52.48	91.63	95.60	78.22	31.84	55.37	60.00	56.39	31.59	79.04	93.08	50.59	75.52	76.23	93.62	62.43	68.92	69.66	77.88	99.36	67.97
ReCLIP (pseudo label)	72.54	74.50	43.25	91.91	96.56	81.40	26.30	59.04	73.36	57.15	36.33	82.55	93.95	60.64	25.11	82.85	94.77	62.46	68.86	77.63	77.66	99.52	70.54

Table 4. ReCLIP pseudo label Quality. Results are generated with vanilla CLIP ViT-L/16 checkpoint, on the first epoch of ReCLIP before the training algorithms update the model weights.

mance. To maintain stable performance, we have turned off label propagation and simply used model predictions as our pseudo labels on datasets with over 500 categories. Studies on how the hyper-parameters influence the label propagation performance on datasets with more than 500 categories will be important future work to further improve ReCLIP.

Another future direction will be the utilization of augmentation consistency. Augmentation Consistency has been shown to be a very powerful unsupervised training signal and has been widely applied in unsupervised methods [4, 5, 14]. Due to the scope and complexity of this project, we have not explored the usage of augmentation consistency in source-free domain adaptation. It will be important future work to explore the combination of the current ReCLIP with augmentation consistency to further improve the adaptation performance.

References

[1] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2014. 1, 3, 5

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 1, 3, 5

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 5

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 5

[6] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 1, 3, 5

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 3, 5

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 1, 3, 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 3, 5

[10] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 1, 3, 5

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl-

vain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 1

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 5

[15] 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*, 12(7):2217–2226, 2019. 1, 3, 5

[16] 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. 3

[17] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 1

[18] 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. *Advances in Neural Information Processing Systems*, 33:2611–2624, 2020. 1

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 1, 3, 5

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 3, 5

[21] Li, Andreeto, Ranzato, and Perona. Caltech 101, Apr 2022. 1, 3, 5

[22] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 1, 3, 5

[23] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 1, 3, 5

[24] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 1, 3, 4, 5

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 3, 4, 5

[26] 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. 1

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012. 1, 3, 5

[28] Korawat Tanwisuth, Shujian Zhang, Huangjie Zheng, Pengcheng He, and Mingyuan Zhou. Pouf: Prompt-oriented unsupervised fine-tuning for large pre-trained models. *arXiv preprint arXiv:2305.00350*, 2023. 3

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[30] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018. 1, 3, 5

[31] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 3

[32] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017. 1, 3, 5

[33] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 1, 3, 5

[34] Shiqi Yang, Shangling Jui, Joost van de Weijer, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. *Advances in Neural Information Processing Systems*, 35:5802–5815, 2022. 3

[35] Lutfiah Zahara, Purnawarman Musa, Eri Prasetyo Wibowo, Irwan Karim, and Saiful Bahri Musa. The facial emotion recognition (fer-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (cnn) algorithm based raspberry pi. In *2020 Fifth international conference on informatics and computing (ICIC)*, pages 1–9. IEEE, 2020. 1, 3, 5