Towards Fine-Grained Adaptation of CLIP via a Self-Trained Alignment Score

Supplementary Material

This supplementary material complements the main paper by providing additional insights and details. Secs. 1.1 and 1.2 provide implementation details and dataset statistics to support reproducibility. Secs. 2.1 to 2.3 discuss extended experimental results, including analyses of zero-shot techniques in pseudo-labeling, comparisons with other VLMs, and an evaluation of model complexity. Lastly, Sec. 3 outlines the limitations of our approach and potential directions for future research. We provide the detailed pseudo-code of our method in Algorithm 1, along with a list of symbols and notations in Sec. 4.

1. Additional Implementation Details

1.1. Implementation and Computation Details:

In our unsupervised fine-tuning approach, we focus on adjusting the layer normalization weights of the image encoder and CDA. This method has been shown to be both effective and stable for adapting models under noisy supervision [2, 14]. Input images are standardized to a size of 224 × 224. During training, we use RandomResizedCrop, Flip, and RandAugment [3] as strong augmentation methods, and CenterCrop as the only weak augmentation for `FAIR`, since we augment the image features for pseudo-labeling using RandomCrop, and RandomResizedCrop for `FAIR`-g. We utilize the AdamW optimizer [11] with a cosine learning rate schedule. For all experiments, we set the learning rate to 1×10^{-4} for all datasets, except for Food101 and SUN397, where it is set to 1×10^{-6}. We use a batch size of 32 for all datasets, training for 15 epochs. For all crop-based experiments, we set the hyperparameters to $\alpha = 0.5$ and $\beta = 0.9$, as used in WCA [10]. Additionally, we use $(N, k) = (16, 4)$, where N denotes the number of crops generated per image, and k represents the top-k crops selected from a set of N randomly sampled crops. Our method is implemented using PyTorch, and all experiments are conducted on a single NVIDIA A100-SXM4-40GB GPU. The LLM descriptions used in our study are derived from CuPL [13], which automates description generation using carefully designed prompts for LLMs. To ensure fair comparisons, we reproduce the results of SOTA methods using their official codebases. We use VISSL [4] to standardize dataset splits across all SOTA methods, ensuring consistency, as different methods use varying dataset partitions. In both the main paper's ablation experiments and the supplementary experiments, we evaluate six of the thirteen datasets, excluding Caltech101, AID, CUB, RESIS45, CIFAR-100, Food101, and SUN397. Following ReCLIP's procedure [8], we select six diverse datasets—EuroSAT (satellite imagery), UCF-101 (action recognition), Flowers and Cars (fine-grained recognition), DTD (texture), and Pets (intra-class variation)—to balance variety and experimental depth. These smaller datasets enable more extensive experiments while covering diverse domains and difficulty levels for robust generalization evaluation. A comparison of `FAIR` using RN50 is not feasible, as both `FAIR` and the relevant baselines (e.g., LaFTer [12], ReCLIP [8], DPA [1]) are specifically designed for transformer-based image encoders.

1.2. Dataset Statistics and Splits:

We conduct experiments on 13 diverse datasets, summarized in Tab. 1, which outline key details such as the number of text descriptions per class, the number of classes, and the sizes of the training and test sets.

Dataset	Abbr.	Desc/Class	Classes	Train	Test
Caltech101	Caltech	30	100	4,403	6,645
DTD	DTD	60	47	3,760	1,880
EuroSAT	ESAT	25	10	10,000	5,000
Food101	Food	30	101	75,750	25,250
Flowers102	Flower	20	102	4,093	2,463
Oxford Pets	OxPets	20	37	3,680	3,669
SUN397	SUN	30	397	76,129	21,758
Stanford Cars	StCars	90	196	8,144	8,041
CIFAR10	CIFAR10	30	10	50,000	10,000
CIFAR100	CIFAR100	40	100	50,000	10,000
UCF101	UCF	50	101	9,537	3,783
CUB-200-2011	CUB	31-40	200	5,994	5,794
RESISC45	UCF	50	45	25,200	6,300
AID	AID	50	30	7,000	3,000

Table 1. Detailed dataset statistics.

2. Additional Experiments and Comparisons

2.1. Empirical Study of Zero-Shot Techniques in Pseudo-labeling

We conduct experiments with CODER [16], a recent method that has demonstrated SOTA performance in zero-shot learning, to further evaluate the effectiveness of these techniques for pseudo-labeling, particularly focusing on the benefits of fine-grained interactions achieved through localized image features and self-learned class description anchors in our approach. CODER employs an LLM to generate five distinct types of prompts, which are then used to construct dense, class-specific textual representations. These descriptions are systematically compared to global

image features, enabling a unimodal fine-grained interaction. CODER's standout contribution lies in using an Auto Text Generator (ATG) to effectively utilize these rich textual descriptions. Compared to CuPL [13], CODER utilizes denser and more systematic textual descriptions for each class. This detailed representation provides a performance edge in tasks requiring fine-grained image classification. For example, CODER achieves an average improvement of 1.38% over FAIR-g (68.87% to 70.25%), as shown in Tab. 2. This improvement highlights the value of dense textual neighbors and their integration with visual features, akin to our FAIR-g approach but with better-calibrated textual embeddings. While the dense textual embeddings improve CODER's performance as a pseudo-labeler, our proposed method, FAIR, achieves SOTA results, surpassing CODER by a significant margin (5.13% average accuracy). This improvement is attributed to the fine-grained interactions between localized image features and learned CDA during pseudo-labeling. Our method's robust design enables it to outperform other approaches across multiple benchmarks, demonstrating the efficacy of FAIR as a generalized pseudo-labeling framework.

Method	DTD	ESAT	Flowers	OxPets	StCars	UCF	Avg
FAIR (w/ WCA PL)	55.16	66.10	71.86	89.45	60.68	69.39	68.77
FAIR-g	53.98	67.92	72.47	89.94	59.72	69.18	68.87
FAIR (w/ CODER PL)	55.80	77.30	73.69	86.56	59.33	68.81	70.25
FAIR (ours)	**62.07**	**91.92**	**75.72**	**90.52**	**61.83**	**73.54**	**75.93**

Table 2. Comparison of average top-1 accuracy between FAIR and other SOTA zero-shot methods used as pseudo-labelers. Notably, FAIR-g, our simplified baseline that uses global views instead of localized crops, incorporates CuPL [13] as its pseudo-labeling method. FAIR 's pseudo-labeling weight has been used in all three other methods for a fair comparison. In contrast to the other three methods, FAIR refines the alignment score function using fine-grained interactions during pseudo-labeling.

2.2. Comparison with latest VLMs

Table 3 presents a comparison between our method, FAIR, and state-of-the-art (SOTA) methods using a different vision-language model (VLM), MetaCLIP [15] (ViT-B/32). As shown in Table 3, FAIR consistently outperforms SOTA methods across the ablation datasets. Notably, FAIR maintains its effectiveness even when the underlying VLM changes, demonstrating robustness and adaptability across diverse VLMs.

Method	DTD	ESAT	Flowers	Pets	Cars	UCF	Avg
ZS CuPL	60.96	51.29	69.91	88.50	68.23	64.10	67.17
DPA	56.90	88.14	**76.86**	89.80	69.40	72.10	75.53
FAIR (ours)	**70.05**	**89.72**	74.38	**90.73**	**70.87**	**78.01**	**78.96**

Table 3. Performance comparison of methods using MetaCLIP (ViT-B/32)

2.3. Comparison of Model Complexity

Table 4 provides a detailed comparison of FAIR with SOTA methods in terms of computational cost, measured by the total number of optimized parameters. Since FAIR backpropagates only through the top-k most informative image patches, FAIR enables competitive performance with fewer optimized parameters. Unlike ReCLIP, which requires backpropagation through two encoders and label propagation, FAIR trains a single encoder and achieves faster convergence (15 vs. 50 epochs), significantly improving computational efficiency even when accounting for the image cropping overhead.

Furthermore, Table 5 highlights that both fine-tuning strategies—FAIR-LN and FAIR-KAdaptation—yield nearly identical performance across datasets, confirming the adaptability and robustness of our approach across various parameter-efficient configurations.

Method	#Params	Acc. (%)
Zero-shot CLIP	0	43.84
UPL	8.2K	51.88
POUF	2.0K	62.90
LaFTer	1.14M	69.96
ReCLIP	65.5K	70.80
DPA	45.1K	79.94
FAIR (ours)	45.1K	91.92

Table 4. Accuracy and trainable parameters on EuroSAT (ViT-B/32).

Method	DTD	ESAT	Flower	OxPets	StCars	UCF101	Avg
FAIR-LoRA [7]	58.30	71.82	72.51	89.78	57.41	71.27	70.18
FAIR-KAdaptation [5]	61.86	90.62	**76.86**	**90.81**	60.50	72.80	75.58
FAIR-LN (ours)	**62.07**	**91.92**	75.72	90.52	**61.83**	**73.54**	**75.93**

Table 5. Accuracy (%) comparison of CLIP fine-tuning techniques on FAIR. All experiments use ViT-B/32 as the backbone.

3. Limitations and Future Work

While FAIR demonstrates robust performance on fine-grained datasets such as EuroSAT [6], it exhibits limitations on the StanfordCars dataset [9]. This dataset presents unique challenges stemming from its detailed intra-class variations (e.g., different car models from the same manufacturer) and high inter-class similarities (e.g., visually analogous cars from different manufacturers). In the absence of labeled data, pseudo-labeling becomes unreliable when differentiating between visually similar classes, potentially causing cascading errors during training. These difficulties are compounded by domain-specific biases in pretrained models, which are typically optimized for more

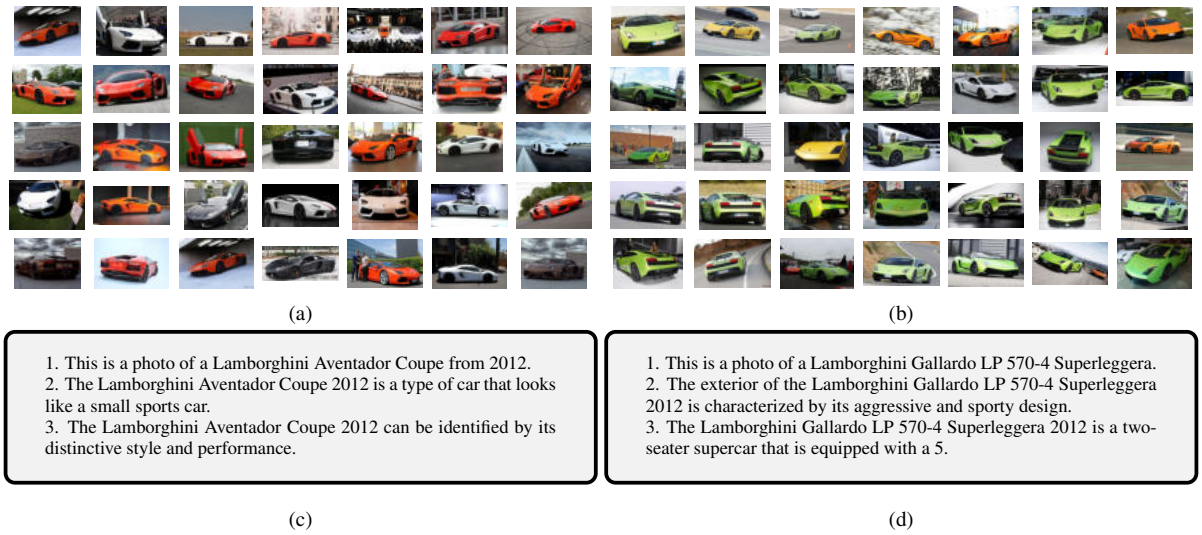

Figure 1. Comparison of images between two visually similar car models from the same manufacturer: (a) Lamborghini Aventador Coupe 2012 and (b) Lamborghini Gallardo LP 570-4 Superleggera 2012. These images highlight subtle design differences, posing challenges for accurate classification in fine-grained recognition tasks. (c) and (d) show the top three descriptions that best match a randomly selected image from each class.

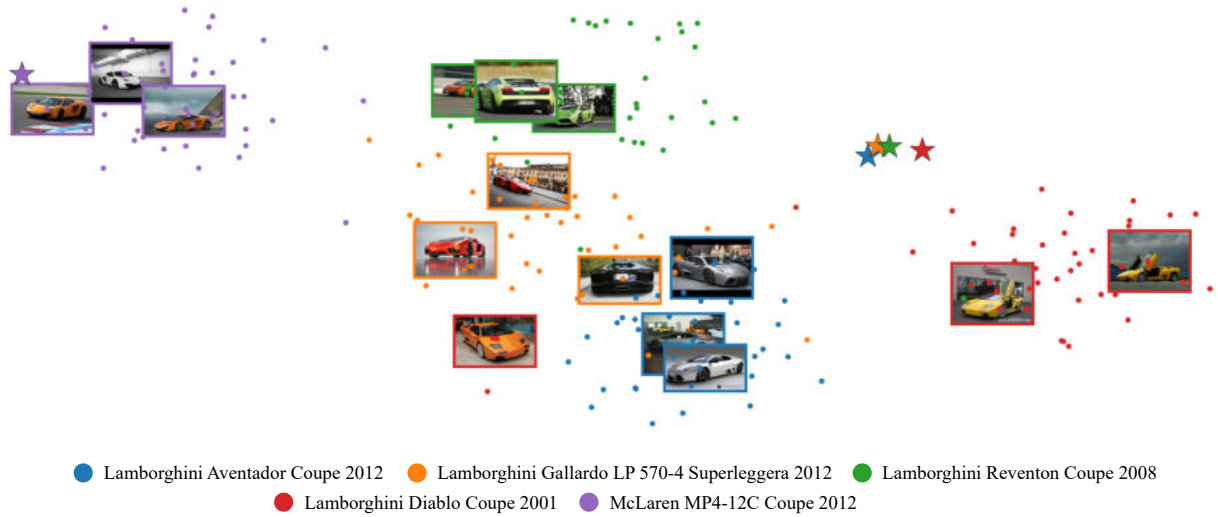

Figure 2. t-SNE visualization of embeddings from five similar car classes in the StanfordCars [9] dataset. Dots (●) represent image embeddings, while stars (★) denote Class Description Anchors. Overlaid images correspond to three random points per class, illustrating the overlap and proximity between embeddings of visually similar car models.

common object categories and thus struggle to capture the nuanced distinctions required in fine-grained recognition tasks like those in the StanfordCars dataset. Consequently, these factors collectively constrain our method's generalizability and performance in highly specialized visual recognition contexts.

An example of the challenges faced in fine-grained classification is the comparison between Lamborghini Aventador Coupe 2012' and Lamborghini Gallardo LP 570-4 Superleggera 2012', as shown in Fig. 1a and Fig. 1b, respectively. These Lamborghini models exhibit striking stylistic similarities, with distinctions often limited to subtle design elements such as headlight shapes or air intake structures. This difficulty is further illustrated in Fig. 2, where embeddings of these two classes, alongside three other visually similar car models, are projected in a t-SNE space. The figure demonstrates significant overlap in embedding space, with multiple instances of inter-class confu-

sion. This overlap is exacerbated in self-trained settings, as pseudo-labeling struggles to delineate fine-grained distinctions without labeled data. The embedding proximity highlights the inherent challenge of disentangling subtle visual differences when relying on self-supervised pseudo-labels.

To mitigate these limitations, future work should focus on generating denser, fine-grained descriptions that capture subtle class-specific features, such as detailed structural components or stylistic cues, and on improving pseudo-label generation for fine-grained recognition tasks. As shown in Fig. 1c and Fig. 1d, current LLM-generated descriptions [13] lack the necessary specificity to differentiate between visually similar subclasses, thereby exacerbating intra-class ambiguity. Further advancements could involve domain-adaptive fine-tuning to address the domain-specific biases in pretrained models, as well as incorporating attention mechanisms to refine the model's focus on relevant details.

Algorithm 1 `FAIR` self-training

Require: CLIP vision encoder, E_v^{Θ} where Θ represents all the affine parameters in the LayerNorm layers;
Unlabeled images of a target dataset $\mathcal{X}_t = \{x_i\}_{i=1}^{N}$;
An LLM model $h(\cdot)$;
Set of class names $\mathcal{Y}$;
Class names to integer map, $C : \mathcal{Y} \rightarrow 1 \cdots C$;
Weak augmentation $\alpha(\cdot)$;
Strong augmentation $\mathcal{A}(\cdot)$;
Number of epochs `MaxEpochs`;
Batch size `B`

function INITCDA(E_t, $\mathcal{Y}$, h)
  $\mathbf{Z}^* \leftarrow \{\emptyset\}_{j=1}^{C}$
  for each $y \in \mathcal{Y}$ **do**
    $\mathbf{t} \leftarrow h(y)$ ▷ Prompt the LLM to extract M number of descriptions for class y
    $\mathbf{Z}_j \leftarrow \frac{1}{M} \sum_{i=1}^{M} E_t(\mathbf{t})$ ▷ Take the average of the description embedding $Z_j^* \in \mathbb{R}^{1 \times d}$ for class y
  return $\mathbf{Z}^*$

function ADAPTIVEWEIGHT($\psi_{\texttt{FAIR}}$)
  $S_{i,x}, S_{j,x} \leftarrow$ top-2($\psi_{\texttt{FAIR}}$) ▷ Get the top-2 logits by evaluating the `FAIR` similarity function
  $\gamma_x \leftarrow S_{i,x} \cdot (S_{i,x} - S_{j,x})$
  return γ_x

$\mathbf{Z}^* \leftarrow$ INITCDA(E_t, $\mathcal{Y}$, h) ▷ Initialize CDA
for `epoch` $\leftarrow 1$ to `MaxEpochs` **do**
  $\mathbf{x} \leftarrow$ SAMPLEMINIBATCH($\mathcal{X}_t$, B) ▷ $\mathbf{x} \in \mathbb{R}^{B \times W \times H \times 3}$

  With **no Back-Propagation**:
    $\mathbf{p}(\mathbf{x}) \leftarrow \{p_i = \phi(\alpha(\mathbf{x}), \lambda_i \min(W, H)) \mid i = 1, \ldots, N\}$ ▷ Extract N random crops from the weakly augmented image $\alpha(x)$
    $\mathbf{f}, \mathbf{f}^{[\texttt{CLS}]} \leftarrow E_v^{\Theta}(p(\mathbf{x}))$ ▷ Local view representation features $\in \mathbb{R}^{B \times d}$
    $\widetilde{\Theta} \leftarrow \begin{bmatrix} \text{sim}(\mathbf{f}_1, \mathbf{Z}_1^*) & \cdots & \text{sim}(\mathbf{f}_1, \mathbf{Z}_C^*) \\ \vdots & \ddots & \vdots \\ \text{sim}(\mathbf{f}_N, \mathbf{Z}_1^*) & \cdots & \text{sim}(\mathbf{f}_N, \mathbf{Z}_C^*) \end{bmatrix}$
    $\widetilde{w_i} \leftarrow \frac{\text{sim}(f^{[\texttt{CLS}]}, f_i^{[\texttt{CLS}]})}{\sum_{l=1}^{N} \text{sim}(f^{[\texttt{CLS}]}, f_l^{[\texttt{CLS}]})}$
    $\mathcal{I}_k \leftarrow \text{argsort}(\widetilde{\mathcal{W}})[:k]$ ▷ Select the top-k crop indices based on $\widetilde{\mathcal{W}}$
    $\psi_{\texttt{FAIR}}(x, y|\mathbf{p}, E_v, \mathbf{Z}^*, C) \leftarrow \sum_{i=1}^{N} \widetilde{w_i} \widetilde{\Theta}_{ij}|_{j=C(y)} \mathbb{I}_{\{i \in \mathcal{I}_k\}}$ ▷ Evaluate `FAIR` similarity function
    $\hat{y} \leftarrow \arg\max_{y \in \mathcal{Y}} (\psi_{\texttt{FAIR}}(x, y|\mathbf{p}, E_v, \mathbf{Z}^*, C)$ ▷ Compute the pseudo-labels
    $\gamma_{\mathbf{x}} \leftarrow$ ADAPTIVEWEIGHT($\psi_{\texttt{FAIR}}(x, y|\mathbf{p}, E_v, \mathbf{Z}^*, C)$) ▷ Compute the adaptive weights

  $p_{\mathcal{A}(x)} \leftarrow \text{softmax}(\text{sim}(E_v^{\Theta}(\mathcal{A}(\mathbf{x})), \mathbf{Z}^{*T}), axis = 1)$ ▷ Strongly-augmented counterpart
  $\mathcal{L}_{st} \leftarrow \gamma_x \cdot \text{cross\_entropy}(p_{\mathcal{A}(x)}, \hat{\mathbf{y}})$ ▷ Self-training loss
  $\mathcal{L}_{reg} \leftarrow -\frac{1}{C} \sum_{j=1}^{C} \log(\bar{p}_{\mathcal{A}(x),j})$ ▷ Fairness regularization loss
  $\mathcal{L} \leftarrow \mathcal{L}_{st} + \mathcal{L}_{reg}$
  Back-Propagate over Θ and $\mathbf{Z}^*$ on $\mathcal{L}$

4. List of Symbols and Notation

Symbol	Description
	List of Symbols
E_v	The visual encoder of CLIP
E_t	The natural language encoder of CLIP
$\mathcal{D}_t$	The target dataset
$\mathcal{X}_t$	The set of unlabeled images in the target dataset
x_i	An arbitrary unlabeled image sampled from $\mathcal{X}_t$ in $\mathcal{D}_t$
$\mathcal{Y}$	The set of unique class names
y	The class name corresponding to the image x_i, sampled from $\mathcal{Y}$
c	The number of classes in $\mathcal{D}_t$
H	The height of the image x
W	The width of the image x
$\mathbf{t}$	Hand-crafted prompts
$\mathbf{Z}$	The natural language embeddings generated from $Z = E_t(\mathbf{t})$
f	The visual embeddings generated from $f = E_v(\mathbf{x})$
$\mathbf{Z}^*$	The learnable description embeddings
d	The dimensionality of the feature space
ψ	The cosine similarity function
ψ_{clip}	The cosine similarity function for CLIP
ψ_{CuPL}	The cosine similarity function for CuPL
ψ_{WCA}	The cosine similarity function for WCA
$\psi_{\texttt{FAIR}}$	The learnable cosine similarity function for `FAIR`
$h(\cdot)$	The LLM model
M	The number of generated descriptions per class
N	The number of crops generated per image
$\mathbf{p}(x)$	The localized crops generated from the image x
$\phi(\cdot)$	The function performs a random crop on the input image
λ_i	A random variable drawn from a uniform distribution $\lambda_i \sim U(\alpha, \beta)$
α	The lower bound for the uniform distribution U
β	The upper bound for the uniform distribution U
Θ	The crop-description similarity matrix
$\mathcal{W}$	The set of weights for image crops
$\mathcal{V}$	The set of weights for text descriptions
$\mathcal{A}(.)$	The strongly-augmented function
$\bar{p}_{\mathcal{A}(x)}$	The model's average prediction from the strongly augmented images across the batch
$S_{i,x}$	The highest similarity score derived from the learnable similarity function $\psi_{\texttt{FAIR}}$
$S_{j,x}$	The second-highest similarity score from the learnable similarity function $\psi_{\texttt{FAIR}}$
γ_{x}	The weight of the pseudo-label for image x
$\mathcal{L}_{\text{st}}$	The self-training loss function
$\mathcal{L}_{\text{reg}}$	The fairness regularization loss function

References

[1] Eman Ali, Sathira Silva, and Muhammad Haris Khan. Dpa: Dual prototypes alignment for unsupervised adaptation of vision-language models. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 6083–6093, February 2025. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 1

[4] Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. Vissl. https://github.com/facebookresearch/vissl, 2021. 1

[5] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient model adaptation for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 817–825, 2023. 2

[6] 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. 2

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022. 2

[8] Xuefeng Hu, Ke Zhang, Lu Xia, Albert Chen, Jiajia Luo, Yuyin Sun, Ken Wang, Nan Qiao, Xiao Zeng, Min Sun, et al. Reclip: Refine contrastive language image pre-training with source free domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2994–3003, 2024. 1

[9] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 2, 3

[10] Jinhao Li, Haopeng Li, Sarah Erfani, Lei Feng, James Bailey, and Feng Liu. Visual-text cross alignment: Refining the similarity score in vision-language models. In *International Conference on Machine Learning*, 2024. 1

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 1

[12] Muhammad Jehanzeb Mirza, Leonid Karlinsky, Wei Lin, Horst Possegger, Mateusz Kozinski, Rogerio Feris, and Horst Bischof. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. *Advances in Neural Information Processing Systems*, 36:5765–5777, 2023. 1

[13] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 1, 2, 4

[14] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1

[15] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. 2

[16] Chao Yi, Lu Ren, De-Chuan Zhan, and Han-Jia Ye. Leveraging cross-modal neighbor representation for improved clip classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27402–27411, 2024. 1