Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 264 (2025) 167–176

Procedia Computer Science

www.elsevier.com/locate/procedia

International Neural Network Society Workshop on Deep Learning Innovations and Applications

CLIP-DoRA: Weight-decomposed Low-rank Adaptation for Efficient Vision-Language Models

Jesús M. Rodríguez-de-Vera[a,b,*], Imanol G. Estepa[a,b], Bhalaji Nagarajan[c], Petia Radeva[a,d]

[a]*Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007-Barcelona, Spain*
[b]*Computer Vision Center, Cerdanyonla del Vallès, Barcelona, Spain*
[c]*Barcelona Supercomputing Center (BSC)*
[d]*Institut de Neurosciències, Universitat de Barcelona, Passeig de la Vall d'Hebron 171, 08035-Barcelona, Spain*

Abstract

Foundational Vision-Language (V-L) models, such as CLIP, have progressed computer vision research by providing general "shared" latent spaces for image and text modalities. However, training these models, or even fine-tuning them, demands substantial computational resources and has a high environmental cost. In this work, we propose **CLIP-DoRA**, a method leveraging weight-decomposed low-rank adaptation for parameter-efficient V-L fine-tuning. We explore the potential of less explored low-rank-based methods for V-L fine-tuning, and provide empirical proof of the benefits of weight-decomposed fine-tuning. Our extensive experiments across 11 few-shot datasets and 4 domain generalization benchmarks demonstrate that CLIP-DoRA outperforms existing PEFT methods with average improvements of up to 0.28% over the previous state-of-the-art. Furthermore, CLIP-DoRA shows competitive results in complex tasks like medical image segmentation using only 1.5% of the trainable parameters, proving its potential as a more sustainable and accessible solution for V-L model adaptation. These findings highlight the robustness and versatility of CLIP-DoRA in developing efficient and environmentally friendly computer vision solutions.

Keywords: Vision-Language Models; Parameter-Efficient Fine-Tuning; Low-Rank Adaptation

1. Introduction

Foundational models have become a key to the recent advent of Artificial Intelligence, particularly in Deep Learning. The big scalability of models (up to trillions of parameters) enables the exploitation of web-scale data and enables them to maintain high performance in most computer vision tasks. Large language models (LLMs) and Multi-modal models (MMMs) are two prominent types of foundational models. LLMs, such as Llama [1] and Mistral [2], are designed to process and generate human language. MMMs, like vision-language (V-L) models, provide a "shared" latent space for different modalities (e.g., images and text). V-L Models such as CLIP [3] and ALIGN [4] produce

* Corresponding author.
E-mail address: j.molina.rdv@ub.edu

1877-0509
10.1016/j.procs.2025.07.128

semantically rich representations that can be used in several image and text downstream tasks. For this reason, they play a pivotal role in the current progress of the computer vision field.

Despite its success, large foundational models pose several challenges. First, training such large models (ranging from 150M to 1.37B in the case of CLIP [3]) requires enormous computational resources, which results in high energy consumption. In addition, according to the neural scaling laws of LLMs [5] and vision transformers (ViTs) [6], the performance of these models increases with their size. The larger the models, the more effectively they can use large-scale datasets. In other words, current development requires an ever-increasing amount of computation for training, resulting in a constantly growing impact on the environment [7]. Ultimately, the high amount of resources required just to use these models on inference makes it impossible for small and medium-sized research centers to create and pre-train these foundational models. In some cases, even fine-tuning requires several high-end GPUs. For instance, Alpaca [8] required 8×80GB NVIDIA A100s to fine-tune Llama-7B, the smallest variant.

In recent years, Parameter-Efficient Fine-Tuning (PEFT) has emerged as a solution to address these challenges. PEFT methods are characterized by keeping most of the parameters of a large model frozen, while only training a small amount of weights for any given downstream task [9]. This reduces the computational burden of fine-tuning process. Recent surveys classify PEFT methods in three main categories: (1) *selective*, where a subset of the original weights is trained; (2) *additive*, where new elements are added to the architecture and trained; and (3) *reparameterized*, where the original weights are represented in a more computationally efficient way [9]. Although there is extensive literature on PEFT methods under low data regimes, V-L models have mainly focused on prompt-based and adapter-based methods, overlooking potentially useful approaches like Low-Rank Adaptation (LoRA) [10]. Recent methods such as CLIP-LoRA [11] proved that LoRA can outperform existing V-L PEFT methods in few-shot settings.

Motivation and Contributions. While CLIP-LORA demonstrated that low-rank adaptation outperforms existing PEFT methods in few-shot settings, its suitability for other tasks was not explored. It remained unknown if more advanced alternatives could further improve the performance obtained by CLIP-LoRA even more [11]. Motivated by its success, we aimed to explore and enhance the potential of low-rank adaptation methods for efficient fine-tuning of V-L models. Our contributions are listed as follows: **(1) CLIP-DoRA.** We introduce the method *CLIP-DoRA*, which leverages weight decomposition [12] to achieve more stable and faster convergence. CLIP-DoRA does not require heavy hyperparameter tuning and can be used in any transformer-based pre-trained V-L model. **(2) Extensive few-shot analysis.** We perform a detailed analysis and comparison of CLIP-DoRA with other methods in few-shot settings. Our results show that CLIP-DoRA consistently outperforms the previous SoTA, CLIP-LoRA, across different backbones and varying numbers of shots, establishing a new benchmark in few-shot learning scenarios. **(3) Domain Generalization Evaluation.** We conduct an in-depth evaluation of CLIP-DoRA's performance in domain generalization tasks and indicate that CLIP-DoRA surpasses existing PEFT methods under the same data regimes, proving the generalization ability of low-rank methods for V-L fine-tuning. **(4) Complex V-L task analysis.** We prove the suitability of CLIP-DoRA for medical semantic segmentation, a complex V-L task. Our results highlight the potential impact of this new family of V-L PEFT methods and show that low-resource fine-tuning is feasible in complex scenarios.

2. Related Works

Vision-Language Models. Recent V-L models like CLIP [3], ALIGN [4], and SigLIP [13] learn combined image-language representations in a self-supervised way using large-scale web data. For instance, CLIP and ALIGN train a multimodal network with approximately 400M and 1B image-text pairs, respectively. They have shown impressive performance as foundational models for a wide variety of tasks, including zero-shot classification [14]. Given that these models have been trained with extensive datasets, they have learned highly generalizable features [3, 13]. Due to this generalization capability, beyond zero-shot classification, V-L models are also used as pre-trained baselines to adapt to other tasks such as object detection or semantic segmentation (e.g. PromptDet [15] or CLIPSeg [16]).

PEFT of V-L Models. Despite their impressive adaptability, V-L models are inherently large and fully fine-tuning them is computationally intensive. PEFT provides a practical solution by efficiently adjusting large models for various downstream tasks [9]. Following existing taxonomy [9], we can distinguish different types of PEFT methods: additive, selective and parameterized. In the context of V-L fine-tuning, most solutions are additive methods, focused on open-vocabulary image classification. However, selective and parameterized approaches have been less explored.

Prompt-based PEFT methods learn useful context tokens, evolving from single modality approaches like CoOp [17] and CoCoOp [18] to multimodal ones like MaPLe [19]. Advanced techniques like PLOT [20] improve prompt learning through optimal transport, while KgCoOp [21] enhances generalizability by minimizing discrepancy between learned and hand-crafted prompts. **Adapter-based methods** insert trainable adapter layers within transformer blocks, as seen in CLIPAdapter [22] and Tip-Adapter [23]. Most of these works focus on few-shot learning [9].

Reparameterized PEFT. Contrary to the additive methods, Reparameterized PEFT, also called "low-rank adaptation", involves transforming model parameters into a low-dimensional representation for training, which is then integrated back into the original model for inference [9]. These methods are designed to fine-tune LLMs of very large dimensions. LoRA is the most well-known reparameterized PEFT method [10]. Given its success, different variants have appeared in the last few years. VeRA [24] uses a single pair of frozen random matrices shared across all layers, modifying them with trainable scaling vectors. DyLoRA [25] dynamically trains low-rank adapters across multiple ranks simultaneously. AdaLoRA [26] further advances this approach by adaptively allocating the parameter budget among weight matrices based on their importance, optimizing the rank of incremental matrices to control their budget effectively. Laplace-LoRA [27] applies a Bayesian approach to fine-tuning, treating the task as a posterior inference problem and providing a probabilistic interpretation of low-rank adaptations. QLoRA [28] integrates quantization with low-rank adaptation to significantly reduce memory usage while maintaining performance. Most recently, DoRA [12] introduced a novel approach to weight decomposition and reparameterization, improving learning capacity, enhancing convergence speed, and increasing training stability. Despite their success in LLM fine-tuning, these methods remain largely unexplored for V-L models. Only recently, CLIP-LoRA [11] was proposed to tackle CLIP few-shot fine-tuning using vanilla LoRA, showing improvements to existing additive PEFT methods.

In this work, we explore DoRA as a more advanced and faster-convergent alternative to vanilla LoRA. Additionally, we explore the performance of DoRA relative to other techniques in various contexts beyond few-shot learning.

3. Weight-Decomposed Low-Rank Adaptation for Vision-Language Model Fine-tuning

3.1. Contrastive Language-Image Pre-training (CLIP)

Our solution is built on top of any transformer-based [29] V-L model, like CLIP [3]. It consists of a pair of encoders - a visual encoder or ViT [30] and a text encoder - which share a common V-L embedding or latent space.

Visual encoder. The ViT or visual encoder, $\mathcal{V}$, comprises K transformer layers $\{\mathcal{V}\}_{i=1}^{K}$. The input image is divided into N equal-sized patches. Each patch is flattened and projected into a patch embedding of dimension $d_{\mathcal{V}}$ using a trainable linear layer. In this way, we obtain $\mathbf{E}_0 \in \mathbb{R}^{N\times d_{\mathcal{V}}} = \{\mathbf{E}_0^j\}_{j=1}^{N}$. A learnable *class token* $\mathbf{c}_0$ is prepended to $\mathbf{E}_0$ to obtain the input of $\mathcal{V}_1$. In general, the model works sequentially as follows: $x_i = [\mathbf{c}_i, \mathbf{E}_i] = \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}]); i = 1, 2, \ldots, K$. The transformer encoder comprises a series of stacked multi-head attention (MHA) modules and linear layers with residual connections. A single head can be represented as: $\mathbf{x}'_i = \text{Softmax}\left(\frac{\mathbf{x}_{i-1}\mathbf{W}_{q_i}(\mathbf{x}_{i-1}\mathbf{W}_{k_i})^\top}{\sqrt{d}}\right)(\mathbf{x}_{i-1}\mathbf{W}_{v_i}) + \mathbf{x}_{i-1}; \mathbf{x}_i = LN(x'_i) * \mathbf{W}_{o_i} + \mathbf{x}'_i$; where $\mathbf{W}_{q_i}, \mathbf{W}_{k_i}, \mathbf{W}_{v_i}, \mathbf{W}_{o_i} \in \mathbb{R}^{d_{\mathcal{V}}\times d_{\mathcal{V}}}$ are learnable matrices and LN represents the Layernorm operation [31]. The final image representation, $\mathbf{f}$, is obtained by projecting c_N into a common V-L embedding space using a learnable linear layer $\mathbf{P}_i \in \mathbb{R}^{d_{\mathcal{V}}\times d_{vl}}$: $\mathbf{f} = \mathbf{c}_K\mathbf{P}_i$.

Text encoder. The text encoder $\mathcal{T}$ is also composed by a sequence of transformer layers $\{\mathcal{T}_i\}_{i=1}^{Q}$. The input text is tokenized and projected into word embeddings $\mathbf{U}_0 = [\mathbf{u}_0^1, \mathbf{u}_0^2, \cdots, \mathbf{u}_0^M] \in \mathbb{R}^{M\times d_{\mathcal{T}}}$. In contrast to the visual encoder, a class embedding is not added here. Thus, we have the sequence of operations as: $\mathbf{U}_i = \mathcal{T}_i(\mathbf{U}_{i-1}); i = 1, 2, \ldots, Q$. The transformers of $\mathcal{T}_i$ have the same architecture as the ones of $\mathcal{V}_i$ apart from potential changes in the dimensions of the matrices and vectors. The final text representation $\mathbf{z}$ is obtained by projecting the last output corresponding to the end-of-text token in the common V-L space, i.e., $\mathbf{z} = \mathbf{u}_Q^M\mathbf{P}_T$, where $\mathbf{P}_T \in \mathbb{R}^{d_{\mathcal{T}}\times d_{vl}}$. Thus, CLIP projects images and text into a common space.

CLIP for image classification. CLIP enables zero-shot classification thanks to its common V-L embedding space. Given a set of C class labels $y \in \{1, 2, \ldots, C\}$, a hand-crafted prompt template (e.g. "a photo of a `<category>`") is used to obtain the text features for each class $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_C\}$. For any image, its feature representation $\mathbf{f}$ is obtained using the visual encoder. The prediction logits for each category are computed as the cosine similarity $l_j = \mathbf{f}^\top\mathbf{t}_j$ (for simplicity,

we assume $\mathbf{f}$ and $\mathbf{t}_j$ are normalized). The probabilities are then computed as usual using the softmax operator

$$p_j = \frac{\exp(l_j/\tau)}{\sum_k^C \exp(l_k/\tau)}, \tag{1}$$

where τ is a temperature hyperparameter. The prediction of CLIP is $\arg\max_j(p_j)$. While CLIP demonstrates strong zero-shot performance, sometimes it is necessary to fine-tune it to achieve optimal performance on a given downstream task. However, large transformer encoders are computationally intensive, making full fine-tuning unfeasible or very inefficient. Moreover, when dealing with limited data, tuning fewer parameters is often more effective to avoid overfitting. PEFT methods, like low-rank adaption, are crucial to tackle this issue.

3.2. Low-Rank Adaptation

Low-Rank Adaptation (LoRA) is a PEFT technique based on the idea that the updates made to the weights of a large model during fine-tuning exhibit a low "intrinsic rank" [10]. Given a pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$, its updates during fine-tuning, $\Delta\mathbf{W}$, are constrained through low-rank decomposition: $\Delta\mathbf{W} = \mathbf{BA}$, where $\mathbf{B} \in \mathbb{R}^{d\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times k}$ and $r \ll \min(d,k)$ is the rank. The updated weights can accept inputs of the same shape as the original $\mathbf{W}_0$. The result of the modified forward pass is very easy to compute as $(\mathbf{W}_0 + \Delta\mathbf{W})x = \mathbf{W}_0 x + \Delta\mathbf{W}x = \mathbf{W}_0 x + \mathbf{BA}x$. If $\mathbf{W}_0$ is kept frozen during training (i.e., it does not receive any gradient update) and only $\mathbf{A}$ and $\mathbf{B}$ are trained, the number of trainable parameters is reduced from $d \cdot k$ to $r \cdot (d + k)$, leading to a much more efficient fine-tuning. After fine-tuning the model is done, the matrix $\Delta\mathbf{W}$ can be merged into $\mathbf{W}_0$ as $\mathbf{W}' = \mathbf{W}_0 + \Delta\mathbf{W}$, leading to a model with the exact architecture as the original one. Thus, LoRA methods do not introduce any overhead during inference.

3.3. CLIP-DoRA

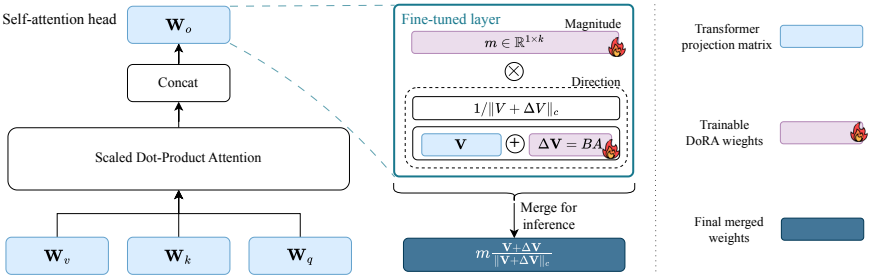

Fig. 1: Application of **CLIP-DoRA** in all self-attention blocks of a transformer-based V-L model. The original weights are frozen, and only low-rank matrices (direction) and a vector (magnitude) are trained for each transformer projection layer.

Weight-Decomposed Low-Rank Adaptation (DoRA) [12] is a recently proposed alternative to LoRA that takes advantage of weight matrix decomposition [32] for PEFT. In DoRA, pre-trained matrices are divided into two separate components, magnitude and direction. Formally, $\mathbf{W}_0 = m\frac{\mathbf{V}}{\|\mathbf{V}\|_c} = \|\mathbf{W}_0\|\frac{\mathbf{W}_0}{\|\mathbf{W}_0\|_c}$, where $m \in \mathbb{R}^{1\times k}$ is the magnitude vector, $\mathbf{V} \in \mathbb{R}^{d\times k}$ is the directional matrix and $\cdot/\|\cdot\|_c$ is the column-wise normalization operator. This decomposition ensures that each column remains a unit vector. The magnitude represents the scale of the pre-trained weights, while the direction captures their orientation in the multidimensional space. By separating these components, DoRA allows for more efficient fine-tuning and more stable and faster convergence. This is because the model can adjust the direction of the weights without altering their scale, enhancing learning efficiency and stability.

Similar to LoRA the original pre-trained matrix $\mathbf{W}_0$ is frozen during training. Its updates are reparameterized as

$$\mathbf{W}' = \underline{m}\frac{\mathbf{V}+\Delta\mathbf{V}}{|||\mathbf{V}+\Delta\mathbf{V}||_c} = \underline{m}\frac{\mathbf{W}_0+\underline{\mathbf{BA}}}{|||\mathbf{W}_0+\underline{\mathbf{BA}}||_c}, \tag{2}$$

where only underlined elements are updated during fine-tuning. Thus, this provides a more efficient alternative to full fine-tuning due to the reduction in computations and memory needed. More concretely, the number of trainable parameters is reduced from $d \cdot k$ to $k + r \cdot (d + k)$. In CLIP-DoRA, we take advantage of this improved reparametrization for low-rank adaptation to provide a high-performance and resource-efficient fine-tuning paradigm for V-L models.

To further enhance the efficiency of our method, we incorporate an implementation trick that reduces memory consumption during backpropagation [12]. Specifically, in DoRA, the low-rank adaptation is redirected towards the directional component, which results in the gradient of the low-rank updates differs from that of $\mathbf{W}'$, as shown in Equation (3):

$$\nabla_{\mathbf{V}'}\mathcal{L} = \frac{m}{||\mathbf{V}'||_c}\left(I - \frac{\mathbf{V}'\mathbf{V}'^{\top}}{||\mathbf{V}'||_c^2}\right)\nabla_{\mathbf{W}'}\mathcal{L}, \quad \nabla_m\mathcal{L} = \frac{\nabla_{\mathbf{W}'}\mathcal{L}\cdot\mathbf{V}'}{||\mathbf{V}'||_c}, \tag{3}$$

where $\mathbf{V}' = \mathbf{V} + \Delta\mathbf{V}$ and $\mathcal{L}$ is the training loss. This divergence consumes extra memory during backpropagation. To mitigate this, we treat $||\mathbf{V} + \Delta\mathbf{V}||_c$ in Equation (2) as a constant, detaching it from the gradient graph. This approach ensures that while $||\mathbf{V} + \Delta\mathbf{V}||_c$ dynamically reflects the updates of $\Delta\mathbf{V}$, it will not receive any gradient during backpropagation. Consequently, the gradient with respect to m does not change, and $\nabla_{\mathbf{V}}\mathcal{L}$ is redefined as $\nabla_{\mathbf{V}}\mathcal{L} = \frac{m}{C}\mathbf{W}_0'\nabla_{\mathbf{W}}\mathcal{L}$, where $C = ||\mathbf{W}_0'||_c$. As illustrated in Figure 1, we apply this reparameterization to all the $\mathbf{W}_{q_i}$, $\mathbf{W}_{k_i}$, $\mathbf{W}_{v_i}$, and $\mathbf{W}_{o_i}$ of all transformer layers of $\mathcal{V}$ and $\mathcal{T}$. Therefore, in CLIP-DoRA, all the original CLIP weights are kept frozen. Only the newly added reparametrization representation (m, $\mathbf{A}$ and $\mathbf{B}$ matrices in (2)) are updated. After the fine-tuning is finished, the matrix $\Delta\mathbf{W}$ can be merged into $\mathbf{W}_0$: $\mathbf{W}' = m \cdot (\mathbf{V} + \Delta\mathbf{V})/||\mathbf{V} + \Delta\mathbf{V}||_c$.

4. Results and Discussion

In this section, we validate the effectiveness of CLIP-DoRA through few-shot classification and domain generalization experiments. These tasks highlight the model's ability to perform well and produce general representations with limited data and further emphasize the efficiency of our method. Although we compare with prompt-based and adapter-based PEFT methods, we perform the main comparisons with CLIP-LoRA [11], since it is the only other V-L low-rank PEFT method and the current state-of-the-art.

4.1. Setup

Datasets. We use 11 datasets to evaluate the **few-shot** training capabilities of CLIP-DoRA as is commonly done in the literature [19, 18, 17, 11]. In few-shot learning, a support set of S elements or shots (maximum 16) per category is used to fine-tune the model. For **domain generalization**, we follow the evaluation of ODG-CLIP [33]. In particular, we use 4 benchmark datasets: PACS [34], OfficeHome [35], Digits-DG [36] and Mini-DomainNet [37]. Each of these benchmarks contains a series of subsets, differentiated from each other by some domain shift. We use the same splits as ODG-CLIP [33].

Implementation details. To ensure a fair comparison, for **few-shot** classification we keep all hyperparameters the same as in CLIP-LoRA [11], unless explicitly stated otherwise. More concretely, we fix a rank $r = 2$ for DoRA, a dropout of $p = 0.25$, a cosine scheduler with an initial learning rate of $2 \cdot 10^{-4}$ and $500 \times S$ training steps (where S is the number of shots), and a batch size of 32. We adopt the widely employed prompt from the literature, "a photo of a `<class>`" [3]. For **domain generalization** we maintain the setup, except we set the scheduler to 10 epochs per

experiment. For both tasks, the logits or probabilities computed using Equation (1) are used to train the network with cross-entropy loss.

Evaluation. In **few-shot learning**, we use Top-1 accuracy on the complete test set of the dataset used to perform the fine-tuning. For **domain-generalization**, we employ a *leave-one-out strategy*: we train on all but one of the subsets of every benchmark, and evaluate the remaining one. The average Top-1 accuracy for every benchmark is reported. All the results reported are obtained as the average of executing with three random seeds. The same hyperparameters are kept for all the datasets and both tasks.

4.2. Few-shot Classification Fine-Tuning

We show the results for few-shot learning on the 11 datasets considered for different numbers of shots and different CLIP PEFT methods (with ViT-B/16) in Table 1. Low-rank methods are clearly superior to "traditional" PEFT methods in terms of average accuracy. More specifically, the proposed CLIP-DoRA outperforms the best non-low-rank method by 1.76%, 0.70% and 1.15% for 1, 4 and 16 shots, respectively. CLIP-DoRA is overall the best-performing method in the three regimes, with an average advantage over CLIP-LoRA ranging from 0.13% to 0.28%. It is noticeable that the difference increases with the number of shots, indicating that it can generally use more data effectively. As seen in Table 1, Low-rank methods perform worse than other PEFT methods in the fine-grained datasets Food101 and OxfordPets. It is remarkable that, for these two datasets, some methods experience degradation as the number of shots and training increases. This indicates that the pre-trained model has "forgotten" previous knowledge, which explains why methods like KgCoOp [21] (designed to deal with this particular problem) excel in these datasets. More detailed results for ViT-B/16, ViT-B/32 and ViT-L/14 are found in Table 2.

4.3. Domain Generalization

Table 3 contains the results for domain generalization. Since these results were not reported in [11], we kept their default hyperparameters and used them to evaluate both CLIP-LoRA (using the official implementation) and our proposed method with three different random seeds. In addition, we also include the results of different CLIP-PEFT methods, which were extracted from [33]. We can see that low-rank-based methods generally obtain much better results in domain generalization than baseline pre-trained CLIP, CLIP features combined with OpenMax [38], CLIP features paired with OSDA-BP [39], and a variety of prompt-based methods. They only fall slightly (-0.02%) behind MaPLe when it is trained with additional synthetic images, as described in [33], which requires additional computation and more data. This result suggests the ability of low-rank based methods to learn features that are useful and robust to domain shifts. If we focus only on low-rank methods, we can see that CLIP-DoRA beats CLIP-LoRA in 3 out of 4 benchmarks while the overall average is nearly identical.

4.4. CLIP-DoRA in Image Segmentation

In addition to the classical classification tasks in which PEFT methods are typically tested, we explore other potential applications of CLIP-DoRA for the adaptation of V-L models to other challenging tasks. We consider the task of fine-tuning CLIPSeg [16] for medical imaging segmentation. In medical VLSM[40], the authors explore the potential of large V-L foundational models to be fine-tuned for these tasks. We compare the results achieved through full fine-tuning (FFT) as reported in medical VLSM [40] with the results achieved by fine-tuning using our proposed CLIP-DoRA. We use the same hyperparameters as in [40], with the only exception that the whole CLIPSeg model is frozen except for the DoRA-related weights ($r = 8$) and the transposed convolution responsible for the upscale from the CLIP latent space. We evaluate on four datasets: Kvasir-SEG [41], ClinicDB [42], ISIC [43], BUSI [44]. We use the average dice score as the evaluation metric. In Table 4, we show the results of CLIPSeg fine-tuning to medical image segmentation. As is expected, FFT performs better on the different benchmarks. However, with only a fraction of the trainable parameters (1.53%), CLIP-DoRA achieves competitive results in all the datasets. In particular, the difference in the mean dice score is lower than 1.0 for 3 datasets. The average difference is only 0.7. This result highlights the potential of CLIP-DoRA as an efficient alternative for a wider variety of CLIP-adaptation tasks.

Table 1: Top-1 accuracy (Mean of 3 random seeds) of different methods on few-shot learning regimes (zero-shot, 1, 4 and 16 shots) in the 11 considered datasets with CLIP ViT-B/16. The last column contains the average over all the datasets.

	Method	ImageNet	SUN	FGVC	SAT	Cars	Food	Pets	Flowers	Caltech	DTD	UCF	Avg
	CLIP (Zero-shot)	66.7	62.6	24.7	47.5	65.3	86.1	89.1	71.4	92.9	43.6	66.7	65.15
1 shot	CoOp (4)	68.0	67.3	26.2	50.9	67.1	82.6	90.3	72.7	93.2	50.1	70.7	67.19
	CoOp (16)	65.7	67.0	20.8	56.4	67.5	84.3	90.2	78.3	92.5	50.1	71.2	67.64
	CoCoOp	69.4	68.7	28.1	55.4	67.6	84.9	91.9	73.4	94.1	52.6	70.4	68.77
	TIP-Adapter-F	69.4	67.2	28.8	67.8	67.1	85.8	90.6	83.8	94.0	51.6	73.4	70.86
	CLIP-Adapter	67.9	65.4	25.2	49.3	65.7	86.1	89.0	71.3	92.0	44.2	66.9	65.73
	PLOT++	66.5	66.8	28.6	65.4	68.8	86.2	91.9	80.5	**94.3**	**54.6**	74.3	70.72
	KgCoOp	68.9	68.4	26.8	61.9	66.7	**86.4**	92.1	74.7	94.2	52.7	72.8	69.60
	TaskRes	69.6	68.1	**31.3**	65.4	68.8	84.6	90.2	81.7	93.6	53.8	71.7	70.80
	MaPLe	69.7	69.3	28.1	29.1	67.6	85.4	91.4	74.9	93.6	50.0	71.1	66.38
	ProGrad	67.0	67.0	28.8	57.0	68.2	84.9	91.4	80.9	93.5	52.8	73.3	69.53
	CLIP-LoRA	70.4	**70.4**	30.2	72.3	70.1	84.3	**92.3**	83.2	93.7	54.3	76.3	72.50
	CLIP-DoRA	**70.5**	70.0	29.0	**72.9**	**70.7**	84.5	91.6	**85.1**	93.7	54.5	**76.4**	**72.63**
4 shots	CoOp (4)	69.7	70.6	29.7	65.8	73.4	83.5	92.3	86.6	94.5	58.5	78.1	72.97
	CoOp (16)	68.8	69.7	30.9	69.7	74.4	84.5	92.5	92.2	94.5	59.5	77.6	74.03
	CoCoOp	70.6	70.4	30.6	61.7	69.5	86.3	92.7	81.5	94.8	55.7	75.3	71.74
	TIP-Adapter-F	70.7	70.8	35.7	76.8	74.1	86.5	91.9	92.1	94.8	59.8	78.1	75.57
	CLIP-Adapter	68.6	68.0	27.9	51.2	67.5	86.5	90.8	73.1	94.0	46.1	70.6	67.66
	PLOT++	70.4	71.7	35.3	83.2	76.3	86.5	92.6	92.9	95.1	62.4	79.8	76.93
	KgCoOp	69.9	71.5	32.2	71.8	69.5	**86.9**	92.6	87.0	95.0	58.7	77.6	73.88
	TaskRes	71.0	72.7	33.4	74.2	76.0	86.0	91.9	85.0	95.0	60.1	76.2	74.68
	MaPLe	70.6	71.4	30.1	69.9	70.1	86.7	**93.3**	84.9	95.0	59.0	77.1	73.46
	ProGrad	70.2	71.7	34.1	69.6	75.0	85.4	92.1	91.1	94.4	59.7	77.9	74.65
	CLIP-LoRA	**71.4**	**72.8**	**37.9**	84.9	77.4	82.7	91.0	93.7	**95.2**	63.8	81.1	77.45
	CLIP-DoRA	**71.4**	72.5	37.5	**86.1**	**77.9**	82.9	90.4	**94.6**	95.0	**64.0**	**81.6**	**77.63**
16 shots	CoOp (4)	71.5	74.6	40.1	83.5	79.1	85.1	92.4	96.4	95.5	69.2	81.9	79.03
	CoOp (16)	71.9	74.9	43.2	85.0	82.9	84.2	92.0	96.8	95.8	69.7	83.1	79.95
	CoCoOp	71.1	72.6	33.3	73.6	72.3	**87.4**	93.4	89.1	95.1	63.7	77.2	75.35
	TIP-Adapter-F	73.4	76.0	44.6	85.9	82.3	86.8	92.6	96.2	95.7	70.8	83.9	80.75
	CLIP-Adapter	69.8	74.2	34.2	71.4	74.0	87.1	92.3	92.9	94.9	59.4	80.2	75.49
	PLOT++	72.6	76.0	46.7	92.0	84.6	87.1	**93.6**	97.6	96.0	71.4	85.3	82.08
	KgCoOp	70.4	73.3	36.5	76.2	74.8	87.2	93.2	93.4	95.2	68.7	81.7	77.33
	TaskRes	73.0	**76.1**	44.9	82.7	83.5	86.9	92.4	97.5	95.8	71.5	84.0	80.75
	MaPLe	71.9	74.5	36.8	87.5	74.3	**87.4**	93.2	94.2	95.4	68.4	81.4	78.64
	ProGrad	72.1	75.1	43.0	83.6	82.9	85.8	92.8	96.6	95.9	68.8	82.7	79.94
	CLIP-LoRA	73.6	**76.1**	54.7	92.1	86.3	84.2	92.4	98.0	**96.4**	72.0	**86.7**	82.95
	CLIP-DoRA	**73.8**	76.0	**56.4**	**92.6**	**86.9**	84.2	92.3	**98.2**	96.3	**72.4**	86.4	**83.23**

4.5. Convergence Study

DoRA [12] is a low-rank adaptation alternative to vanilla LoRA [10] that provides faster convergence thanks to the weight-matrix decomposition discussed in Section 3. In this subsection, we analyze the behavior of the loss curves during the few-shot training of different datasets with different numbers of shots per category. As illustrated in Figure 2, CLIP-DoRA consistently converges faster and with loss values lower than those of CLIP-LoRA. This proves our hypothesis that the weight decomposition [32], which had proved to be useful in NLP tasks, could also be beneficial for V-L models. As evidenced in Figure 2, this pattern is present regardless of (1) the number of shots; (2) the domain of the dataset; (3) the number of epochs (the number of steps is determined only by the number of shots per class, so datasets with more classes will be trained for fewer epochs).

Table 2: Results of CLIP-DoRA with different visual encoders in few-shot learning.

	ViT-B/16					ViT-B/32					ViT-L/14				
Dataset	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
Cal101	93.7	94.8	95.0	95.5	96.3	93.0	93.9	93.8	94.7	95.3	95.6	96.8	97.2	97.0	97.0
DTD	54.5	60.1	64.0	67.1	72.4	49.3	56.9	60.3	63.8	68.6	60.8	67.8	70.3	73.2	77.0
EuroSAT	72.9	81.9	86.1	88.8	92.6	65.6	78.9	83.5	88.8	91.6	74.1	84.2	85.2	89.2	92.7
FGVC	29.0	33.1	37.5	45.8	56.4	22.0	23.4	27.7	35.3	47.0	40.5	43.3	49.7	57.4	66.8
Food101	84.5	83.2	82.9	83.2	84.2	78.7	76.3	75.6	76.7	78.0	90.5	89.9	89.8	89.6	89.8
INet	70.5	70.9	71.4	72.5	73.8	65.2	65.8	66.2	67.4	68.8	77.2	77.6	77.8	78.8	79.8
Flowers	85.1	90.4	94.6	96.3	98.2	78.4	85.7	90.7	93.4	96.6	91.3	95.3	97.6	98.4	99.1
OxPets	91.6	90.6	90.4	91.1	92.3	87.8	86.8	85.7	87.1	88.6	94.4	93.9	94.3	94.5	94.8
Cars	70.7	73.9	77.9	82.5	86.9	63.4	64.8	68.9	75.1	80.9	81.3	82.8	85.1	89.0	91.1
SUN397	70.0	71.0	72.5	74.4	76.0	67.3	68.1	70.2	72.1	74.0	74.4	75.3	76.7	78.2	79.7
UCF101	76.4	79.9	81.6	83.7	86.4	72.4	75.3	76.3	80.4	82.5	83.1	84.8	86.6	88.6	90.4
Avg	72.63	75.44	77.63	80.08	83.23	67.55	70.54	72.63	75.89	79.26	78.47	81.06	82.75	84.90	87.11

Table 3: Domain generalization of different CNN-based methods and CLIP-based PEFT methods in 4 benchmarks. All but the last 4 rows have been taken from [33]. Reported average results for three random seeds. ViT-B/32 is used.

Methods	PACS	OH	D-DG	M-DN	Avg
CLIP	95.16	81.43	77.08	84.50	84.54
CLIP + OpenMax	93.45	81.00	76.93	81.89	83.32
CLIP + OSDA	92.62	82.58	80.53	82.00	84.43
CoCoOp	85.76	75.38	52.77	60.63	68.64
MaPLe	93.97	79.47	70.54	74.67	79.66
PromptSRC	94.53	80.21	75.34	73.60	80.92
CLIPN	96.24	84.55	81.70	77.38	84.97
MaPLe + SD	91.47	85.02	79.92	83.79	85.05
CLIP-LoRA	95.09	82.52	81.26	81.21	85.02
CLIP-DoRA	95.35	83.19	81.43	80.15	85.03

Table 4: Comparison of full fine-tuning of CLIPSeg [16] and our proposed CLIP-DoRA in medical image segmentation. The number of trained parameters as well as their relative size with respect to the total size of the original model are reported. We report the average dice score per dataset.

Metric	Full Fine-tuning	CLIP-DoRA	Difference
Params (M) / %	150 (100%)	2.3 (1.53%)	-147.7 (-98.47%)
Kvasir	89.51	88.37	-1.14
ClinicDB	88.74	87.98	-0.76
ISIC	92.12	91.90	-0.22
BUSI	64.32	63.64	-0.68
Avg	83.67	82.97	-0.70

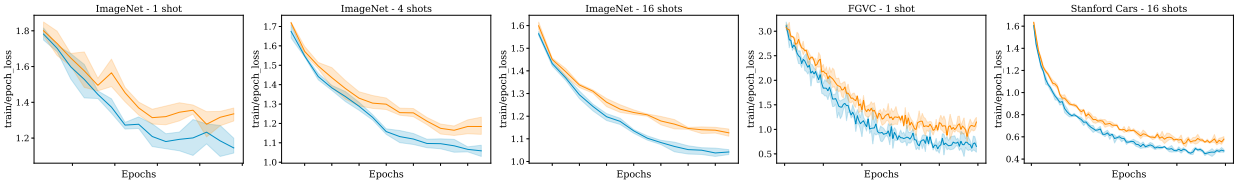

Fig. 2: Comparison of loss curves (cross-entropy) for CLIP-LoRA ( ) and CLIP-DoRA ( ) for different datasets and number of shots with CLIP ViT-B/16. Each line represents the average of three runs with different random seeds.

5. Conclusion

In this work, we proposed CLIP-DoRA, an innovative approach to efficient fine-tuning of vision-language models through weight-decomposed low-rank adaptation. Our extensive experiments demonstrate that CLIP-DoRA leverages the faster convergence of weight matrix decomposition during training. In turn, it consistently outperforms existing PEFT methods in few-shot classification, establishing a new state-of-the-art across different backbones and regimes. Additionally, CLIP-DoRA achieves competitive results in domain generalization compared to other families of PEFT methods for vision-language models, highlighting its robustness and versatility. Our method also proved its effectiveness in more complex vision-language tasks, such as medical image segmentation, where it performed competitively with only a fraction of the trainable parameters. These results underscore the potential impact of CLIP-DoRA in help-

ing to develop more efficient computer vision solutions. Despite the promising results and the new insights extracted from our research, we outline some key questions for **future research.** (1) The current approach treats both text and visual encoder of CLIP identically without considering their particularities and interactions, which might be suboptimal for some domains. A natural question is how we can better adapt the DoRA method to the intricacies of the multimodal space of V-L models to improve their performance. (2) Since we have proved the faster convergence of CLIP-DoRA over other methods, another promising research direction is exploring the potential for training models more efficiently with fewer epochs, reducing computational costs and training time even further. (3) Finally, although we stick to CLIP since it is the *de facto* standard for V-L PEFT evaluation, it is worth exploring how CLIP-DoRA behaves when applied to other V-L models.

Acknowledgements

This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00) and IDEATE (AEI-MICINN, PID2022-141566NB-I00). J. M. Rodríguez-de-Vera and Imanol G. Estepa acknowledge the support of FPU Becas with code FPU22/03116 and FPU23/02822 respectively, Ministry of Universities, Spain. B. Nagarajan acknowledges AI4S fellowship within the "Generación D" initiative by Red.es, Ministerio para la Transformación Digital y de la Función Pública, for talent attraction (C005/24-ED CV1), funded by NextGenerationEU through PRTR. The authors thankfully acknowledge EuroHPC Joint Undertaking (EHPC-DEV-2023D12-059) for awarding us access to Leonardo at CINECA, Italy and Spanish Supercomputing Network (RES) (IM-2023-3-0019) for awarding us access to MareNostrum5 at BSC, Spain.

References

[1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[2] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: ICML, PMLR, 2021, pp. 8748–8763.

[4] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: ICML, PMLR, 2021, pp. 4904–4916.

[5] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361 (2020).

[6] X. Zhai, A. Kolesnikov, N. Houlsby, L. Beyer, Scaling vision transformers, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 12104–12113.

[7] J. Castaño, S. Martínez-Fernández, X. Franch, J. Bogner, Exploring the carbon footprint of hugging face's ml models: A repository mining study, in: 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2023, pp. 1–12.

[8] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto. Stanford alpaca: An instruction-following llama model [online] (2023). Last accessed 2024-04-07.

[9] Z. Han, C. Gao, J. Liu, S. Q. Zhang, et al., Parameter-efficient fine-tuning for large models: A comprehensive survey, arXiv preprint arXiv:2403.14608 (2024).

[10] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: ICLR, 2022.

[11] M. Zanella, I. Ben Ayed, Low-rank few-shot adaptation of vision-language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2024, pp. 1593–1603.

[12] S. yang Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, M.-H. Chen, DoRA: Weight-decomposed low-rank adaptation, in: Forty-first ICML, 2024.
URL https://openreview.net/forum?id=3d5CIRG1n2

[13] X. Zhai, B. Mustafa, A. Kolesnikov, L. Beyer, Sigmoid loss for language image pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.

[14] Z. Guo, R. Zhang, L. Qiu, X. Ma, X. Miao, X. He, B. Cui, Calip: Zero-shot enhancement of clip with parameter-free attention, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 746–754.

[15] C. Feng, Y. Zhong, Z. Jie, X. Chu, H. Ren, X. Wei, W. Xie, L. Ma, Promptdet: Towards open-vocabulary detection using uncurated images, in: European Conference on Computer Vision, Springer, 2022, pp. 701–717.

[16] T. Lüddecke, A. Ecker, Image segmentation using text and image prompts, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 7086–7096.
[17] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Learning to prompt for vision-language models, International Journal of Computer Vision 130 (9) (2022) 2337–2348.
[18] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16816–16825.
[19] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, F. S. Khan, Maple: Multi-modal prompt learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19113–19122.
[20] G. Chen, W. Yao, X. Song, X. Li, Y. Rao, K. Zhang, PLOT: Prompt learning with optimal transport for vision-language models, in: The Eleventh ICLR, 2023.
URL https://openreview.net/forum?id=zqwryBoXYnh
[21] H. Yao, R. Zhang, C. Xu, Visual-language prompt tuning with knowledge-guided context optimization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 6757–6767.
[22] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, Y. Qiao, Clip-adapter: Better vision-language models with feature adapters, International Journal of Computer Vision 132 (2) (2024) 581–595.
[23] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, H. Li, Tip-adapter: Training-free adaption of clip for few-shot classification, in: European conference on computer vision, Springer, 2022, pp. 493–510.
[24] D. J. Kopiczko, T. Blankevoort, Y. M. Asano, VeRA: Vector-based random matrix adaptation, in: The Twelfth ICLR, 2024.
[25] M. Valipour, M. Rezagholizadeh, I. Kobyzev, A. Ghodsi, DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation, in: A. Vlachos, I. Augenstein (Eds.), Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 3274–3287. doi:10.18653/v1/2023.eacl-main.239.
URL https://aclanthology.org/2023.eacl-main.239
[26] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, T. Zhao, Adaptive budget allocation for parameter-efficient fine-tuning, in: The Eleventh ICLR, 2023.
[27] A. X. Yang, M. Robeyns, X. Wang, L. Aitchison, Bayesian low-rank adaptation for large language models, in: The Twelfth ICLR, 2024.
URL https://openreview.net/forum?id=FJiUyzOF1m
[28] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, Advances in Neural Information Processing Systems 36 (2024).
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, 2021.
[31] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, L. S. Chao, Learning deep transformer models for machine translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1810–1822.
[32] T. Salimans, D. P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, Advances in neural information processing systems 29 (2016).
[33] M. Singha, A. Jha, S. Bose, A. Nair, M. Abdar, B. Banerjee, Unknown prompt the only lacuna: Unveiling clip's potential for open domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 13309–13319.
[34] D. Li, Y. Yang, Y.-Z. Song, T. M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 5542–5550.
[35] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5018–5027.
[36] K. Zhou, Y. Yang, T. Hospedales, T. Xiang, Learning to generate novel domains for domain generalization, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16, Springer, 2020, pp. 561–578.
[37] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1406–1415.
[38] A. Bendale, T. E. Boult, Towards open set deep networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1563–1572.
[39] P. Panareda Busto, J. Gall, Open set domain adaptation, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 754–763.
[40] K. Poudel, M. Dhakal, P. Bhandari, R. Adhikari, S. Thapaliya, B. Khanal, Exploring transfer learning in medical image segmentation using vision-language models, in: Medical Imaging with Deep Learning, 2024.
[41] D. Jha, P. Smedsrud, M. Riegler, et al., Kvasir-seg: a segmented polyp dataset, multimedia modeling: 26th international conference (2020).
[42] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, F. Vilariño, Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians, Computerized medical imaging and graphics 43 (2015) 99–111.
[43] D. Gutman, N. C. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic), arXiv preprint arXiv:1605.01397 (2016).
[44] W. Al-Dhabyani, M. Gomaa, H. Khaled, A. Fahmy, Dataset of breast ultrasound images, Data in brief 28 (2020) 104863.