# Chapter 1

# Few-shot Learning with CLIP

## 1.1 Motivation

Complete fine-tuning of vision-language models demands significant resources both in terms of data and computation. However, with the emergence of CLIP with its remarkable capabilities of zero-shot classification we can attain strong performance without the need for training data or the actual training of a model by transfering CLIP pre-trained knowledge to a new task Yu (2022). Nevertheless, access to few-shot examples per class for few-shot fine-tuning remains a crucial step to unlock CLIP's full transfer learning potential as zero-shot performance can significantly decrease when pre-training distribution of the data CLIP was trained on is different from the target distribution Zhou et al. (2022b) or when downstream tasks entail specific objectives such as fine-grained classification, region or pixel-level recognition and more because CLIP's training objective is task-agnostic Radford et al. (2021).

## 1.2 Organisation

In this chapter, we will begin by providing a taxonomy of few-shot learning methods that use CLIP. We will delve into a more detailed description of the methods for which we intend to propose improvements, while providing a relatively less detailed descriptions of other methods to offer a comprehensive understanding of the landscape. In Section 2.5, we will outline the challenge of the intra-modal overlap Udandarao, Gupta and Albanie (2023), which we aim to mitigate in the experiments detailed in Chapter 6 where we will demonstrate how addressing it can enhance the performance of CLIP's few-shot classification abilities.

## 1.3 CLIP Zero-shot classification

Because most of the methods in this and in subsequent chapters are based on CLIP model, we are going to describe first how CLIP is used to perform zero-shot classification. CLIP zero-shot classification refers to the capability of the CLIP model to classify objects in images without the need for specific training examples of those objects. Such capability has arisen thanks to the large amount of 0.4 billion image-text pairs that CLIP was trained on and contrastive loss adopted during CLIP pre-training that pulls together similar image-text pairs to have high similarity scores in a shared text-image representation space while pushing dissimilar pairs apart. This shared representation space where textual prompts and images can be compared directly enables CLIP to perform zero-shot classification.

**Zero-shot CLIP prediction** Given $N$ textual classes, we incorporate them into a contextual prompt $PR_i$ "*A photo of a {class}*" where $\{class\}$ is replaced by the ground-truth text label and encode it with CLIP's textual encoder $TE$ to obtain text embedding for each class:

$$w_i = TE(PR_i), w_i \in R^{1\times d} \tag{1.1}$$

where $d$ is the embedding dimension. By concatenating the embeddings for all the classes we obtain a classifier weight matrix $W \in R^{N\times d}$. Subsequently, when presented with a test image $I_j$, it is encoded using CLIP image encoder $VE$:

$$T_j = VE(I_j), T_j \in R^{1\times d} \tag{1.2}$$

And the predicted probability for class $i$ of image $I_j$ is defined as follows:

$$p(y = i|w_i) = \frac{exp(sim(T_j, w_i)/\tau)}{\sum_{k=1}^{N} exp(sim(T_j, w_k)/\tau)} \tag{1.3}$$

Where $sim$ is the cosine similarity, $\tau$ is the learned temperature of CLIP and $exp$ the exponential operation. The whole probability vector is given by concatenating $p(y = i|w_i)$ or $p_i$ for short: $P = \{p_i\}_{i=1}^{N}$. We can also define unnormalized zero-shot classification logits for test image $I_j$ as the dot product between $W$ matrix and the embedded image giving us the zero-shot classification logits for the test image $I_j$:

$$\text{CLIPlogits} = T_j W^T, \text{CLIPlogits} \in R^{1\times N} \tag{1.4}$$

| Category | Method |
|---|---|
| Requires training | Clip-Adapter<br>CoOp<br>CoCoOp<br>TaskRes<br>MaPLe<br>UPT<br>SVL-Adapter<br>ProDA<br>ProGrad<br>VPT |
| Training-free methods (* if have trainable alternative) | Tip-Adapter*<br>Tip-X<br>CaFO* |
| Zero-shot methods (* if have trainable alternative) | Zero-shot CLIP<br>CALIP*<br>SuS-X (Variation of Tip-X) |
| Generative methods (* if have trainable alternative) | SuS-X (Variation of Tip-X)<br>CuPL<br>VisDesc<br>CaFo* |

Table 1.1: Table with Categories and Methods

## 1.4 Taxonomy of adaptation methods

We can broadly group existing few-shot learning methods into four categories:

- Methods that require training: This family of methods uses few-shot examples to tune additional parameters to CLIP model.
- Training-free methods: These methods require access to the few-shot labelled data but do not require any training. Some of them have a trainable alternative that requires parameters tuning.
- Zero-shot methods: These methods can classify objects in images without the need for specific training examples or additional parameters fine-tuning. Some of them also have a trainable alternatives requiring few-shot examples.
- Generative methods: These methods do not require specific training examples or additional parameters fine-tuning as they use external models/databases to generate or retrieve examples from given classes and generate rich textual prompts.

Note that some methods are in multiple categories as they may have characteristics of multiple ones.

## 1.5 Methods that require training

### 1.5.1 Hierarchical visual features for Task Residual method

Before introducing task residual (TaskRes) Yu et al. (2023) we need to understand adapter tuning and prompt tuning methods. Adapter tuning involves refining the frozen features extracted from CLIP by adjusting them through new tunable parameters (adapters), either on top of the visual or language encoder. These adapters learn new knowledge from few-shot examples, while the original CLIP maintains its general knowledge. An example of such methods is CLIP-Adapter Gao et al. (2021). Prompt tuning methods, on the other hand, stem from the recognition that a well-crafted prompt can enhance the prediction accuracy of zero-shot CLIP Sun et al. (2023). However, manually tuning prompts is cumbersome, time-consuming, and often requires domain expertise. Hence, prompt tuning methods learn context words for each specific task instead of defining them manually. Examples of these methods include CoOp Sun et al. (2023) and CoCoOp Zhou et al. (2022a).
Task residual paper Yu et al. (2023) criticizes adapter tuning methods like CLIP-Adapter and prompt tuning methods like CoOp because in the first case, as the features extractors are frozen the adapters do not learn any new features but try to exploit the old features to predict the task. In the second case, by replacing with task-specific learnable parameters the fixed prompt we change the CLIP text-based classifier learnt during long pre-training that can lead to old knowledge forgetting and damaging pre-trained classification boundaries resulting in lower performance especially when only a few-shot training examples are available. To correct these problems TaskRes paper proposes to keep fixed the text classifier and learn new knowledge that is independent of CLIP extracted features by adding in a residual fashion learnable parameters to pre-trained text features.

**Task Residual hierarchical visual features** TaskRes primarily focuses on the modification of textual features and in that sense it is similar to CoOp with the difference that CoOp incorporates learnable parameters before the textual features extractor while TaskRes after it. However, TaskRes may not fully leverage the potential for enhancing performance on the image side. The authors attempted to introduce learnable parameters to pre-trained image features in a residual fashion mirroring the process on the text side but this effort faced challenges due to the substantial diversity among image embeddings that is not found in text embedding as the classes are fixed. This diversity proved to be a hurdle for the experiment. Our approach differs significantly. We suggest utilizing CLIP visual features in a hierarchical manner. In the standard training of CLIP, the last layers of image and text encoders are aligned within the same image-text space. However, due to the modification of the text space in TaskRes through the introduction of independent parameters, the last layer from the image

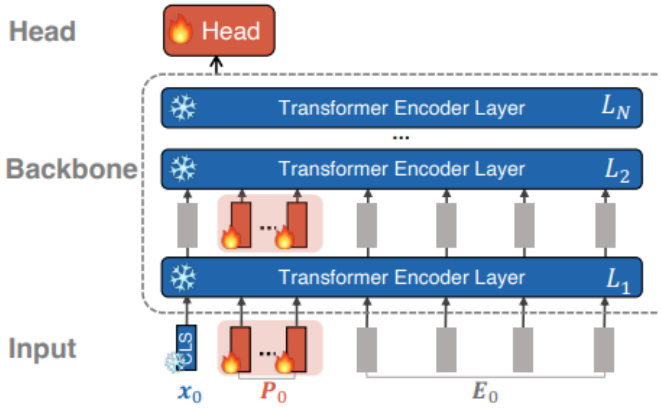



Figure 1-1: Latent space prompting. Illustration from Jia et al. (2022)

encoder may not be the ideal choice anymore for computing similarity with the TaskRes layer in the text encoder. Therefore, our proposal involves learning attention weights for different visual layers enabling the weighting of image features from different layers. We also expect that hierarchical features approach allows the network to automatically determine whether fine-grained features are more relevant for predicting certain classes or if coarse-grained features are sufficient.

### 1.5.2 Exploring pixel vs latent space prompting

Methods in this subsection mainly apply to the Visual Transformer (ViT) Dosovitskiy et al. (2020) model and are not easily adapted to convolutions-based backbones like Residual Network (ResNet) He et al. (2015). ViT is based on the transformer architecture which consists of a stack of transformer encoder layers with each layer made of self-attention mechanisms and feedforward neural networks. It treats images as sequences of patches and processes them through the encoder layers where self-attention mechanisms enable interactions between different patches. On the other hand ResNet is a convolutional neural network architecture that introduces skip connections, or residual connections, between layers. The main idea of the latent space prompting is to insert learnable prompts in the encoder layers after the embedding layer (but sometimes also in the input space together with text prompts like CoOp or patches) concatenating them to the patches embeddings. This is illustrated in fig. 1-1. Because of a different structure of ResNet it's clear that adapting this method to the convolutional backbone is more involved. In VPT method Jia et al. (2022) the authors applied learnable latent-space tokens to the ViT model handling only unimodal visual tasks. Multimodal methods such as MaPLe Jia et al. (2022) and UPT Zang et al. (2022) extended it to the multimodal realm. MaPLe first incorporate learnable prompts in the latent space of the textual branch. Then, for the visual branch, instead of defining learnable tokens

Figure 1-2: **On the left**: Image from OxfordFlowers dataset Nilsback and Zisserman (2008) **On the right**: Image from OxfordFlowers dataset with 10% most important pixels highlighted using a prompt "point to the most important parts in this object"

as in the text branch they make visual prompts function of the text prompts using a coupling linear layer function. This setting facilitates the interaction between the prompts in the vision and text branches. UPT Zang et al. (2022) is similar to MaPLe, however, instead of using a coupling function between vision and language branches for the interaction they learn a unified prompt matrix for each encoder layer that is passed through a learnable Transformer layer and split into text and visual prompts that are inserted into the text and vision encoders respectively.

**Pixel space prompting** None of these methods allow direct interaction with the pixels of the images. In Bahng et al. (2022) authors modify pixel space by adding a learnable padding pixels around the image improving the accuracy on many tasks. Motivated by this finding our idea is to investigate how adding parameters to the pixel space affects the accuracy. To be more specific, rather than randomly altering pixels one idea we could potentially explore is to use CLIP pre-trained model to identify e.g. 10% most important pixels in an image through a task-dependent prompt *"point to most important object in the image"* employing techniques like GradCam Selvaraju et al. (2020) or similar methods - an example is shown in Figure 1-2. Following this identification we consider the incorporation of these newly learned pixels as a residual sum to the already identified most important pixels in the image and potentially blur not important areas.

### 1.5.3 Other methods that require training

ProGrad Zhu et al. (2023) addresses the overfitting issue observed in CoOp, which can lead to forgetting the general knowledge of zero-shot CLIP. It introduces a refined training procedure where the prompt parameters of CoOp are updated based on the gradient of the cross-entropy loss between CoOp predictions and true labels. If this update aligns with the gradient update of the general knowledge direction, computed using the Kullback-Leibler (KL) divergence

between CoOp and zero-shot CLIP predictions, the parameters are updated in the usual manner. However, if the update is in a different direction, indicating potential forgetting of pre-trained CLIP general knowledge, the parameter update is adjusted to prevent an increase in KL loss. ProDA Zhang et al. (2021a) focuses on addressing the time-consuming process of manually creating diverse and informative prompts for various tasks. It suggests learning the distributions of these prompts, noting the challenge of modeling these distributions at the text embedding level due to the dispersed nature of different descriptions for the same class in the embedding space. Instead, ProDA leverages the closely clustered embeddings from the text encoder for different descriptions of the same class, which can be effectively modeled with simple distributions like a multivariate Gaussian. These distributions are used to sample the representation of each class in the textual embedding space, with the mean serving as the textual representation for the classes in practice.

## 1.6 Training-free methods

### 1.6.1 Intra-modal overlap correction through adapters

In Zhang et al. (2021b), Tip-Adapter is introduced as a training-free method leveraging CLIP. It operates by encoding images from a given dataset with CLIP's image encoder to form a key-value cached model. The images are used as keys, and their corresponding one-hot encoded labels serve as values. This cached model encapsulates new knowledge from the few-shot training examples without requiring any training process. During testing, the test image acts as a query. An affinity matrix is computed to represent the similarity between the test image and the training images. Tip-Adapter logits are then calculated by combining the new knowledge from the cached model with the prior knowledge from CLIP. An extension named Tip-Adapter F allows the adjustment of the initialized cache model by making the keys learnable parameters.

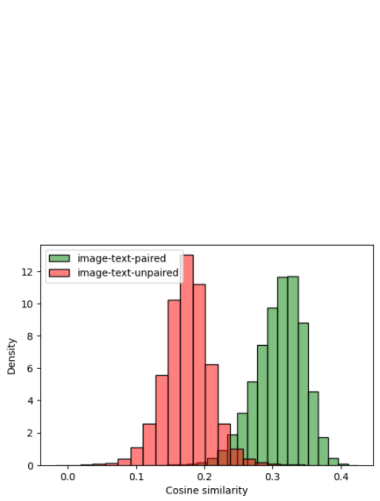
Multiple research papers Liang et al. (2022); Udandarao, Gupta and Albanie (2023) point out that there exists a modality gap in CLIP model, which affects the similarity computed in the image space (intra-modal) resulting in high overlap between paired and unpaired images, meaning that Tip-Adapter, that computes the affinities in the image space is affected by it. This intra-modal overlap arises from the contrastive loss that CLIP is trained with that maximizes the cosine similarity between paired image and text (inter-modal), but ignores the intra-modal similarity (image-image). This is illustrated in Figure 1-3, where we can observe that the overlap in image-text space 1-3a between unpaired and paired classes is relatively small while it is relatively larger in image space 1-3b.

CaFo Zhang et al. (2023) is another training-free method that does not only exploit the knowledge of CLIP, but also of other foundational models. GPT-3 is used to generate $K$

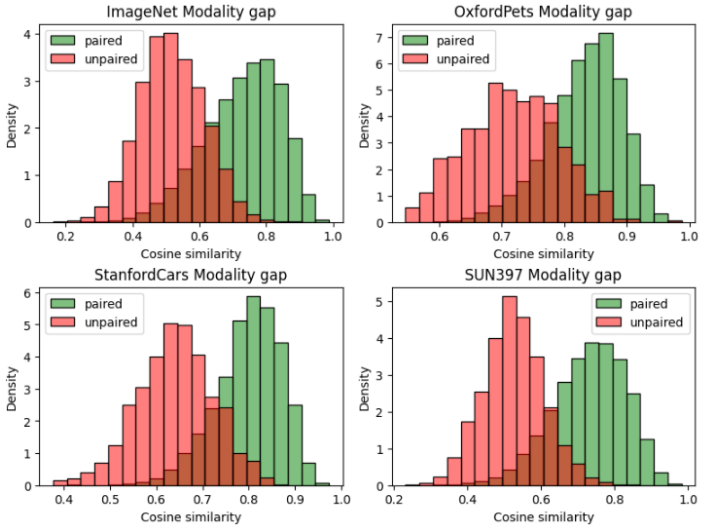
additional specialized prompts by inputting into the model 5 template questions. These prompts are then encoded with CLIP and ensembled. With DALL-E they augment the few-shot examples by generating additional images filtering them with CLIP. Finally, self-supervised vision-only DINOv2 Oquab et al. (2023) model, together with CLIP vision encoder are used to create cache models like in Tip-Adapter (that can also be learnable) from both DALL-E generated and few-shot training examples. The final prediction is given by the sum of CLIP zero-shot and weighted cache model derived from DINOv2 and CLIP.

CaFo is less susceptible to the intra-modal overlap (IMO) primarily because they incorporate a DINOv2 cached model in addition to a Tip-Adapter-based cached model. DINOv2, being trained exclusively on images, is inherently not affected by IMO. The assumption is that logits' weights of DINOv2 and Tip-Adapter favor DINOv2 due to IMO issue. However, it is important to note that the paper does not provide any empirical evidence for this, thus, further investigation and analysis would be necessary to validate this claim.

Another problem with CaFo is that it needs to keep two foundational models at the same time in memory which can be expensive. Dropping DINOv2, as shown in the paper, affects the accuracy by around 1-3% depending on the number of shots. We believe that correcting IMO by introducing adapters into the CLIP model would allow to generate distinct features from the original CLIP producing a different cached model, serving thus similar purpose to DINOv2 but at a much smaller cost.




(a) Inter-modal similarity for ImageNet validation




(b) Intra-modal similarity for different validation datasets

Figure 1-3: Intra-modal and inter-modal CLIP cosine similarities

Authors of Tip-X method Udandarao, Gupta and Albanie (2023) propose to use inter-modal distances as a bridge to handle intra-modal overlap (IMO) problem. They construct an

affinity matrix similarly to Tip-Adapter but in image-text space where the similarity measure between two images is given by the KL divergence between CLIP logits (Eq. 1.4) instead of cosine similarity. The method then combines this affinity matrix in image-text space with CLIP zero-shot initial knowledge and the affinity matrix in image space, like in Tip-Adapter, to predict the label.

**Intra-modal correction with supervised training** While the authors of Tip-X may have achieved superior results compared to the original Tip-Adapter, they still incorporate Tip-Adapter logits into the final prediction which are influenced by IMO. Thus, we propose to correct the intra-modal overlap directly within the image space and replace image-image affinity matrix component with IMO corrected features that are compatible with both Tip-Adapter and Tip-X. To do this we could incorporate a bottleneck adapter Chen et al. (2022) into CLIP visual encoder and fine-tune it in a supervised manner on selected images and classes, for example from Google Open-Image dataset Open Images Dataset V5 (2019). It's worth noting that adapters are lightweight and would introduce approximately 0.80% (around 1 million) new parameters to the model. Such a solution would be more cost-effective than maintaining a new vision model in memory like Dinov2 used in CaFo. Our experiments on the intra-modal overlap correction are shown in chapter 6.

**Intra-modal overlap correction with self-supervised training** Another research question that has not been explored in the literature to the best of our knowledge involves investigating whether training the adapters in a self-supervised manner, rather than relying on labeled data, can effectively reduce this IMO and improve performance in general. In a prior study, known as SVL-Adapter Pantazis et al. (2022), the authors pursued a self-supervised pre-training approach using SimCLR Chen et al. (2020). They initiated a visual encoder based on CLIP to speed up convergence and trained on unlabeled images taken in the wild that are distinct from those used in CLIP's training dataset. During prediction, given few-shot training data, they combined predictions from zero-shot CLIP with a new, frozen visual encoder training an adapter on top of it that outputs the probability of classes. This approach yielded improved results in few-shot learning tasks, particularly on datasets significantly different from the curated internet data. In our research we are going to investigate a slightly different question. Rather than training a visual encoder from scratch, we will focus solely on training the bottleneck adapters. Among the self-supervised methods under consideration are SimCLR, BYOL Grill et al. (2020), DINO Caron et al. (2021), and others. This investigation aims to determine if self-supervised training of adapters, without training the entire visual encoder, can help to reduce IMO and achieve similar performance to supervised adapters training.

## 1.7 Zero-shot methods

Zero-shot methods enable object classification in images without requiring specific training examples or additional parameter fine-tuning. Earlier in this chapter we introduced the zero-shot CLIP method that classifies images by embedding an image and textual description of classes within a predefined textual prompt to then compute the cosine similarity between them, with the highest similarity corresponding to the predicted class.
CALIP Guo et al. (2022) improves on the standard zero-shot CLIP method by introducing a parameter-free cross-attention module. The approach leverages spatial visual features and textual features from the final encoder layers to facilitate interaction between the textual and visual modalities. Since the features from these last layers are already in a shared embedding space, attention computation can be performed directly without the need for query, key, and value projection matrices. This eliminates the necessity for any learnable parameters.
SuS-X Udandarao, Gupta and Albanie (2023), a variation of Tip-X, uses Tip-X to compute the similarity between the support set and test images. The support set is formed by either generating or sampling images from an external database. In case of external database sampling, LAION-5B[1] database is used which is a vast collection with 5.85 billion image-text pairs. Task class names are used to retrieve the support images. In case of the generated support set the Stable Diffusion Rombach et al. (2022) model is used employing Large Language Models (LLMs) to generate rich class descriptions. Notably, since this method does not use few-shot examples from the target dataset it can be categorized as zero-shot.

## 1.8 Generative methods

Standard prompting templates like *"A picture of {class}"* have major limitations - they require considerable human effort to create, lack specificity to be applicable across all categories in a given dataset and necessitate prior knowledge about a given dataset to generate high-quality prompts. To correct these limitations authors of CuPL Pratt et al. (2022) introduced a solution by using LLMs to produce customized prompts. This approach differentiates between two types of prompts: LLM-prompts which guide the LLM model to generate descriptions for the dataset categories and image-prompts which stem from LLM prompts and offer detailed descriptions for each category. By leveraging such highly specific and detailed prompts they enhanced CLIP's ability to emphasize image regions critical for accurate classification.
VisDesc Menon and Vondrick (2022) is similar to CuPL as it also utilizes LLM to generate prompts at scale for CLIP. However, rather than producing descriptive captions for specific class categories VisDesc generates descriptive features containing detailed descriptions for

[1] https://laion.ai/blog/laion-5b/

individual features of each class. These descriptive features are employed to calculate similarity with a provided image embedding and the predicted class is determined by selecting the highest average similarity among these descriptive features across all classes. This method inherently provides explainability by looking at the most similar features for the predicted class.
As explained in more detail in the previous section, SuS-X utilizes the LLM to generate detailed class descriptions which are then inputted into the Stable Diffusion model to generate class images that can be utilized to perform zero-shot classification with CLIP.

### 1.8.1 Generating negative examples for better differentiation between similar classes

To the best of our knowledge most generative approaches in vision-language few-shot learning tend to focus solely on positive examples and the concept of generating negative or counterfactual examples remains underexplored. The goal is to create images that closely relate to a specific class while differing in particular attributes. By employing this method, the model can potentially gain a clearer understanding of class boundaries, fostering a more sophisticated differentiation between different categories. One related work is the CPL He et al. (2022) model, where they explore the counterfactual generation process within the feature space. This involves extracting the closest negative examples in the text space, given that semantic concepts in text space are typically less complex and more easily extractable. They then employ a weight vector to merge the positive and negative examples in the feature space. This vector learns the minimum change in feature space required to alter the label. Our proposed approach involves directly generating such images rather than focusing solely on the feature space. Generating images directly offers the advantage of understanding the model's reasoning process through direct visualization of the generated images. Additionally, this method allows for the expansion to multiple negative examples, in contrast to the CPL method which works with only one negative example.

# Bibliography


Bahng, H., Jahanian, A., Sankaranarayanan, S. and Isola, P., 2022. Exploring visual prompts for adapting large-scale models [Online]. Available from: https://doi.org/10.48550/arXiv.2203.17274 [Accessed 2024-01-16].

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P. and Joulin, A., 2021. Emerging properties in self-supervised vision transformers. *arxiv:2104.14294 [cs]* [Online]. Available from: https://arxiv.org/abs/2104.14294.

Chen, S., Ge, C., Zhan, T., Wang, J., Song, Y., Wang, J. and Luo, P., 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *arxiv (cornell university)* [Online]. Available from: https://doi.org/10.48550/arxiv.2205.13535 [Accessed 2023-09-03].

Chen, T., Kornblith, S., Norouzi, M. and Hinton, G., 2020. A simple framework for contrastive learning of visual representations. *arxiv:2002.05709 [cs, stat]* [Online]. Available from: https://arxiv.org/abs/2002.05709.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *Cvpr*.

Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H. and Qiao, Y., 2021. Clip-adapter: Better vision-language models with feature adapters [Online]. Available from: https://doi.org/10.48550/arXiv.2110.04544 [Accessed 2024-01-03].

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R. and Valko, M., 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arxiv:2006.07733 [cs, stat]* [Online]. Available from: https://arxiv.org/abs/2006.07733.

Guo, Z., Zhang, R., Qiu, L., Ma, X., Miao, X., He, X. and Cui, B., 2022. Calip: Zero-shot enhancement of clip with parameter-free attention [Online]. Available from: https://doi.org/10.48550/arXiv.2209.14169 [Accessed 2024-01-03].

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Deep residual learning for image recognition [Online]. Available from: https://arxiv.org/abs/1512.03385.

He, X., Yang, D., Feng, W., Fu, T.J., Akula, A., Jampani, V., Narayana, P., Basu, S., Wang, W.Y. and Wang, X., 2022. Cpl: Counterfactual prompt learning for vision and language models [Online]. Available from: https://doi.org/10.18653/v1/2022.emnlp-main.224 [Accessed 2024-01-22].

Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B. and Lim, S.N., 2022. Visual prompt tuning [Online]. Available from: https://doi.org/10.48550/arXiv.2203.12119.

Liang, W., Zhang, Y., Kwon, Y., Yeung, S. and Zou, J., 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning [Online]. Available from: https://doi.org/10.48550/arXiv.2203.02053 [Accessed 2024-01-03].

Menon, S. and Vondrick, C., 2022. Visual classification via description from large language models [Online]. Available from: https://openreview.net/forum?id=jlAjNL8z5cs [Accessed 2024-01-22].

Nilsback, M. and Zisserman, A., 2008. Automated flower classification over a large number of classes. *2008 sixth indian conference on computer vision, graphics image processing* [Online]. Available from: https://www.semanticscholar.org/paper/Automated-Flower-Classification-over-a-Large-Number-Nilsback-Zisserman/02b28f3b71138a06e40dbd614abf8568420ae183 [Accessed 2024-01-03].

Open images dataset v5, 2019. [Online]. Available from: https://storage.googleapis.com/openimages/web/index.html.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A. and Bojanowski, P., 2023. Dinov2: Learning robust visual features without supervision [Online]. Available from: https://doi.org/10.48550/arXiv.2304.07193.

Pantazis, O., Brostow, G., Jones, K. and Mac Aodha, O., 2022. Svl-adapter: Self-supervised

adapter for vision-language pretrained models [Online]. Available from: https://doi.org/10.48550/arXiv.2210.03794 [Accessed 2024-01-03].

Pratt, S., Covert, I., Liu, R. and Farhadi, A., 2022. What does a platypus look like? generating customized prompts for zero-shot image classification [Online]. Available from: https://openaccess.thecvf.com/content/ICCV2023/papers/Pratt_What_Does_a_Platypus_Look_Like_Generating_Customized_Prompts_for_ICCV_2023_paper.pdf [Accessed 2024-01-22].

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I., 2021. Learning transferable visual models from natural language supervision. *arxiv:2103.00020 [cs]* [Online]. Available from: https://arxiv.org/abs/2103.00020.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. *Cvpr* [Online]. Available from: https://arxiv.org/abs/2112.10752.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2020. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International journal of computer vision* [Online], 128, p.336–359. Available from: https://doi.org/10.1007/s11263-019-01228-7.

Sun, J., Qin, J., Lin, Z. and Chen, C., 2023. Prompt tuning based adapter for vision-language model adaption [Online]. Available from: https://doi.org/10.48550/arXiv.2303.15234 [Accessed 2024-01-03].

Udandarao, V., Gupta, A. and Albanie, S., 2023. Sus-x: Training-free name-only transfer of vision-language models [Online]. Available from: https://doi.org/10.48550/arXiv.2211.16198 [Accessed 2024-01-03].

Yu, T., Lu, Z., Jin, X., Chen, Z. and Wang, X., 2023. Task residual for tuning vision-language models [Online]. Available from: https://doi.org/10.48550/arXiv.2211.10277 [Accessed 2024-01-03].

Yu, Z., 2022. A survey on clip-guided vision-language tasks. *Highlights in science, engineering and technology* [Online], 12, pp.153–159. Available from: https://doi.org/10.54097/hset.v12i.1418 [Accessed 2022-11-28].

Zang, Y., Li, W., Zhou, K., Huang, C. and Loy, C.C., 2022. Unified vision and language prompt learning [Online]. Available from: https://doi.org/10.48550/arXiv.2210.07225 [Accessed 2024-01-03].

Zhang, P., Zhang, B., Zhang, T., Chen, D. and Wang, Y., 2021a. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation [Online]. Available from: https://openaccess.thecvf.com/content/CVPR2021/papers/Zhang_Prototypical_Pseudo_Label_Denoising_and_Target_Structure_Learning_for_Domain_CVPR_2021_paper.pdf [Accessed 2024-01-22].

Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y. and Li, H., 2021b. Tip-adapter: Training-free clip-adapter for better vision-language modeling [Online]. Available from: https://doi.org/10.48550/arXiv.2111.03930 [Accessed 2024-01-16].

Zhang, R., Hu, X., Li, B., Huang, S., Deng, H., Li, H., Qiao, Y. and Gao, P., 2023. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners [Online]. Available from: https://doi.org/10.48550/arXiv.2303.02151 [Accessed 2024-01-03].

Zhou, K., Yang, J., Loy, C.C. and Liu, Z., 2022a. Conditional prompt learning for vision-language models [Online]. Available from: https://doi.org/10.48550/arXiv.2203.05557 [Accessed 2024-01-03].

Zhou, K., Yang, J., Loy, C.C. and Liu, Z., 2022b. Learning to prompt for vision-language models. *arxiv:2109.01134 [cs]* [Online]. Available from: https://arxiv.org/abs/2109.01134.

Zhu, B., Niu, Y., Han, Y., Wu, Y. and Zhang, H., 2023. Prompt-aligned gradient for prompt tuning [Online]. Available from: https://doi.org/10.48550/arXiv.2205.14865 [Accessed 2024-01-03].