COL8628/COL828 Assignment 1
Transformer Models for Implant Classification

Aastha A K Verma
Entry Number: 2022CS11607
Indian Institute of Technology Delhi

September 30, 2025

Contents

1 Introduction **3**

2 Related Work **3**
2.1 Vision Transformers . . . 3
2.2 CLIP and Vision-Language Models . . . 3
2.3 Prompt Learning . . . 3

3 Datasets **4**
3.1 Orthonet Dataset . . . 4
3.2 Pacemaker Dataset . . . 4
3.3 Preprocessing . . . 5

4 Methodology **6**
4.1 Model Architectures . . . 6
4.1.1 Vision Transformer (ViT) . . . 6
4.1.2 Prompt Learning Methods . . . 6
4.2 Metrics . . . 6

5 Training Configuration **6**

6 Experimental Results **7**
6.1 Task 1: Orthonet Dataset . . . 7
6.1.1 Fine-tuning Pretrained ViT Variants (Subtask 1.1) . . . 7
6.1.2 Zero-Shot Classification with CLIP (Subtask 1.2) . . . 8
6.1.3 Prompt Learning Methods (Subtask 1.3) . . . 9
6.2 Task 2: Pacemaker Dataset . . . 10
6.2.1 Fine-tuning Pretrained ViT Variants (Subtask 2.1) . . . 10
6.2.2 Zero-Shot Classification with CLIP (Subtask 2.2) . . . 11
6.2.3 Non-Hierarchical Prompt Learning . . . 11
6.2.4 Hierarchical Prompt Learning (Subtask 2.3) . . . 12

7 Analysis and Discussion **13**
7.1 Comparison of Pretraining Strategies . . . 13
7.2 Zero-Shot vs Fine-Tuning Performance . . . 13
7.3 Prompt Learning Effectiveness . . . 14
7.4 Hierarchical Learning Benefits . . . 14
7.5 Medical Imaging Specific Observations . . . 16

8 Implementation Details **16**
8.1 Models and other details 16
8.2 Code Structure 16
8.2.1 Core implementational files 16
8.2.2 Running scripts 17
8.3 Model Weights 17
8.4 External Libraries 17

Abstract

This report presents an experimental evaluation of Vision Transformer (ViT) models and prompt-learning techniques for medical implant classification on two datasets: Orthonet and Pacemakers. We investigate the transfer learning capabilities of different pretrained ViT variants (ImageNet-21k, CLIP, DINOv2), explore zero-shot classification using CLIP, and analyze advanced prompt-learning methods including CoOp, CoCoOp, and MaPLe. Additionally, we investigate a hierarchical prompt learning approach for fine-grained classification on the Pacemaker dataset. The findings demonstrate that the finetuning and zero-shot have a trade-off for specificity and generalizability. However Prompt Learning methods provide a sweet spot between these in a parameter efficient manner.

1 Introduction

Medical image classification has gained significant importance with the advancement of deep learning techniques. Vision Transformers (ViTs) have emerged as powerful alternatives to convolutional neural networks, particularly when leveraging large-scale pretraining. This assignment explores the application of different pretrained ViT models and prompt-learning techniques for implant classification using X-ray images.

The work is divided into two main tasks:

- **Task 1**: Classification on the Orthonet dataset (orthopedic knee implants)
- **Task 2**: Classification on the Pacemakers dataset (cardiac devices with hierarchical taxonomy)

Each task involves fine-tuning pretrained models, zero-shot classification using CLIP, and advanced prompt-learning methods. The Pacemaker dataset additionally explores **hierarchical prompt learning** to leverage the manufacturer-series-model taxonomy.

2 Related Work

2.1 Vision Transformers

Vision Transformers [1] have revolutionized computer vision by adapting the transformer architecture originally designed for natural language processing. Unlike CNNs, ViTs process images as sequences of patches, enabling global attention mechanisms that capture long-range dependencies and achieve state-of-the-art performance on various benchmarks.

2.2 CLIP and Vision-Language Models

CLIP [4] introduces a novel approach to learning visual representations through natural language supervision. By training on image-text pairs, CLIP learns transferable visual features that can be applied to various downstream tasks through zero-shot classification, bridging the gap between vision and language understanding.

2.3 Prompt Learning

Recent advances in prompt learning for vision-language models have shown promising results. CoOp [7] introduces learnable context optimization, while CoCoOp [6] adds conditional prompting. MaPLe [2] extends this to multi-modal prompt learning, enabling more flexible adaptation to diverse tasks with minimal labeled data.

3 Datasets

3.1 Orthonet Dataset

The Orthonet dataset consists of X-ray images of orthopedic knee implants. There are two types of implants (Hip, 8 classes and Knee, 4 classes), but have different manufacturers and models.

Figure 1: Some class examples from the Orthonet dataset

The dataset consists of 12 classes. It is split using a stratified split to maintain class balance.

Set	Samples
Train	935
Validation	234
Test	180
Total Classes	12

Figure 2: Orthonet Dataset statistics with train/validation/test split.

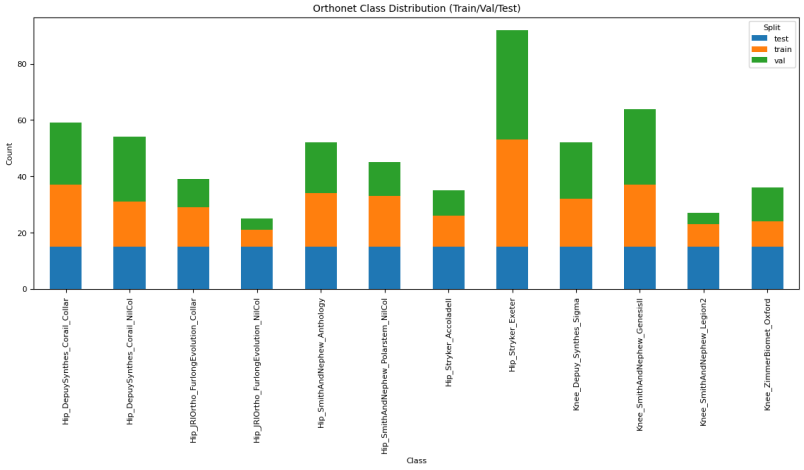

Figure 3: Orthonet dataset distribution.

3.2 Pacemaker Dataset

The Pacemaker dataset contains chest radiographs with cardiac implants. It features a hierarchical taxonomy with 45 classes organized into 5 manufacturers: BIO, BOS, MDT, SOR, and STJ. The dataset includes [number] training and [number] test images...

Set	Samples
Train	1160
Validation	291
Test	225
Total Classes	45

Table 1: Dataset statistics with train/validation/test split.

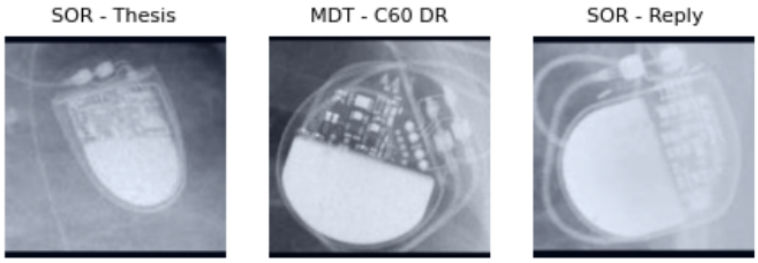

Figure 4: Some class examples from the Pacemaker dataset

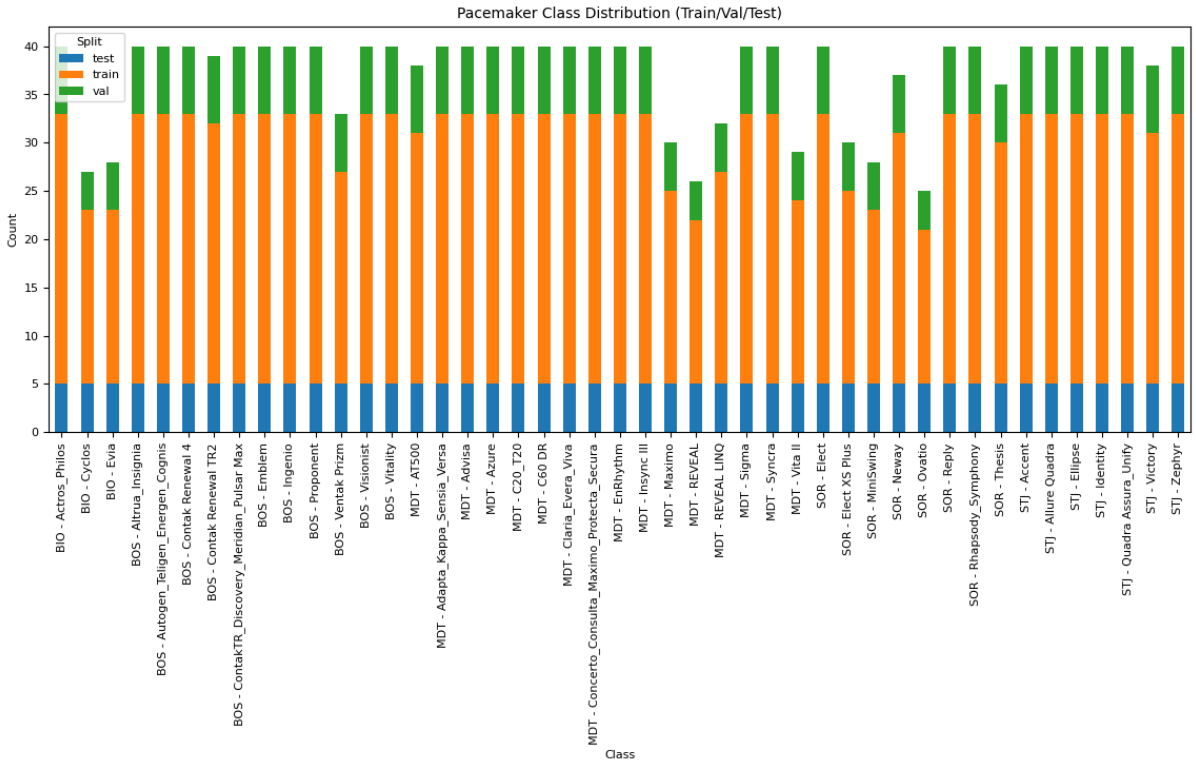

Figure 5: Pacemakers dataset distribution

3.3 Preprocessing

Both datasets were split into **80% train and 20% validation** using **stratified sampling** to maintain class balance. All images were loaded in RGB, resized to match the model input (**224×224 for CLIP/Imagenet, 518×518 for DINOv2**), normalized with model-specific mean and std, and optionally augmented with small rotations, flips, and contrast adjustments

during training. Further, we tried fine preprocessing with CLAHE, black border cropping, square padding etc.

4 Methodology

4.1 Model Architectures

4.1.1 Vision Transformer (ViT)

We utilize the ViT architecture with different pretraining strategies:

- **ImageNet-21k**: Standard supervised pretraining on ImageNet-21k
- **CLIP**: Contrastive language-image pretraining
- **DINOv2**: Self-supervised learning with knowledge distillation

For every dataset, we test three strategies:

- Finetuning the ViT with preloaded weights.
- Zero shot classification with CLIP, using some hand engineered text prompts.
- Prompt Learning methods on CLIP weights, as described below.

4.1.2 Prompt Learning Methods

We implement three prompt learning approaches:

- **CoOp**: Context Optimization with learnable continuous prompts
- **CoCoOp**: Conditional Context Optimization with input-dependent prompts
- **MaPLe**: Multi-modal Prompt Learning across multiple layers

Hierarchical Prompt Learning: For Pacemaker dataset, we have implemented both single-pass and double-pass prompt learning. The former is the usual version. For the latter, the training is first done on the 5 manufacturer superclasses. Then in the second stage, we further tune the learnt prompts in a finer grained manner on all the 45 classes.

4.2 Metrics

The classification quality is measured with **Top-k accuracy**, **F1** score, and **AUC-ROC**.

In the multiclass setting, the AUC-ROC was computed using a one-vs-rest (OvR) strategy with weighted averaging, which evaluates how well the model ranks the true class against all others across decision thresholds, providing a measure of ranking quality that is less sensitive to class imbalance than accuracy.

5 Training Configuration

Variant	lr	epochs	batch size
imagenet	1e-4	20	32
clip	1e-5	20	32
dinov2	1e-5	20	32

Table 2: Finetune configurations for Orthonet.

Type	Prompt Template
Hip	“an X-ray of a {} hip implant”
Knee	“an X-ray of a {} knee implant”

Table 3: Zero-shot CLIP prompt templates used for Orthonet dataset

Method	n_ctx	ctx_init	class token	depth	lr	epochs	batch size
CoOp	16	""	end	-	0.003	100	32
CoCoOp	16	""	end	–	0.002	100	32
MaPLe	2	"a photo of a"	end	9	0.0035	50	32

Table 4: Prompt-based configurations (CoOp, CoCoOp, MaPLe) for Orthonet.

Variant	lr	epochs	batch size
imagenet	1e-4	20	32
clip	1e-5	20	32
dinov2	1e-5	20	32

Table 5: Finetune configurations for Pacemakers.

Prompt Template
“an image of a {} implant”

Table 6: Zero-shot CLIP prompt templates used for Pacemaker dataset

Method	n_ctx	ctx_init	class token	depth	lr	epochs	batch size
CoOp	16	""	end	-	0.003	100	32
CoCoOp	16	""	end	–	0.002	100	32
MaPLe	2	"a photo of a"	end	9	0.0035	50	32

Table 7: Prompt-based configurations (CoOp, CoCoOp, MaPLe) for Pacemakers.

6 Experimental Results

6.1 Task 1: Orthonet Dataset

6.1.1 Fine-tuning Pretrained ViT Variants (Subtask 1.1)

Table 8 shows the performance of different pretrained ViT variants on the Orthonet dataset.

Table 8: Performance comparison of pretrained ViT variants on Orthonet dataset

Model	Top-1 Accuracy (%)	F1-Score	AUC-ROC
ViT (ImageNet-21k)	80.00	0.7953	0.9788
ViT (CLIP)	87.78	0.8694	0.9936
ViT (DINOv2)	93.33	0.9318	0.9982

Figure 6: Finetuning training curve for ImageNet weights on Orthonet dataset

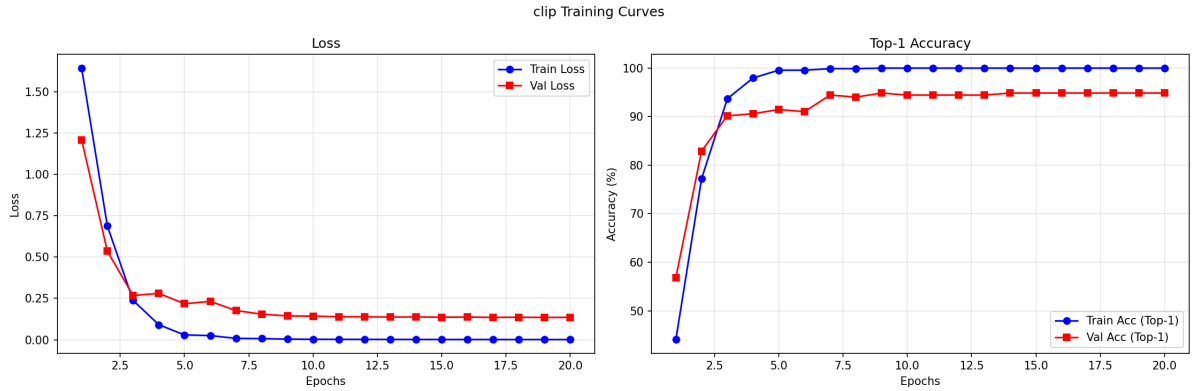

Figure 7: Finetuning training curve for CLIP weights on Orthonet dataset

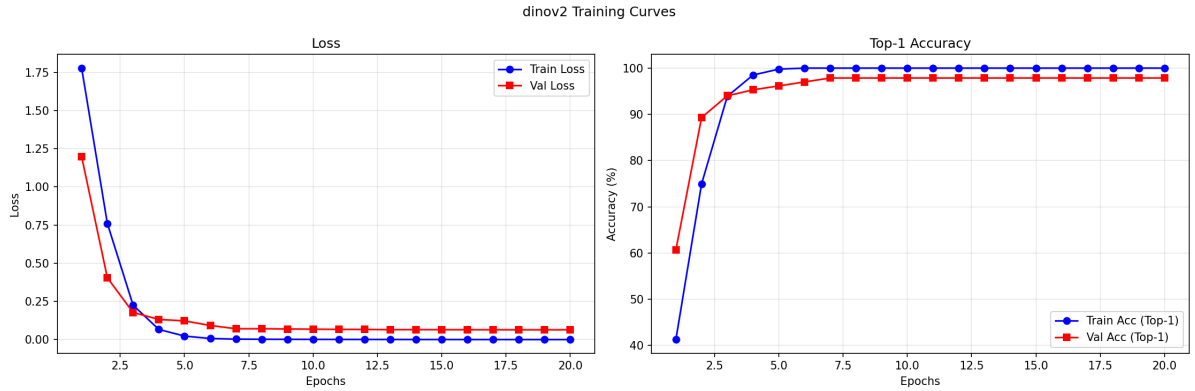

Figure 8: Finetuning training curve for DINOv2 weights on Orthonet dataset

6.1.2 Zero-Shot Classification with CLIP (Subtask 1.2)

The zero-shot classification results using CLIP are presented in Table 9.

Table 9: Zero-shot classification performance on Orthonet dataset

Method	Top-1 Accuracy (%)	F1-Score	AUC-ROC
CLIP Zero-shot	23.89	0.1390	0.7410

6.1.3 Prompt Learning Methods (Subtask 1.3)

Table 10 compares the performance of different prompt learning methods.

Table 10: Prompt learning methods performance on Orthonet dataset

Method	Top-1 Accuracy (%)	F1-Score	AUC-ROC
CoOp	47.22	0.4391	0.8988
CoCoOp	44.44	0.4019	0.8863
MaPLe	73.89	0.7319	0.9545

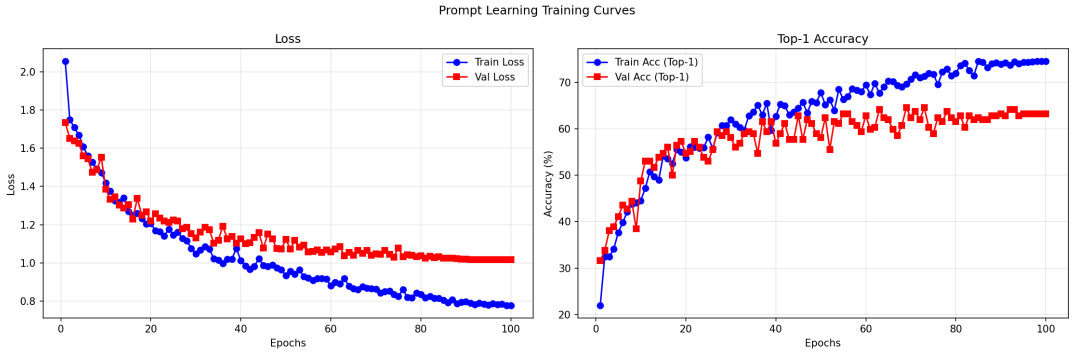

Figure 9: Training curve for CoOp on Orthonet dataset

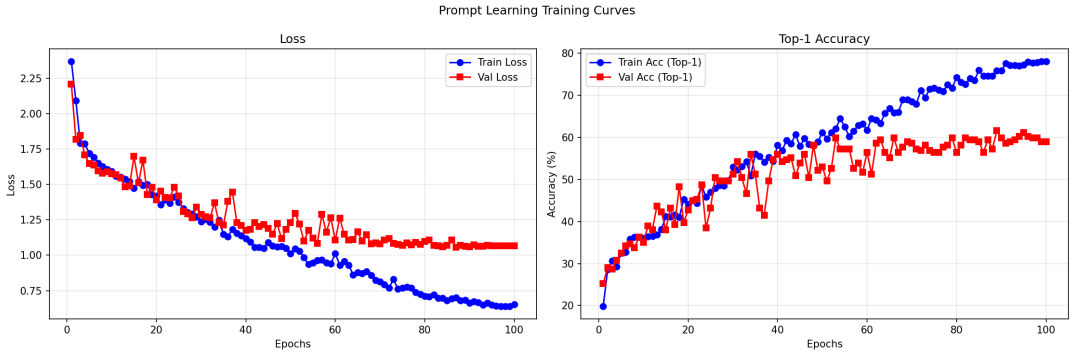

Figure 10: Training curve for CoCoOp on Orthonet dataset

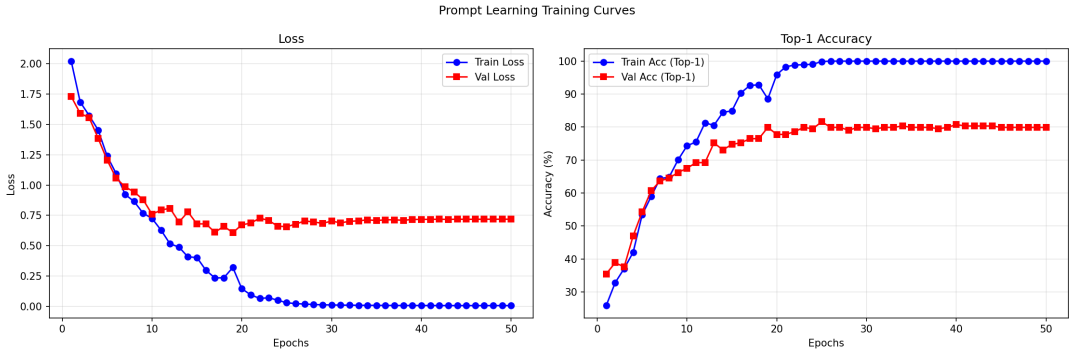

Figure 11: Training curve for MaPLe on Orthonet dataset

6.2 Task 2: Pacemaker Dataset

6.2.1 Fine-tuning Pretrained ViT Variants (Subtask 2.1)

Table 11 presents the results for ViT fine-tuning on the Pacemaker dataset.

Table 11: Performance comparison of pretrained ViT variants on Pacemaker dataset

Model	Top-1 Acc (%)	Top-3 Acc (%)	F1-Score	AUC-ROC
ViT (ImageNet-21k)	85.78	95.56	0.8448	0.9944
ViT (CLIP)	93.78	97.33	0.9374	0.9989
ViT (DINOv2)	86.67	96.00	0.8648	0.9962

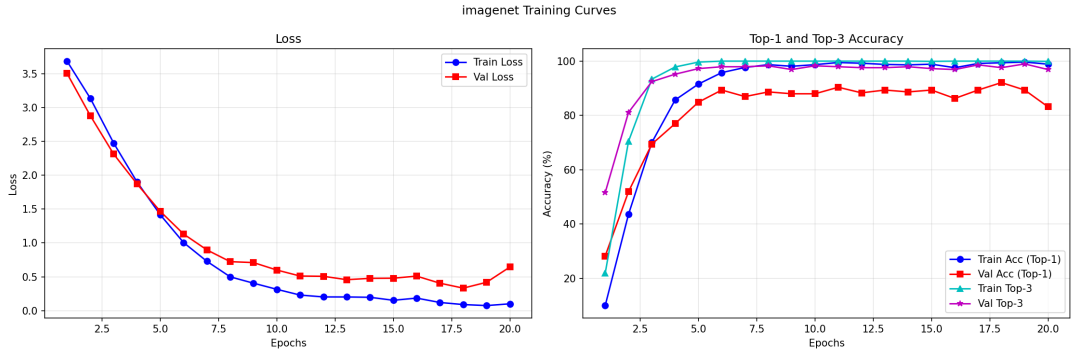

Figure 12: Finetuning training curve for ImageNet weights on Pacemaker dataset

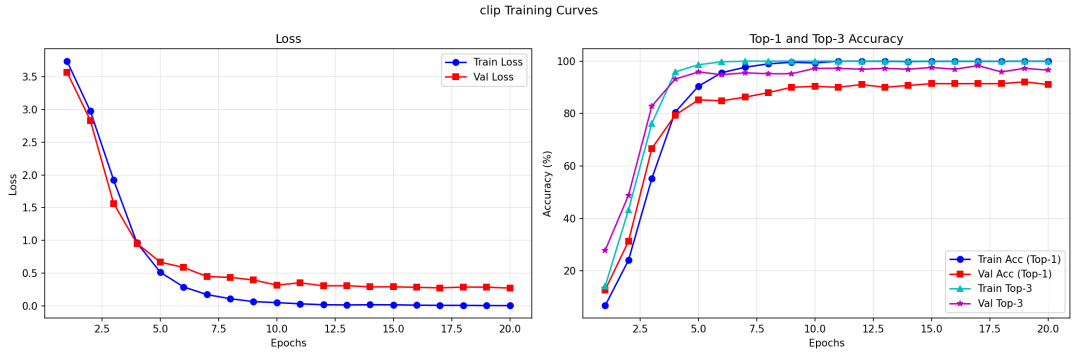

Figure 13: Finetuning training curve for CLIP weights on Pacemaker dataset

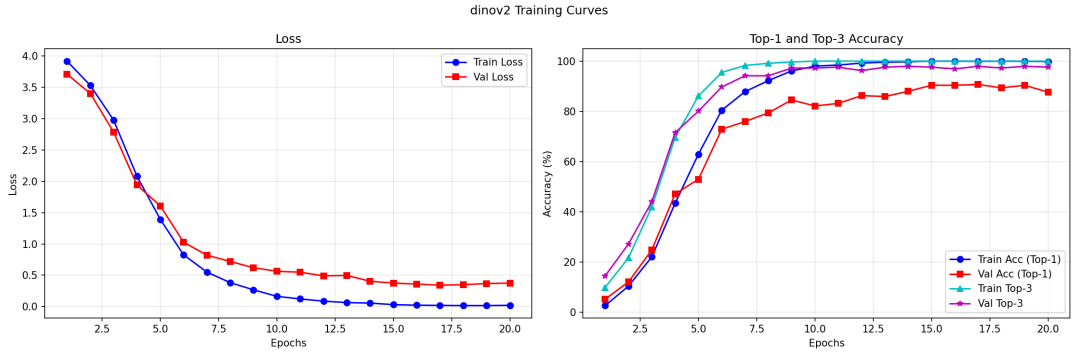

Figure 14: Finetuning training curve for DINOv2 weights on Pacemaker dataset

6.2.2 Zero-Shot Classification with CLIP (Subtask 2.2)

Table 12: Zero-shot classification performance on Pacemaker dataset

Method	Top-1 Acc (%)	Top-3 Acc (%)	F1-Score	AUC-ROC
CLIP Zero-shot	2.22	6.22	0.0065	0.4825

6.2.3 Non-Hierarchical Prompt Learning

Table 13: Prompt learning methods performance on Pacemaker dataset

Method	Top-1 Acc (%)	Top-3 Acc (%)	F1-Score	AUC-ROC
CoOp	37.78	69.78	0.3682	0.9430
CoCoOp	43.11	70.22	0.4078	0.9455
MaPLe	62.67	85.33	0.6251	0.9802

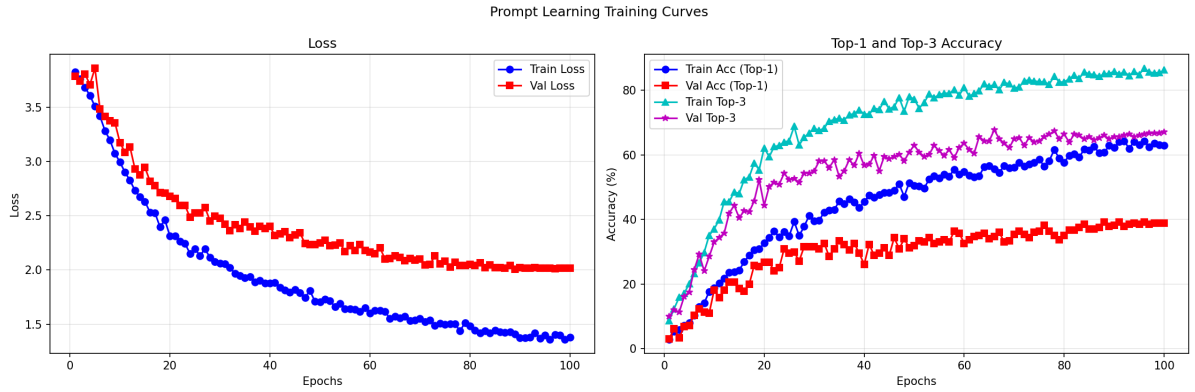

Figure 15: Training curve for single-pass CoOp on Pacemaker dataset

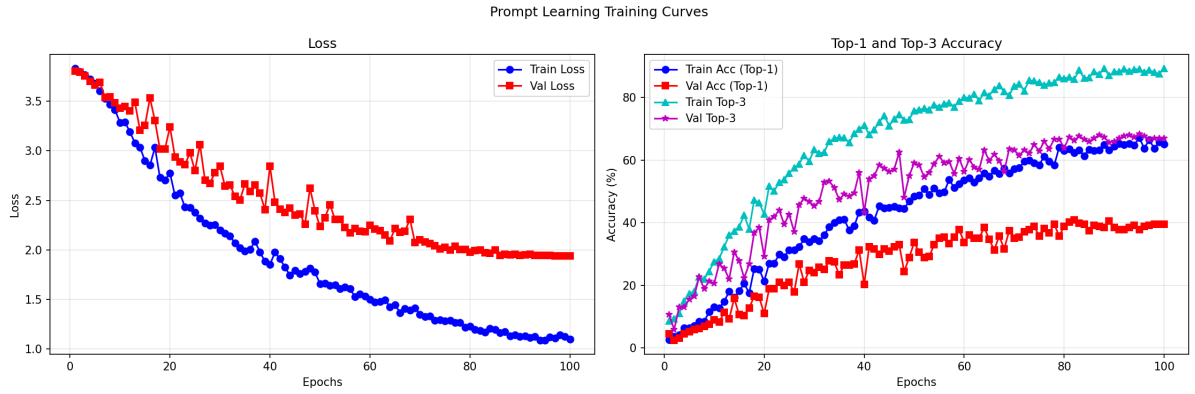

Figure 16: Training curve for single-pass CoCoOp on Pacemaker dataset

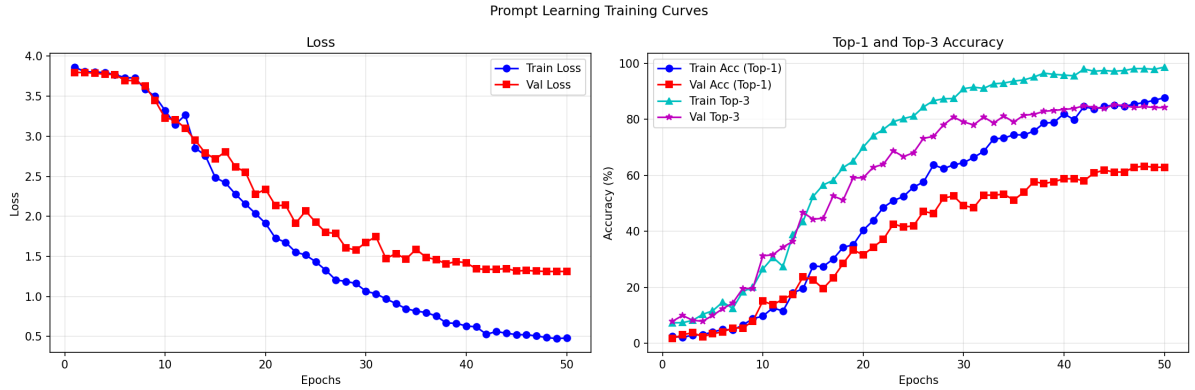

Figure 17: Training curve for single-pass MaPLe on Pacemaker dataset

6.2.4 Hierarchical Prompt Learning (Subtask 2.3)

Stage 1: Manufacturer-level Classification

Table 14: Manufacturer-level hierarchical prompt learning results on Pacemaker dataset (Stage 1: 5 classes)

Method	Top-1 Acc (%)	Top-3 Acc (%)	F1-Score	AUC-ROC
CoOp (Hierarchical)	65.33	93.33	0.6462	0.8843
CoCoOp (Hierarchical)	64.44	93.78	0.6406	0.8943
MaPLe (Hierarchical)	88.00	99.11	0.8770	0.9862

Stage 2: Fine-grained Classification

Table 15: Hierarchical prompt learning on Pacemaker dataset (Stage 2: Fine-grained, 45 classes)

Method	Top-1 Acc (%)	Top-3 Acc (%)	F1-Score	AUC-ROC
CoOp (Hierarchical)	42.22	64.89	0.3887	0.9325
CoCoOp (Hierarchical)	48.00	76.00	0.4704	0.9516
MaPLe (Hierarchical)	75.11	92.00	0.7427	0.9913

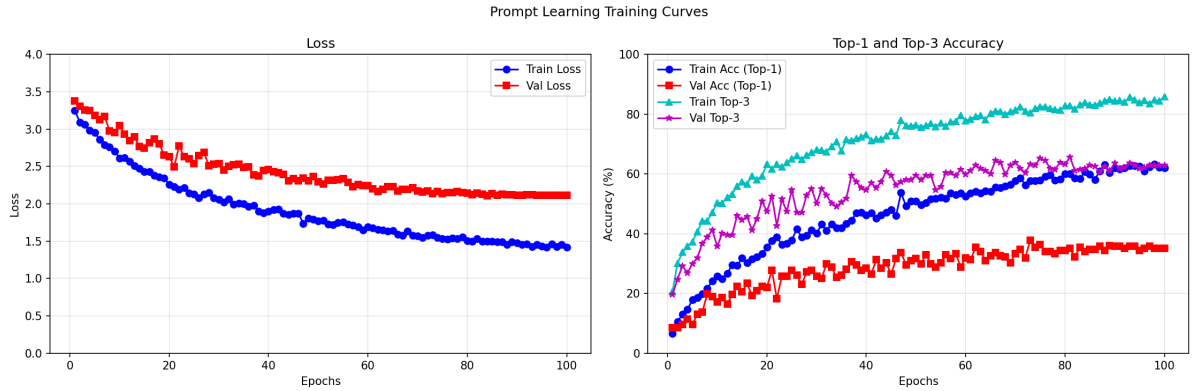

Figure 18: Stage 2 CoOp training curve for Pacemaker dataset

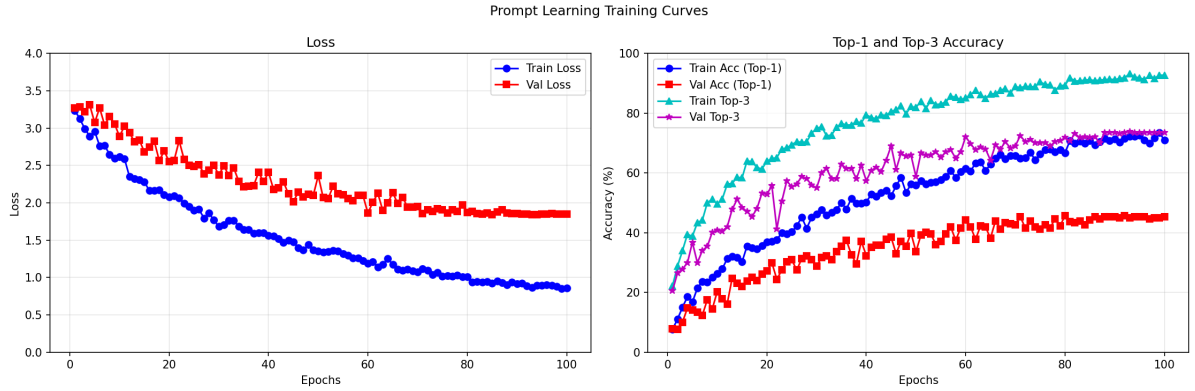

Figure 19: Stage 2 CoCoOp training curve for Pacemaker dataset

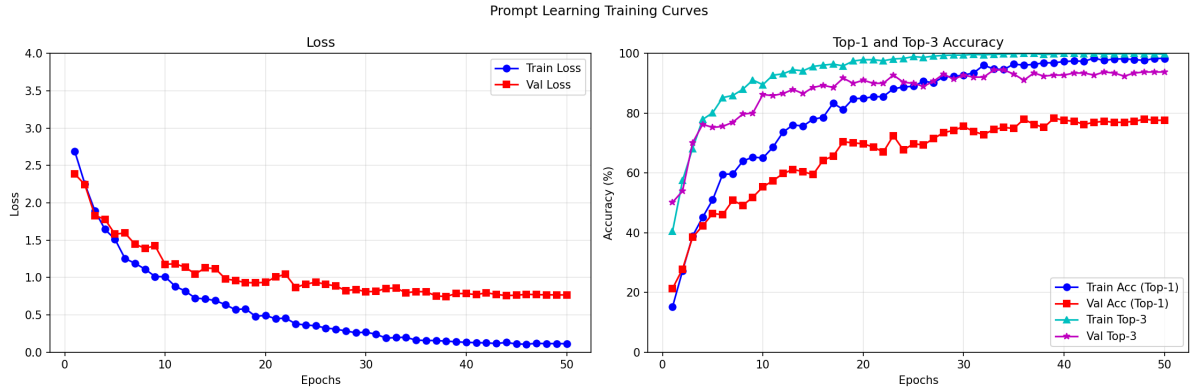

Figure 20: Stage 2 MaPLe training curve for Pacemaker dataset

7 Analysis and Discussion

7.1 Comparison of Pretraining Strategies

The results demonstrate varying effectiveness of different pretraining approaches across the two datasets. On the **Orthonet** dataset, the best performance was achieved by **DINOv2 fine-tuning**, with **93.3% accuracy** and an AUC of 0.998. On the **Pacemaker** dataset, the **fine-tuned CLIP model** performed best, reaching **93.8% accuracy**, **97.3% Top-3 accuracy**, and an AUC of 0.999. Notably, all the three models had similar top 3 accuracies for Pacemakers. In both cases, full fine-tuning of pretrained ViTs substantially outperformed alternative strategies, confirming the value of adapting large pretrained models to medical imaging domains. There is also the problem of manually finding the best text prompt template for the classes.

7.2 Zero-Shot vs Fine-Tuning Performance

Comparing zero-shot classification with fine-tuned models reveals a large performance gap between generalization and task-specific adaptation. The **zero-shot CLIP model** achieved **23.9% accuracy** on Orthonet and only **2.2% accuracy (6.2% Top-3)** on Pacemakers, whereas the best fine-tuned models reached **93.3% and 93.8% accuracy** with **97.3% Top-3** on Pacemakers. This shows that while CLIP's general-purpose vision–language representations have some ability to transfer, they are insufficient for fine-grained implant classification without adaptation.

7.3 Prompt Learning Effectiveness

Among the prompt learning methods, **MaPLe consistently outperformed CoOp and CoCoOp** across both datasets, achieving **73.9% accuracy on Orthonet** and **62.7% (85.3% Top-3) on Pacemakers [non-hierarchical]**. CoOp and CoCoOp trailed behind with accuracies in the **38–43% range** and Top-3 values around **65–70%**. This suggests that distributing prompts across both encoders and multiple transformer layers is particularly beneficial for adapting CLIP to specialized medical imaging tasks. However, even **MaPLe remained notably weaker than fine-tuning**, highlighting the limits of prompt-only adaptation. We can also see that CoCoOp, being more flexible, performs better than CoOp as well. Another notable difference is that **MaPLe has faster convergence and shows more stable learning.**

7.4 Hierarchical Learning Benefits

The hierarchical prompt learning experiments on the Pacemaker dataset reveal different trends across methods. For **CoOp**, hierarchical initialization resulted in only a slight improvement over flat prompts, rising from **37.8% (Top-3: 69.8%)** to **42.2% (Top-3: 64.9%)**, with marginal changes in F1-score and AUC. In contrast, **CoCoOp clearly benefited from hierarchical learning**, improving from **43.1% (Top-3: 70.2%)** to **48.0% (Top-3: 76.0%)**, showing that hierarchical initialization helped capture finer distinctions among classes. **The most significant gains were observed with MaPLe**, which improved from **62.7% (Top-3: 85.3%)** to **75.1% (Top-3: 92.0%)**, accompanied by notable increases in F1-score and AUC. These results suggest that while CoOp shows limited benefit, both CoCoOp and especially MaPLe can leverage hierarchical prompts effectively, with MaPLe demonstrating the strongest synergy between hierarchical initialization and multi-level prompt distribution. **The convergence was faster** when starting from better initialized prompts. The training **starts out with lower loss**, and the final accuracy is better.

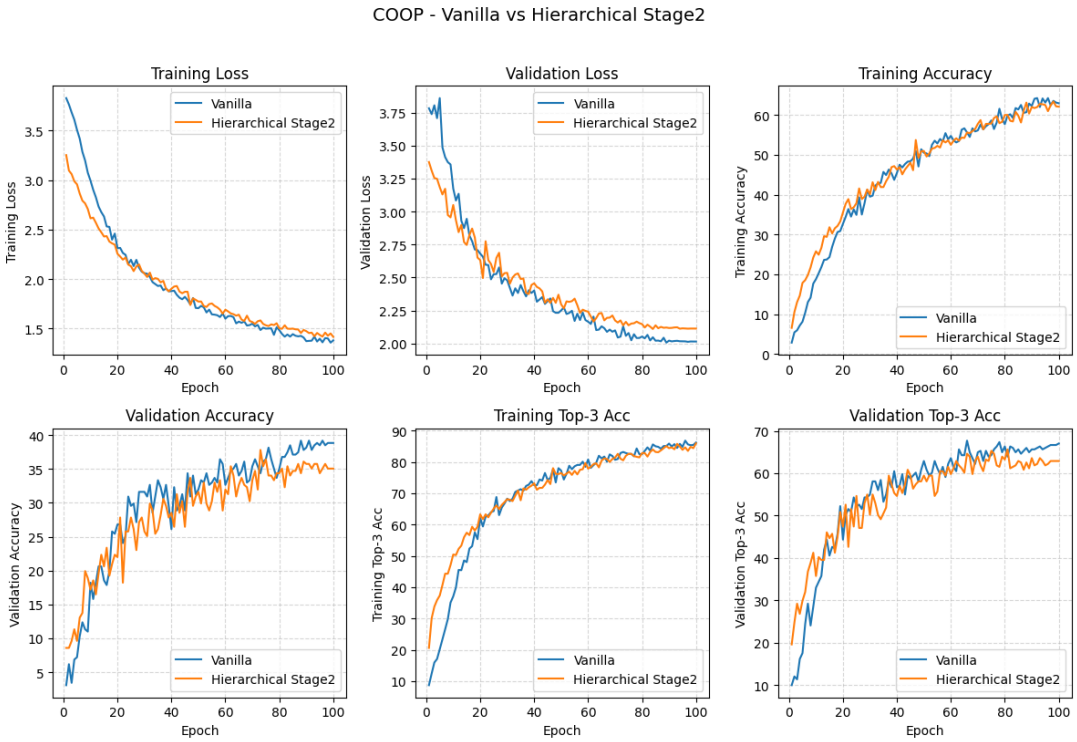

Figure 21: Comparison of Vanilla vs Hierarchical Prompt Learning for CoOp

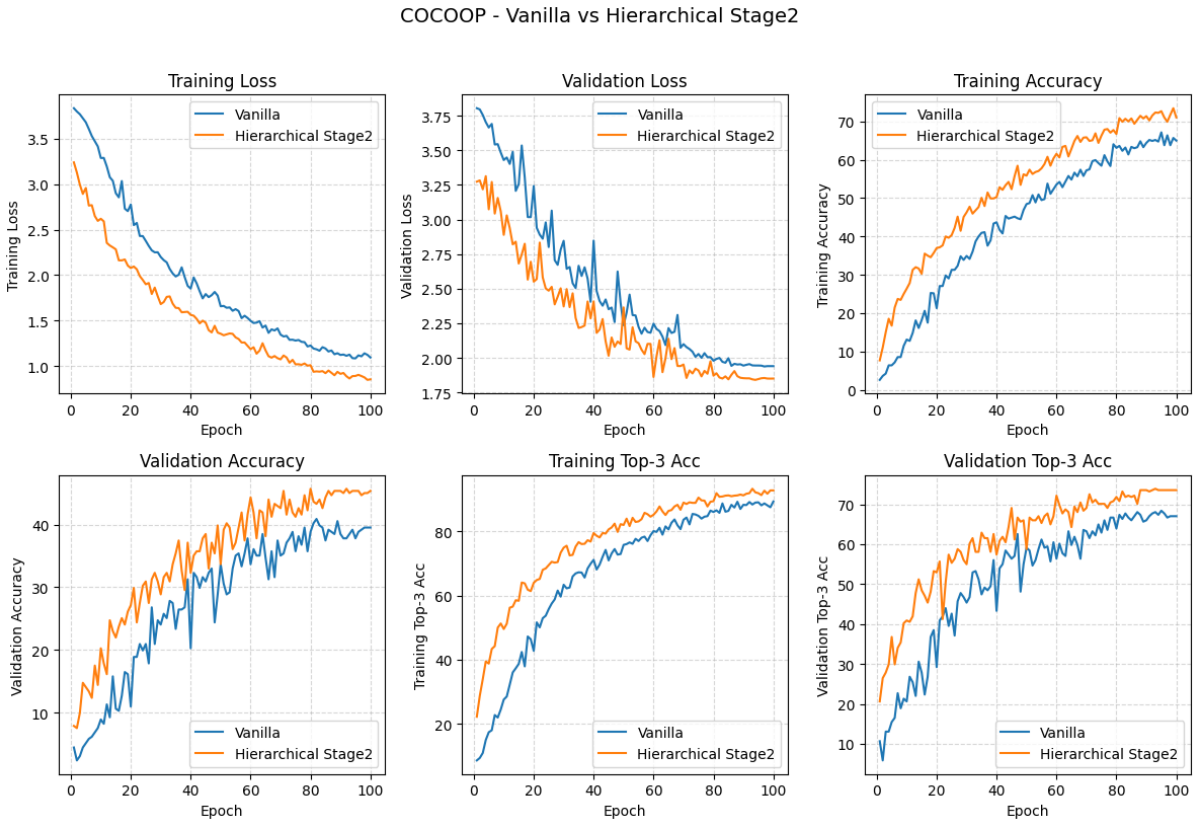

Figure 22: Comparison of Vanilla vs Hierarchical Prompt Learning for CoCoOp

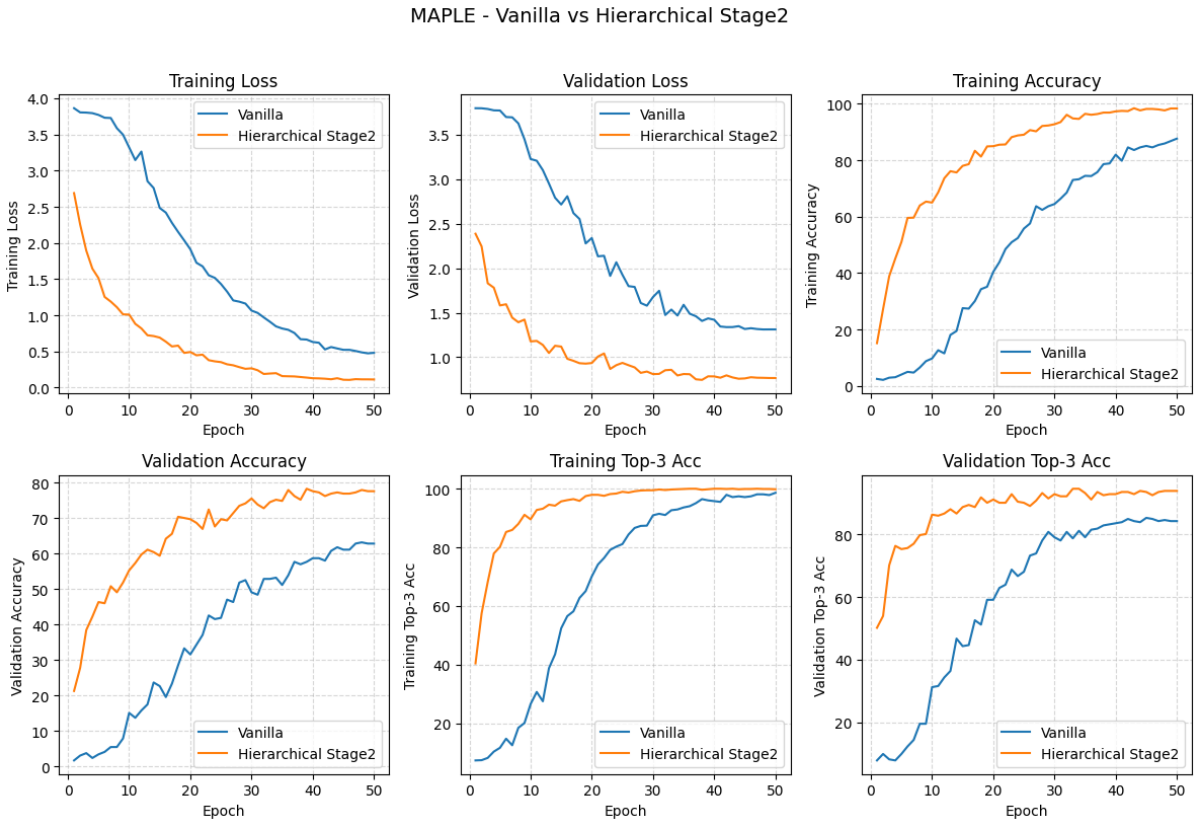

Figure 23: Comparison of Vanilla vs Hierarchical Prompt Learning for CoCoOp

7.5 Medical Imaging Specific Observations

Several domain-specific observations emerge from this study:

- The effectiveness of pretraining strategies varies: **DINOv2 excelled on orthopedic imaging (Orthonet)**, while **CLIP fine-tuning excelled on cardiac imaging (Pacemakers)**.
- **Fine-grained classification** (Pacemakers, 45 classes) i**s significantly harder** than coarse-grained classification (Orthonet, fewer classes), making hierarchical approaches attractive.
- CLIP's zero-shot capabilities remain limited in medical imaging but show potential if paired with **better prompt engineering** and possibly **domain-specific pretraining**.

8 Implementation Details

8.1 Models and other details

- The ViTClassifier is adapted for claasification. For ImageNet and DINOv2, the classifier head is modified, but for CLIP, the classifier head is added.
- The finetuning is done with Adam optimizer.
- The zero shot backbone is CLIP `ViT-L/14@336px`.
- Prompt Learning backbone is CLIP `ViT-B/32`.
- Prompt Learning methods use SGD with cosine annealing.

8.2 Code Structure

8.2.1 Core implementational files

We tried to keep the codebase as modular as possible. There are separate dataset folders. Apart from this, we have

- Three directories for the three types of classifiers. The prompt learning folder contains the relevant portion of the MaPLe repo for custom CLIP implementation.
- Utility folder for ViT+classifier classes and dataloaders.
- There is a `Runner` class for each classification method, which encapsulates all the details, and can be run with minimal setup.
- The `results` folder contains all experimental results including training curves and classification reports.
- The dictionary in `config.py` contains the tuned hyperparamters. `metrics_summary.csv` contains the final test metrics for all experiments.
- `README.md` contains the command to run the experiments.

8.2.2 Running scripts

The implementation follows the required submission format with separate training and testing scripts for each subtask:

- `train_subtask_1_1.py` / `test_subtask_1_1.py`: ViT fine-tuning on Orthonet
- `test_subtask_1_2.py`: CLIP zero-shot on Orthonet
- `train_subtask_1_3.py` / `test_subtask_1_3.py`: Prompt learning on Orthonet
- `train_subtask_2_1.py` / `test_subtask_2_1.py`: ViT fine-tuning on Pacemaker
- `test_subtask_2_2.py`: CLIP zero-shot on Pacemaker
- `train_subtask_2_3.py` / `test_subtask_2_3.py`: Hierarchical prompt learning

8.3 Model Weights

The trained model weights can be found here.

8.4 External Libraries

We utilized the following publicly available repositories:

- Hugging Face Transformers for ViT implementations
- OpenAI CLIP for pretrained models, `timm` for DINOv2.
- Custom implementations of CoOp, CoCoOp, and MaPLe based on original papers, from their original repositories.
- Kaggle CLI for downloading the datasets.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Muhammad Uzair Khattak, Imran Ahmed, Salman H Khan, Fahad Shahbaz Khan, and Bernard Ghanem. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[3] Maxime Oquab, Timothée Darcet, Theo Moutakanni, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[5] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pre-training for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[6] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022.