COL828 A1
Transformer models for implant classification

Yash Bansal (2022CS51133)

Contents

1 Introduction

This assignment is related to implementing and evaluating different vision techniques on the **OrthoNet** and **Pacemakers** datasets. Both of these datasets contain X-ray images of implants in different body organs.

For our study, we performed the following experiments:

1.1 OrthoNet

For the OrthoNet dataset, we first fine-tune a pretrained ViT model using ImageNet-21k, CLIP, and DINOv2 weights. We conduct both (i) full model fine-tuning and (ii) training only the classification head, and report metrics such as Top-1 accuracy, F1-score, and AUC-ROC.

Next, we explore the zero-shot detection capability of the CLIP model. Since handcrafted prompts are not very expressive, we also learn continuous prompt embeddings using the following methods:

- CoOp (Context Optimization)
- CoCoOp (Conditional Context Optimization)
- MaPLe (Multimodal Prompt Learning)

For each of these methods, we experiment with both a single shared prompt for all classes and class-specific prompts for each class, and evaluate the performance across different metrics.

1.2 Pacemakers

For the Pacemakers dataset, we again fine-tune pretrained ViT models using ImageNet-21k, CLIP, and DINOv2 weights. As before, we try both full model fine-tuning and training only the classification head.

Since this dataset contains 45 classes, with each implant belonging to one of five manufacturers, we also report **Top-1 accuracy** and **Manufacturer-level accuracy** to determine whether misclassifications happen within the same manufacturer or across different ones. We also explore zero-shot CLIP on this dataset.

For prompt learning, we design both flat and hierarchical prompts:

- **Flat prompts**: learned in both class-specific and non-class-specific manners.
- **Hierarchical prompts**: first learn manufacturer-level prompts, then initialize fine-grained class-level prompts from them. Both class-specific and shared variants are evaluated.

1.3 References and resources

For CoOp, CoCoOp, and MaPLe, we used the official implementations provided by the authors:

`https://github.com/muzairkhattak/multimodal-prompt-learning/tree/main`

The trained models are saved at the following link: Trained Models Link

2 OrthoNet

2.1 Fine-Tuning Pretrained ViT

In this part, we fine-tune pretrained ViT models with different initialization weights: **ImageNet-21k**, **CLIP**, and **DINOv2**. We experiment with two strategies: full model fine-tuning and training only the classification head.

2.1.1 Full Fine-Tuning

In this experiment, the entire ViT backbone along with the classification head is fine-tuned for each of the three pretrained weights.

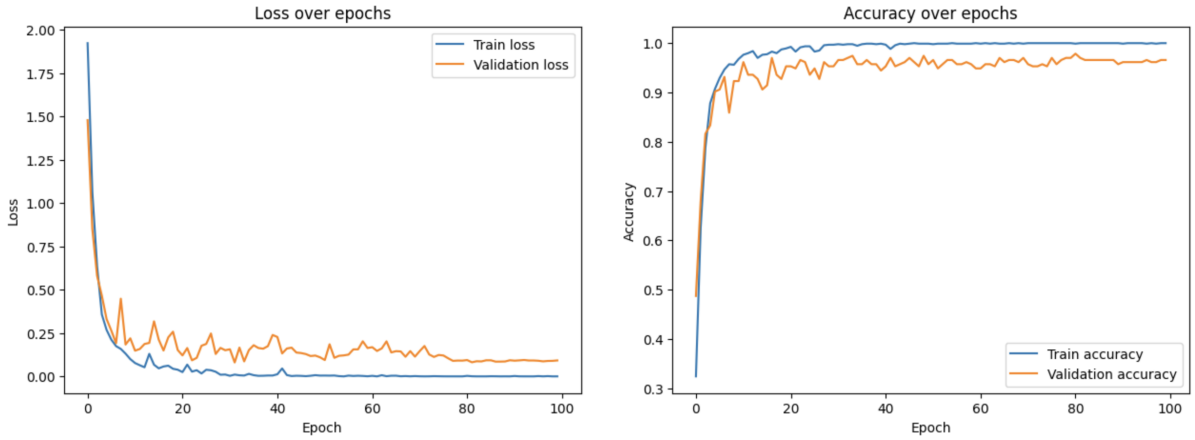

Figure 1: Training loss (left) and accuracy (right) curves for full fine-tuning with ImageNet-21k weights.

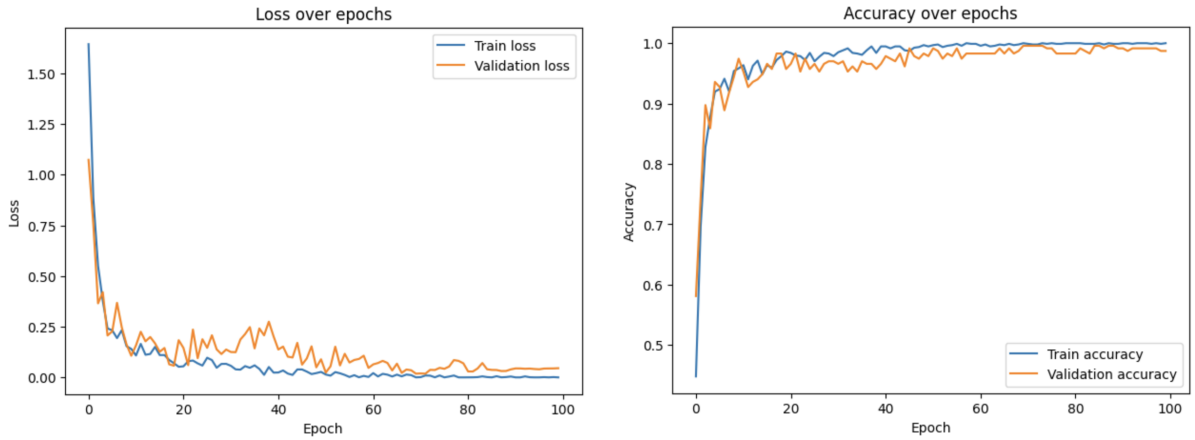

Figure 2: Training loss (left) and accuracy (right) curves for full fine-tuning with CLIP weights.

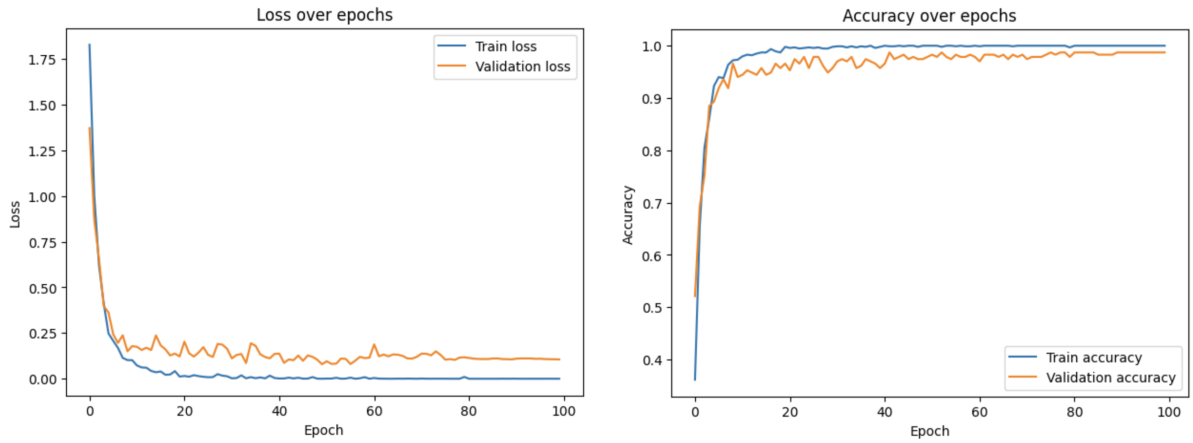

Figure 3: Training loss (left) and accuracy (right) curves for full fine-tuning with DINOv2 weights.

Final test results (full fine-tuning):

- ImageNet-21k: Accuracy = 0.911, F1 = 0.904, AUC-ROC = 0.997
- CLIP: Accuracy = 0.961, F1 = 0.961, AUC-ROC = 0.999
- DINOv2: Accuracy = 0.911, F1 = 0.910, AUC-ROC = 0.995

Analysis: From the results, CLIP-pretrained weights clearly outperform both ImageNet-21k and DINOv2, achieving the highest accuracy (96.1%) and F1-score (0.961). Both ImageNet-21k and DINOv2 achieve similar performance (around 91.1% accuracy), indicating that while these features transfer reasonably well, they are less suited for X-ray implants compared to CLIP's multimodal pretraining. The consistently high AUC-ROC (> 0.99) across all models suggests strong separability between classes when fine-tuning is applied.

2.1.2 Training Only the Classifier Layer

In this experiment, only the final classification head is trained, while the pretrained ViT backbone is frozen.

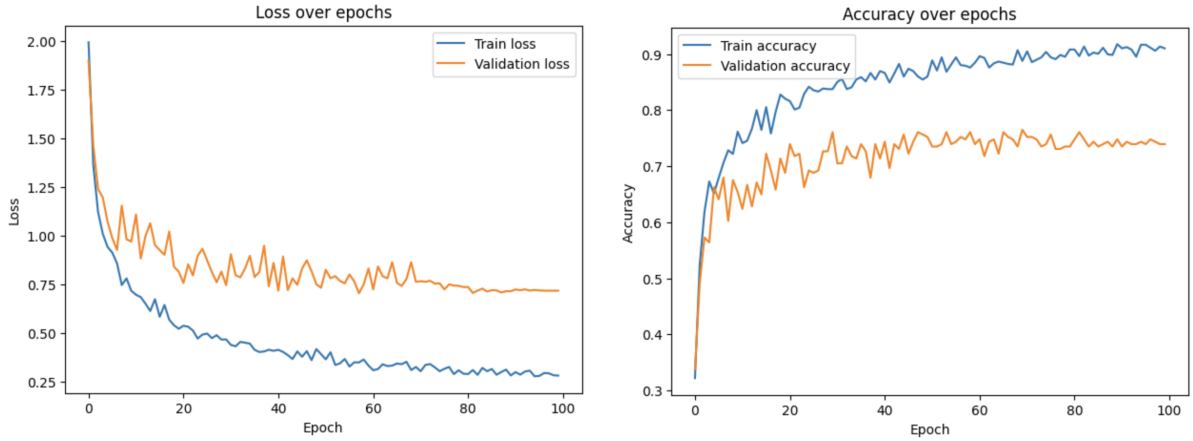

Figure 4: Training loss (left) and accuracy (right) curves for classifier-only training with ImageNet-21k weights.

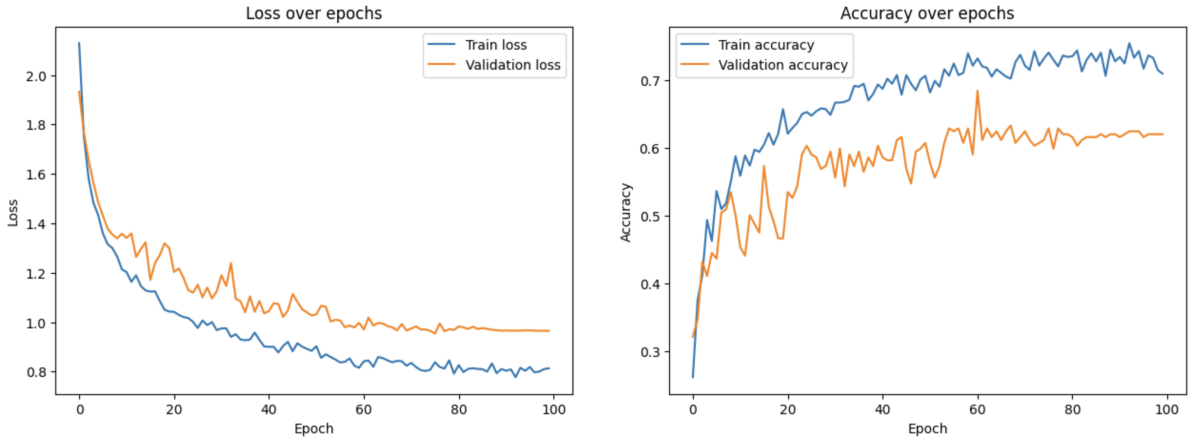

Figure 5: Training loss (left) and accuracy (right) curves for classifier-only training with CLIP weights.

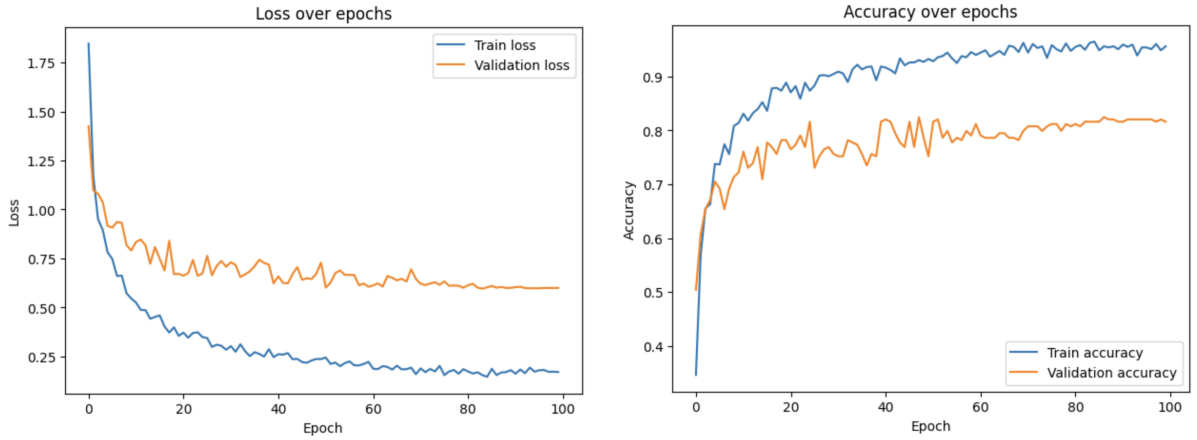

Figure 6: Training loss (left) and accuracy (right) curves for classifier-only training with DINOv2 weights.

Final test results (classifier-only training):

- ImageNet-21k: Accuracy = 0.650, F1 = 0.621, AUC-ROC = 0.946
- CLIP: Accuracy = 0.561, F1 = 0.512, AUC-ROC = 0.940
- DINOv2: Accuracy = 0.644, F1 = 0.606, AUC-ROC = 0.942

Analysis: Performance drops significantly when only the classifier head is trained. This suggests that the pretrained embeddings are not well aligned with the X-ray implant domain, and adapting the backbone via full fine-tuning is necessary. Among the frozen-backbone settings, ImageNet-21k and DINOv2 perform slightly better (around 65% accuracy) compared to CLIP (56%). This indicates that while CLIP excels under full fine-tuning (likely due to its rich multimodal pretraining), its frozen representations are less effective for this specialized medical task.

2.2 Zero-shot Classification with CLIP

In this part, we evaluate the zero-shot classification capabilities of CLIP on the OrthoNet X-ray implant dataset. We use the following handcrafted prompt, which gave the best results among different prompt choices:

"A grayscale X-ray showing orthopedic implant of {class} "

For each image, we pass it through the CLIP image encoder and compute its embedding. In parallel, we generate 12 class-specific text embeddings by filling in the class name in the prompt above. The predicted class for each image is then obtained by selecting the class with the highest cosine similarity between the image embedding and text embeddings.

The results are summarized below:

Train set: Accuracy: 0.180, F1: 0.113, AUC-ROC: 0.739

Test set: Accuracy: 0.139, F1: 0.074, AUC-ROC: 0.721

As we can see, the zero-shot performance of CLIP on this task is poor. The main reasons for this are:

- The domain gap between CLIP's pretraining data (natural images and captions) and our dataset (medical grayscale X-rays of implants).
- Handcrafted prompts are not expressive enough to capture the subtle visual and semantic differences between implant classes.
- Some implant categories are visually very similar in X-rays, making it difficult for CLIP to distinguish them without task-specific adaptation.

These limitations highlight that while CLIP demonstrates strong zero-shot performance in natural image domains, it struggles in highly specialized domains such as medical imaging. To overcome this, we next explore prompt tuning approaches such as **CoOp**, **CoCoOp**, and **MaPLe**, which allow learning continuous prompt embeddings tailored to our dataset. These methods aim to bridge the domain gap and improve classification performance.

2.3 Context Optimization (CoOp)

In this part, we replace handcrafted prompt tokens with learnable continuous prompt embeddings using the CoOp framework. The model backbone remains completely frozen, and only the prompt embeddings are trained.

We conducted several experiments with both class-specific and non-class-specific prompts, using different initialization strategies:

- **Non-class specific prompts:**
  - Random initialization
  - Text-initialized
- **Class-specific prompts:**
  - Random initialization
  - Text-initialized

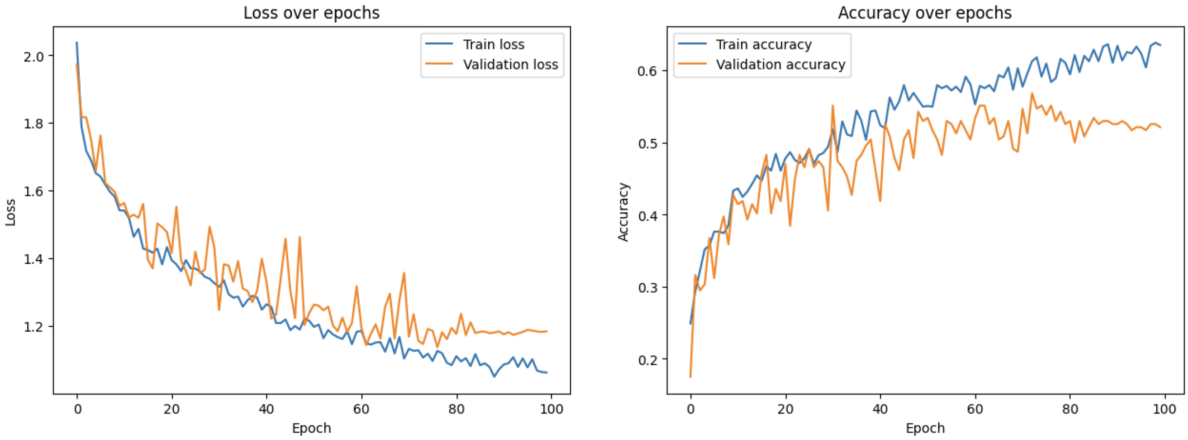

Figure 7: Training loss (left) and accuracy (right) curves for CoOp with text-initialized class-specific prompts (best performing setting).

The final test results (accuracy, F1 score, and AUC-ROC) for each setting are as follows:

Setting	**Accuracy**	**F1**	**AUC-ROC**
Non-class specific (Random)	0.400	0.348	0.899
Non-class specific (Text init.)	0.433	0.377	0.906
Class-specific (Random)	0.483	0.464	0.932
Class-specific (Text init.)	0.556	0.514	0.929

Table 1: Performance of CoOp under different initialization strategies.

Analysis:

- **Text-initialized prompts consistently outperform randomly initialized ones.** This is because textual initialization provides the model with a semantically meaningful starting point that is aligned with the target domain, while random prompts start from arbitrary embeddings that require more optimization.
- **Class-specific prompts perform better than non-class specific prompts.** This is expected because each implant category in OrthoNet has distinct visual cues (e.g., shape, placement, density in X-rays). A single shared prompt struggles to capture these fine-grained differences, whereas class-specific prompts can adapt to the characteristics of each category.
- The best-performing configuration (**class-specific, text-initialized**) achieves an accuracy of 55.6%, which is a significant improvement over zero-shot CLIP results, showing the effectiveness of prompt tuning in reducing the domain gap.

2.4 Conditional Context Optimization (CoCoOp)

In this part, we implement conditional prompt embeddings based on image features, introducing dynamic adaptation of the prompts to the input. Specifically, we learn the prompt context vector (similar to CoOp), but also train a mapping from image features to the prompt vectors using a 2-layer MLP. The image embedding is added to the prompt context vectors to condition the prompts on the input.

For class-specific prompts, **only the prompt context vectors are class-specific**, while the image encoder remains shared across all classes. This design ensures that the visual features extracted

from X-rays are consistent and comparable across classes, while the learned prompts can capture class-specific nuances. If the image encoder were class-specific, the model would need to learn separate visual representations for each class, which is inefficient and could lead to overfitting given limited data per class.

We perform two main experiments:

- Non-class specific embeddings
- Class-specific embeddings

Both are initialized randomly. Text initialization does not provide much improvement over random initialization because CoCoOp already dynamically adapts prompts based on image features, which provides a strong inductive bias and reduces the dependence on the initial prompt.

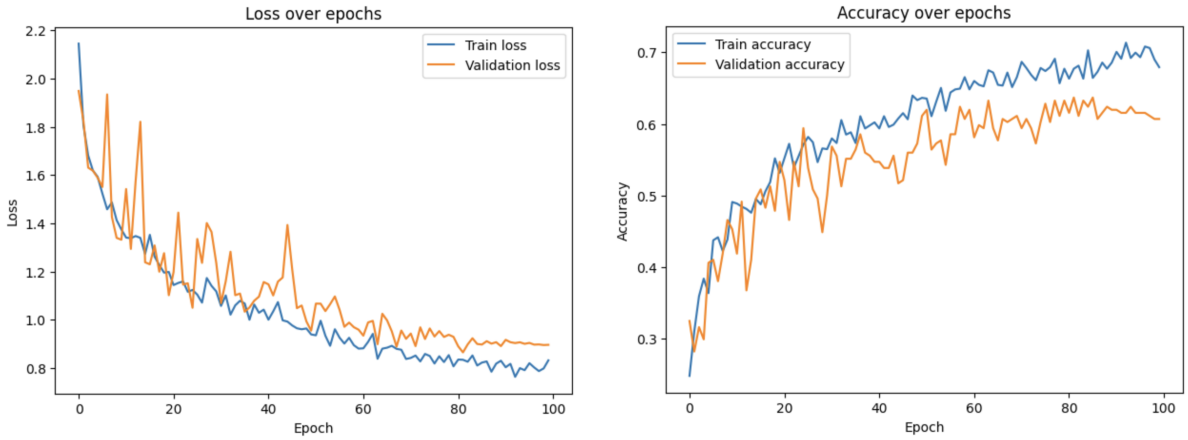

Figure 8: Training loss (left) and accuracy (right) curves for CoCoOp (best performing setting: non-class specific, random init).

The final test results (accuracy, F1, and AUC-ROC) are:

Setting	**Accuracy**	**F1**	**AUC-ROC**
Non-class specific (Random)	0.550	0.539	0.925
Class-specific (Random)	0.528	0.508	0.920

Table 2: Performance of CoCoOp under different initialization strategies.

Analysis:

- Non-class specific CoCoOp performs slightly better than class-specific in this setting. This may be because the dynamic conditioning on image features allows the model to adapt to each input, reducing the need for separate class-specific prompts.
- Compared to CoOp, CoCoOp shows improved performance over non-class specific prompts (accuracy 55.0% vs 43.3%). This demonstrates the effectiveness of conditioning prompts on image features: the model can adjust shared prompts dynamically rather than relying solely on a fixed embedding.
- Text initialization provides minimal gain in CoCoOp because the image-conditioned adaptation dominates the prompt representation, making the starting point less critical.

- For class-specific prompts, only the prompt context vectors are class-specific, while the image encoder is shared. This ensures consistent visual representations across classes and prevents overfitting on limited data.
- Overall, CoCoOp narrows the gap between non-class specific and class-specific prompts and provides more robust performance on unseen inputs, highlighting the benefits of conditional prompt learning over fixed prompt embeddings.

2.5 Multimodal Prompt Learning (MaPLe)

In this part, learnable prompts are distributed along both the text and image encoders, as well as multiple transformer layers. This allows the model to improve embeddings in both modalities, potentially leading to better performance compared to single-modality prompt tuning.

For class-specific prompts, **only the initial prompt context vector is class-specific**, while all other encoder weights remain shared across classes. This design ensures consistent visual and textual representations while allowing class-specific adaptation through the prompt context.

We perform the following main experiments:

- Non-class specific prompts only in the first transformer layer
- Non-class specific prompts in all 12 transformer layers
- Class-specific prompts in all 12 transformer layers

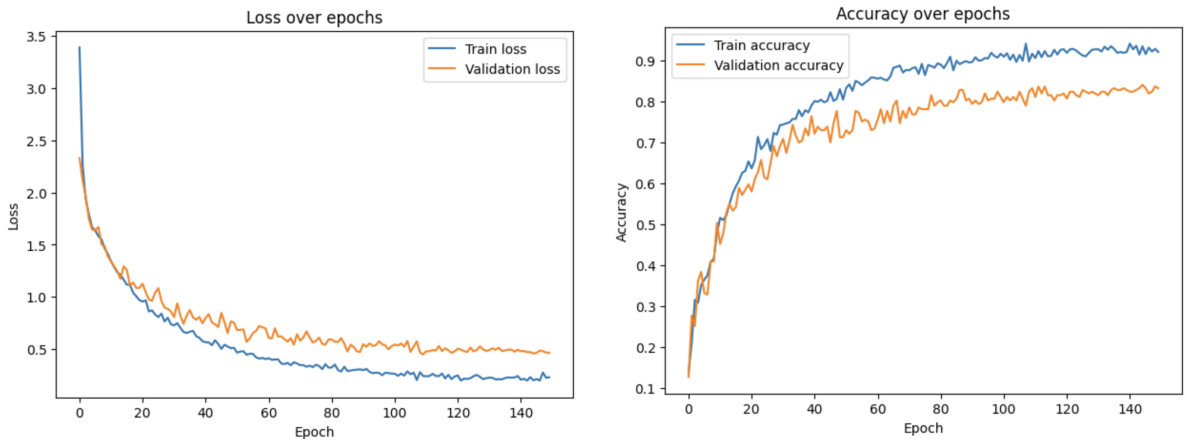

Figure 9: Training loss (left) and accuracy (right) curves for MaPLe (best performing setting: class-specific prompts in all transformer layers).

The final test results (accuracy and F1) are summarized below:

Setting	Accuracy	F1
Non-class specific (first layer)	0.656	0.602
Non-class specific (all layers)	0.767	0.746
Class-specific (all layers)	0.783	0.778

Table 3: Performance of MaPLe under different prompt configurations.

Analysis:

- MaPLe outperforms both CoOp and CoCoOp by a significant margin, demonstrating the benefit of distributing learnable prompts across both text and image encoders.
- Allowing prompts in multiple transformer layers helps the model adapt at different levels of abstraction, improving both visual and textual feature alignment.
- Class-specific prompts in all layers perform the best, achieving 78.3% accuracy and 0.778 F1-score. This is because the prompts can capture subtle class-specific variations at multiple levels in the transformer.
- Non-class specific prompts in all layers also perform well (76.7% accuracy), suggesting that deep-layer adaptation allows the shared prompts to sufficiently differentiate classes without being explicitly class-specific.
- Non-class specific prompts only in the first layer perform worst among MaPLe settings (65.6% accuracy), indicating that single-layer adaptation is insufficient to capture complex multimodal interactions between image and text embeddings.
- Overall, MaPLe demonstrates the importance of learning prompts across both modalities and multiple layers, and highlights how class-specific adaptation can further boost performance in fine-grained medical classification tasks like OrthoNet.

3 Pacemakers

In this section, we apply different fine-tuning techniques on the Pacemakers dataset. Since the dataset is hierarchical in nature, with 45 implant classes grouped into 5 manufacturers, we evaluate models using **Top-1 Accuracy**, **Top-3 Accuracy**, **F1-score**, **AUC-ROC**, and **Manufacturer Accuracy**.

Top-3 Accuracy provides insight into whether misclassifications occur within the same manufacturer group. For example, if an implant of manufacturer A is misclassified as another implant from the same manufacturer, Top-3 Accuracy may still be high, reflecting that the model is capturing manufacturer-level structure even if fine-grained class distinctions are imperfect.

3.1 Fine-Tuning Pretrained ViT

We fine-tune pretrained ViT models initialized with ImageNet-21k, CLIP, and DINOv2 weights. Two strategies are evaluated:

- **Full model fine-tuning:** All layers of the ViT are updated.
- **Training only the classification head:** Only the final classification layer is updated, while the rest of the model is frozen.

3.1.1 Full Fine-Tuning

Here, all layers of the ViT model are fine-tuned. The following figures show the loss and accuracy curves for each weight:

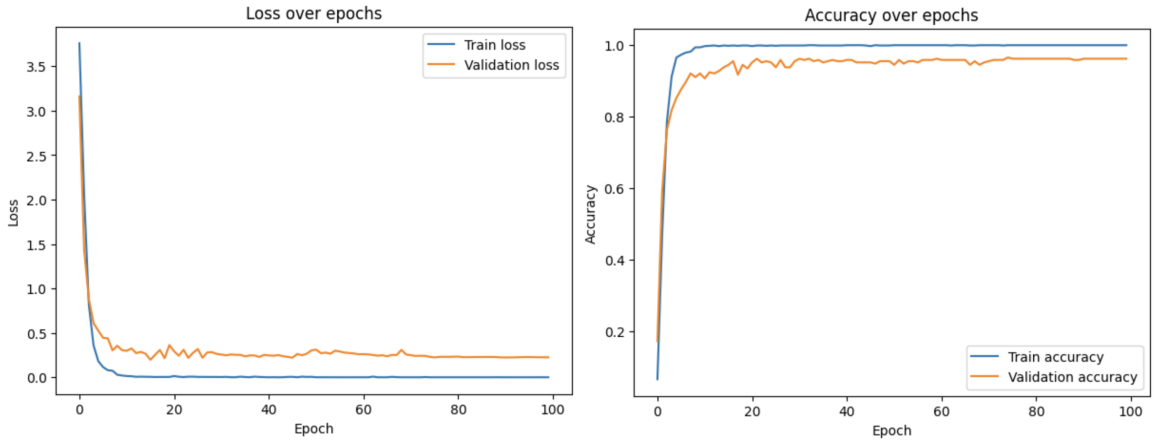

Figure 10: Training loss (left) and accuracy (right) curves for full fine-tuning of ViT with ImageNet-21k weights on the Pacemakers dataset.

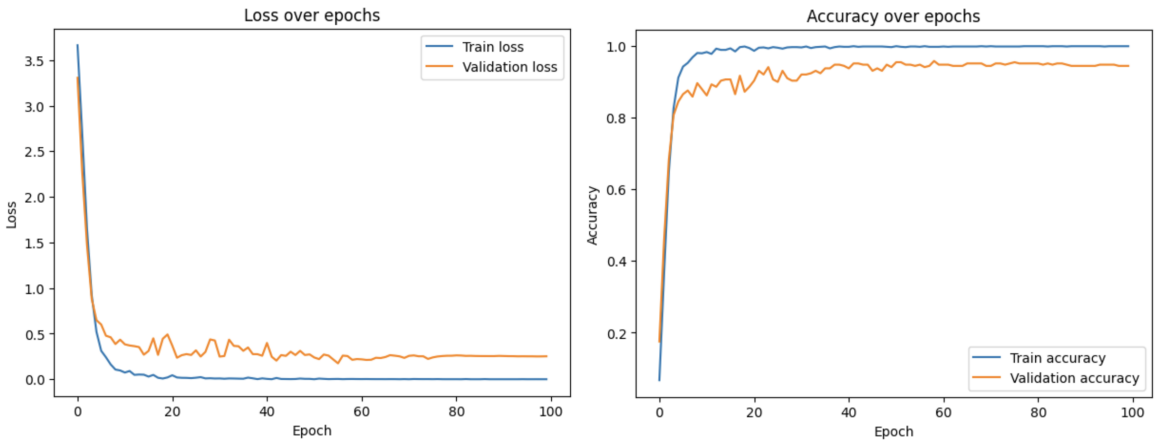

Figure 11: Training loss (left) and accuracy (right) curves for full fine-tuning of ViT with CLIP weights on the Pacemakers dataset.

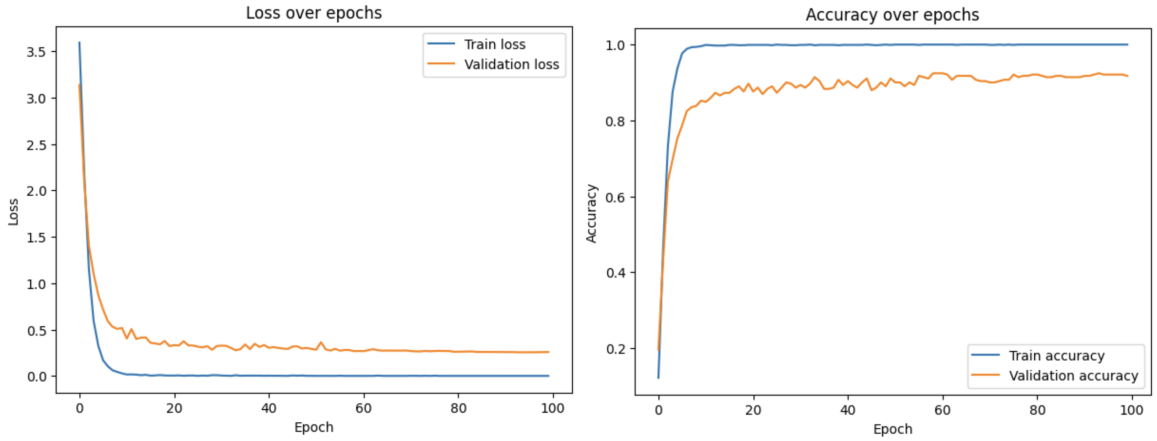

Figure 12: Training loss (left) and accuracy (right) curves for full fine-tuning of ViT with DINOv2 weights on the Pacemakers dataset.

Final test results (Accuracy, Top-3 Accuracy, F1, AUC-ROC, Manufacturer Accuracy):

Weights	**Acc**	**Top-3 Acc**	**F1**	**AUC-ROC**	**Manufacturer Acc**
ImageNet-21k	0.938	0.978	0.938	0.999	0.991
CLIP	0.956	0.982	0.955	0.999	0.987
DINOv2	0.916	0.973	0.916	0.997	0.973

Table 4: Full fine-tuning performance of ViT models on the Pacemakers dataset.

Analysis:

- All models achieve high performance, demonstrating that fine-tuning allows ViT to adapt well to the X-ray domain of pacemakers.
- CLIP-initialized weights perform the best, likely due to their robust multimodal pretraining which helps in distinguishing visually similar implants.

- Manufacturer accuracy is consistently higher than Top-1 Accuracy, indicating that most misclassifications occur within the same manufacturer group. This aligns with the hierarchical nature of the dataset.
- DINOv2 performs slightly worse than ImageNet-21k and CLIP, possibly because its self-supervised pretraining is less aligned with fine-grained implant distinctions.

3.1.2 Training Only the Classification Head

When only the final classification layer is trained and the backbone is frozen, performance decreases, especially for CLIP:

Weights	Acc	Top-3 Acc	F1	AUC-ROC	Manufacturer Acc
ImageNet-21k	0.862	0.964	0.860	0.997	0.947
CLIP	0.564	0.778	0.547	0.962	0.760
DINOv2	0.853	0.951	0.852	0.995	0.929

Table 5: Performance when training only the classification head on the Pacemakers dataset.

Analysis:

- Training only the classification head is insufficient, particularly for CLIP, whose frozen embeddings are not fully aligned with the fine-grained pacemaker classes.
- ImageNet-21k and DINOv2 maintain moderate performance because their pretrained features are more aligned with general X-ray patterns.
- Manufacturer accuracy remains higher than Top-1 Accuracy, indicating that even when misclassifications occur, they often remain within the correct manufacturer group.
- Overall, full fine-tuning is essential for optimal performance on this hierarchical dataset due to the subtle visual differences between classes.

3.2 Zero-Shot Classification with CLIP

In this experiment, we evaluate the zero-shot capabilities of the CLIP model on the Pacemakers dataset using the prompt:

"Cardiac implant X-ray by {class} implant".

The metrics obtained are as follows:

Dataset	Accuracy	Top-3 Acc	F1	AUC-ROC	Manufacturer Acc
Train	0.015	0.057	0.005	0.483	0.325
Test	0.027	0.067	0.004	0.493	0.280

Table 6: Zero-shot performance of CLIP on Pacemakers dataset using the prompt *"Cardiac implant X-ray by {class} implant".*

Analysis:

- The zero-shot performance is extremely poor, with Top-1 Accuracy below 3% and F1-score near zero, indicating that the pretrained CLIP embeddings are not aligned with the Pacemakers domain.

- The dataset contains fine-grained distinctions between 45 implant classes across 5 manufacturers, including subtle differences in design and series. These are not captured by CLIP's general pretraining on broad web images.
- Manufacturer-level accuracy is higher than overall accuracy (0.325 on train, 0.28 on test), showing that some misclassifications still occur within the correct manufacturer group.
- Overall, this demonstrates that zero-shot CLIP cannot handle such fine-grained medical classification tasks, emphasizing the need for domain-specific fine-tuning or prompt learning approaches such as CoOp, CoCoOp, or MaPLe.

3.3 Context Optimization (CoOp)

In this part, instead of using handcrafted prompt embeddings, we learn continuous prompt embeddings. Since the Pacemakers dataset is hierarchical in nature, we design two main strategies:

- **Hierarchical CoOp:** We first learn manufacturer-level prompt embeddings in a class-specific manner (using the loss from manufacturer prediction). Then, we split these 5 manufacturer prompts into 45 class prompts, where each class inherits the initialization of its manufacturer. These are fine-tuned further in a class-specific manner for the 45-way classification task. This ensures that the learned prompts encode coarse manufacturer-level distinctions while adapting to finer model-level differences.
- **Flat CoOp:** We directly learn prompts for all 45 classes either in a non-class-specific or class-specific setting, without leveraging the hierarchical structure.
- **Non-class-specific variant:** As a control experiment, we also trained a single manufacturer-level prompt in a non-class-specific manner, and then fine-tuned it for all 45 classes again in a non-class-specific setting. This is compared with learning flat non-class-specific prompts.

The final test results are summarized below:

Method	Accuracy	Top-3 Acc	F1	AUC-ROC	Manuf. Acc
Flat CoOp (non-class specific)	0.360	0.649	0.328	0.937	0.587
Hier. CoOp (non-class specific)	0.320	0.547	0.281	0.911	0.587
Flat CoOp (class specific)	**0.560**	0.804	0.545	0.967	0.742
Hier. CoOp (stage 1: manuf. level)	0.649	0.947	0.630	0.885	—
Hier. CoOp (class specific, stage 2)	0.551	**0.818**	**0.555**	**0.970**	0.733

Table 7: Test results for CoOp on the Pacemakers dataset. Stage 1 = manufacturer-level pretraining, Stage 2 = fine-grained class-level training.

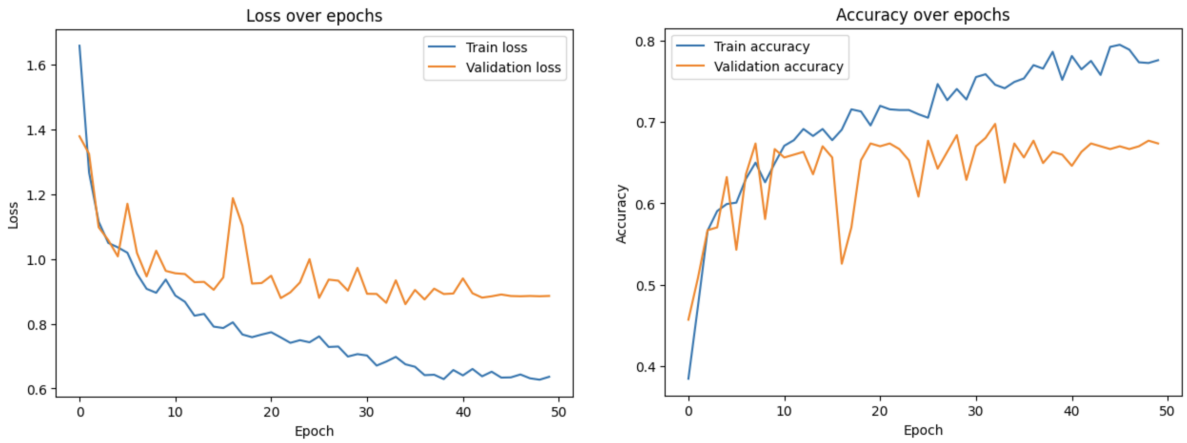

Figure 13: Stage 1: Training loss (left) and accuracy (right) for manufacturer-level CoOp prompts.

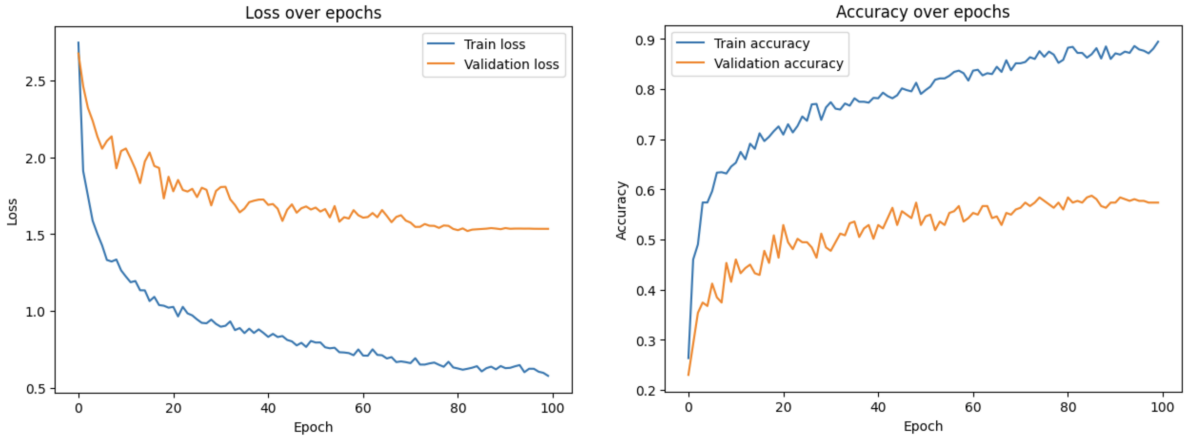

Figure 14: Stage 2: Training loss (left) and accuracy (right) for class-level fine-grained CoOp prompts.

Analysis:

- Flat CoOp with class-specific prompts achieves the highest accuracy (56%), showing that class-specific prompts help capture fine distinctions between implant models.
- Hierarchical CoOp performs slightly worse in terms of accuracy (55%), but achieves higher Top-3 Accuracy (81.8%) and F1-score (0.555), and comparable manufacturer accuracy.
- Importantly, hierarchical prompts show better **generalization**. For flat prompts, test results drop by 3–4% compared to validation, whereas hierarchical prompts achieve test performance comparable to validation. This indicates that the hierarchical initialization acts as a strong inductive bias, preserving coarse manufacturer-level information and preventing overfitting to specific classes.
- Non-class-specific settings underperform significantly (32–36% accuracy), confirming that class-specific prompt learning is crucial in this fine-grained classification task.

Overall, CoOp demonstrates that while flat class-specific prompts give the best raw accuracy, hierarchical prompts provide improved robustness and generalization. This motivates extending hierarchical strategies with more advanced prompt learning approaches such as CoCoOp and MaPLe.

3.4 Conditional Context Optimization (CoCoOp)

We perform similar experiments as in CoOp, but now using CoCoOp, which conditions the learned prompts on the input image features. As in CoOp, we explore both *flat* and *hierarchical* designs, and compare class-specific and non-class-specific variants. For the class-specific variants, the learned prompts are applied only to the prompt context tokens, and not to the image meta encoder.
The final test results are summarized in Table 8.

Method	Accuracy	Top-3 Acc	F1	AUC-ROC	Manuf. Acc
Flat CoCoOp (non-class specific)	0.511	0.756	0.502	0.953	0.720
Hier. CoCoOp (non-class specific)	0.471	0.747	0.459	0.955	0.738
Flat CoCoOp (class specific)	0.569	0.787	0.556	0.968	0.764
Hier. CoCoOp (stage 1: manuf. level)	0.702	0.947	0.671	0.924	—
Hier. CoCoOp (class specific, stage 2)	0.613	0.800	0.611	0.964	0.791

Table 8: Test results for CoCoOp on the Pacemakers dataset. Stage 1 = manufacturer-level pretraining, Stage 2 = class-level fine-grained training.

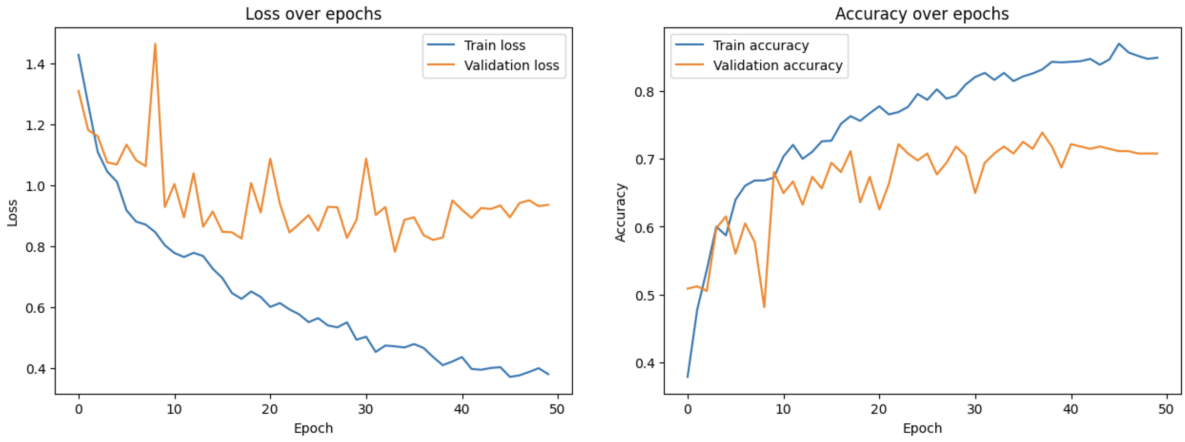

Figure 15: Stage 1: Training loss (left) and accuracy (right) for manufacturer-level CoCoOp prompts.

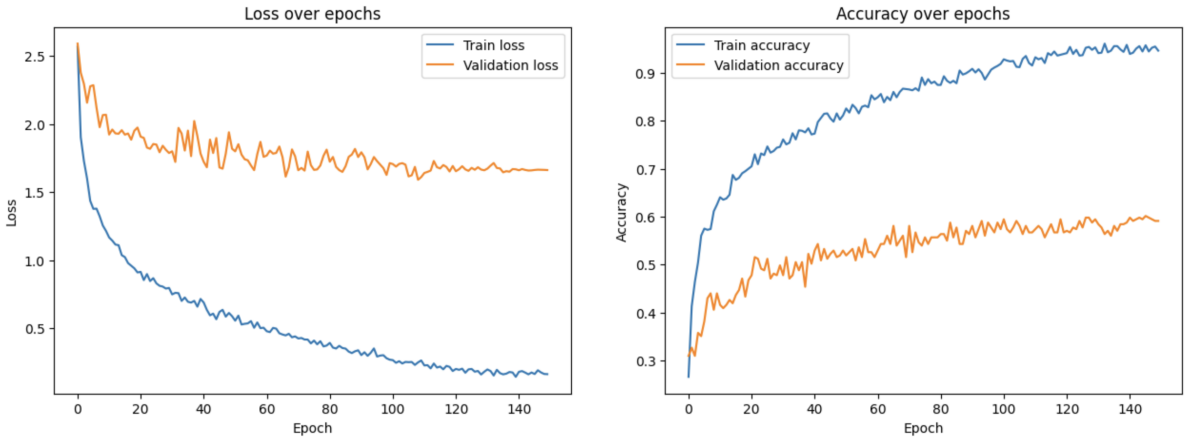

Figure 16: Stage 2: Training loss (left) and accuracy (right) for class-level fine-grained CoCoOp prompts.

Analysis:

- As in CoOp, non-class-specific settings perform relatively poorly (47–51% accuracy), showing that conditioning alone cannot replace class-specific prompt tuning.
- Flat CoCoOp with class-specific prompts achieves 56.9% accuracy, a modest gain over flat CoOp (56.0%), confirming the benefit of conditional prompt adaptation.
- Hierarchical CoCoOp with class-specific prompts achieves the best overall performance with 61.3% accuracy, 80.0% top-3 accuracy, and an F1-score of 0.611. This demonstrates that hierarchical initialization synergizes well with conditional adaptation.
- Manufacturer-level results in Stage 1 reach 70.2% accuracy, which shows that the hierarchical setup captures coarse-level distinctions very effectively. Fine-tuning in Stage 2 adapts this knowledge for fine-grained class classification.
- Importantly, as with CoOp, hierarchical prompts provide better **generalization**. While flat prompts show a noticeable drop from validation to test, hierarchical CoCoOp achieves test results nearly on par with validation, suggesting a stronger inductive bias against overfitting.
- Compared to CoOp, the gains are more pronounced in the hierarchical class-specific setting, highlighting the strength of image-conditioned adaptation when combined with hierarchical structure.

Overall, CoCoOp confirms the trend seen with CoOp: hierarchical prompts are not always the best in raw accuracy, but they provide significantly improved generalization. When combined with conditional context optimization, however, hierarchical prompts outperform flat prompts, suggesting that CoCoOp particularly benefits from hierarchical structure.

3.5 Multimodal Prompt Learning (MaPLe)

We next evaluate MaPLe on the Pacemakers dataset. As in CoOp and CoCoOp, we explore both flat and hierarchical variants, as well as class-specific and non-class-specific prompts. Recall that in MaPLe, learnable prompts are distributed across both text and image encoders, as well as multiple transformer layers. This multimodal distribution helps enrich the embeddings of both modalities. As in previous methods, for the class-specific variants, only the initial prompt context is class-specific, while the image encoder prompts remain non-class-specific.
The final test results are summarized in Table 9.

Method	**Accuracy**	**Top-3 Acc**	**F1**	**Manuf. Acc**
Flat MaPLe (non-class specific)	0.751	0.929	0.742	0.880
Hier. MaPLe (non-class specific)	**0.804**	0.947	**0.804**	**0.924**
Flat MaPLe (class specific)	0.671	0.893	0.669	0.831
Hier. MaPLe (class specific, stage 1: manuf.)	0.880	0.987	0.864	—
Hier. MaPLe (class specific, stage 2)	**0.804**	**0.951**	**0.804**	0.920

Table 9: Test results for MaPLe on the Pacemakers dataset. Stage 1 = manufacturer-level pretraining, Stage 2 = class-level fine-grained training.

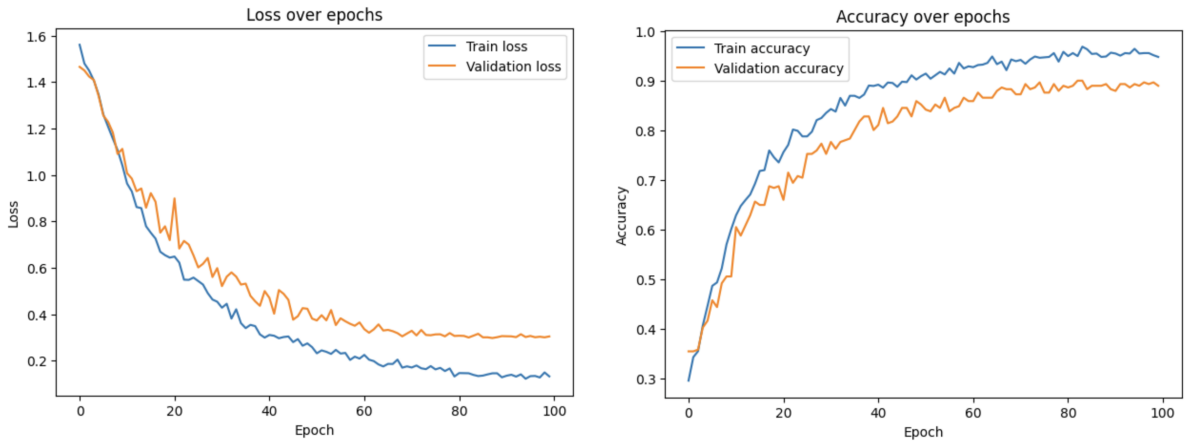

Figure 17: Stage 1: Training loss (left) and accuracy (right) for manufacturer-level MaPLe prompts.

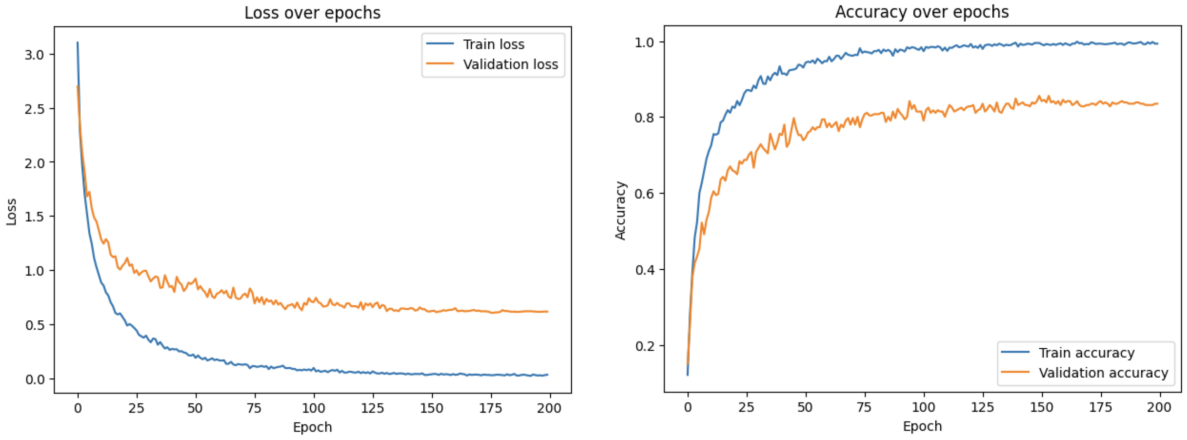

Figure 18: Stage 2: Training loss (left) and accuracy (right) for class-level fine-grained MaPLe prompts.

Analysis:

- MaPLe achieves significantly higher performance than both CoOp and CoCoOp. For example, hierarchical class-specific MaPLe achieves 80.4% accuracy compared to 61.3% for hierarchical class-specific CoCoOp and 55.1% for hierarchical class-specific CoOp.
- Non-class-specific MaPLe already performs strongly (75.1–80.4% accuracy), showing that distributing prompts across both encoders and multiple layers captures rich multimodal alignment even without class-specific tuning.
- The hierarchical setup provides consistent improvements over flat prompts. In the class-specific setting, accuracy rises from 67.1% (flat) to 80.4% (hierarchical). This demonstrates that hierarchical initialization provides strong coarse-to-fine inductive bias.
- Stage 1 (manufacturer-level) training achieves 88.0% accuracy, confirming that the model learns coarse manufacturer distinctions very effectively before fine-grained adaptation.
- Importantly, MaPLe shows both high accuracy and robustness: test results are well aligned with validation performance, showing limited overfitting. This indicates that multimodal prompt learning helps regularize training by enforcing consistency across modalities.

- **Why class-specific prompts perform better:** Class-specific prompts provide each implant class with its own learnable prompt tokens, allowing the model to capture subtle differences between visually similar devices (e.g., pacemakers from the same manufacturer but different series). Non-class-specific prompts share a common context across all classes, which helps generalization but reduces the model's ability to distinguish fine-grained details. Thus, class-specific prompts better adapt to intra-manufacturer variations, improving fine-grained classification metrics like Top-1 accuracy and F1-score.
- Compared to CoOp and CoCoOp, MaPLe's advantage lies in allowing mutual reinforcement of text and image embeddings across layers. By enriching both encoders, MaPLe avoids the imbalance of earlier methods, where adaptation was concentrated only in the text encoder.

Overall, MaPLe demonstrates the strongest results across all experiments on Pacemakers. Its hierarchical class-specific variant not only outperforms CoOp and CoCoOp but also shows excellent generalization, making it the most effective prompt learning approach for fine-grained medical implant classification.