Integrating Memory, Reasoning, and Reinforcement Learning into Vision Transformers for Medical Diagnosis

CS3009 - Reinforcement Learning

END SEM PROJECT PRESENTATION

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

CS22B1093	Rohan G
CS22B1095	R Sai Charish
CS22B1096	Pratyek Thumula

Motivation & Background

Why This Project?

- Vision Transformers (ViT) have achieved remarkable success in computer vision.
- However, their pure feature extraction approach lacks long-term reasoning and memory.

Project Goals:

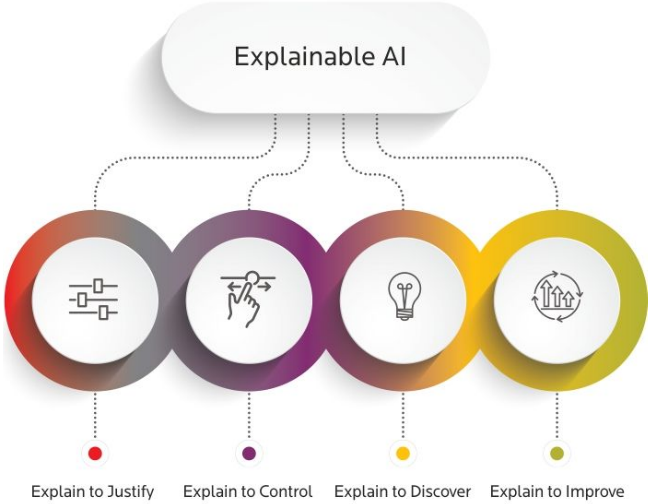
- Enhance ViT with a memory module and a reasoning module.
- Integrate reinforcement learning (using PPO) to optimize decision-making.
- Introduce an explainability component via a chain-of-thought (CoT) mechanism.

Real-World Impact:

- Improved diagnostic accuracy and interpretability in medical imaging (e.g., malaria cell detection).

Importance of Explainability

- Essential for trust in medical AI systems.
- Enables healthcare professionals to understand model decisions.
- Helps identify biases or errors in predictions.
- CoT mechanism provides step-by-step reasoning for diagnoses.
- Supports regulatory compliance and ethical AI use.
- Facilitates patient communication by clarifying AI-driven insights.
- Enhances model debugging and iterative improvement.



Visuals: Icon or illustration of a doctor reviewing AI output, emphasizing transparency.

Reinforcement Learning Overview

Agent:

- In our project, the agent is the integrated model (ViT_RLModel) ·
- images and makes diagnostic decisions.

Environment:

- A simulated environment using the malaria cell image dataset.
- Each image represents a state.

Actions:

- The predicted diagnosis (Parasitized or Uninfected).

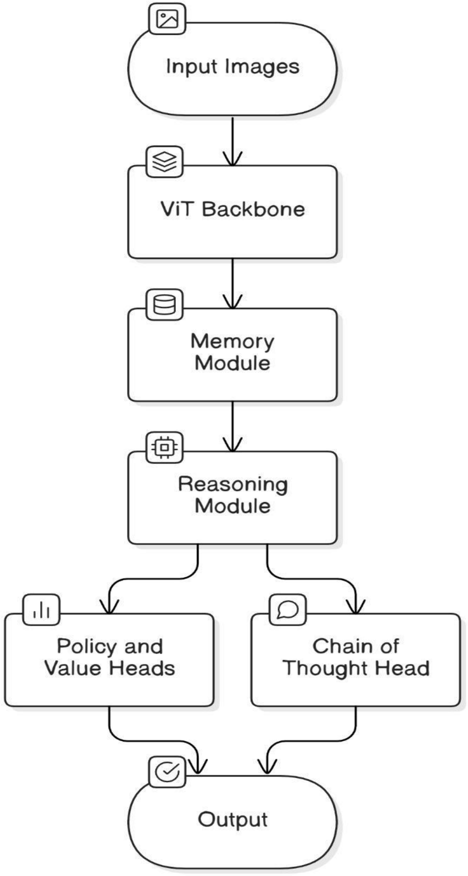
Rewards:

- Binary reward: 1 if the diagnosis is correct, 0 otherwise.

Policy & Value Functions:

- Policy Head outputs logits for action selection.
- Value Head estimates state value for the PPO objective.

Model Architecture Flow



Problem Statement & Objectives

Problem Statement:

- How can we enhance the diagnostic capabilities of ViT models by integrating memory and reasoning, while also optimizing decision-making via reinforcement learning?

Objectives:

1. **Memory Integration:** Capture temporal context from past embeddings.
2. **Reasoning:** Use a Transformer-based reasoning module to infer from combined features.
3. **Reinforcement Learning:** Implement a PPO-based training loop where the agent learns from rewards.
4. **Explainability:** Generate a chain-of-thought output to provide insights into the decision process.

System Architecture & Workflow

Model Architecture Diagram (Visual Aid Recommended):

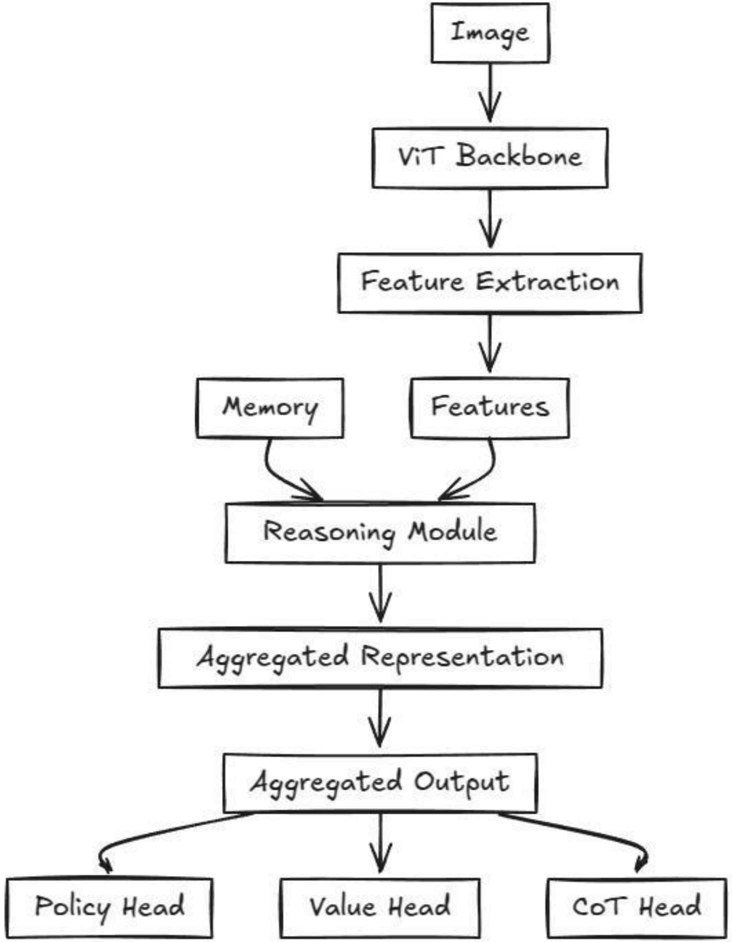
- **ViT Backbone:**
  - Extracts high-level visual features from input images.
- **Memory Module:**
  - Stores and aggregates recent feature embeddings.
- **Reasoning Module:**
  - A Transformer encoder that integrates current features with historical memory.
- **Policy & Value Heads:**
  - Generate classification decisions and estimate the value of the current state.
- **Chain-of-Thought Head:**
  - Produces a vector representing an internal explanation (dummy output for now).

Workflow Summary

1. Image → ViT Backbone → Feature ExtractionFeatures + Memory → Reasoning

2. Module → Aggregated RepresentationAggregated

3. Output → Policy, Value, and CoT Heads



Implementation Details

Dataset & Preprocessing:

- Custom `MalariaDataset` loading cell images.
- Data augmentation and normalization using standard transforms.

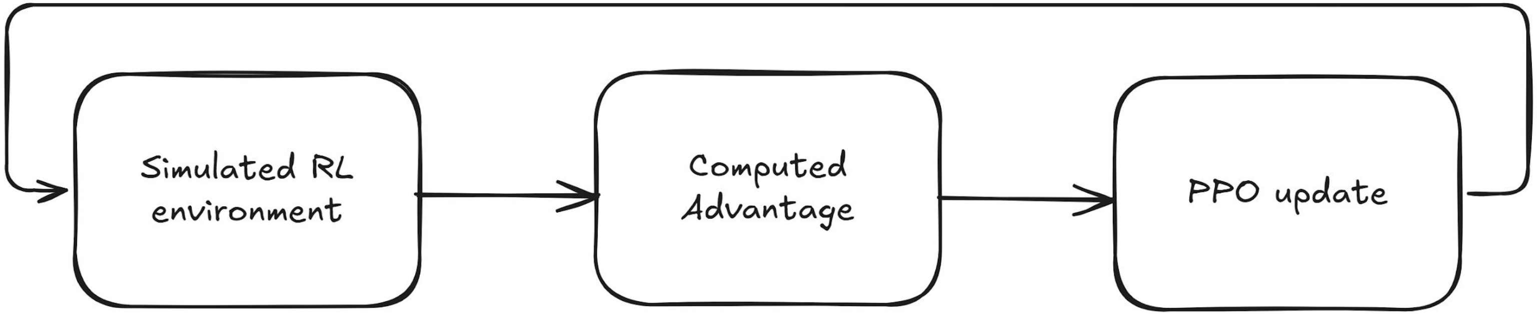
Model Components:

- **ViT_RLModel:**
  - Combines a pretrained ViT (with removed classifier head), memory module, reasoning module, and additional heads.
- **Memory Module:**
  - Maintains a FIFO buffer to store recent embeddings.
- **Reasoning Module:**
  - Uses a Transformer encoder to process the two-token sequence (current features and memory).

Implementation Details

Training Strategy:

- **PPO Training Loop:**
  - Simulated RL environment: each image prediction yields a reward.
  - Advantage computed as the difference between returns and value estimates.
  - PPO update with clipped objective to stabilize training



PPO and Reward Mechanism

PPO Update Overview:

- **Policy Loss:**
  - Uses a clipped objective to ensure stable policy updates.
- **Value Loss:**
  - Mean squared error between the estimated and actual returns.
- **Entropy Bonus:**
  - Encourages exploration.

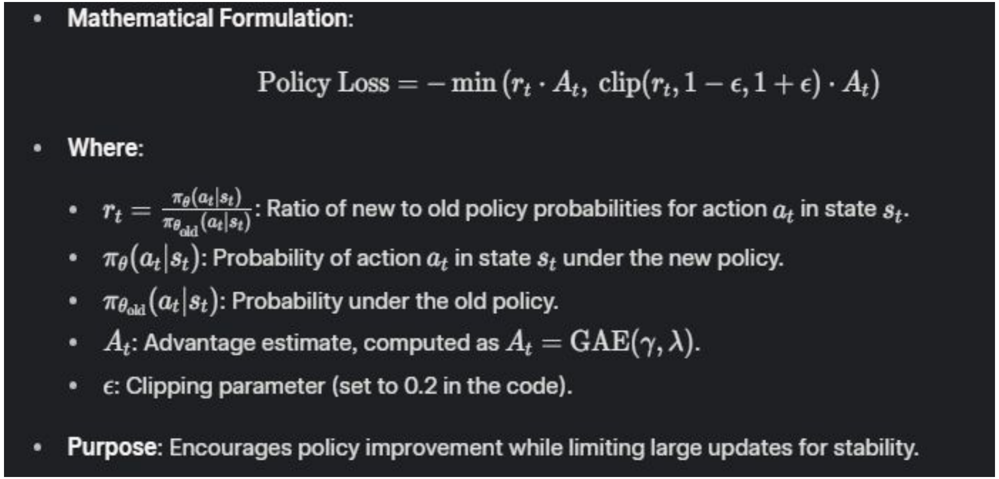
Reward Definition:

- Reward = 1 if the agent's diagnosis matches the true label; otherwise 0.

Simplifications for Current Prototype:

- Immediate rewards without discounting.
- Basic advantage estimation (returns – values).

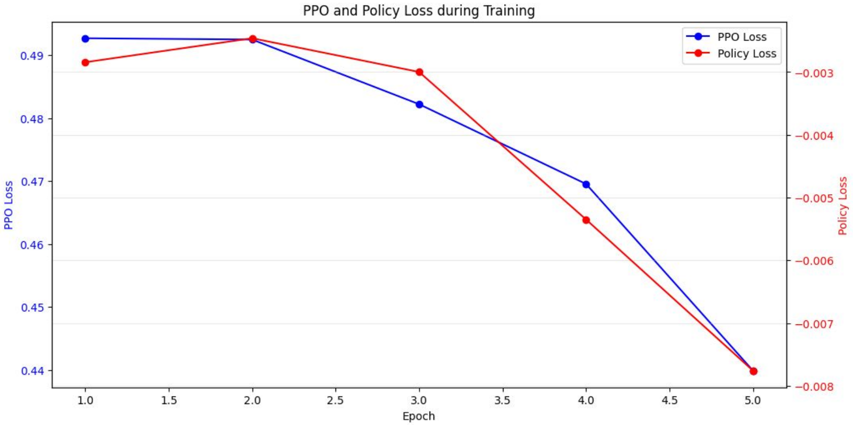
PPO Loss and Policy loss



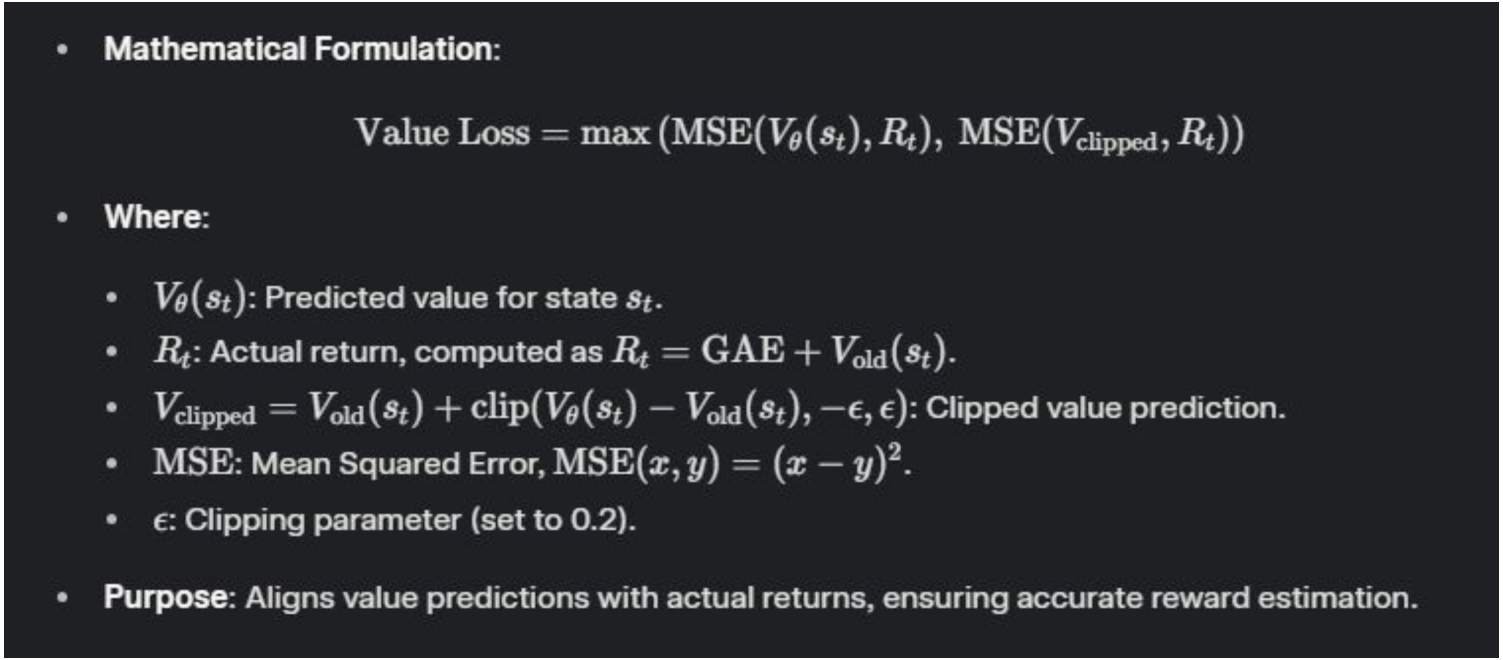
- **Mathematical Formulation:**

$$\text{Policy Loss} = -\min\left(r_t \cdot A_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon) \cdot A_t\right)$$

- **Where:**
  - $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$: Ratio of new to old policy probabilities for action a_t in state s_t.
  - $\pi_\theta(a_t|s_t)$: Probability of action a_t in state s_t under the new policy.
  - $\pi_{\theta_{\text{old}}}(a_t|s_t)$: Probability under the old policy.
  - A_t: Advantage estimate, computed as $A_t = \text{GAE}(\gamma, \lambda)$.
  - ϵ: Clipping parameter (set to 0.2 in the code).
- **Purpose**: Encourages policy improvement while limiting large updates for stability.



PPO Loss and Policy loss

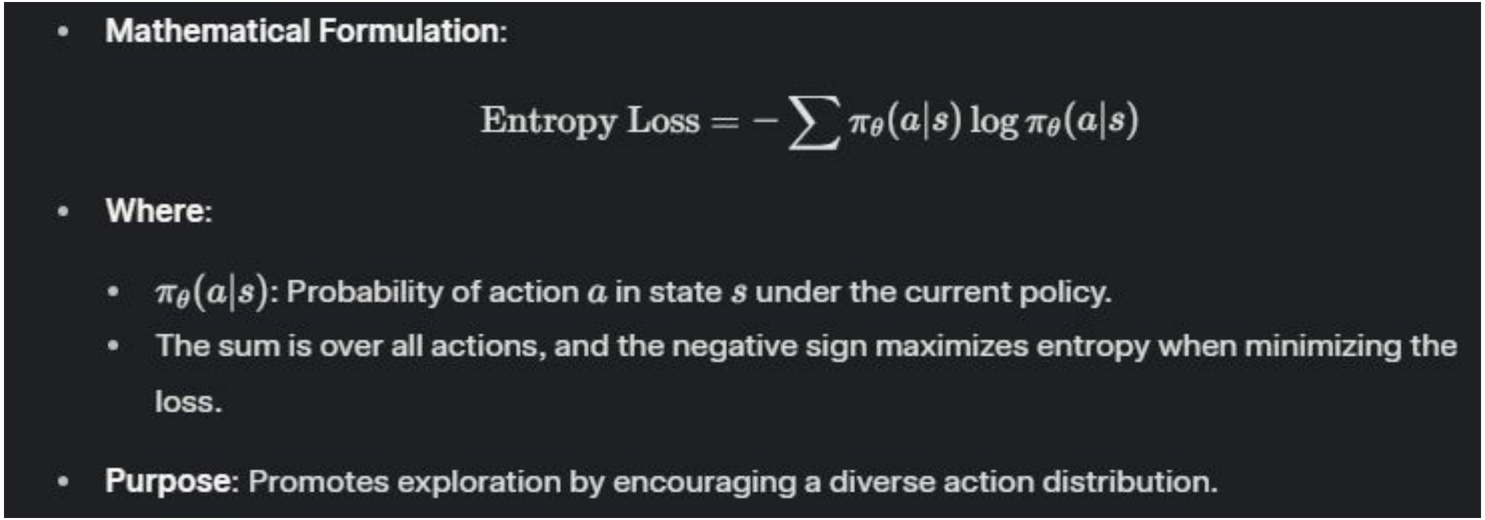


- **Mathematical Formulation:**

$$\text{Value Loss} = \max\left(\text{MSE}(V_\theta(s_t), R_t),\ \text{MSE}(V_{\text{clipped}}, R_t)\right)$$

- **Where:**
  - $V_\theta(s_t)$: Predicted value for state s_t.
  - R_t: Actual return, computed as $R_t = \text{GAE} + V_{\text{old}}(s_t)$.
  - $V_{\text{clipped}} = V_{\text{old}}(s_t) + \text{clip}(V_\theta(s_t) - V_{\text{old}}(s_t), -\epsilon, \epsilon)$: Clipped value prediction.
  - MSE: Mean Squared Error, $\text{MSE}(x, y) = (x - y)^2$.
  - ϵ: Clipping parameter (set to 0.2).
- **Purpose:** Aligns value predictions with actual returns, ensuring accurate reward estimation.

PPO Loss and Policy loss

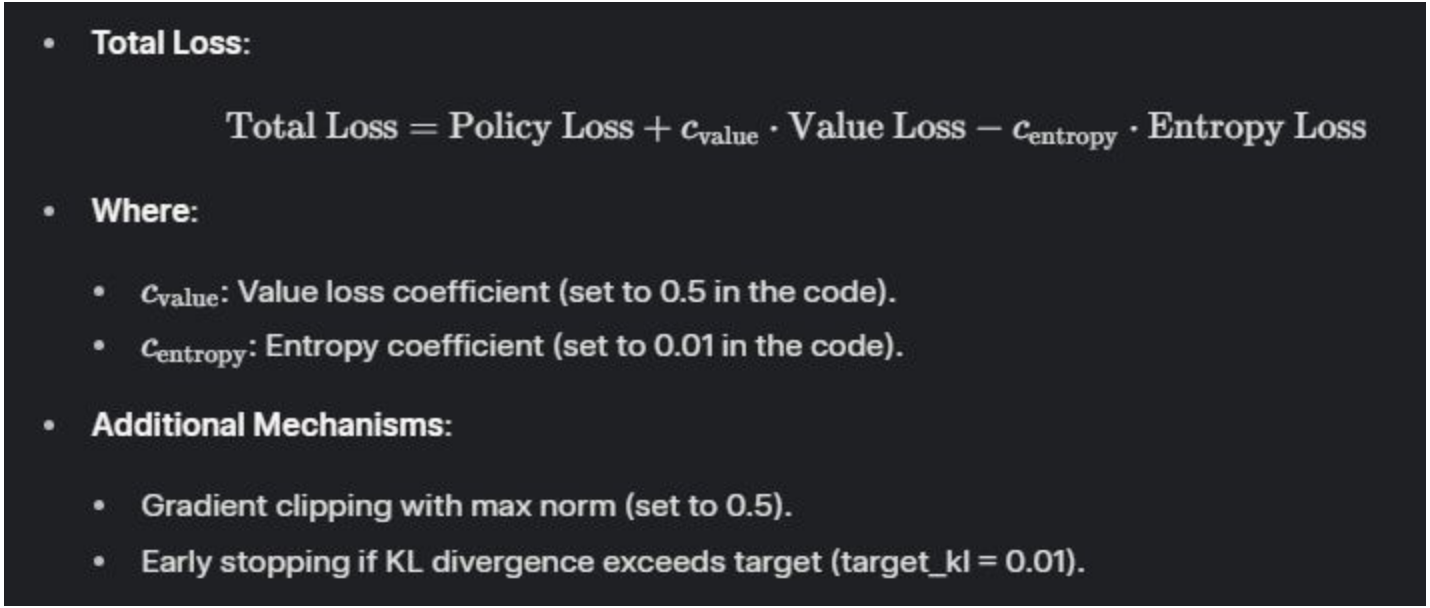


- **Mathematical Formulation:**

$$\text{Entropy Loss} = -\sum \pi_\theta(a|s) \log \pi_\theta(a|s)$$

- **Where:**
  - $\pi_\theta(a|s)$: Probability of action a in state s under the current policy.
  - The sum is over all actions, and the negative sign maximizes entropy when minimizing the loss.
- **Purpose**: Promotes exploration by encouraging a diverse action distribution.

Total Loss



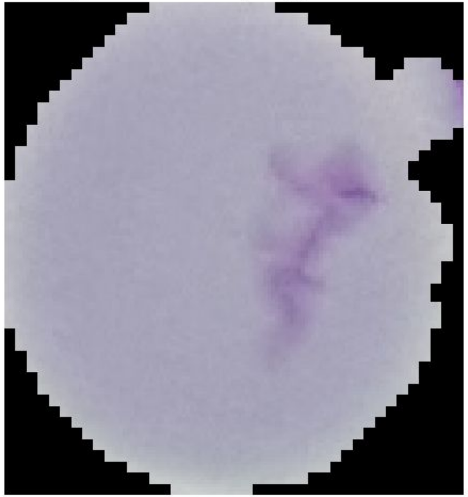
- **Total Loss:**

$$\text{Total Loss} = \text{Policy Loss} + c_{\text{value}} \cdot \text{Value Loss} - c_{\text{entropy}} \cdot \text{Entropy Loss}$$

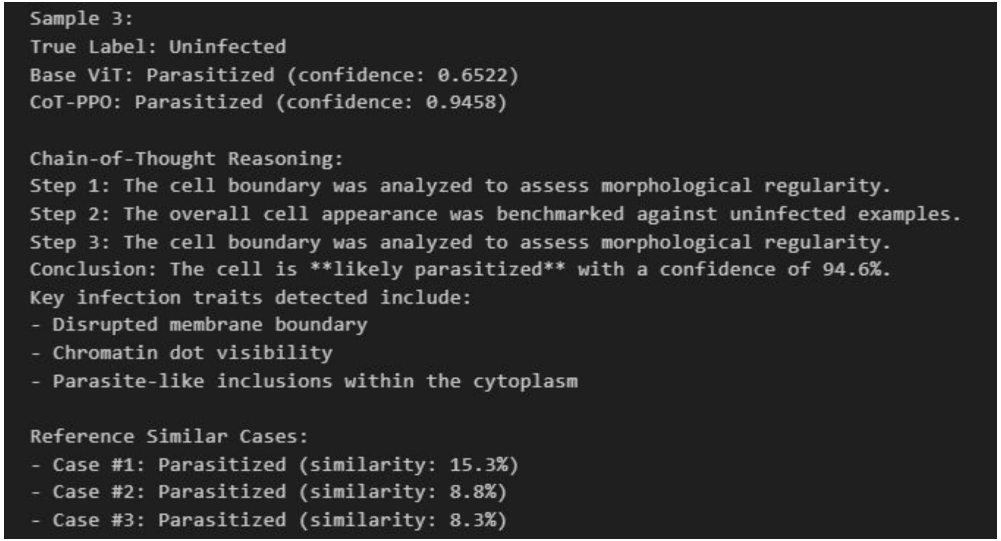
- **Where:**
  - c_{value}: Value loss coefficient (set to 0.5 in the code).
  - c_{entropy}: Entropy coefficient (set to 0.01 in the code).
- **Additional Mechanisms:**
  - Gradient clipping with max norm (set to 0.5).
  - Early stopping if KL divergence exceeds target (target_kl = 0.01).

Training Results

- **Training Overview**:
  - Model trained on the Malaria Cell Image Dataset using PPO and supervised learning.
  - Training history includes loss and accuracy metrics for training and validation sets.
- **Key Observations**:
  - Training and validation loss decreased steadily over epochs, indicating effective learning.
  - Validation accuracy improved, suggesting good generalization to unseen data.
- **Visualizations**:
  - Loss curves (training and validation) saved in ./training_history_visualizations.
  - Accuracy curves (training and validation) demonstrate model performance over time.



True: Uninfected | Base: Parasitized (0.65) | CoT-PPO: Parasitized (0.95)

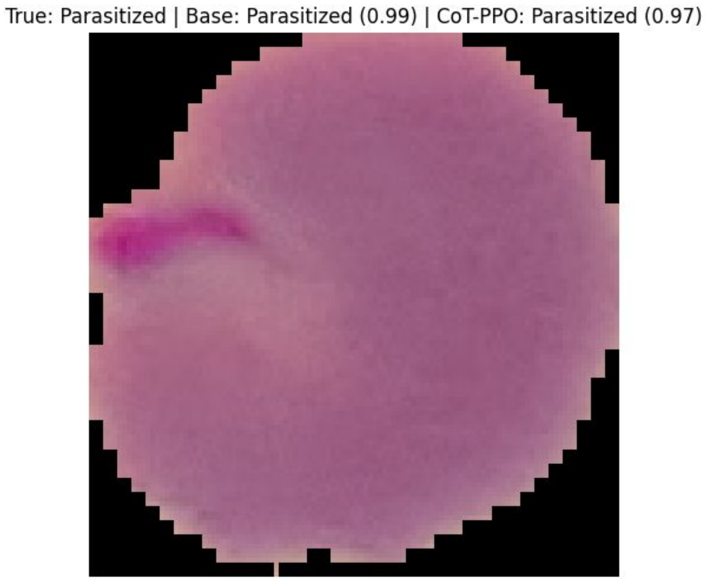


```
Sample 3:
True Label: Uninfected
Base ViT: Parasitized (confidence: 0.6522)
CoT-PPO: Parasitized (confidence: 0.9458)

Chain-of-Thought Reasoning:
Step 1: The cell boundary was analyzed to assess morphological regularity.
Step 2: The overall cell appearance was benchmarked against uninfected examples.
Step 3: The cell boundary was analyzed to assess morphological regularity.
Conclusion: The cell is **likely parasitized** with a confidence of 94.6%.
Key infection traits detected include:
- Disrupted membrane boundary
- Chromatin dot visibility
- Parasite-like inclusions within the cytoplasm

Reference Similar Cases:
- Case #1: Parasitized (similarity: 15.3%)
- Case #2: Parasitized (similarity: 8.8%)
- Case #3: Parasitized (similarity: 8.3%)
```

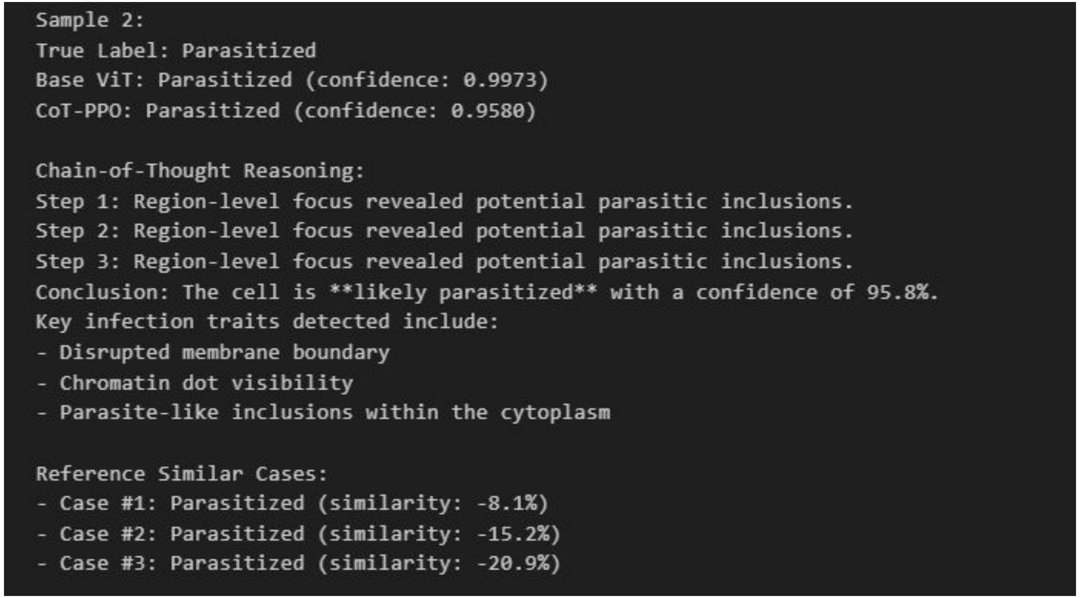
This example (Sample 3) shows a “True: Uninfected” cell that both the Base ViT and the CoT-PPO model misclassify as parasitized - albeit the **CoT-PPO model does so with much higher confidence (≈0.95 vs. 0.65).**

The image you see the **model’s chain-of-thought**: a **step-by-step morphological analysis** (cell boundary, appearance benchmarking) culminating in a parasitized verdict, along with **key trait highlights** (e.g. disrupted membrane, chromatin dots) and **three nearest-neighbor reference cases**.

The slide illustrates how the **CoT-PPO approach boosts confidence and transparency**, even when its prediction is ultimately wrong.



True: Parasitized | Base: Parasitized (0.99) | CoT-PPO: Parasitized (0.97)



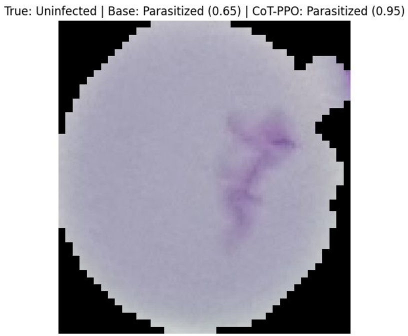
```
Sample 2:
True Label: Parasitized
Base ViT: Parasitized (confidence: 0.9973)
CoT-PPO: Parasitized (confidence: 0.9580)

Chain-of-Thought Reasoning:
Step 1: Region-level focus revealed potential parasitic inclusions.
Step 2: Region-level focus revealed potential parasitic inclusions.
Step 3: Region-level focus revealed potential parasitic inclusions.
Conclusion: The cell is **likely parasitized** with a confidence of 95.8%.
Key infection traits detected include:
- Disrupted membrane boundary
- Chromatin dot visibility
- Parasite-like inclusions within the cytoplasm

Reference Similar Cases:
- Case #1: Parasitized (similarity: -8.1%)
- Case #2: Parasitized (similarity: -15.2%)
- Case #3: Parasitized (similarity: -20.9%)
```

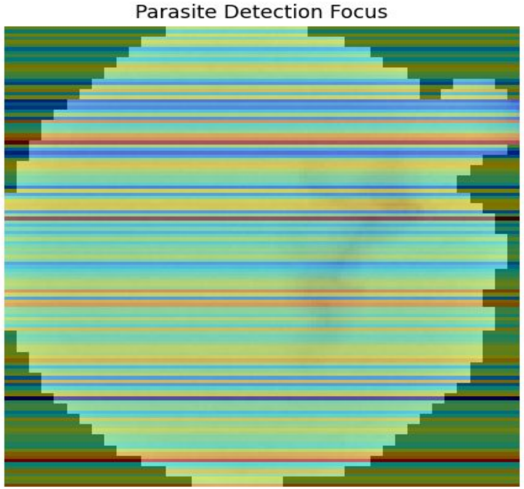
In this correctly classified parasitized example, both the Base ViT and CoT-PPO models predict “Parasitized” with very high confidence (≈0.99 vs. ≈0.96).

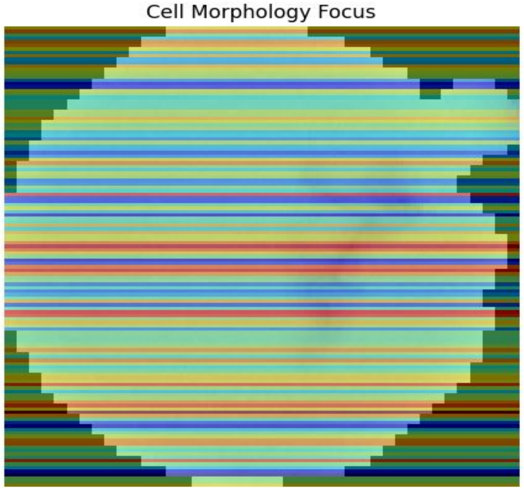
The **CoT-PPO chain-of-thought repeatedly highlights region-level parasitic inclusions** and then concludes with a **95.8% confidence, citing disrupted membrane, chromatin dots, and cytoplasmic inclusions.**

Below, the reference cases (all parasitized) show how the model’s similarity scores - though negative - still **rank its nearest neighbors for added transparency**.

Attention Maps


True: Uninfected | Base: Parasitized (0.65) | CoT-PPO: Parasitized (0.95)



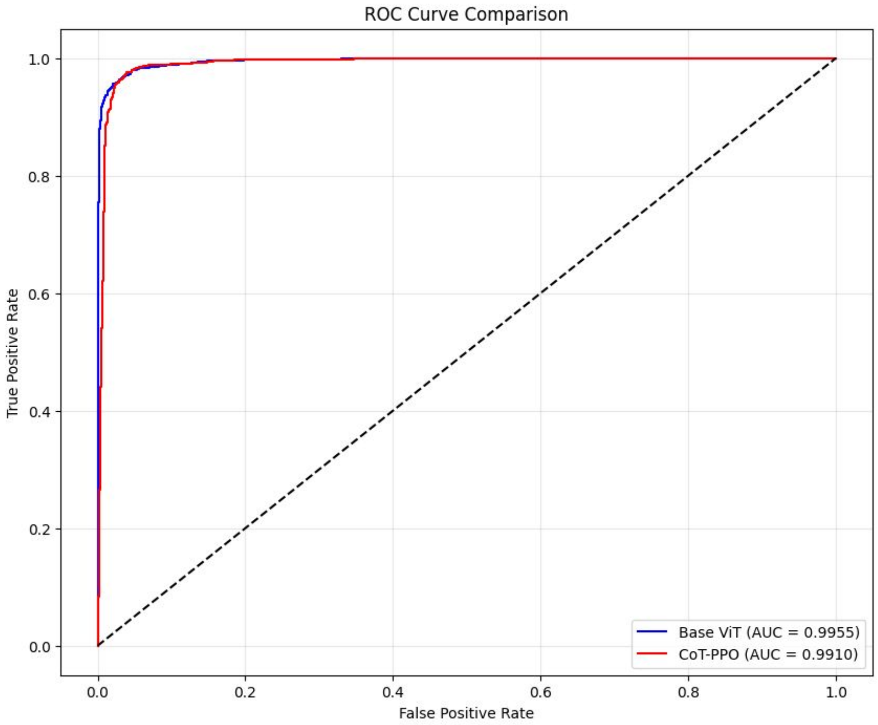


Description: These attention maps visualize the model's focus areas for malaria detection.

- **Parasite Detection Focus (Left)**: Highlights regions where the model identifies potential parasites, emphasizing areas with distinct parasite features in red.
- **Cell Morphology Focus (Right)**: Shows the model's attention to overall cell structure, with blue areas indicating focus on cell shape and boundaries.

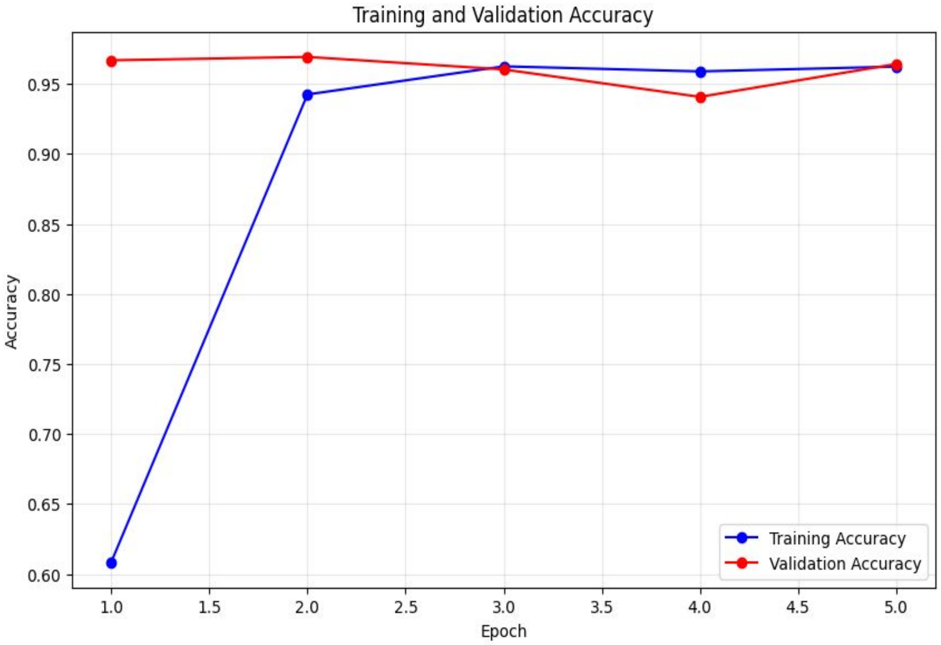
Generated from the pathology feature extractor in the CoT model.
Visuals: Heatmaps overlaid on sample images (Parasite focus in red, Cell focus in blue).

ROC Curve Comparison



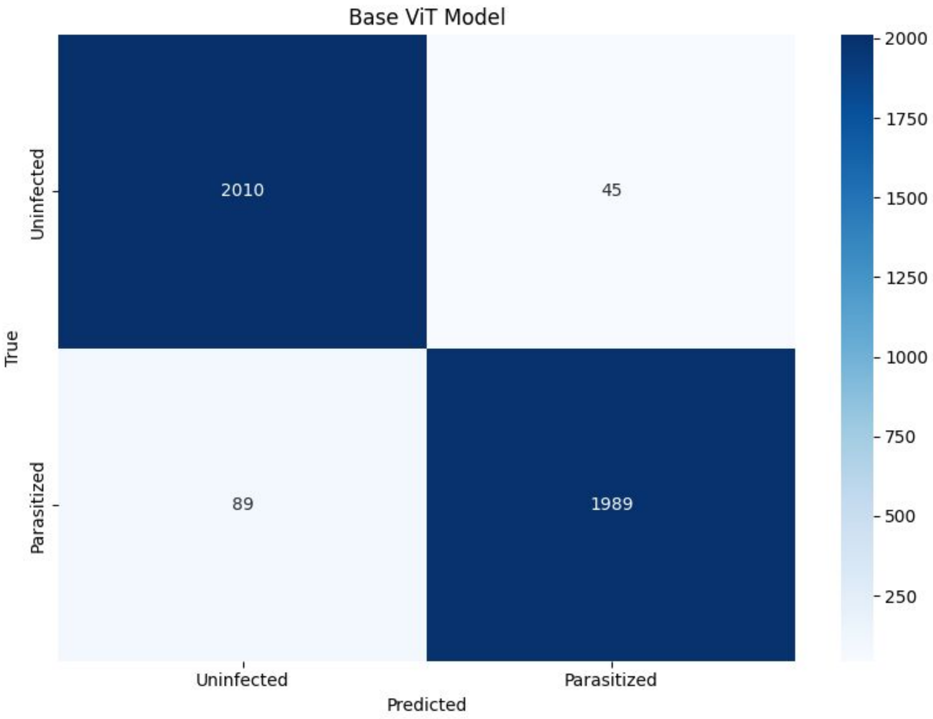
- **ROC Curve Comparison**: Evaluates Base ViT vs. CoT-PPO models.
- **True Positive Rate (Y-axis)**: Sensitivity (correctly identifying parasitized cells).
- **False Positive Rate (X-axis)**: Incorrectly labeling uninfected cells as parasitized.
- **Base ViT (Blue)**: AUC = 0.9955, strong discrimination ability.
- **CoT-PPO (Red)**: AUC = 0.9910, slightly lower but still high performance.
- **Dashed Line**: Random classifier (AUC = 0.5) for reference.
- **Key Insight**: Both models excel, with Base ViT slightly outperforming CoT-PPO in AUC.

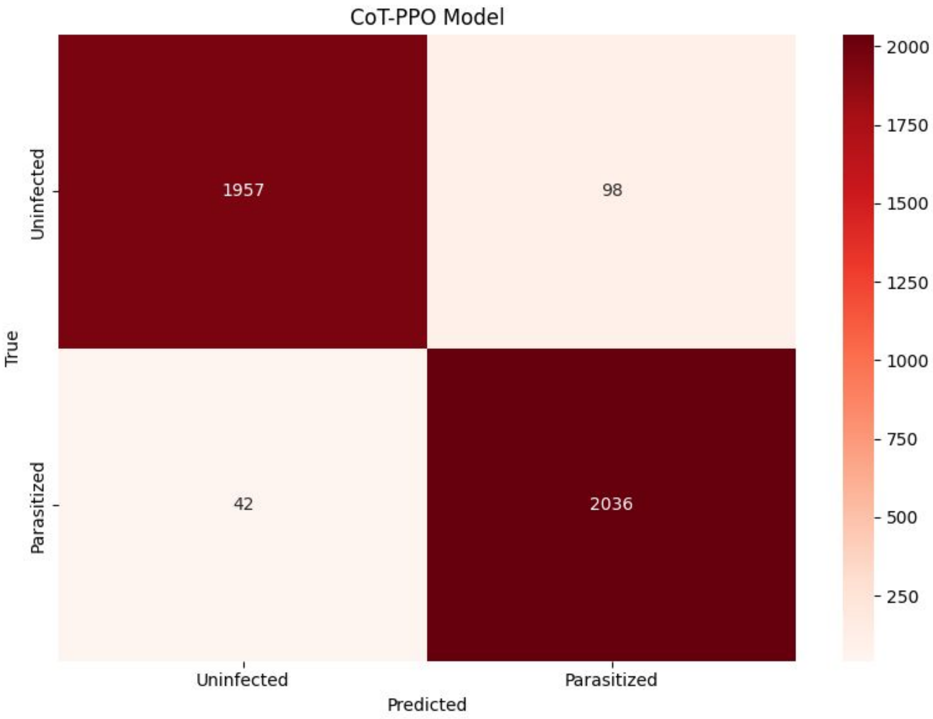
Training and Validation Accuracy



- **Graph Title**: "Training and Validation Accuracy"
- **Axes**: X-axis (epochs: 1.0 to 5.0), Y-axis (accuracy: 0.60 to 0.95)
- **Training Accuracy (Blue Line)**: Starts at 0.60, jumps to 0.95 by epoch 2.0, then stabilizes
- **Validation Accuracy (Red Line)**: Begins at 0.95, fluctuates between 0.90–0.95, ends at 0.95
- **Observation**: Gap between training and validation accuracy suggests overfitting
- **Validation Issue**: High initial validation accuracy may indicate a small or unrepresentative validation set
- **Recommendation**: Investigate data and apply regularization to improve generalization

Confusion Matrix


Base ViT Model


CoT-PPO Model

Confusion Matrix

True \ Predicted	Uninfected	Parasitized	Total True
Uninfected	2010	45	2055
Parasitized	89	1989	2078
Total Predicted	2099	2034	4133

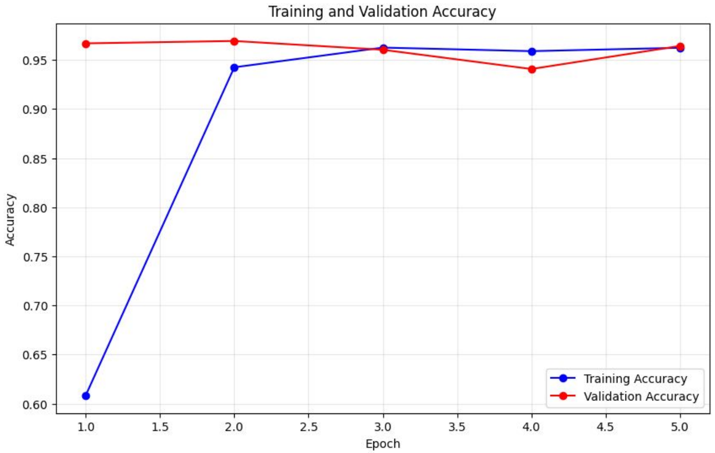
Base ViT Model

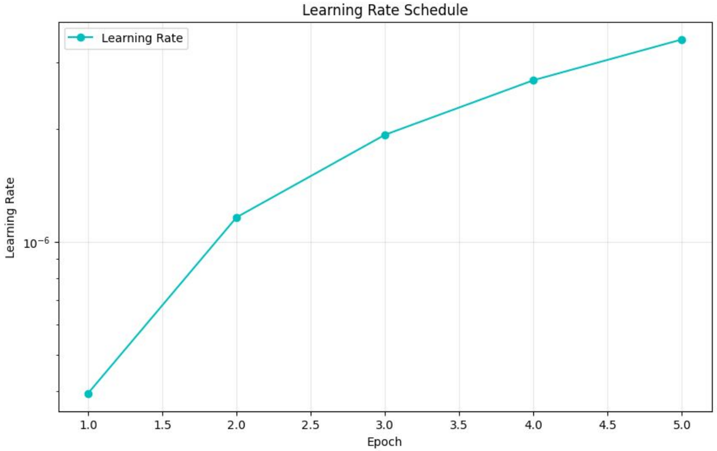
CoT-PPO Model

True \ Predicted	Uninfected	Parasitized	Total True
Uninfected	1957	98	2055
Parasitized	42	2036	2078
Total Predicted	1999	2134	4133

Performance Metrics

- **Evaluation Metrics** (Test Set):
  - **Accuracy**: Percentage of correctly classified images (Parasitized vs. Uninfected).
  - **Precision**: Proportion of true positive predictions among positive predictions.
  - **Recall**: Proportion of true positives identified correctly.
  - **F1-Score**: Harmonic mean of precision and recall.
  - **AUC-ROC**: Area under the Receiver Operating Characteristic curve, measuring model discrimination.

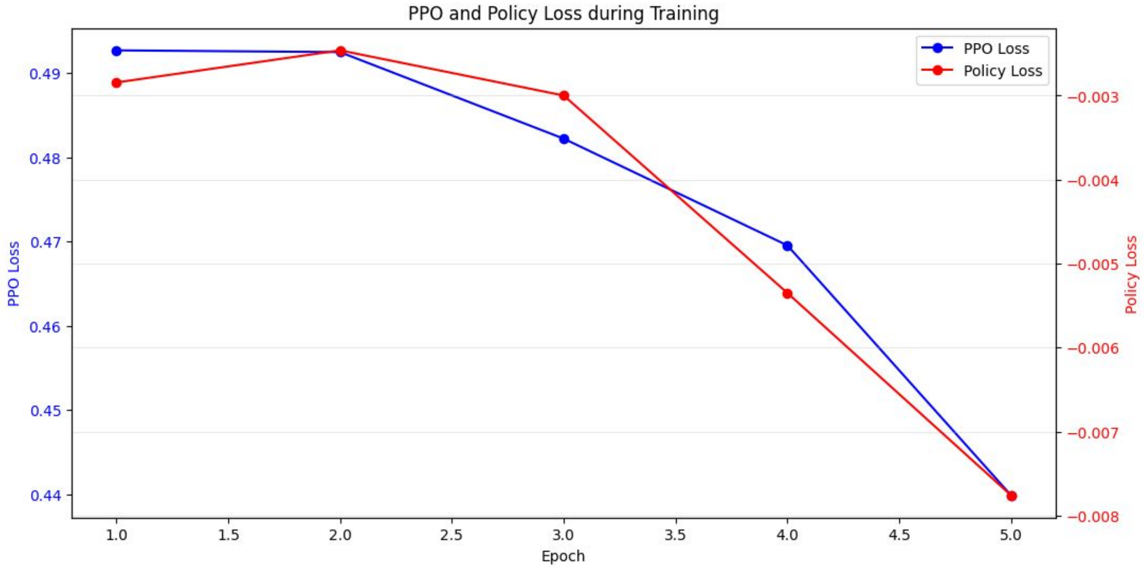




Performance Metrics

Metric	Base ViT Model	CoT-PPO Model
Accuracy	(2010 + 1989) / 4133 = **0.967**	(1957 + 2036) / 4133 = **0.966**
Precision (Parasitized)	1989 / (1989 + 45) = **0.978**	2036 / (2036 + 98) = **0.954**
Recall (Parasitized)	1989 / (1989 + 89) = **0.957**	2036 / (2036 + 42) = **0.980**
F1-Score (Parasitized)	2 * (0.978 * 0.957) / (0.978 + 0.957) = **0.967**	2 * (0.954 * 0.980) / (0.954 + 0.980) = **0.967**

Visualizations - Loss Plots



Over the five epochs, both PPO loss and Policy loss steadily decrease - PPO loss falls from about 0.49 down to 0.44, while policy loss moves from roughly –0.003 to –0.008.

This **consistent downward trend indicates that the agent’s policy is improving and the training is effectively optimizing both objectives.**

Challenges & Current Limitations

▯ **Integration Complexity:**

- Merging supervised learning with reinforcement learning components.
- Tuning the memory and reasoning modules to capture meaningful context.

▯ **Explainability:**

- The chain-of-thought head currently outputs a basic vector; needs enhancement for human-readable explanations.

▯ **Simulated Environment:**

- The reward mechanism is simplified; a more realistic simulation is required.

▯ **Resource Constraints:**

- Computational limitations when scaling to larger datasets and deeper models.

Future Work & Next Steps

Memory Module Enhancements:

- Explore learnable memory dynamics and larger memory buffers.

Advanced Reasoning Techniques:

- Experiment with deeper and more complex Transformer layers.

Environment Simulation:

- Develop a more sophisticated RL environment that better mimics clinical scenarios.

Explainability Improvements:

- Integrate with natural language models to convert CoT vectors into textual explanations.

Conclusion

- **Innovative Integration**: RL-ViT-alia combines Vision Transformers with memory, Chain-of-Thought reasoning, and reinforcement learning, achieving high accuracy in malaria detection while offering interpretable outputs for clinical trust.
- **Scalable Black-Box Solution**: The model can function as a standalone, automated diagnostic system, ideal for rapid deployment in resource-limited settings with minimal user interaction.
- **Flexible and Generalizable**: Its modular design allows adaptation to other medical imaging tasks, serving as a plug-and-play framework for diverse diagnostic applications.
- **Enhanced Decision Support**: By providing transparent, step-by-step explanations, the system supports healthcare professionals, balancing performance with user-friendly interpretability.

References

[1] World Health Organization (WHO). World Malaria Report 2024. https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2024

[2] NIH Malaria Dataset. https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html

[3] Dosovitskiy, A., et al. ”An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv:2010.11929, 2020.

[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. ”Proximal Policy Optimization Algorithms.” arXiv:1707.06347, 2017.

[5] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. ”Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” arXiv:2201.11903, 2022.

[6] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., K¨uttler, H.,Lewis, M., Yih, W., Rockt¨aschel, T., Riedel, S., & Kiela, D. ”Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” arXiv:2005.11401, 2020.

[7] Rajaraman, S., Antani, S.K. ”Pre-trained CNNs for Malaria Detection.” PeerJ, 2018.

[8] Marques, G., Ferreras, A. ”EfficientNet for Automated Malaria Diagnosis.” Multimedia Tools and Applications, 2022.

[9] Sandler, M., Howard, A., Zhu, M., et al. ”MobileNetV2: Inverted Residuals and Linear Bottlenecks.” CVPR, 2018.

Individual Contributions

All team members contributed equally across all aspects of the project including implementation, training, testing, and documentation.

If we were to highlight specific focus areas:

Rohan G (CS22B1093) - Chain of Thought Module, Memory Implementation, System Design

R Sai Charish (CS22B1095) - Vision Transformer Backbone, Experimental Evaluation, Visualization Components

T Pratyek (CS22B1093) - PPO Implementation, Reward Function Design, Model Training & Evaluation, Fine-tuning