Pending Tasks and Challenges

- *Dimensionality Mismatch: There is an unresolved issue with a dimensionality mismatch: "The size of tensor a (128) must match the size of tensor b (512) at non-singleton dimension 2". This error occurs within the MoE Wait-k policy during tensor operations involving token embeddings and expert weights.*

  Plan to Tackle: Adjust the tensor shapes for better broadcasting in PyTorch. If the issue persists, the plan is to explore a Keras implementation to leverage TensorFlow's handling of tensor operations, which may offer better flexibility or debugging insights.

- *Reward Function Tuning: The reward function balancing translation quality (BLEU/ROUGE) and latency metrics needs further refinement.*

  Plan to Tackle: While continuing with the current PyTorch implementation, we may experiment with Keras to simplify the policy learning and gradient optimization for the reinforcement learning module.

- *Optimization of RL Agent: The policy gradient RL agent needs further fine-tuning to better predict the optimal k value, minimizing latency without sacrificing translation quality.*

  Plan to Tackle: If PyTorch-based optimization doesn't yield the expected improvements, we will test out Keras' RL libraries for easier integration and faster iterations.

- *Evaluation on Larger Datasets: After fixing the dimensional issues and optimizing the reward function, we plan to scale up testing on larger datasets.*

  Plan to Tackle: Transitioning to Keras may simplify data pipeline integration with TensorFlow's Dataset API for larger-scale evaluation if PyTorch performance or debugging becomes too complex.

Alternative Frameworks: *If issues with PyTorch persist, we will consider switching to TensorFlow/Keras for the model implementation. This may involve rewriting parts of the code to adapt to the new framework.*

While significant challenges remain, particularly regarding tensor operations and syntax, the approach taken shows potential for effective translation capabilities. Future efforts

will focus on addressing these challenges, ensuring the model operates smoothly, and exploring TensorFlow/Keras as an alternative if necessary.