1. Refined Problem Statement

Problem statement: How can we improve the accuracy of prompt tuning to help prevent bias and misinformation in Large Language Models?

Current research questions:
Can we find some differentiation between success and failure cases during prompt tuning to predict failures before they occur?
Can we create some sort of post-hoc calibration that will allow the model to learn from its mistakes during training?

2. Updated Literature Review

Choi, J., Kim, J., Park, J., Mok, W., & Lee, S. (2023). SMoP: Towards Efficient and Effective Prompt Tuning with Sparse Mixture-of-Prompts. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 14306–14316. https://doi.org/10.18653/v1/2023.emnlp-main.884

In this article, the researchers propose a Sparse Mixture of Prompts (SMoP) as a new method of prompt-tuning which uses short soft prompts that maintains performance gains while improving efficiency. They achieve this by using a gating mechanism that trains multiple smaller soft prompts that handle different subsets of data rather than a single long soft prompt. They have shown through experiments that SMoP outperforms fine tuning while also being more efficient.

We could adopt a similar gate mechanism, where the gate trains itself how to filter between good and bad sample predictions, which would be a more elegant solution than the proposed solution detailed in my writing above because we wouldn't have to figure out the

relationship between a good/bad sample and the training model, making this a more generalized solution.

ZeroTuning: Unlocking the Initial Token's Power to Enhance Large Language Models Without Training:

Han, F., Yu, X., Tang, J., & Ungar, L. (2025, May 16). *ZeroTuning: Unlocking the initial token's power to enhance large language models without training*. arXiv.org. https://arxiv.org/abs/2505.11739

Research Question:

Is it possible to improve model performance by tuning the attention to a universal and task-agnostic token without relying on task-specific token identification?

Findings:

Tuning the initial token consistently improves LLM performance more than any other token. The tuning effect propagates through all the layers, meaning that the initial and middle layers contribute more than the later ones. Tuning all the layers jointly yields the best results. Some attention heads respond positively while others respond negatively to initial token p-tuning. They found that selectively tuning them outperforms tuning them uniformly.

They created the ZeroTuning method, a training-free method that improves the LLM's performance through recalibrating the initial token's attention without requiring task-specific token identification.

Hu, J., Zhang, Z., Chen, G., Wen, X., Shuai, C., Luo, W., Xiao, B., Li, Y., & Tan, M. (2025, May 27). *Test-Time learning for large language models*. arXiv.org. https://arxiv.org/abs/2505.20633

In this paper, they propose Test Time Learning for LLMs, or TLM for short. They've observed that high-perplexity samples are more informative for model optimization than low-perplexity samples, so they created the Sample Efficient Learning Strategy which uses

those high-perplexity samples for test-time updates. They found that TLM improves the performance of original LLM's by at least 20% on domain knowledge adaptation. The researchers used LoRA (Low-Rank Adaptation), as they found it was more effective at preventing catastrophic forgetting than full parameter updates.

It could be possible to use a perplexity-defined strategy like this to create a gate, where we filter out prompts of a certain threshold of perplexity into a different untrained model. This would mean that we are inferring that a higher perplexity has a correlation to a less accurate response. While they use LoRA, we could probably adapt their method for a prompt tuning approach.

Ji, K., Xu, J., Liang, T., Liu, Q., He, Z., Chen, X., Liu, X., Wang, Z., Chen, J., Wang, B., Tu, Z., Mi, H., & Yu, D. (2025). The first few tokens are all you need: an efficient and effective unsupervised prefix Fine-Tuning method for reasoning models. *arXiv.org*. https://doi.org/10.13140/RG.2.2.33772.07043

Research Question:

In this paper, they propose an unsupervised fine-tuning method that requires only a single pass of model-generated responses per question, coupled with prefix-based fine-tuning.

Findings:

Just by using prefix substrings for guidance, they were able to outperform full-token fine-tuning approaches. By taking this approach, the training time and inference time is greatly reduced.

Important findings for my research question:

Because this approach uses unsupervised fine-tuning, most methods will not be applicable to the supervised fine-tuning that we plan to conduct for our research question. However, the prefix-based fine-tuning used in this paper will provide important insights to how we can improve fine tuning in our approach.

Efficient Test-Time Adaptation of Vision-Learning Models:

Karmanov, A., Guan, D., Lu, S., Saddik, A. E., & Xing, E. (2024, March 27). *Efficient Test-Time adaptation of Vision-Language models*. arXiv.org. https://arxiv.org/abs/2403.18293

In this paper, they proposed to utilize a positive and negative cache to store the keys and values of either confidence or not confidence samples. Then, the cache is used alongside the CLIP predictions to create the final prediction. This way, they minimize the chance that lower confidence labels will be selected.
To do this, they filter out a certain proportion of the samples that are deemed less confident than expected, and those would go to the negative cache.
This differs from our research in terms of genre, as they focus on Vision-Learning Models while we are focusing on Large Language Models.

P-Tuning v2:

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2021, October 14). *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*. arXiv.org. https://arxiv.org/abs/2110.07602

Research findings:

Properly optimized prompt tuning can be universally effective across a wide range of model scales and NLU tasks. They can match the performance of fine-tuning, while only having to use 0.1%-3% tuned parameters.

Important findings for my research question:

The discovery that prompt tuning could possibly match the accuracy of fine-tuning is very important for my research question. Because my hope is to create an accurate LLM that only uses prompt tuning with no further refinement, this paper reinforces the possibility of this method. Since their github and datasets are public, I will be building off of their resources to experiment and hopefully find some insightful results.

Efficient Streaming Language Models with Attention Sinks:

Xiao, G., Tian, Y., Chen, B., Han, S., & Lewis, M. (2023, September 29). *Efficient Streaming Language Models with Attention Sinks*. arXiv.org. https://arxiv.org/abs/2309.17453

Research question:

Can we deploy an LLM for infinite-length inputs without sacrificing efficiency and performance?
Findings:

They created StreamingLLM, an efficient framework that enables LLMs trained with a finite length attention window to generalize to infinite sequence length without any fine-tuning.

This was done by taking use of the attention sink of several initial tokens. These initial tokens are considered sink tokens because they are visible to all following tokens. A large amount of attention is being disproportionately spent on the first few tokens, and dense and window attention levels collapse after a certain training point. By reintroducing ~4 initial tokens as attention sinks, the model perplexity returns to normal and can accurately represent text over 4 million tokens.

Important findings for my research question:

The main use of this paper is their discovery of the attention sink in the first few tokens. We will be testing how adjusting the soft prompt prepended onto the frozen tokens, so knowing the importance of attention sink in initial tokens will be very useful for gauging the effect of prompt tuning with a gate.

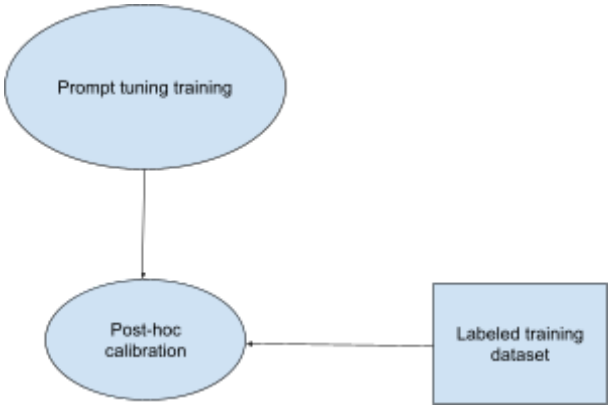
Test-Time prompt tuning for Zero-Shot generalization in Vision-Language models:

Shu, M., Nie, W., Huang, D., Yu, Z., Goldstein, T., Anandkumar, A., & Xiao, C. (2022, September 15). *Test-Time prompt tuning for Zero-Shot generalization in Vision-Language models*. arXiv.org. https://arxiv.org/abs/2209.07511

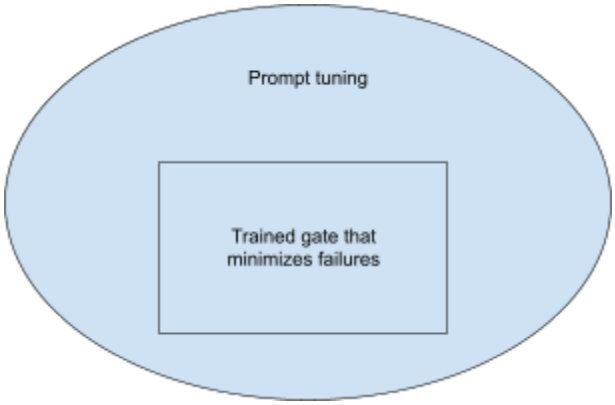
In this research for Vision-Language Models, they are able to make prompt tuning more accurate by discarding views that are less confident, reducing noise from the views and making the results less chaotic. Additionally their prompt tuning is done without a training stage. Instead, the model goes through unsupervised learning to change the prompt for each individual test image.
While we plan to use supervised learning and will perform a testing stage, their approach at reducing noise could be very valuable to us.

3. Conceptual Diagrams

Plan A:



Plan B:



4. Experimental Design

Baseline: Prompt tuning v2:
Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2021, October 14). *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*. arXiv.org. https://arxiv.org/abs/2110.07602

I plan to recreate all of their scripts. Once that is done, I will build off their code to report improvements upon their work. Their baseline can be found at that link.

Datasets include CB, COPA, ConLL04, ConLL05, WSC, WiC, and more. Many of these datasets are Hugging Face.
The main metric to measure is accuracy.
We are aiming to either do some post-hoc calibration or gating method.

5. Preliminary Results and Key Findings

I have so far recreated 6/8 of the BERT scripts, and will soon move on to the RoBERTa scripts. Here are my results:
Recreated two scripts last week:

1. BERT + COPA: 71.0, comparable to their 73.0

2. 2. BERT + BoolQ: 73.73, comparable to their 75.8

New results:

1. WSC + Bert: 65.38 (their results - 68.3)
2. RTE + Bert: 76.53 (their results - 80.1)
3. WiC + Bert: 72.88 (their results: 75.1)
4. CoNLL04 + Bert: 82.21 (their results: 84.5)

While there is a discrepancy, I've found out the reason why – my testing environment is not exactly the same as theirs, leading to a worse accuracy. I am actively testing different parameters to try to recreate their results as much as possible.