1.What challenges might we encounter if we try to apply a probability-based detection approach, such as the one proposed by Mitchell et al. in this paper, to the output of models that are not downloadable, and are only accessible via APIs, such as GPT-4? (A sentence or two should be sufficient here)

Mitchel el al's approach is based on a white-box assumption - which means they are relying on the model's provided algorithm/function to evaluate log probabilities of the models text output. There is no hidden part that hinder the probability evaluation process. However, if the model are not downloadable, and are only accessible via APIs, then it will cost money to access that api.

2. We can always change the LLM so that its output is no longer detectable by a given detection approach. Describe how you would change GPT-2 or any LLM so that its output will no longer be detectable by the Mitchell et al. approach. Your answer should be more specific than "I will change the training" or "I will change the decoding strategy". Be specific about exactly what you will change. Feel free to use illustrative elements in your response, such as examples, equations or pseudo-code. (A short paragraph is sufficient here, but you are welcome to elaborate.)

I think I will try to add more data in the training process, basically incorporate more human write sentence in the training process of the model. The training text can come from more diverse source, like social media, which more likely to incorporate urban slang and new trendy word/sentence pattern among young population. Although it may lose some professional tone/ and accuracy in generated response, but it will more likely to pass the detector. The model needs to have hyper-parameters to adjust how much the model wants to fit these human-written text vs answer accuracy. Some round of adversarial training against the detector mechanism and some round of fine tuning to adjust accuracy can be useful to generate the suitable model against LLM detection mechanism.

3.Describe ways in which you (and more generally students) should be able to use large language models (LLMs) in your courses in a productive way. That is, LLMs should not interfere with learning outcomes. Here is what ChatGPT has to say. Provide your own thoughts. (A couple of sentences should be sufficient here)

I think some good ways to use LLM in course work are:

1. **Use it to generate examples for concepts: It is very useful for me over the graduate study, found certain term hard to understand, I often use chatgpt to help me understand certain concept by asking it "Give me an example on how xxx work over xxx/how to use xxx". A lot of time it gives me very helpful response.**
2. **Help me to fix grammar and debug: As a non-native English speaker it is very convenient to use LLM to fix my grammar mistake, make my wording more confident and professional; as a cs student, a lot of typo in command or problem in program can be fixed by having LLM check the code I write, saves a lot of time on some trivial mistake.**
3. **Use it as a browser with advance setting: sometime traditional browser will have ad content, or content favor your opinions they figure out from your past search, which**

can cause some bias searching result. We can use LLM and explicitly asks it to find information from different source, and ask it provides positive and negative opinions on a same topic. This way we can mitigate some search bias and develop skeptical thinking.