rough ideas

### Overarching principles to keep sight of:

* Begin with an interesting story of AI to hook the audience
* Clearly establish the goals of the presentation and the “why you need to listen”
  + Empowerment parameters ✅
* Frequent checkpoints to the goals of the presentation
  + Cycle on the problem without coming across as condescending ✅
  + Repeat the intro and goal of the talk slide so that we can jump back on board at each new idea ✅
  + Ask a question at the end of each section so that we can get people back on track and engaged
    - Not too easy or hard ✅
* Make several onramp points for if someone loses concentration
* Take questions
* Lock in the keypoints in a short intuitive summary
* Always exhibit passion
* Express vision and passion in the first 5mins
  + This is a miracle of modern science
* Have one really cool demo that does something amazing!
  + Pretrained bible translator?
  + Joke maker?

## Overview and structure

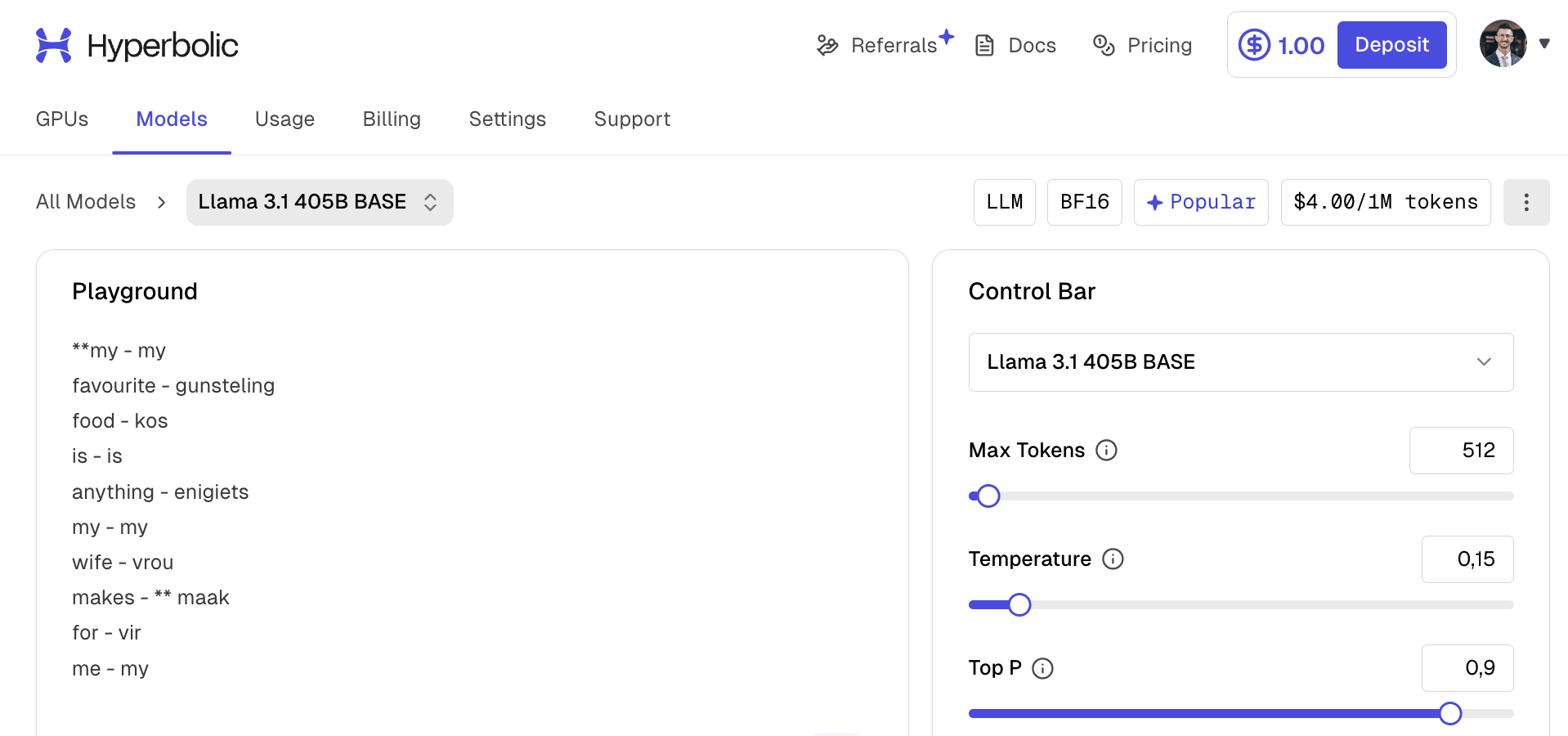
* Intro
  + Start with a story of AI or LLMs
    - Move 37 of alphaGo?
  + Empowerment params
    - What AI is and where LLMs fit in in the AI landscape
    - Basic intuition of how LLMs work
    - Common LLM usage pitfalls
      * Be able to have a sort of gut-feel for what tasks you can or cannot trust LLMs on
* Landscape slide -
  + How’re we going to do this
    - Transformer architecture
      * What is the attention mechanism
    - The common steps gone through for creating a basic LLM model
      * Training an LLM model
        + Tokenization
        + Pre-training
        + Post-training
      * Generating useful outputs
        + Autoregressive decoding
        + What is temperature?
    - Let’s build a (micro) LLM from scratch

### Start of main talk

* Question: What is AI
  + The AI we think of is called supervised learning because it learns from examples, we have inputs and outputs and the mathematical equations tune weights and biases that are little numbers that are tuned to create a mathematical mapping from those inputs to outputs in order to teach them how to predict, classify and do things that we find helpful
    - Common AI models and a landscape overview - figure
    - Where do LLMs fit into this landscape
* Landscape slide - cycle
* Transformer architecture - attention demo from NLP with Python textbook
  + Explain attention - (very shallow explanation)
  + Autoregressive decoding
  + Attention graph?
* Landscape slide - cycle
* Tokenization - how does this model *know* what a word or text is?
  + Question: How would we feed in this information about the internet or whatever our sources is into something like an AI model that can only work with numbers
    - Character level vs word level quick overview
    - Figure to demo this, char vs word level vs hybrid
  + Show [tiktoken](https://tiktokenizer.vercel.app/) example
    - My superhero name is Max Verstappen
    - Learning LLMs is as easy as 12.34567
    - Explain how typos are really bad
      * my faorite food is any food my wife makes... or chocolate
* Landscape slide - cycle
* Pretraining - how to speak, factual knowledge - glorified autocomplete
  + Question: How would we build a model that has information about everything? Where would we start? -> internet, books, private data
    - Lossy zip file of the internet
    - Compress the info of all the world’s knowledge
    - Show [fine web dataset](https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1), explain that it’s 44TB in size
    - Try to get a model that can memorise not just the whole internet but learn the concepts of what things in the real-world are
    - We want to build a model that has a lot of general world-knowledge and can answer questions for us
  + End off by showing that it’s only a text completer on [hyperbolic](https://app.hyperbolic.ai/models/llama31-405b-base-bf-16)
  + Question: How would we turn this base-model into an assistant that can answer questions
  + Hyperbolic demo
* Landscape slide - cycle
* Post training - how to answer questions, how to think, how to apply the knowledge it has as an autocomplete to autocomplete QA pairs
  + Hyperbolic demo of completing question answer
    - Question: what is the capital of japan? Answer:
    - Question: How do you get a Dr to use an e-scripting platform instead of paper scripts? Answer:
    - Question: Please repeat this number back exactly 123456 with no other text. Answer:
* Landscape slide - cycle
* Generation - How to answer MY question
  + [Ai-studio](https://aistudio.google.com/prompts/new_chat) - what’s behind this screen?
  + Temperature Allows us to get creative (random answers)
    - How how on [ai-studio](https://aistudio.google.com/prompts/new_chat) the answers change when you give different questions
      * Give me a short poem
      * Indicate how randomness increases the possibility of non-perfect next token
      * “Perfect” next token is just a statistical determination of what word occurs next to the preceding word in the dataset given the info before
* Landscape slide - cycle
* Let’s build our own model -
  + Tokenization
  + Pretrain
  + Post-train
  + Decode and show temperature’s effects
* Landscape slide - cycle
* Question: Which of these tasks do you think LLMs will be good at
  + Summarization
  + Copy-pasting
    - Show temperature issue
  + NER
    - Concept recognition
  + Generating starter code for a project
    - Many examples of that in the training data
  + Humanlike interaction
  + Mathematics
    - Show the tokenization concept breaking what numbers are
  + Factual retrieval
    - Pretrain date
    - Current date
  + Regenerating a response multiple times
    - Show own example of temperature
  + Grammar

## Not in this talk

* Goal of the talk
  + Teach you all the details of what an LLM is, where it fits into the broader topic of “AI” and learn how it works intuitively
  + Then we’re going to build up a tiny LLM model together in code
  + Know how to spot common pitfalls of using LLMs and the reasons why they occur to empower you to use LLMs more effectively next week when Marius and Peter Cressey give the next LLM masterclass
* How to use LLMs effectively
  + RAG/context engineering
  + How to use as a dev
    - Templates
    - AI Studio
  + Multiple attempts
  + New chats
    - Try to get it in one-shot
      * Multiple questions create confounding attention
* LLM pitfalls
  + Copy-pasting
    - Copy a very large list and show that it’s not correct
  + Mental math
    - Big numbers multiplication breaks
  + Hallucinations and factual retrieval
  + Stochasticity
    - Ask for a poem multiple times, different results
    - Temperature increases randomness
* What LLMs are very good at
  + Summarization
    - attention mechanism can capture the semantic meaning/vibe
  + Grammar
    - Pretrained on high quality data without grammar “errors”
  + Code syntax
    - Pretrained on so much code
  + Coding overall
    - Getting better every day
    - Good at common tasks or even hard tasks that are broken down easily
    - Making code work together



my - my

favourite - gunsteling

food - kos

is - is

anything - enigiets

my - my

wife - vrou

makes -