# Introduction to LLMs

Understanding the technology behind modern AI text generation

## Definition

Large Language Models are deep learning algorithms designed to recognize, summarize, translate, predict, and generate human-like text.

## Foundation

These models are built upon massive, diverse datasets and utilize complex neural network architectures, typically Transformers.

## Key Examples

Prominent models currently shaping the AI landscape include GPT-4, Llama 3, and Claude.

# Core Architecture: Transformers

The Foundation of Modern LLMs

### Encoder-Decoder Structure

A dual-component design where the encoder processes the input sequence and the decoder generates the output, enabling complex language tasks.

### Attention Mechanisms

Utilizes self-attention to dynamically weigh the relevance of words, capturing long-range dependencies more effectively than traditional models.

### Positional Encoding

Injects information about token positions to compensate for the lack of inherent sequential order in parallel processing.

### Massive Parallelization

Facilitates parallel data processing to reduce training times and overcome memory and latency constraints of RNNs.

Transformers facilitate massive parallelization, significantly reducing training times compared to Recurrent Neural Network (RNN) architectures.

# Processing Text: Tokens and Embeddings

Foundational concepts in natural language processing

### Tokenization

Breaking raw text into smaller units like words, sub-words, or characters to enable model processing.

### Embeddings

Converting tokens into high-dimensional numerical vectors that capture semantic meaning.

### Vector Reasoning

Demonstrating how algebraic manipulation, such as 'King' - 'Man' + 'Woman' ≈ 'Queen', captures relationships.

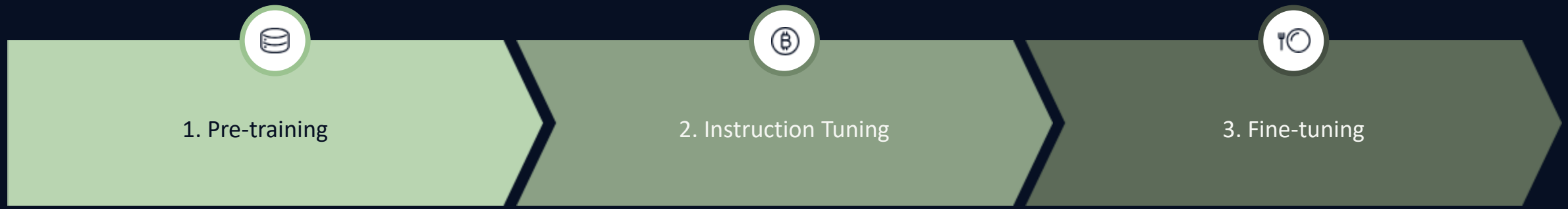
Self-Attention

# The Attention Mechanism

- Mechanism: Allows the model to weigh the importance of different words in a sequence, regardless of their distance.
- Disambiguation: Enables the model to distinguish word meanings based on surrounding context.
- Example:
  - 'River bank': High attention weight on 'river'.
  - 'Bank account': High attention weight on 'account'.
- Impact: Captures nuanced relationships and long-range dependencies efficiently.

# Training Lifecycle

The three-stage process for developing language models




Learning general language patterns from massive corpora using unsupervised methods.

Training the model to effectively follow and respond to specific user prompts.

Adapting the model for specialized domains or specific functional tasks.

# Controlling LLM Output

Fine-tuning Inference Parameters

- **Temperature**

  Controls creativity versus determinism: low values (0.1–0.3) provide precise, predictable results, while high values (0.7–1.0+) increase randomness and diversity.

- **Top-K Sampling**

  Limits next-token selection to the top K most likely words, effectively pruning incoherent tokens from the long tail.

- **Top-P (Nucleus) Sampling**

  Filters vocabulary based on cumulative probability mass, allowing for dynamic adaptation by selecting the smallest set of tokens that sum to P.

# Ecosystem and Deployment

Overview of LLM integration and deployment models

- **LLM APIs**

  Leveraging managed services such as OpenAI API and Anthropic API for high-performance and scalable application integration.

- **Open-Source Models**

  Utilizing models hosted on platforms like Hugging Face, allowing for greater customization and community-driven innovation.

- **Local LLMs**

  Running models directly on consumer hardware using tools like Ollama and LM Studio, offering enhanced data privacy and offline capability.

# LLM Limitations and Evaluation

A overview of core challenges and validation frameworks

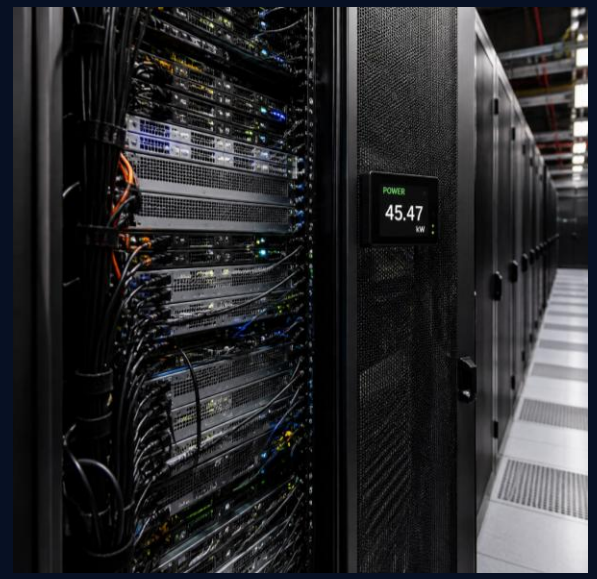
## Hallucinations

Risk of generating plausible but factually incorrect information.

## Bias and Privacy

Inheritance of stereotypes and challenges in protecting sensitive data.

## Compute Requirements



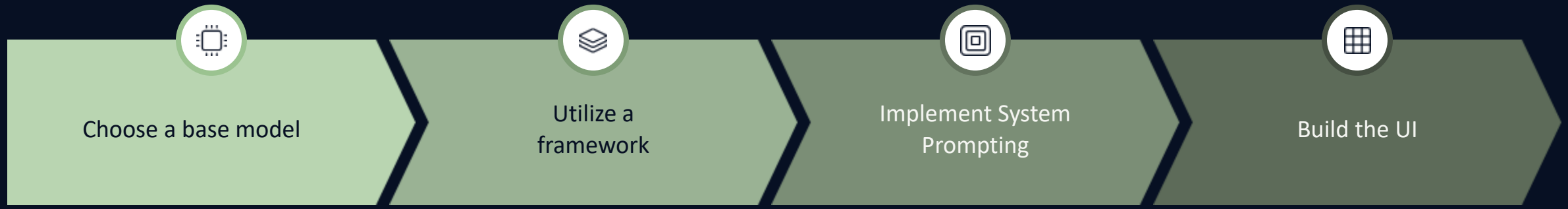

High energy and hardware costs for large-scale training.

## Evaluation Methods


Using benchmarks (MMLU, HumanEval) and Human-in-the-loop review.

# Mini Project: Building an LLM Chatbot

A roadmap for creating an interactive application




Select an optimized model like Llama 3 based on performance requirements.

Integrate LangChain or LlamaIndex for robust data orchestration.

Define the persona, tone, and operational boundaries of the chatbot.

Develop the frontend using Streamlit or Gradio for an interactive experience.

Goal: Create an interactive application.