LLMs & Wrappers

Understanding how to connect and control language models in LangChain through standardised interfaces and decoding parameters.

What is an LLM (Large Language Model)?

LLMs are sophisticated AI models trained on vast amounts of text data to understand and generate human-like language. These powerful systems have revolutionised how we interact with technology.

Popular Examples

GPT, **BERT**, **Falcon**, **LLaMA**, **Mistral**, and **Claude** are leading the charge in natural language processing.

Text Generation

Create coherent, contextual content

Summarisation

Condense information efficiently

Translation

Bridge language barriers seamlessly

Question Answering

Provide accurate, relevant responses

Reasoning Tasks

Perform complex logical operations

🧠 **LLMs are the "brain"** behind modern AI applications, powering intelligent interactions across industries.

What is an LLM Wrapper?

Definition

A *wrapper* is a **standardised interface** that allows different LLMs to be used within the same framework, regardless of their underlying implementation or API structure.

The Problem

Every LLM has its own unique API style, parameters, and implementation requirements.

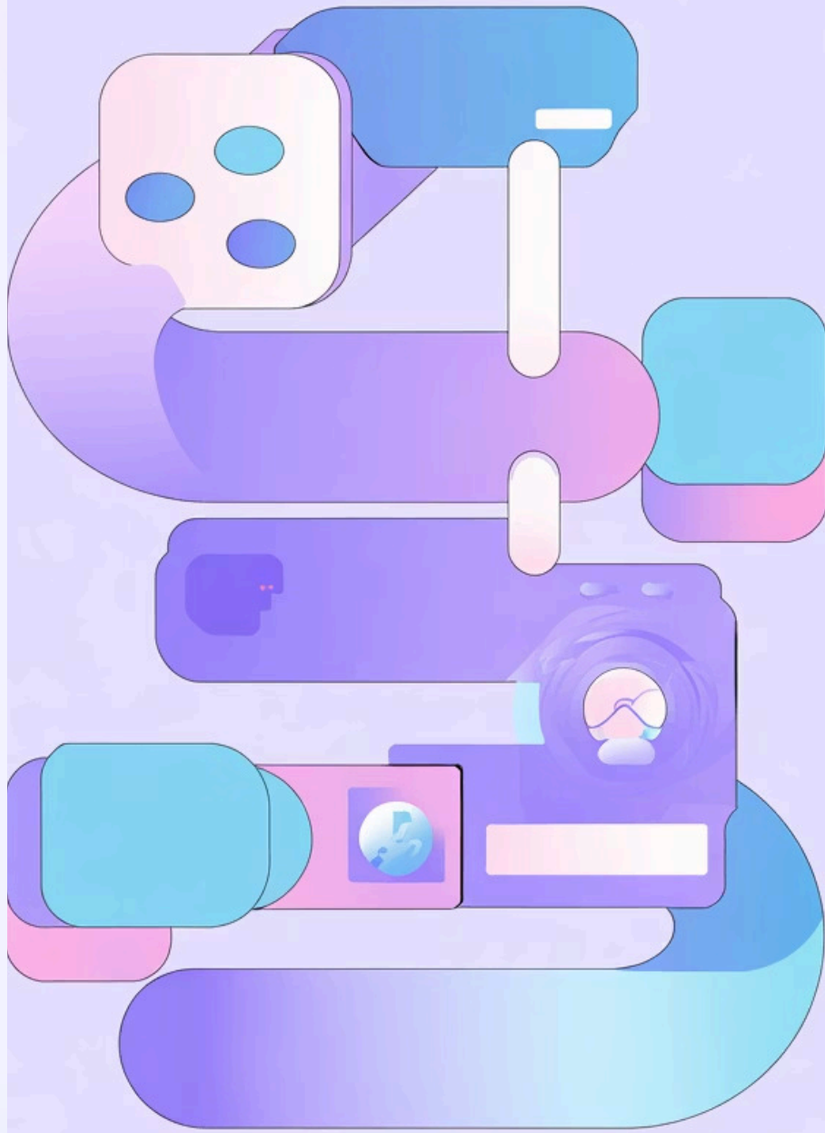
The Solution

Wrappers unify diverse models into a **common format** that developers can use consistently.

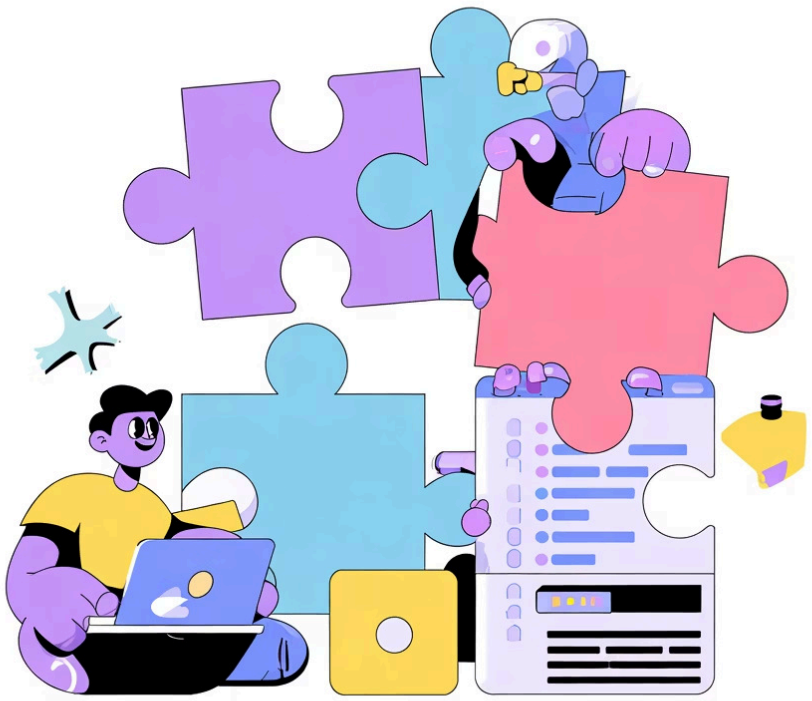
The Result

Seamless switching between models without changing application logic.

In LangChain: You can effortlessly switch from OpenAI → Hugging Face → Cohere → local models without rewriting your codebase.



Why Use LLM Wrappers?



LLM wrappers solve critical integration challenges that developers face when working with multiple AI models across different platforms and providers.

Challenge	How Wrappers Help
Different model APIs	Provide consistent, unified interface
Integration complexity	Simplify setup and usage dramatically
Experimenting with models	Enable easy model swapping
Need for modularity	Make pipelines flexible and scalable

🧩 **In short:** Wrappers make LLMs *plug-and-play* for developers, accelerating development and reducing technical overhead.

Hugging Face Integration — HuggingFacePipeline

What is Hugging Face?

A popular open-source platform hosting over 100,000 pre-trained models for natural language processing, computer vision, and audio tasks.

HuggingFacePipeline in LangChain

- Acts as a wrapper for Hugging Face models
- Supports text generation, summarisation, and translation
- Enables local model execution

Code Example

```
from langchain.llms import HuggingFacePipeline
from transformers import pipeline

pipe = pipeline("text-generation", model="gpt2")
hf = HuggingFacePipeline(pipeline=pipe)
```

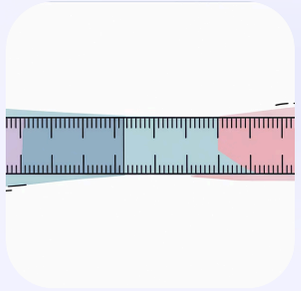
When to Use: Choose HuggingFacePipeline when you want to run **local or open-source** models instead of cloud APIs like OpenAI, giving you greater control and privacy.

Key Decoding Parameters

Fine-tuning these parameters allows you to precisely control how your language model generates responses, balancing creativity with consistency.

Temperature

Controls randomness in output. **High values** (e.g., 1.0) produce creative, diverse text. **Low values** (e.g., 0.2) generate focused, deterministic responses.

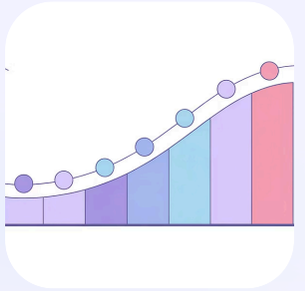


Max Tokens

Limits the output length by specifying maximum number of tokens. Controls how long the generated response can be, preventing excessive verbosity.

Top_k

Chooses from the top k most likely next words. **Small k** values create predictable output. **Large k** values enable diverse, exploratory text.



Top_p (Nucleus Sampling)

Selects words until cumulative probability reaches p. A value of **p=0.9** provides balanced creativity whilst maintaining coherence.

> 🎯 **Use these parameters to fine-tune model behaviour** and achieve the perfect balance for your specific use case.

Practical Example — Effect of Parameters

Prompt: "AI will change..."

See how different parameter settings dramatically alter the model's output style and content.

Temperature = 0.2

Focused & Deterministic

"AI will change the way people work."

Temperature = 1.0

Creative & Varied

"AI will reshape creativity, education, and human thought."

Top_k = 5

Narrow Choices

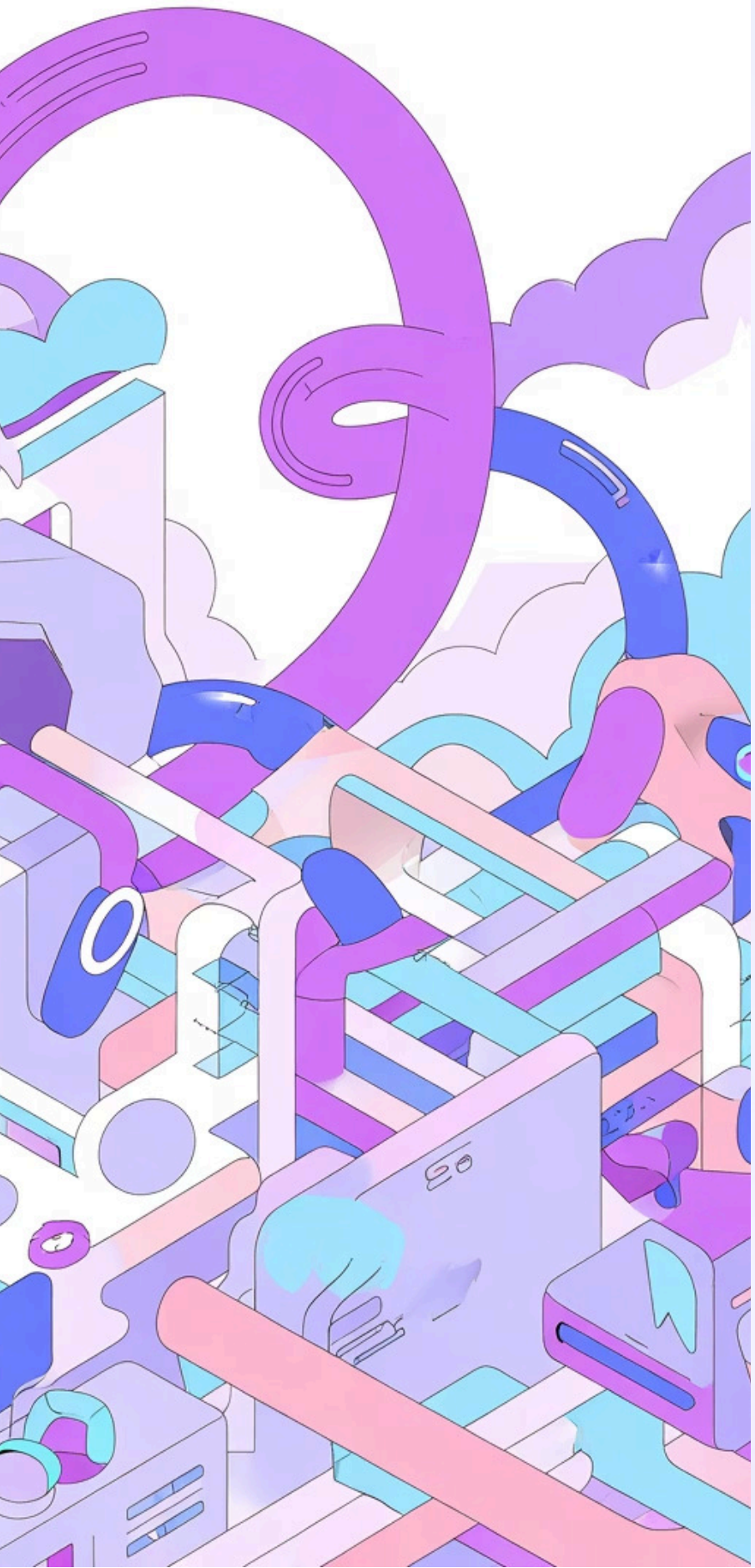
Repetitive but safe outputs with high predictability.

Top_k = 100

Broad Choices

Diverse outputs with potential for occasional nonsensical text.

🧠 **Pro Tip:** Start with temperature = 0.7 and top_p = 0.9 for balanced output that combines creativity with coherence.



Summary — LLMs & Wrappers

LLM

Core model that generates text and processes natural language

Wrapper

Standardised interface for any LLM, enabling seamless integration

HuggingFacePipeline

Integrates Hugging Face models directly into LangChain

Decoding Parameters

Fine-tune creativity, length, and randomness in model outputs

✅ **Wrappers = "Universal Adapter" for LLMs.** They make model switching and experimentation smooth, scalable, and developer-friendly.