# Understanding Temperature and Top-P in Large Language Models

## 1. Introduction

When interacting with Large Language Models (LLMs), two parameters—**temperature** and **top_p**—play a major role in shaping how creative, predictable, or diverse the AI's output becomes. During my implementation of the Bedrock-based chat system, I experimented with both parameters to fine-tune the type of responses the model generates.

## 2. What Is Temperature?

### 2.1 Concept

Temperature controls the **level of randomness** in the model's token (word) selection process.

- **Lower temperature (0.0 – 0.3)** makes responses more focused, deterministic, and factual.
- **Medium temperature (0.4 – 0.7)** introduces slight variation without losing coherence.
- **Higher temperature (0.8 – 1.0+)** increases creativity, variability, and unpredictability.

### 2.2 How I Use It

In practice, I choose:

- **Low temperature** when I want the model to be *precise*, especially for technical or knowledge-base-supported answers.
- **Higher temperature** when I need *creative suggestions, brainstorming,* or non-critical conversational output.

## 3. What Is Top-P (Nucleus Sampling)?

### 3.1 Concept

Top_p sets a **probability cutoff** that controls which tokens the model is allowed to consider.

Example:

- **top_p = 0.1** → Model selects from the top 10% most probable tokens.
- **top_p = 1.0** → Model can choose from *all* tokens (maximum diversity).

Unlike temperature—which changes *how* randomness is applied—top_p changes *how wide the selection pool is.*

### 3.2 How I Use It

I adjust top_p depending on how tightly I want the model to follow the most likely responses:

- **Low top_p** for structured, predictable answers (summaries, factual explanations).
- **High top_p** for flexible, more exploratory responses.

## 4. How They Work Together

Temperature and top_p both affect randomness, but in different ways. I treat them like two knobs:

- **Temperature = How creative?**
- **Top_p = How wide is the vocabulary pool?**

I keep both low for strict accuracy, and raise one or both when I want more variety in responses. A well-balanced combination ensures responses remain coherent while still offering some creativity.

Together, these settings give me fine control over how the model behaves based on the task.