Assignment

Course: GenAI Core Essentials for QA Engineers [AI-Powered Testing Mastery]

Topic: Deep Dive into GPT, DeepSeek, Llama

(LLM architecture deeply)

Live Session Date: 8th Feb 2026

Q. 1: From open router ai: https://openrouter.ai/models Context window, input token and output token

- Open AI: 3 models
- Claude: 3 models
- Qwen: 3 models

Service Provider	Model Name	Weekly Tokens	Input ($/1M)	Output ($/1M)	Context window
OpenAI	GPT Audio	5.05M	$2.50	$10	128000
	GPT Audio Mini	65.3M	$0.60	$2.40	128000
	GPT-5.2-Codex	67.4B	$1.75	$14	400000
Anthropic	Claude Sonnet 4.6	76.7B	$3	$15	1000000
	Claude Opus 4.6	655B	$5	$25	1000000
	Claude Opus 4.5	180B	$5	$25	200000
Alibaba	Qwen3.5 Plus 2026-02-15	4.59B	$0.40	$2.40	1000000
	Qwen3.5 397B A17B	8.75B	$0.60	$3.60	262144
	Qwen3 Max Thinking	1.13B	$1.20	$6	262144

Q. 2: Check for the Moderation models in Groq (https://console.groq.com/) Safety / Content Moderation Models in Groq

What is Content Moderation

⊙ **Issue in prompts:**

- User prompts can sometimes include -
  - × **Harmful** content,
  - × **inappropriate** content , or
  - × **policy-violating** content
- This can be used to **exploit models** in production to generate **unsafe content**.

⊙ **Solution:** To address this issue, we can utilize **safeguard models** for content moderation.

⊙ **How:** Content moderation for models involves **detecting** and **filtering** **harmful** or **unwanted** content **in user prompts** and **model responses**.

⊙ **Need:**

- This is essential to ensure safe and responsible use of models.
- By integrating robust content moderation, we can –
  - ✓ **build trust** with users,
  - ✓ **comply** with **regulatory standards**, and
  - ✓ maintain a **safe environment**.

⊙ Groq offers multiple models for content moderation:

Model Name	Provider	Type	Purpose
Safety GPT OSS 20B	**Open AI**	**Policy-following moderation**	Flexible safety classification using custom policy definitions
Llama Prompt Guard 2 (86M)	**Meta**	**Prompt/community guard**	Lightweight detection of unsafe prompt content
Llama Prompt Guard 2 (22M)	**Meta**	**Ultra-light safety guard**	Very minimal safety prompt filtering

---

Q. 3: Qwen / GPT using transformer or MOE?

Both **Qwen** and **GPT** are built on the **Transformer architecture**, but some variants use **Mixture-of-Experts (MoE)** on top of Transformers.

Model Family	Transformer	MoE Used?	Notes
Qwen Dense	✅ Yes	❌ No	Standard decoder Transformer
Qwen-MoE	✅ Yes	✅ Yes	Efficient scaling via expert routing
GPT-3	✅ Yes	❌ No	Fully dense
GPT-4	✅ Yes	Likely	Not officially confirmed

🧠 **Technical Summary**

- **Transformer = Base Architecture**
- **MoE = Scaling Strategy on top of Transformer**
- Both Qwen and GPT fundamentally rely on **self-attention mechanisms**
- MoE improves **compute efficiency per token** while increasing total parameter count

🎯 **Transformer vs MoE — Systems Architecture Perspective**

A. Dense Transformer (Standard GPT-style)

- **Concept:** Every token passes through **all layers and all parameters**.
- **Execution Flow:** Token → Embedding → Self-Attention → FFN → ... (repeat N layers) → Output

B. Mixture of Experts (MoE)

- **Concept:** A gating network routes each token to a **subset of expert FFNs** instead of all FFNs.
- **Execution Flow:** Token → Attention → Router → Top-k Experts → Combine → Next Layer

⊙ **Architectural Visualization**

- **Dense Transformer**

[GPU 1] → Full model execution

- **MoE (Distributed Experts)**

```
                ┌── Expert 1 (GPU A)
Token → Router  ├── Expert 7 (GPU C)
                └── Expert 12 (GPU F)
```

🎯 **Executive-Level Conclusion:** If you are designing:

🏢 **Enterprise AI Platform (Controlled Scale)** → Choose Dense Transformer

☁️ **Hyperscale LLM Service (Millions of requests/sec)** → MoE becomes economically attractive

---

Q.4: List down the Available Models that supports FNN, MoE

Since, in this topic we are studying the **LLM architecture deeply (GPT, DeepSeek, LLaMA),**
we should analyse this from a **model architecture classification perspective** — specifically focusing on:

- **FNN (Feedforward Neural Network based Transformer – Dense models)**
- **MoE (Mixture of Experts models)**

Note: In modern LLM context, "FNN" usually refers to **Dense Transformer models**, where *all parameters are activated for every token*.

🔷 1️⃣ Models Supporting FNN (Dense Transformer Architecture)

These models use a **standard Transformer architecture** where:

- Every token passes through **all layers**
- All parameters are active during inference
- No expert routing mechanism

✅ OpenAI GPT Series (Dense)

- OpenAI **GPT-3**
- OpenAI **GPT-3.5**
- OpenAI **GPT-4 (initial versions)**

👉 Architecture: Decoder-only Transformer (Dense)

✅ Meta LLaMA Series (Dense)

Meta models:

- Meta AI **LLaMA 1**
- Meta AI **LLaMA 2**
- Meta AI **LLaMA 3**

👉 Architecture: Dense Transformer

👉 No expert routing

👉 All parameters active per token

✅ DeepSeek Dense Models

DeepSeek:

- DeepSeek-LLM (Base versions)
- DeepSeek-Coder (Base versions)

👉 Dense architecture (non-MoE versions)

✅ Other Popular Dense LLMs

- Google **PaLM**
- Google **Gemini (Base Dense variants)**
- Anthropic **Claude 1 / 2 (earlier dense models)**

🔷 2️⃣ Models Supporting MoE (Mixture of Experts Architecture)

MoE models:

- Contain multiple "expert" feedforward blocks
- A router selects a small subset of experts per token
- Only some parameters are activated

- Efficient scaling (huge parameter count, lower compute per token)

✅ **GPT-4 (MoE-based versions)**

OpenAI **GPT-4 (later architecture versions)**

👉 Believed to use MoE internally

👉 Massive parameter count

👉 Expert routing per token

(Exact architecture not publicly disclosed, but widely accepted as MoE-based.)

✅ **DeepSeek MoE Models**

DeepSeek:

- **DeepSeek-V2**
- **DeepSeek-V3**
- DeepSeek-MoE

👉 Explicitly MoE architecture

👉 Very large expert count

👉 Efficient inference vs dense scaling

✅ **Mixtral (MoE)**

Mistral AI:

- **Mixtral 8x7B**
- **Mixtral 8x22B**

👉 8 experts per layer

👉 Top-2 routing

👉 High performance with lower compute cost

✅ **Google Switch Transformer**

Google:

- Switch Transformer

👉 One of the first large-scale MoE LLMs

👉 Single expert routing (Switch)

✅ **GLaM (Google MoE)**

Google:

- GLaM (Generalist Language Model)

👉 MoE architecture

👉 Sparse activation

🔷 **Summary Table (For Your Assignment)**

Model Family	Example Models	Architecture Type
GPT-3 / 3.5	GPT-3, GPT-3.5	Dense (FNN-based Transformer)
GPT-4 (newer)	GPT-4 (MoE variant)	MoE
LLaMA 1/2/3	LLaMA series	Dense
DeepSeek	DeepSeek-LLM	Dense
DeepSeek-V2/V3	DeepSeek MoE	MoE
Mixtral	8x7B, 8x22B	MoE

Model Family	Example Models	Architecture Type
Switch Transformer	Google	MoE
GLaM	Google	MoE

🎯 **Architect-Level Insight (Important for QA Engineers)**

From a **testing & inference engineering perspective**:

Dense (FNN)	MoE
Stable latency	Variable latency (expert routing)
All weights active	Sparse activation
Easier deployment	Complex distributed routing
Predictable memory	Higher total params but lower active compute

---