Best Practices for Logging in LLM (GPT-3.5 / GPT-4-O) Production Environment

# 1. System and Model Interaction Logs

Capture logs related to infrastructure (CPU, memory, GPU usage), availability, network latency, as well as user requests and model outputs.

# 2. Error and Performance Logs

Ensure logging of both model and system-level errors. Track token usage, response times, and inference latencies. Ensure alerts are set up for high latency or anomalous behaviors.

# 3. Security Logging

Log authentication events, access control violations, and abnormal behavior patterns. If dealing with sensitive inputs, ensure proper anonymization and encryption.

# 4. Centralized Logging and Visualization

Use centralized logging tools like ELK Stack, Prometheus, or Datadog. These provide scalability and allow real-time log monitoring and visualization.

# 5. Compliance with Regulations

Ensure logging practices align with GDPR, HIPAA, or other regulatory frameworks, especially when handling sensitive data. Anonymize logs where necessary.

# 6. Log Sampling and Retention Policy

Sample logs to prevent excessive volume in high-traffic environments. Define log retention policies to automatically prune older logs.

# 7. Distributed Tracing and Model Versioning

Implement distributed tracing (e.g., OpenTelemetry) for tracking input-output behavior across microservices. Always log model versions and configuration details used during inference.

# 8. Explainability and Bias Detection

For sensitive applications, log model decision-making processes to provide transparency. Additionally, monitor logs for bias and fairness across demographic groups.

# 9. Fallbacks and Retries

Log model retries, fallbacks, and the conditions that triggered them, especially if using a simpler backup model or response.

# 10. Continuous Monitoring and Alerts

Set up alerting mechanisms to track significant performance changes, such as token usage spikes, throughput drops, or error rate increases.