# **SOF/C Tactical OPE Toolkit: Model Training Evaluation**

# **Purpose**

The purpose of the Model Training Evaluation document is to assess, compare, and communicate the performance of the models selected for the Special Operations Forces / Cyber (SOF/C) Operational Preparation of the Environment (OPE) Toolkit. This evaluation ensures that both the Large Language Model (LLM), GPT-40-Mini, and the Vision-Language Model (VLM), GPT-4 Turbo, meet the system's operational requirements for summarization and image analysis. Clear documentation of model behavior contributes to the development of reliable, transparent, and accountable AI-driven systems.

**Training Objective**

The primary training objective for the LLM (GPT-40-Mini) was to produce accurate and concise natural language summaries from user reviews gathered via the Google Maps API. This aligns with the Toolkit’s goal of condensing large volumes of textual data into actionable insights. GPT-40-Mini, a transformer-based model optimized for next-token prediction, was selected based on its balance of cost-efficiency, latency, and quality of output​.

For the VLM (GPT-4 Turbo), the objective was to generate structured, coherent descriptions of images, supporting tasks such as scene understanding and object detection. GPT-4 Turbo was selected due to its superior multimodal processing capabilities, delivering high accuracy, low latency, and scalability for analyzing operational imagery.

**Datasets**

Both models leveraged extensive pretrained datasets managed by OpenAI and were deployed in an inference-only capacity without further fine-tuning.

* **GPT-40-Mini** was pretrained on a diverse dataset including books, articles, web pages, and conversational dialogues. Evaluation involved test data comprising five textual reviews each from five different establishments, sourced via the Google Maps API. This provided a realistic and varied set of textual inputs to assess summarization capabilities.
* **GPT-4 Turbo** was pretrained on a large multimodal dataset combining images, structured data, and text. Local testing used 261 images across four categories (bank interiors, police scenes, restaurant interiors, and security camera stock images) to validate image understanding and structured output generation.

**Details:**

* **Data sources and collection methodology**: Reviews collected via Google Maps API (for LLM); Stock images curated manually for diverse operational scenarios (for VLM), as well as Google Maps images for final testing and validation.
* **Raw data statistics**: Five sets of reviews for five establishments (text) and 261 categorized images (visual data).
* **Train/validation/test splits**: Not applicable; models used pretrained weights, and local testing was conducted using designated evaluation datasets. For the VLM testing, initial runs identified 18 images that consistently failed processing due to content moderation. These 18 images were used as part of the validation testing to ensure prompt engineering was successful. Additionally, 50 images from Google Maps were utilized as final testing data to simulate real-world usage.
* **Data augmentation**: None performed; testing utilized real-world reviews and stock imagery without synthetic modifications to maintain operational realism.

**Training Hyperparameters**

The SOF/C Tactical OPE Toolkit utilizes two distinct OpenAI models: GPT-40-Mini for LLM tasks and GPT-4 Turbo for VLM tasks​. As both models are accessed via Azure OpenAI Services APIs, we did not directly modify traditional training hyperparameters such as learning rate, batch size, or number of epochs. However, several important inference-time parameters were tuned during local testing to optimize application performance:

* **GPT-40-Mini (LLM, text summarization)**: Prompt engineering techniques were applied to improve the relevance and conciseness of review summaries from the Google Maps API. No temperature adjustments or concurrency changes were necessary for LLM operations since the workload involved processing a limited number of text inputs sequentially, ensuring minimal latency (200–500 ms).
* **GPT-4 Turbo (VLM, image analysis)**: Significant parameter optimization was conducted during local tests:  
  + **Temperature** was reduced and incrementally tested from 0.7 to 0.3 to improve the determinism and consistency of image descriptions.
  + **Token limits** were reduced from 300 to 150 tokens per image request to control response size and processing costs.
  + **Concurrency management** was implemented using a limit of 25 concurrent asynchronous requests to balance throughput and minimize rate-limit errors.
  + **Prompt engineering** was employed to refine image captioning tasks, ensuring structured and contextually accurate outputs​.

These application-level adjustments were essential to achieving a balance between cost, processing speed, and output quality, even though the underlying model architectures and weights remained fixed. Testing focused heavily on the VLM pipeline, where batch processing and API optimizations had the most tangible impact on system performance.

**Training Curves**

As both GPT-40-Mini and GPT-4 Turbo were accessed through OpenAI's API services and not fine-tuned locally, traditional training curves (e.g., loss over epochs) are not applicable. Instead, model behavior was evaluated through iterative testing using various parameter configurations. For GPT-40-Mini, comparative outputs were assessed using changes in prompt structure and token limits, revealing that a lower temperature (0.3) produced more focused and deterministic summaries. For GPT-4 Turbo, several rounds of local image analysis testing were performed, adjusting token budgets, temperature, and concurrency settings to optimize speed and output quality. These evaluations served as practical, application-level proxies for traditional training diagnostics.

**Performance Evaluation**

Performance evaluations were conducted separately for the LLM and VLM components of the Toolkit:

* **LLM (GPT-40-Mini)**: Compared against GPT-40 and LLaMA 3.2 1B using five sets of reviews from different establishments. GPT-40-Mini achieved the best balance across three categories—speed, cost, and quality. It produced summaries in 200–500 milliseconds, maintained coherence and relevance in output, and had the lowest cost per request due to minimal token consumption. It was ultimately selected as the preferred model due to this optimal tradeoff.
* **VLM (GPT-4 Turbo)**: Evaluated alongside GPT-4 and Florence-2 using 261 categorized stock images. GPT-4 Turbo outperformed others in structured scene description tasks with the highest combined score in speed, cost, and quality. Its multimodal output was consistent and informative, and testing confirmed minimal errors, with no hallucinated content. Concurrency tuning and prompt refinement further improved its throughput without sacrificing response fidelity.

These evaluations confirmed that the selected models met or exceeded performance requirements under realistic operational conditions.

# **Ablation Studies**

While traditional ablation studies typically involve modifying architectural components, loss functions, or data preprocessing methods, such changes were not possible in this project due to the use of pretrained, API-hosted models. Instead, comparative evaluations were conducted across multiple models to simulate the intent of ablation-style analysis by observing how changes in model size and design affect performance.

For the **LLM**, we compared GPT-40, GPT-40-Mini, and LLaMA 3.2 1B:

* **GPT-40** delivered the highest quality summaries but suffered from increased latency and higher token costs.
* **LLaMA 3.2 1B** was the fastest and most lightweight, but its summarization quality was noticeably lower.
* **GPT-40-Mini** provided the best overall balance of speed, cost, and output coherence, making it the most suitable choice for the review summarization task.

For the **VLM**, ablation-style improvements were achieved by iteratively tuning GPT-4 Turbo’s runtime behavior:

* Lowering **temperature** improved consistency.
* Reducing the **token budget** streamlined outputs and reduced cost.
* Adjusting **concurrency limits** reduced rate-limit and server errors.
* Refining **prompt engineering** improved the structure and informativeness of image captions.

These comparisons and tuning efforts served as practical substitutes for architectural ablation, allowing us to empirically justify model selection and runtime optimization.

**Appendix**

### **Glossary**

* **SOF/C**: Special Operations Forces / Cyber
* **OPE**: Operational Preparation of the Environment
* **LLM**: Large Language Model (GPT-40-Mini)
* **VLM**: Vision Language Model (GPT-4 Turbo)

## **References**

* Llama. (n.d.). *Llama official website*. Retrieved February 18, 2025, from<https://www.llama.com/>
* Microsoft. (n.d.). *Azure OpenAI Service models*. Microsoft Learn. Retrieved February 18, 2025, from<https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models?tabs=global-standard%2Cstandard-chat-completions#gpt-4o-and-gpt-4-turbo>
* Microsoft. (n.d.). *Azure OpenAI Service models.* Retrieved February 18, 2025, from<https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models>
* OpenAI. (n.d.). *GPT-4 Turbo Overview.* Retrieved February 18, 2025, from<https://openai.com/research/gpt-4-turbo>
* OpenAI. (n.d.). Introduction to GPT-4o. OpenAI Cookbook. Retrieved February 18, 2025, from<https://cookbook.openai.com/examples/gpt4o/introduction_to_gpt4o>