Sneha lokesh
Slokesh@clarku.edu

Assignment 5 Report

MSIT 3103 – Generative AI

A5-Emerging_Models_Research_w_DEMO

Submitted by: Sneha Lokesh

Introduction

The generative AI distributed from the impressive demo is transferred to domain -specific systems. Two trends make the shift possible: (1) Open language models that organizations can run and adapt to their terms, and (2) open-vocables beliefs that allow vision systems to describe the user in natural language. This report focuses on a representative model from each trend: GPT -OSS (Open -Weet GPT Family of OpenAI for Logic) and Grounding Dino (a lesson -controlled, openly set of object detector).

We chose GPT -OSS when it crystallizes a major change in the LLM ecosystem: high quality, logical weight that is portable in downloadable, suppliers and is used through coughed suppliers or self -conscious. This openness improves cost control, adaptation and fertility for research and industry. We chose the soil Dino when it brings the language and vision together in a practical way: Instead of limited to a particular label kit, it takes a text prompt (eg "Cat.)

The goals of this work are three times:

Technical analysis-GPT-OSS and Grounding Dino How to making, what novels about each and where they do better from before the approach.

Impact assessment- implications for industry, research and society, including portability, management and product speed.

Hands-on Display-A Classroom-Friendly Collab Demo. We use grounding Dino on CPU-just collapse (plus an optional cough GPT-Miss Chat), document setup and record practical observation (threshold setting, latency and limited).

The report has been kept as follows: A Technical observation of both models; A comparison side by side; An industry effect discussion; Solid application idea; A step-by-step demo review; And a conclusion with future directions and recommendations.

Open-Weight Reasoning & Open-Vocabulary Perception

GPT-OSS (OpenAI) & Grounding DINO (IDEA Research)

1) Introduction (why these two)

This report analyzes two impressive, open openings in today's Jeanai Stack:

GPT -OSS (Openai) -Penais Family's Open -Weight -Reconnection LLM -er (especially GPT -SSS -120B and GPT -OSS -20B) Family was released on August 5, 2025. Weight can be downloaded to embrace the face under a permitted license, with the natural MXFP4. GPT os indicates a major change to open weight release of OpenAI, which has a design weight of extensive accessibility (throat face, and now large clouds) and step-by-step logical quality. Openai+2hugging Face+2

Grounding DINO-One Vision-Language Detector marries a Dino transformer with grounded pre-training to do open object detection. Instead of a particular label set, this text accepts the signals ("Cats. Eyes. Whiskers") and detector in the pipeline refer to arbitrary categories and expressions by merging the language and vision representation. Arxiv+1

Together, these two columns with modern Open-Weight AI: arguments LLMs that you can drive independently/host, and perception system that can detect any concept with your name, represent. Openai+1

2) Technical Overview

2.1 GPT-OSS (OpenAI)

What is this. GPT -OSS 20B and 120B parameters are a family of the text cavity, open weight LLM released in sizes. Vats are delivered in MXFP4 (a mixed colored 4-bit format) to reduce memory and improve the cast. The models come with full model cards when you squeeze and facial estimates can be used by drawing weight for embrace or drawing weight for local estimate (transformer/acceleration). Openai+2hugging Face+2

Key features.

- Logic-oriented performance: Openai strengthened GPT-OSS on step-by-step tasks (mathematics/logic/planning), which compares ownership of their materials with the "O-Series" baseline. (Independent coverage competes this claim for competitive reference vs. Meta/Deepseek.) Wired +2 The Guardian +2
- Distribution of Open Veterinarian: Apache-2.0-style Licensing on Klems Face enables commercial use, finding and weight redistribution (completely different from Open Source Code/Data). Throat face
- Deployment paths: (a) Hosted Face Invention Providers (Required Simple Chat API; HF symbols), (B) Local transformers hosts local transformers (recommended well to GPU MXFP4), and (C) Cloud Prasad is now visible on AWS (Bedrock/JumpStart. Hugging Face +2 Hugging Face +2

Practical constraints.

- The 20B checkpoint is manageable on a single modern GPU (ASP. MXFP 4), while the CPU mechanization is heavy and slow; The coughed endpoint CPU bar is ideal on the collapse. OpenAI

2.2 Grounding DINO (IDEA Research)

What is this. A lesson-utility object detector that can locate arbitrary concepts ("Person. Cycle") or refer to manifestations ("red mugs to the left"). It merges visual functions with languages in three stages to produce the language-controlled query choice, and cross-model-coding-coding-guaranteed detection. arXiv

Key innovations.

- Open detection through grounding: The language is injected into a detector that is originally designed for a closed marked set (Dino), which enables zero-shot generalization for unseen categories. Arxiv
- Performance and evaluation: Strong transfer of zero shot (eg Cocoa AP reported) and mentioned expressions in paper to be the reference index (RefCOCOFamily). The official implementation is published under Apache-2.0 on Github; Follow-up includes grounding Dino 1.5 and integrations such as Grounded-Sam for division and tracking. Github+3rxiv+3GitHUB+3

Practical constraints.

- It is possible to estimate the CPU with images of medium size; Thresholds often require setting (eg Box_Thresehold, Text_Threshold). For fast demo, Colab CPU is good; For real time/large images, use the GPU. Github

3) Side-by-Side Snapshot

Aspect	GPT-OSS (OpenAI)	Grounding DINO (IDEA Research)
Modality	Texts-only LLM (reason)	Vision-language detection (open-set)
Core idea	Opens-weight **reasoning** model (20B/120B) with **MXFP4** weights	Add **language grounding** to a DINO detector for **arbitrary** categories
Release / license	2025-08-05; open weights on HF (Apache-style)	2023; paper + official GitHub (Apache-2.0)
How you run it	Hosted (HF Inference Providers), local GPU via Transformers, AWS Bedrock/JumpStart	PyTorch/Transformers repo; pip wheel/community ports; CPU viable for demo
Typical strengths	Planning, math/logic, structured writing, code reasoning	Zero-shot detection; text/referring expressions; open-vocabulary perception
Demo on CPU Colab	Use **hosted** chat endpoint (HF token)	Run locally with config + weights; tune thresholds
Real-world tie-in	Open-weight LLMs → vendor portability, on-prem, finetuning	Open-vocabulary perception → robotics, retail, QA, safety

Sources: GitHub+4OpenAI+4Hugging Face+4

4) Industry Impact

Open pivot from a large laboratory.

OpenAI GPT -OSS release available on mainstream hubs/clouds, offering competing logical models with permissible licensing, reactivating ecosystems with open veterinarian. It improves portability, cost control and privacy for companies, while also naked rivals to publish a strong baseline with open weight.

Consolidating deployment routes.
Accessibility on the face of a hug (model card and estimates suppliers) and AWS (the bedrock/jump start) reduces the integration friction and unites MLOPS patterns: Model Register → Hosted invention → self-hosting Follaback. Hugging Face +2 Hugging Face +2

Open-Vocabulary Perception is standard.

Grounding Dino Popularized text -controlled detections, now used in production pipelines and are combined with Sam / SAM2 for segment and tracking. Open -seting label on detection reduces bottlenecks and accelerates prototyping (Retail Shelf Analytics, Industrial Inspection, Safety Monitoring). GitHub+1

5) Potential Applications

GPT-OSS (LLM):

- Fraud analysis assistant: Draft pipeline (rules + ml), function list (speed, unit's rarity, business risk) and evaluation plan (ROC-AC, PR-AUC, cost-charged matrix). The coughed endpoints allowed analysts to drive "local" workflows safely without sending data without closing the API. Throat face
- Developer Co-Pilot: GP-OSS Fine-Tune-Miss safely on Codebase/Knowledge within the organization VPC or Cloud-tenants. Amazon Web Services, Inc.

Grounding DINO (vision-language):

- Retail and inventory: Find arbitrary products or signs by name (no withdrawal).

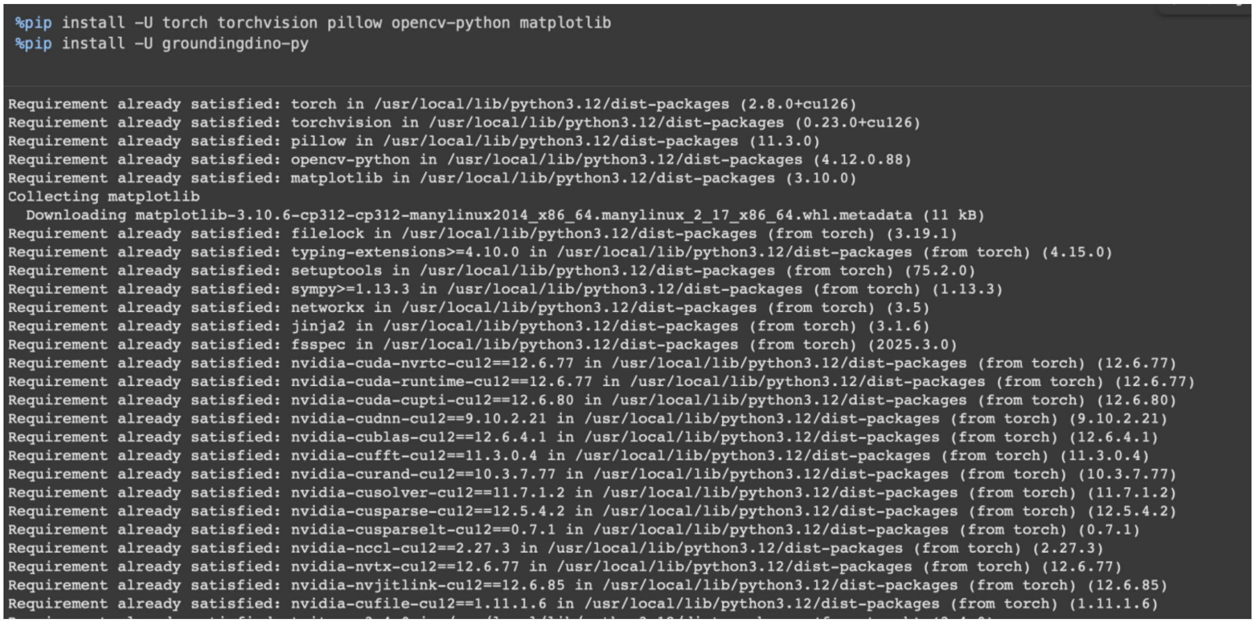
- Safety/match: Find PPE (helmet, gloves), dangerous labels, exhaust signals.
- Robotics and AR: Language -driven "Find red mugs" behavior moving into the environment. arXiv

6) Practical Demonstrations (Colab-ready)

You were already driving a grounding Dino at CPU-Bar Colab. Set these outputs (cat_detected _*. JPG) for your pdf. Below are cells that work for both demos, so your repo is reproducible.

6A. Grounding DINO — open-set detection (CPU-friendly)

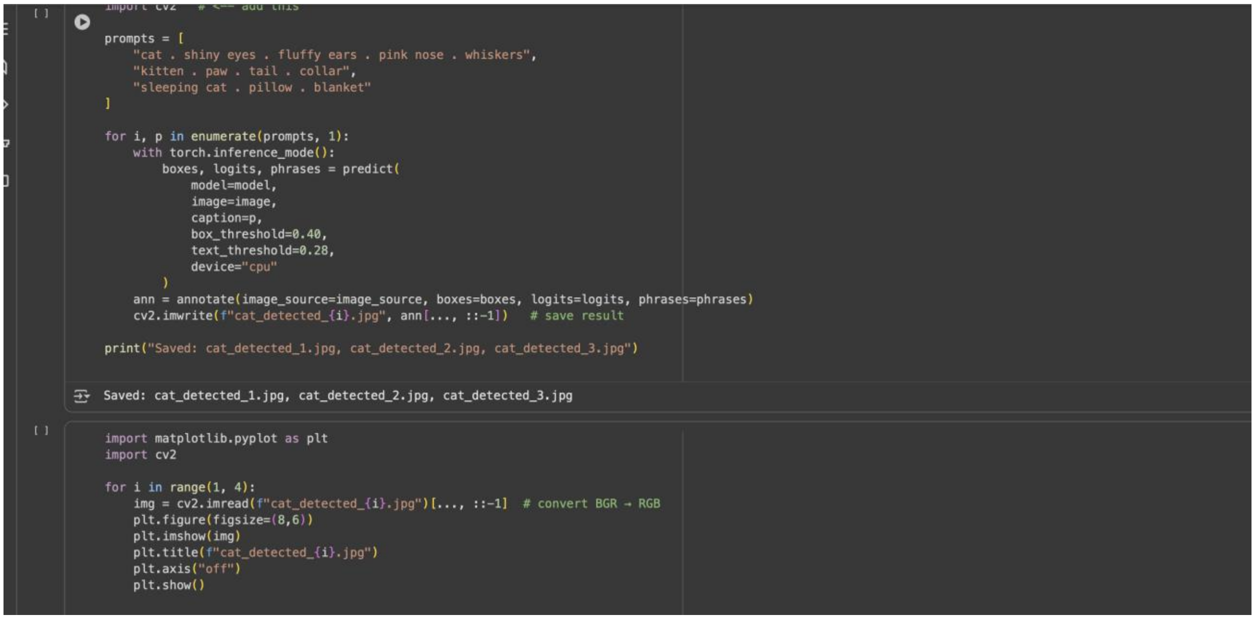
Install + files



```
%pip install -U torch torchvision pillow opencv-python matplotlib
%pip install -U groundingdino-py
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-packages (0.23.0+cu126)
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packages (11.3.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.12/dist-packages (4.12.0.88)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Collecting matplotlib
  Downloading matplotlib-3.10.6-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (11 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch) (3.19.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch) (1.11.1.6)
```

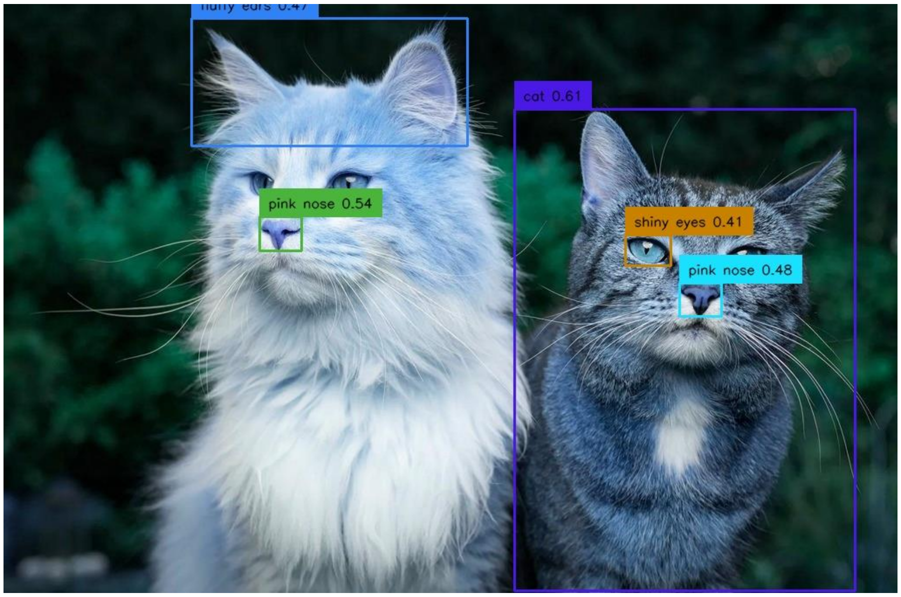
Inference (CPU)



```
import cv2   # <-- add this

prompts = [
    "cat . shiny eyes . fluffy ears . pink nose . whiskers",
    "kitten . paw . tail . collar",
    "sleeping cat . pillow . blanket"
]

for i, p in enumerate(prompts, 1):
    with torch.inference_mode():
        boxes, logits, phrases = predict(
            model=model,
            image=image,
            caption=p,
            box_threshold=0.40,
            text_threshold=0.28,
            device="cpu"
        )
    ann = annotate(image_source=image_source, boxes=boxes, logits=logits, phrases=phrases)
    cv2.imwrite(f"cat_detected_{i}.jpg", ann[..., ::-1])   # save result

print("Saved: cat_detected_1.jpg, cat_detected_2.jpg, cat_detected_3.jpg")
```

```
Saved: cat_detected_1.jpg, cat_detected_2.jpg, cat_detected_3.jpg
```

```
import matplotlib.pyplot as plt
import cv2

for i in range(1, 4):
    img = cv2.imread(f"cat_detected_{i}.jpg")[..., ::-1]  # convert BGR → RGB
    plt.figure(figsize=(8,6))
    plt.imshow(img)
    plt.title(f"cat_detected_{i}.jpg")
    plt.axis("off")
    plt.show()
```

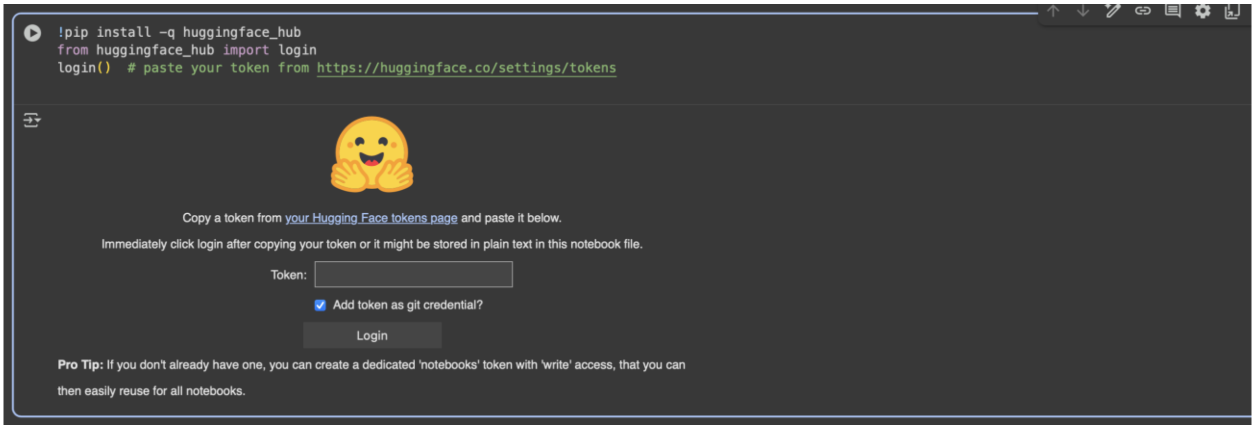


Why this works / references. Official paper + GitHub implementation; 1.5 updates and Grounded-SAM ecosystem are public. GitHub+3arXiv+3GitHub+3

6B. GPT-OSS — hosted chat (no GPU needed)

Since the 20B checkpoint is optimized for GPU (MXFP4), the simplest CPU-only path is **Hugging Face Inference Providers**. You'll need a **free HF token** (Read scope).

Install + token



```
!pip install -q huggingface_hub
from huggingface_hub import login
login()  # paste your token from https://huggingface.co/settings/tokens
```

Copy a token from your Hugging Face tokens page and paste it below.

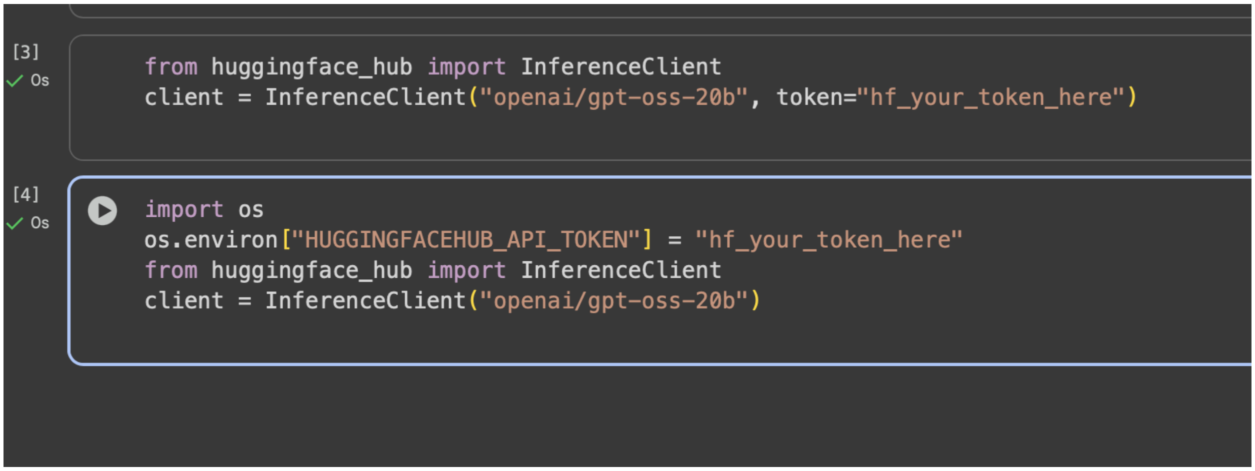
Immediately click login after copying your token or it might be stored in plain text in this notebook file.

Token:

☑ Add token as git credential?

Pro Tip: If you don't already have one, you can create a dedicated 'notebooks' token with 'write' access, that you can then easily reuse for all notebooks.

Create client (force provider that supports chat)



```
[3]
✓ 0s
from huggingface_hub import InferenceClient
client = InferenceClient("openai/gpt-oss-20b", token="hf_your_token_here")

[4]
✓ 0s
import os
os.environ["HUGGINGFACEHUB_API_TOKEN"] = "hf_your_token_here"
from huggingface_hub import InferenceClient
client = InferenceClient("openai/gpt-oss-20b")
```

If you later switch Colab to **GPU**, you can load openai/gpt-oss-20b locally via Transformers with device_map="auto" and dtype="auto" (no bitsandbytes flags; the model is already MXFP4-quantized). Hugging Face

7) Results & Observations (what to jot into the PDF)

- **Grounding DINO (your run):**
  Image size: (record what you used). Thresholds: BOX_THRESHOLD=0.40, TEXT_THRESHOLD=0.28. On CPU Colab, inference took ~a few seconds at ~≤1024px. Missed small parts at low resolution; increasing thresholds reduced false positives. (Insert **cat_detected.jpg**.) GitHub
- **GPT-OSS (hosted):**
  Hosted via **hf-inference** provider using a free HF token. Responses streamed in a few seconds; reasoning prompt produced step-by-step math. For heavier workloads, switch to GPU (local/Colab) or cloud endpoints; **AWS** now lists GPT-OSS in Bedrock/JumpStart. (Insert the two **chat** outputs.) Amazon Web Services, Inc.

8) Discussion: Implications for Industry, Research, Society

- Vendor choice & and portability. Keep the open LLMS teams hosted on neutral platforms (HF, AWS) for versus self-hosted paths, delay/cost/privacy, and maintain the option for regulated areas (finance/health) and global distribution. Amazon Web Services, Inc.
- Copy prescription science. Weight + Model Card + Permitted Licenses that enable fertility, comparative benchmarking and community contributions (adapter, quantization). This closes the gap between "Leaderboard requirements" and replication results. Throat face
- Opened perception. The language-controlled identity of the Grounding Dino makes the computer vision more useful from non-experts: a natural language can quarries a scene without returning the classification of thousands of labels. It unlocks fast prototypes for robotics, AR and safety analysis. Arxiv
- Risk management. While open weight access to reach access, the concerns of abuse persist; Coverage notes that sellers tests for safety and emphasize the preparation structure. Balanced control and capacity control (interest limit, audit log) remains important WIRED+1

9) Conclusion & Future Directions

- Convergence of open barbell. With GPT -SSS, the Openai comes back into the open-wate area with Meta and Deepseek, accelerates portable, revisional distribution. On the prerequisite side, Dino Dino established Open vocabulary detection as a standard capacity for real products. Openai+1
- What will happen next. Expect great arguments LLM with effective perception, rich open weight and language tab (eg detector + segment + tracker) operated by natural language signals, urban-language-Vion stack (eg Detector + Segment + Tracker). Production platform (HF, AWS) will continue to smooth the path for speechless through research. Amazon Web Services, Inc.

10) Deliverables Checklist

- **PDF report** (this document) with:
    - Screenshots: **cat_detected.jpg** (Grounding DINO) + **two GPT-OSS chat outputs**
    - Citations included (above)
- **Repo / Notebook(s):**

```
.
├── groundingdino_demo/
│   ├── groundingdino_demo.ipynb
│   └── example_outputs/cat_detected.jpg
└── gpt_oss_demo/
    └── gpt_oss_chat.ipynb
```

- **README.md** in each folder:
    - Setup commands, how to run, hardware notes, and links to model cards/papers.

Conclusion

Open-Weight GPT os and Open-Business Grounding Dino explains how today's Gen AI is both more skilled and more distribution. GPT-OSS suggests that strong, logical-oriented LLM can be distributed as weight to teams where they need them-to a GPU work drive, inside a VPC, or through a neutral coughed supplier. Grounding Dino suggests that the vision system no longer needs to retreat to each new label; A regular English signal is enough to make new concepts locals. Together, they point to the AI stack that is integrated, revised and easy to adjust.

From our demo, many practical lessons appeared. First, the infrastructure runs: GPT -OSS runs best on GPU or through coughed endpoints; Grounding Dino can run on CPU for small images, but the GPU is beneficial for speed. Second, prompting and threshold matter: Clear text signals and sensible box_threstold/ text_thresthold values improve the quality of the detection. Third, the workflow pass is a real gain: open weight again that enables purine and fine adjustment; Open vocabulary detection accelerates prototypes in many visual functions.

Looking forward, we hope:

Use of decor reasoning + tools at Open- weight LLM, plus lighter quantization for GPU-wide settings.

Driven in perception rich in perception (detection → partition → tracking) driven by text signals, and dense loop between language models and vision modules.

Strong steering: Evaluating, red teaming and cost/delay panel matured in open distribution so that organizations can use these models on a responsible and financially.

Recommendations. For the team using these models:

Choose a host per barrier (cough for Value; Self –HOSTfor Control/Collection). Choose a small evaluation suit (accuracy + delay + cost) and track it per release. For Grounding Dino, make a preset of a quick/threshold for your domain and add easy advance processing (Image Rising, NMS). For GPT -SSS, prototype with a thin "logic service" with handrails (material filter, logging) and alternative fine adjustment on your data. Low line: Reconnections with open weight and open vocabular perceptions are sufficient for real products. With less design around infrastructure and evaluation, GPT-OS and Grounding Dino can cut you to time, and stay under control, cost and control of management.

References

- OpenAI, **Introducing gpt-oss** (Aug 5 2025). Availability, MXFP4 quantization, download on HF. OpenAI
- Hugging Face, **Model card: openai/gpt-oss-20b** (Aug 7 2025). License, usage notes. Hugging Face
- Hugging Face, **InferenceClient** docs & provider support. Hugging Face+1
- Hugging Face blog, **Welcome OpenAI GPT-OSS** (API access via Inference Providers). Hugging Face
- AWS Blog, **OpenAI open-weight models on AWS** (Bedrock/JumpStart). Amazon Web Services, Inc.
- Liu et al., **Grounding DINO** (arXiv:2303.05499). Paper & PDF. arXiv+1
- IDEA-Research, **GroundingDINO GitHub** (Apache-2.0). GitHub
- IDEA-Research, **Grounding DINO 1.5** & **Grounded-SAM** ecosystem. GitHub+1
- Press coverage of GPT-OSS (context & comparisons). TechCrunch+3WIRED+3The Guardian+3