Edge Glass(TBD)

“Towards a Multi-Modal, On-Device Personal Assistant”

(vision + text + audio → interactive assistant)

– Team Member:- Vedaang Chopra

Georgia Tech

How do we design an **open-source, privacy-preserving assistant** that can run **on edge-device** (phone/laptop/glasses), handle **vision + text + audio together**, and provide **interactive outputs** in all modalities?

Outline

- High Level Problem Statement
- Specific Problem Statement
- Proposed Approach
- Key Experiments
- Analysis
- Timeline

High-Level Problem Statement

Goal: Build a lightweight, unified multimodal foundation model that can understand and align text, vision, and audio using frozen open encoders and later a power & memory-aware edge assistant.

PROBLEMS IN CURRENT ASSISTANTS: -

- Modern assistants *can* handle multimodal inputs, but they can't yet do it efficiently, modularly, or locally. Example:- VITA (2024)
- Efficiency gap: Full end-to-end omni models are *too large* for edge or personal deployment.
- Modularity gap: Adding a new modality still means retraining or complex fusion redesign. Example: *LLaVA (2023) → LLaVA-Audio / LLaVA-Video*
- Memory gap: Assistants can't yet recall and reason over past visual/audio/text context seamlessly. Example: - *OpenFlamingo (2023)* and *MiniGPT-4 (2023)*
- Representation gap: Each modality speaks its *own vector language* — preventing unified search, retrieval, and reasoning. Example: *CLIP*, *AudioCLIP*, *ImageBind*

Part-A: - Alignment: Building a Unified Tri-Modal Representation

Low-Level Problem Statement

- **Problem:**
  - Frozen unimodal encoders (ViT for vision, Whisper for audio, MiniLM/Llama-encoder for text) live in separate vector spaces. Example: If “dog photo” ↔ “a dog barking” both relate to the same text caption “a dog,” then their embeddings should also be close in the shared space.
  - To make cross-modal reasoning and retrieval possible, we must align them into one shared embedding space — without retraining the entire model.

- **Open-source examples to cite:** These 3 papers are example of how alignment is being done across different modalities(achieving good results):-
  - *Freeze-Align (CVPR 2025)* — Parameter-efficient image-text alignment with frozen backbones.
  - *ImageBind (Meta 2023)* — Six-modality “image hub” alignment but high compute cost.
  - *AudioCLIP / CLAP* — Partial modality pairing (audio–text only).

Literature Overview: - Unified Encoding (Part-A)

Paper / Model	What it Does (Core Idea)	Limitations / Gaps (What’s Missing)
Freeze-Align (2024)	Keeps strong **unimodal encoders frozen** (like CLIP or DINOv2) and learns **small MLP projectors** to align them, mainly for **image↔text**.	Works only for **one pair (image-text)**. Doesn’t handle **audio** or **tri-modal alignment**. Fails to fix weak **image↔audio** relation.
ImageBind (Meta, 2023)	Trains a large model to **bind six modalities** (image, text, audio, depth, thermal, IMU) into **one shared space**, using **image as the central hub**.	Needs **huge data and compute**. **Encoders are trained end-to-end** (not frozen). Non-hub pairs (like audio↔text) align only **indirectly** → weaker performance.
VALOR (2024)	Builds a **tri-modal (video-audio-text)** model trained end-to-end on **1M videos** with both **contrastive** and **captioning** losses.	Very **heavy and data-hungry**. Designed for **videos**, not easy to adapt or run on **edge devices**. No lightweight path to frozen-encoder alignment.
Unified-IO 2 (2024)	Treats **all modalities as tokens** and trains a single **encoder-decoder** that can **understand & generate** any input/output combination.	A **massive omni-model**, not modular. Needs **joint pretraining from scratch**. No option to use frozen encoders or deploy small pieces.
Chameleon (2024)	Uses **early fusion** of text + image tokens with a unified **transformer decoder**, enabling **joint reasoning and generation**.	End-to-end and **compute-intensive**. No clear **alignment layer** between modalities. Not suitable for **lightweight or frozen setups**.
Matryoshka Multimodal Models (M^3, 2024)	Introduces **nested (small-to-large)** embeddings and **token reduction** for **anytime inference** — trade accuracy vs speed smoothly.	Focuses mostly on **vision-language models**. Doesn’t solve **alignment across different modalities** (like audio↔image).
GRAM (2024)	Proposes a **geometry-aware loss** using **Gramian matrices** to align **multiple modalities together**, beyond pairwise cosine similarity.	The loss is **complex and compute-heavy**. Still requires **joint training**. Not tested on **frozen encoders** or **edge scenarios**.

Proposed Approach

Goal: - It is to build a small, efficient alignment method — using frozen encoders and simple adapter MLPs — that brings **image, text, and audio** into one shared embedding space.

- **Challenge 1:** How to align three modalities (text, image, audio) efficiently without retraining large encoders ?
  - **Our Approach:** Use frozen pretrained encoders with tiny MLP adapter layers that map all modalities into a shared embedding space.
- **Challenge 2:** Maintain semantic alignment while preserving modality-specific details (avoid representation collapse).
  - **Our Approach:** Add a cycle-consistency loss (text - image - audio) so each modality stays distinct but semantically linked.
- **Challenge 3:** Achieve strong cross-modal retrieval across all six directions (t-i, t-a, i-a).
  - **Our Approach:** Train with multi-pair contrastive losses (t-i, t-a, i-a) and shared projection heads to ensure full tri-modal alignment.
- **Challenge 4:** Enable deployment on devices with different compute budgets.
  - **Our Approach:** Incorporate Matryoshka-style variable-width embeddings and token resampling for anytime, budget-aware performance.

Key Experiments Data

Q: What datasets are you using?

- Image–Text: COCO / CC3M / PixMo-Cap (rich human-spoken captions)
- Audio–Text: AudioCaps / Clotho / WavCaps (environmental + everyday sounds)
- Tri-Modal: VGGSound-Cap / YouCook2 / small synthetic (TTS) triplets (5–10 % for I-a alignment)

Together, they provide diverse, semantically aligned samples across vision, audio, and text.

Q: What code / implementation are you using off-the-shelf vs implementing yourself?

- Off-the-shelf (frozen): CLIP / SigLIP (Vision),Whisper (Audio), MiniLM / LLaMA-Enc (Text)
- Implementing: Lightweight adapters + token resampler, contrastive + cycle-consistency losses, Matryoshka heads, full PyTorch training + evaluation pipeline.

Q – How do you define success? What metrics are you using?

- Goal: Unified shared embedding space across all 3 modalities.
- Metrics: Recall@1/5/10, mAP (all six retrieval directions t-i, t-a, i-a), latency, VRAM, robustness.
- Success: Significant increase in i-a Recall@1 vs pairwise baselines with ≤ same compute; t-i and t-a remain competitive.

Q: How are/will you analyze different characteristics of your approach? What ablations will you perform?

- We’ll stress-test quality, efficiency, robustness, and geometry; then isolate causal factors with ablations on losses (cycle/a-v), adapters, token/K, hub choice, data mix, and light tuning/quantization—reporting R@K + latency/VRAM so gains are both accurate and edge-ready.

Part B — Decoder & Model: Making the Assistant Multimodal

Low-Level Problem Statement

- **Problem:** How can we take aligned embeddings (text, image, audio) and design a small, efficient decoder-only assistant that understands all modalities, reasons over them jointly, and runs under tight compute (edge or single-GPU)?

- **Open-source examples to cite:** These 3 models for multimodal models that are for our comparison and use similar techniques as ours (achieving good results):-
  - *Molmo (CVPR 2025)* — Open-source decoder-only VLM (CLIP + LLaMA) trained with PixMo data.
  - BLIP-2 (Salesforce Research) :- Efficient frozen encoder bridge (ViT + LLM via Q-Former).
  - LLaVA / LLaVA-1.5 / LLaVA-Omni :- Vision + text (and later audio) instruction models using small MLP or linear bridges.

What are the key challenges we're facing?

- **Cross-modal alignment gaps** – Image-text or Audio-text work well, but Image-audio remains weak due to limited data
- **High cost of omni-models** – end-to-end training (like Qwen-Omni or VITA)demands massive compute and data.
- **Lack of modularity** – existing models tie encoders and decoders together, making them hard to reuse or deploy on edge.
- **No budget awareness** – current models can't adapt to different hardware constraints (phones vs GPUs).

Literature Overview: - Multimodal Assistant (Part-B)

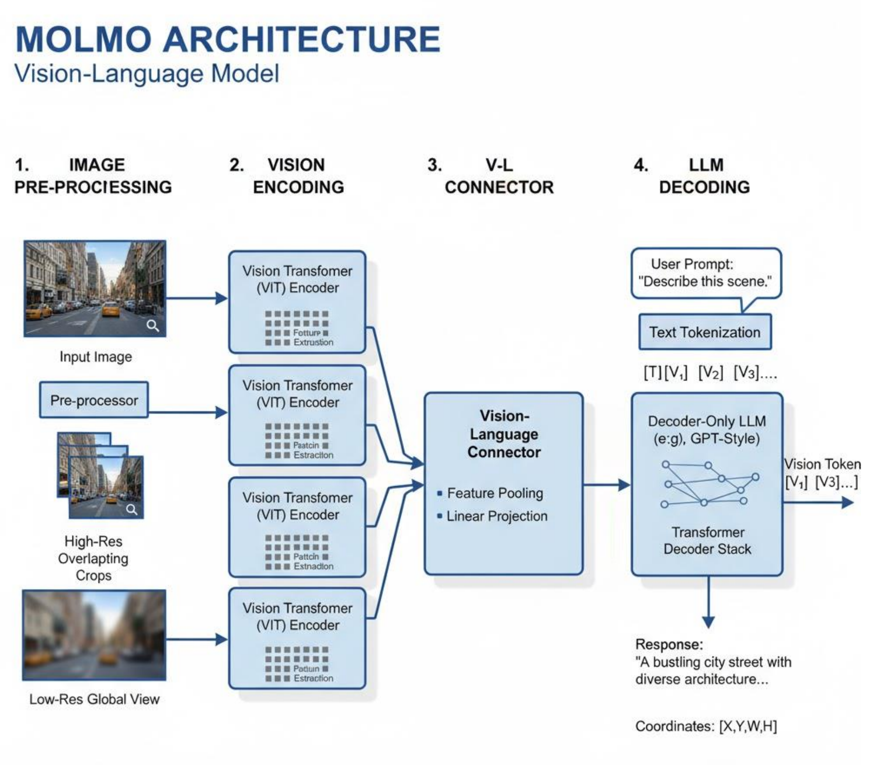
Paper / Model	What it Does (Core Idea)	Limitations / Gaps (What's Missing)
BLIP-2 (Salesforce, 2023)	Connects a **frozen vision encoder** and a **frozen language model** using a **Q-Former bridge**. Learns to translate visual embeddings into LLM-understandable tokens.	Works only on **image + text**, no audio. Doesn't unify modalities — every new modality needs a new bridge. Lacks **budget control** and **tri-modal consistency**.
Qwen2.5-Omni / Qwen3-Omni (2024–25)	A powerful *omni-modal* assistant that understands **text, image, audio, and video**. Uses **Thinker–Talker** architecture for real-time text + speech and **TMRoPE** for precise time alignment across audio/video.	Requires **massive compute and data** for training. Heavy **end-to-end architecture**, not modular. Hard to deploy on edge devices. No clear path for **budget-aware or lightweight usage**.
VITA / Mini-Omni / Mini-Omni-2 (2024)	Real-time **vision–speech–language** assistants. Mini-Omni-2 adds image + audio understanding with **streaming speech output**. Uses **parallel text & audio token generation** for latency reduction.	Excellent latency, but **end-to-end and large**. Needs custom speech codecs. No **frozen-encoder modularity** — hard to adapt or extend. Not designed for **edge or anytime efficiency**.

Literature Overview: - Multimodal Assistant (Part-B)

Paper / Model	What it Does (Core Idea)	Limitations / Gaps (What’s Missing)
Ola (2024)	An **omni-modal LLM** (7B) trained progressively: image → video → audio → speech. Focuses on **cross-modal generation** (e.g., talking videos) with strong temporal modeling.	Very heavy; trained from scratch on massive data. **Couples all modalities tightly**, not modular. Difficult to finetune or export to edge. No **lightweight alignment layer**.
LLaVA-Omni (2024)	Extends LLaVA into **real-time multimodal interaction** (talk + see + hear). Integrates **speech recognition and response** with **visual grounding** using a single decoder.	Focused mainly on **user interaction quality**, not efficient alignment. Still **bi-modal at heart** (image-text + speech I/O). Requires **full retraining** for new modalities.
Molmo (AllenAI, 2024)	A **decoder-only vision-language model** using **frozen ViT + linear projection + LLaMA decoder**. Trained with the **PixMo dataset** for dense captioning and reasoning.	Limited to **image–text** only. No shared embedding or audio support. **End-to-end instruction tuning** is compute-intensive. Not **budget-aware or modular**.

Proposed Approach

Goal: - Design a small, decoder-only model that can read our unified multimodal embeddings and reason across text, image, and audio — efficiently, without end-to-end omni training. *(Kind of like small MOLMO with Audio)*


MOLMO ARCHITECTURE
Vision-Language Model
1. IMAGE PRE-PROCIESSING
2. VISION ENCODING
3. V-L CONNECTOR
4. LLM DECODING
Input Image
Pre-processor
High-Res Overlapping Crops
Low-Res Global View
Vision Transfomer (VIT) Encoder
Fotture Extrustion
Vision Transfomer (VIT) Encoder
Patcin Estraciton
Vision Transfomer (VIT) Encoder
Pattcin Estnadion
Vision Transfomer (VIT) Encoder
Pattuin Estraction
Vision-Language Connector
Feature Pooling
Linear Projection
User Prompt: "Describe this scene."
Text Tokenization
[T][V₁] [V₂] [V3]....
Decoder-Only LLM (e:g), GPT-Style)
Transformer Decoder Stack
Vision Token [V₁] [V3]...]
Response: "A bustling city street with diverse architecture...
Coordinates: [X,Y,W,H]

Q What is our Approach ?

- Keep **encoders frozen**, align all modalities → **shared z-space**.
- Add a **tiny bridge** to connect z-space → decoder:
  – *Prefix-Adapter* (few soft tokens)
  – *Q-Former-Lite* (light cross-attention)
- Train only this bridge + small LoRA on LLM.

Q: How It Differs from other models ?

- No full end-to-end training (unlike Qwen-Omni / LLaVA-Omni).
- Extends **Molmo** idea to include **audio + Matryoshka embeddings**.
- Modular: swap decoders or add new modalities easily.
- Edge-ready and compute-efficient.

Key Experiments Data

Q: How will the datasets be used ?

- **Phase 1:** Train Prefix-Adapter on COCO + AudioCaps (captioning)
- **Phase 2:** Fine-tune on mixed multimodal QA (VQA + Clotho-QA + synthetic triplets)
- **Phase 3:** Instruction-tune with conversational data for assistant behavior

Q: What's our target outcome ?

- Achieve **80–90% of large omni-model accuracy**, with **5–10× lower compute**, modular design, and **smooth anytime performance** across widths/tokens.

Q: What are the benchmarks across omni-models ?

- **Image:** MMBench, MMMU, and ScienceQA — evaluate visual reasoning, grounding, and multimodal question answering.
- **Text:** OpenOrca, ShareGPT4V, and ALLaVA — test instruction following, dialogue, and general reasoning.
- **Audio:** AudioBench, LibriSpeech (WER), and AV-Odyssey — measure speech comprehension, recognition accuracy, and multimodal (audio–visual) understanding.

Q: Which datasets are we using?

- **Image–Text:** COCO Captions, CC3M/CC12M, VQAv2
- **Audio–Text:** AudioCaps, Clotho, WavCaps
- **Tri-Modal (synthetic):** Pair AudioCaps audio + CLIP-retrieved images (filtered by CLIP score + ASR-caption match)
- **Instruction Data:** LLaVA-Instruct, InstructBLIP, Audio-Visual Scene Dialogue

Analysis Literature Review

Part- A: - Alignment

- Freeze-Align shows that keeping strong pretrained models frozen and only training small adapters can achieve near state-of-the-art performance with very low compute and data, making it ideal for edge or lightweight systems.
- ImageBind proves that using images as a central hub allows different modalities (text, audio, depth, etc.) to align naturally, showing that cross-modal understanding can emerge without direct training between all pairs.
- Matryoshka adds the idea of nested embeddings, proving that models can adapt their accuracy vs. speed based on available compute – useful for real-world, dynamic settings.
- Chameleon demonstrates that full early-fusion multimodal models can reason and generate across modalities, but they require huge compute and data, making them best for cloud-scale systems rather than edge devices.

Part-B:- Modelling

- Simpler bridges work best: From BLIP-3, Molmo, and VITA, we learn that small adapters or token samplers can replace complex modules like Q-Former without losing performance — they are faster, easier to train, and scale better.
- Good data beats complex design: Molmo and BLIP-3 showed that using high-quality, diverse, human-collected data gives stronger multimodal alignment than relying on fancy architectures or synthetic datasets.
- Native speech handling is key: Qwen-Omni and VITA proved that directly generating and understanding speech (using codec tokens or small speech decoders) is faster and more natural than using separate ASR or TTS pipelines.
- Unified, modular design wins: All recent Omni models use one central LLM with frozen encoders and lightweight adapters — this setup allows adding new modalities easily while keeping training efficient and performance strong.

Preliminary Exploration -1

Question: Can a pre-trained vision-languagemodel (SigLIP) correctly match images and theircaptions without any training?

Input: We took 12 images and their **human-written captions** from the **PixMo-Cap dataset**.
Each caption describes the image in rich, detailed language.

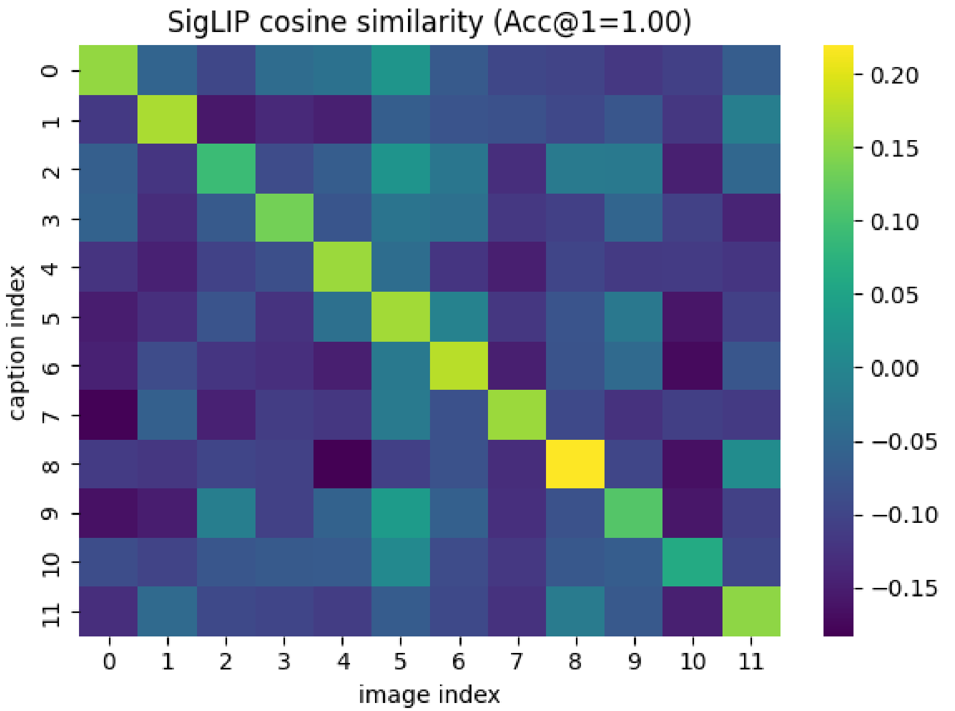
What we did:
We used **SigLIP** (a frozen image–text model) to:

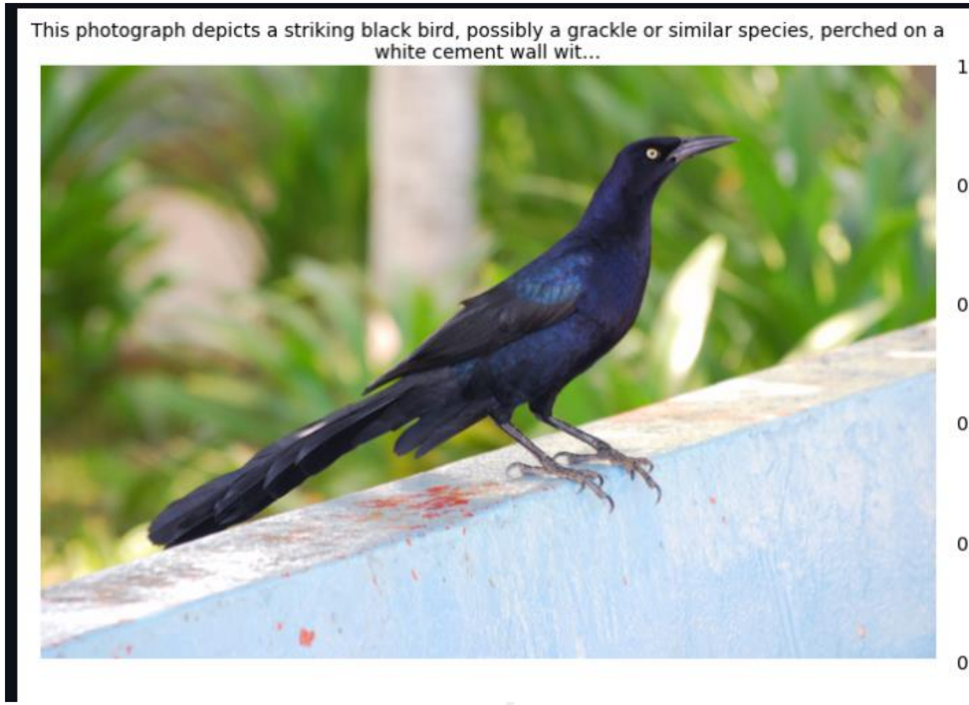
1. Generate **image embeddings** and **text embeddings**.
2. Measure the **cosine similarity** between every image–caption pair.
3. Visualize the results as a **heatmap** and a **top-3 retrieval gallery**.

Output:

- The **heatmap** showed a strong diagonal — meaning each caption matched best with its correct image.
- The **retrieval gallery** confirmed that SigLIP consistently picked the right image for each caption.

Result: SigLIP achieved **100% top-1 accuracy (Acc@1 = 1.00)** on these samples, showing it can understand and align visual and textual information effectively — even on unseen PixMo data.




This photograph depicts a striking black bird, possibly a grackle or similar species, perched on a white cement wall wit...

Ablations Planned

- Part- A: - Alignment
- We will test different image and text encoders (like CLIP and DINOv2) and change the size of the alignment layer to see how these choices affect results.
- We'll look at how well the model matches images with text or audio, and how accuracy changes with compute use when using smaller embeddings.
- We expect that frozen encoders with small adapters can give results close to full models while using much less compute, but we're unsure how well this will work on new, unseen modalities.

Part-B:- Modelling

- What we'll test: We'll compare different connector types(Perceiver sampler vs simple MLP) and different token sizes(32, 64, 128 tokens) to see how they affect accuracy, speed, and memory use on both edge and server setups.
- What we expect: Using more tokens should improve accuracy but make the model slower. The Perceiver connector should work better on detailed images (like OCR),while smaller token sizes should make it faster with only a small drop in performance.
- Early checks: We'll run small tests on image and audio datasets to see how accuracy and latency change with token size and connector type. We expect faster responses with smaller token counts and slightly lower accuracy.

Timeline

Phase	Timeframe	Focus Area	Key Deliverables
Phase 1 — Literature & Setup	Till Oct Mid	• Deep literature review on alignment & omni models • Study encoders (CLIP, Whisper, MiniLM) • Identify datasets (COCO, AudioCaps, PixMo subset) • Reproduce baselines (CLIP, Freeze-Align)	Comprehensive survey summary Baseline reproduction results Problem statement & experiment design slides
Phase 2 — Alignment Prototype (Part A)	Oct Mid → Oct End	• Implement lightweight adapters for text, vision, audio • Train pairwise (t↔i, a↔t) + add consistency (a↔i) Evaluate retrieval across all pairs • Test Matryoshka & token budget variations	Working tri-modal alignment model Retrieval & efficiency plots Ablation report (hub, width, K)
Phase 3 — Decoder & Assistant Integration (Part B)	Nov Early → Mid	• Add small decoder (RAG / prefix / Q-Former-lite) • Test tri-modal QA & assistant behavior • Measure latency on edge hardware	Assistant prototype (text + image + audio) Memory-based retrieval demo Preliminary results slides
Phase 4 — Distillation, Compression & Final Evaluation (Part C)	Nov Mid → End	• Distill unified model to smaller version • Apply quantization (int8/int4) • Benchmark accuracy vs latency • Prepare presentation & final report	Distilled model (edge-ready) Efficiency & accuracy trade-off table Final slides & project report submission