🩺 SurgViVQA-Audio: Audio-Grafted Qwen2-VL for Surgical Video QA

> **Goal:** Engineering a multimodal agent to "hear" live OR audio directly (no ASR) and "see" surgery on consumer hardware (2× RTX 4090).

🛠️ The Engineering Journey

From Proof-of-Concept to Generalization

1. **Feasibility (The "Overfit" Test):**

- **Challenge:** Prove that Whisper audio embeddings could be physically projected into the Qwen2-VL input space without destroying the vision encoder.
- **Method:** Trained on a micro-set (50 samples) until Loss 0.
- **Result:** Validated the "grafting" architecture works; gradients flowed successfully through the custom projector (1,500 audio tokens).

2. **Bias Mitigation (Stratified Scaling):**

- **Challenge:** Medical QA is highly imbalanced (70% of answers are "absent"). Random splits caused the model to collapse into "majority voting."
- **Solution:** Implemented **Question-Type Stratification** to force the model to learn rare classes like `tool_identification` and `blue_dye`.
- **Result:** Achieved **84% accuracy** on safety-critical classes like `occlusion_check`.

3. **Generalization (Held-Out Video):**

- **Challenge:** Prevent the model from memorizing specific patient anatomy.
- **Method:** Evaluated on a completely unseen video (`002-004`) to test true clinical utility.
- **Result: 63.4% Accuracy** (beating the 46% Zero-Shot baseline).

4. **Deployment (Streamlit Demo):**

- **Output:** Built an interactive app with "Flipbook" animation to visualize motion.
- **Latency:** ~1.2 samples/sec (3x faster than traditional ASR pipelines).

📊 Performance Snapshot

Split	Accuracy	Notes
Baseline (Zero-Shot)	46.0%	Audio + Image (Pre-training only)
Stratified Eval (002-001)	**77.6%**	In-domain validation (Seen patient, unseen frames)
Held-Out Test (002-004)	**63.4%**	**True Generalization** (Unseen Patient)

🏗️ Architecture

We bypass the standard ASR (Speech-to-Text) pipeline to reduce latency and privacy risks.

```
graph LR
    A[Audio Waveform] -->|Whisper Large v3| B(Audio Encoder)
    B -->|Projector 1280->3584| C(Audio Tokens)
    D[Surgical Frames] -->|Vision Encoder| E(Visual Tokens)
    C & E --> F[Qwen2-VL Decoder]
    F --> G[Answer]
```

Key Innovation: 1,500 audio tokens are injected directly into the multimodal input sequence.

⚙️ Technical Details

Audio Grafting Strategy

- **Base Model:** Qwen2-VL-7B-Instruct (Vision + Language)
- **Audio Encoder:** Whisper Large v3 Turbo (Frozen)
- **Projector:** Linear Layer (1280 → 3584) trained to map audio features to LLM embedding space.

QLoRA Training Config

- **Precision:** 4-bit Base Model + BF16 Adapters.
- **Rank/Alpha:** r=64, alpha=16.
- **Target Modules:** `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`.
- **Label Masking:** We strictly mask all vision/audio tokens (`-100`), calculating loss **only** on the assistant's text response.

Memory Optimization (RunPod 24GB)

- **Image Resize:** Downsampled to 384x384 (reduces token count by ~70%).
- **Attention:** Used `sdpa` (Scaled Dot Product Attention) for memory efficiency with quantized weights.
- **Gradient Checkpointing:** Enabled (non-reentrant) to fit batch size 1 on 24GB VRAM.

📂 Project Structure

```
SurgViVQA-Audio/
├── src/
│   ├── train_vqa.py           # Main training loop (QLoRA + audio
grafting)
│   ├── app.py                 # Streamlit Demo (Interactive inference)
│   ├── evaluate_checkpoint.py  # Standalone evaluation script
│   ├── create_stratified_split.py # Data stratification logic
│   └── dataset.py             # SurgicalVQADataset class
├── transformers_fork/         # Modified HuggingFace transformers (Audio
support)
├── baselines/                 # Comparison scripts (Text-only, ASR-
pipeline)
├── checkpoints/               # Saved LoRA adapters
```

```
├── scripts/
│   ├── setup_runpod_venv.sh    # Environment bootstrap
│   └── train_002001_stratified.sh # Reproduction script
├── test_set/                   # Small JSONL samples
└── README.md
```

🚀 Quick Start

1. Setup Environment

```
# Clone and setup (using persistent RunPod volume)
git clone https://github.com/your-username/SurgViVQA-Audio
cd SurgViVQA-Audio
source scripts/setup_runpod_venv.sh
```

2. Run the Demo (Streamlit)

Launch the interactive assistant to record audio and analyze surgical frames:

```
streamlit run src/app.py --server.port 8501 --server.address 0.0.0.0
```

3. Reproduction (Training)

To reproduce the stratified training run:

```
./train_002001_stratified.sh
```

Citation

```
@article{abdullah2026surgvivqa,
  title={SurgViVQA-Audio: Audio-Grafted Vision-Language Models for
Surgical Video QA},
  author={Abdullah, Kulsoom},
  journal={GitHub Repository},
  year={2026}
}
```

Acknowledgments

- Base model: Qwen2-VL-7B-Instruct
- Audio encoder: Whisper Large v3 Turbo
- Dataset: EndoVis Challenge

License

Apache 2.0