Vision-Speech Models: Teaching Speech Models to Converse about Images

[Paper] [Github]

1. Model Architecture

MoshiVis augments the Moshi speech LLM with visual processing capabilities through lightweight adaptation modules:

- **Base Model**: Uses Moshi (7B parameters) as the speech backbone and PaliGemma's "stage 2" vision encoder (400M parameters) as the image processor
- **Integration Mechanism**: Adds gated cross-attention layers to each transformer block in Moshi
- **Gating System**: Implements a 2-layer MLP with sigmoid activation that modulates visual information flow, allowing the model to selectively attend to image content
- **Parameter Efficiency**: Freezes both the speech transformer and image encoder weights, training only the adaptation modules (206M parameters)
- **Memory Optimization**: Uses shared cross-attention QKV projection weights across transformer layers to reduce memory footprint

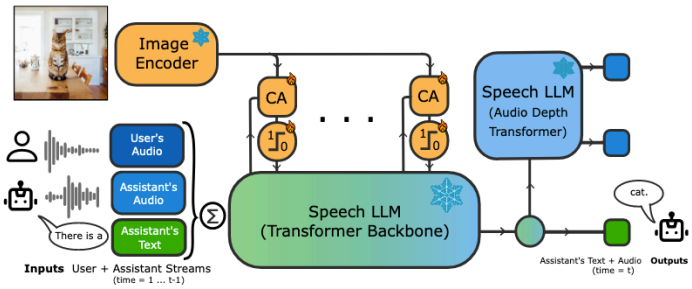

Figure 1. **MoshiVis** is a Vision-Speech model (**VSM**) able to hold full-duplex real-time conversations about an image, and trained with a light data- and compute- budget. For image representations, we use off-the-shelf transformer-based image encoders from the PaliGemma family [5]. For the speech modelling part, we rely on Moshi [8], a recent speech LLM which *jointly* outputs text and audio tokens in real-time, allowing for full-duplex conversations. At its core, Moshi consists of a standard 7B decoder-only transformer taking as inputs *speech tokens* (which are the sums of temporally aligned text tokens and audio tokens extracted from the assistant's and user's streams), rather than only text like a standard LLM. The output of the transformer is then separately decoded in a text token, as well as passed through a small *depth transformer* which auto-regressively produces a hierarchy of audio codebooks, then decoded into audio frames. First, (Sec. 3.1), we detail how we augment the speech LLM's transformer with lightweight visual adaptation modules through cross-attention (CA). We then describe our one-stage finetuning pipeline for these modules: We use a mixture of **(i)** (Sec. 3.2) image+text only data (*"speechless"* data), which, despite incurring a distribution shift due to the lack of audio supervision, allows us to leverage the large body of existing Vision-Language datasets, and **(ii)** (Sec. 3.3) synthetic spoken visual dialogues which we design to mimic realistic discussions about images.

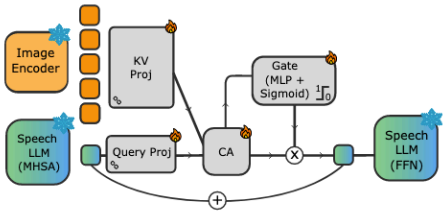

Figure 2. **Adaptation modules.** The image tokens are injected into the current speech token via residual cross-attention (CA) layers, placed between the multi-head self attention (MHSA) and the feedforward network (FFN) in every transformer block. As the cross-attention's QKV projections are shared across layers (🔗), at inference, we only need to compute the keys and values once per image, thus reducing the memory cost needed to store the image embeddings. To enable more context switch, we modulate the output of the cross-attention with a binary gate. The resulting output is fed back into the speech token stream as a residual.

2. Training Process

MoshiVis uses a simple one-stage training pipeline:

- **Mixed-Data Approach**: Combines "speechless" image-text samples with image-speech samples in varying proportions

- **Parameter-Efficient Fine-tuning**: Only adaptation modules are trained while keeping backbone models frozen
- **Ratio Finding**: Experiments showed 25% speech samples and 75% text samples provided the optimal balance between performance and training data requirements
- **Context-Switching Training**: Used dialogue concatenation with unrelated conversations to improve topic-switching abilities
- **Training Efficiency**: 50,000 steps with batch size 64, taking approximately one day on 8 H100 GPUs

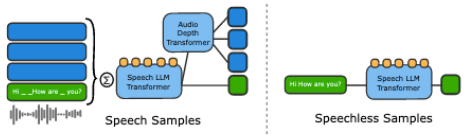

Figure 3. **MoshiVis forward pass during mixed data training**. *Speech samples* are composed of the user's and assistant's audio streams (🟦) and a text stream (🟩) (only for the assistant) containing extra padding tokens (_) to maintain the temporal alignment with speech. The input streams are summed and passed to the transformer. The output audio streams are auto-regressively decoded by a small transformer (Audio Depth Transformer). In practice, *we only train the first two audio streams for speech samples*. This allows for faster training as we need fewer parallel calls to the depth transformer. In contrast, *speechless samples* only contain standard text; in this case, MoshiVis acts as a standard transformer augmented with additional adaptation modules (🟧).

3. Data Used in Each Step

MoshiVis used several types of data in the development process:

- **Image-Text Data**: Leveraged existing VLM datasets for core visual understanding
- **Image-Speech Data**: Used limited speech samples including converted COCO captions
- **Synthetic Visual Dialogues**: Created using a custom dialogue generation pipeline:
  - Used two Mistral-Nemo models in a turn-taking setup (one as user, one as assistant)
  - Both models received image captions as context
  - Generated 8-16 turn conversations covering general questions, detailed inquiries, and misleading questions
  - Converted text dialogues to speech using the same TTS system as Moshi
- **Data Augmentation**: Generated generic spoken dialogues unrelated to images and concatenated them with visual dialogues during training to improve context-switching abilities

4. Novel Contributions and Approaches

Key innovations validated in the research:

- **"Speechless" Data Utilization**: Successfully demonstrated that predominantly text-based training can transfer to the speech domain
- **Gated Cross-Attention**: Proved effective for enabling context switching between visual and non-visual topics
- **Parameter-Efficient Adaptation**: Showed that freezing backbone models and only training lightweight adaptation modules (3% of total parameters) achieved strong performance
- **Mixed-Data Training Strategy**: Confirmed that combining 25% speech samples with 75% text samples provides an optimal balance of performance and training efficiency
- **Real-Time Visual Conversation**: Demonstrated that the adaptation approach maintains the real-time performance of the base model with minimal latency increase

5. Inference in Real-World Use Cases

MoshiVis maintains practical viability for real-world deployment:

- **Latency Performance**: Only 7ms additional latency compared to base Moshi model
  - 51ms per step at conversation start
  - 59ms with a 5-minute context window
  - Well below the 80ms threshold required for real-time conversation
- **Hardware Requirements**: Performs efficiently on both NVIDIA L4 GPU and Apple M4 Pro chip
- **Deployment Options**: Implemented in both Rust and MLX backends for flexibility
- **Practical Applications**: Enables natural spoken conversations about images while maintaining prosodic features and allowing seamless topic transitions

6. Open-Sourced Resources

The researchers have released:

- **Inference Code**: Full implementation available on GitHub (kyutai-labs/moshivis)
- **Image-Speech Evaluation Datasets**: Audio versions of standard vision benchmarks including COCO, VQAv2, and OCR-VQA for evaluating multimodal speech models
- **Training Pipeline**: Details of their parameter-efficient adaptation approach
- **Synthetic Data Generation**: Process for creating realistic visual dialogues

This open-sourced material supports further research in vision-speech models and multimodal conversations.