Multimodal Foundation Models

2025/04/11

NVLM: Open Frontier-Class Multimodal LLMs

Presented by:
Nina Chinnam (fhs9af), Mihika Rao (xsw5kn)

Presentation Outline

- ❖ Multimodal Large Language Models
- ❖ Architectures
- ❖ NVLM-1.0
- ❖ NVLM: Models and Training Methods
- ❖ Training Data
- ❖ Results

Presentation Outline

- ❖ Multimodal Large Language Models
- ❖ Architectures
- ❖ NVLM-1.0
- ❖ NVLM: Models and Training Methods
- ❖ Training Data
- ❖ Results

UNIVERSITY of VIRGINIA

Mihika Rao (xsw5kn)

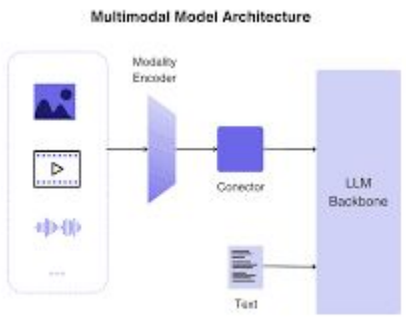
Multimodal Large Language Models

What are Multimodal Large Language Models (MMLMs)?

- **MMLMs** process and understand multiple types of input (text, images, etc.)
- Combine a language model (LLM) with a vision encoder
- Enable capabilities like:
  - Visual question answering
  - OCR (reading text from images)
  - Chart, table, and diagram understanding
  - Math and coding tasks with visual content
- Ex: GPT-4V, Gemini 1.5, NVLM-1.0

Why are MMLMs Important?

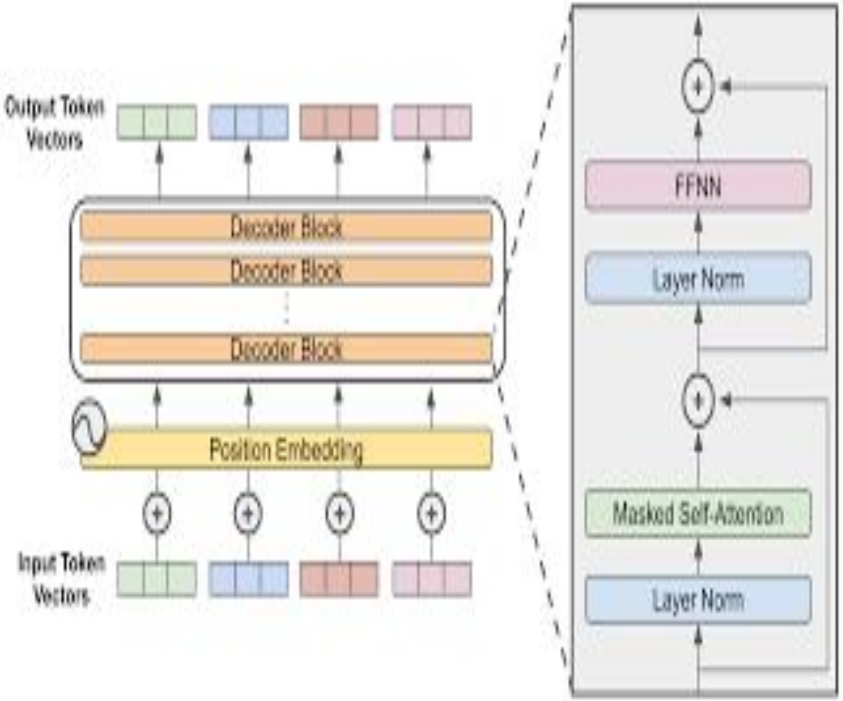
- Combine the strengths of vision + language for richer understanding
- Enable “production-grade multimodality” – strong performance on:
  - Vision-langauge tasks (e.g., OCR, VQA)
  - Text-only tasks (e.g., math, coding)
- Crucial for real-world applications:
  - Document intelligence
  - Assistive AI
- Ex: GPT-4o, Gemini 1.5, NVLM-1.0



Architectures

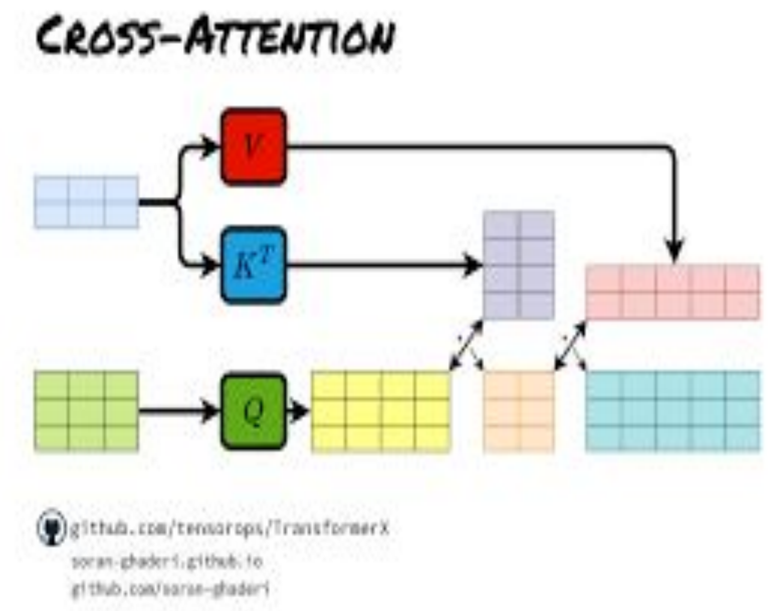
Decoder-Only Architecture

- Image -> Input tokens:
  - Image passed through vision encoder -> image features
  - Image features -> projector (MLP)
  - Projected image vectors + text tokens -> input token vectors



Cross-Attention-Based Architecture

- Source modality:
  - Keys (blue) -> attention scores
  - Values (red) -> content vectors
- Target modality:
  - Queries (Green)
- Q x K^T -> attention scores (orange)
- Attention scores x V -> final cross-attended representation (teal block)



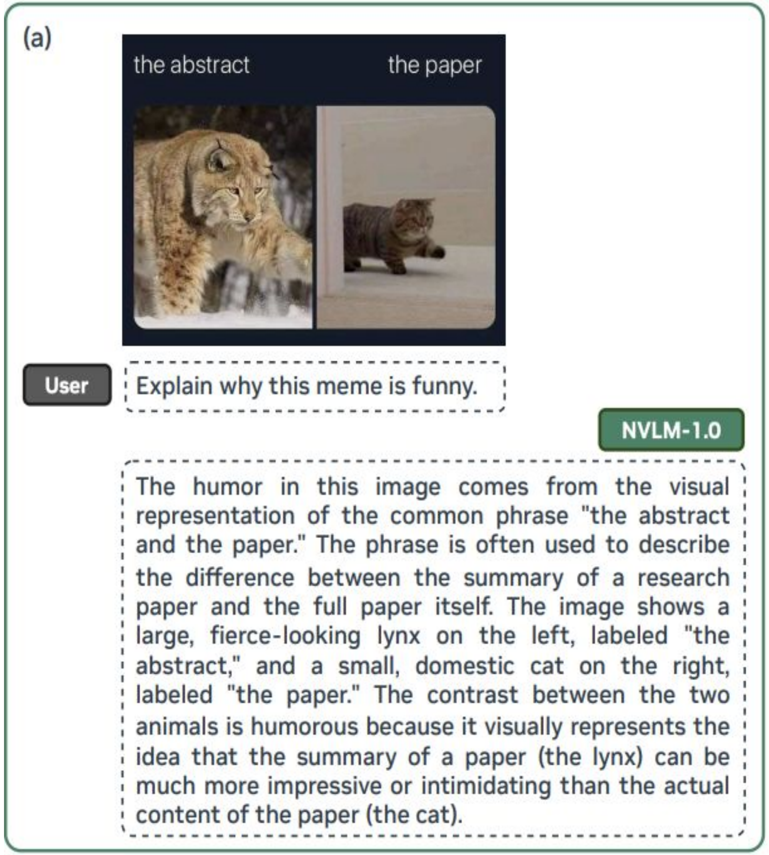
NVLM-1.0

NVLM-1.0

- Developed by NVIDA to explore and improve MMLM architectures
- Includes 3 versions:
    - **NVLM-D:** Decoder-only
    - **NVLM-X:** Cross-attention-based
    - **NVLM-H:** Hybrid of both
- Goals:
    - Compare architectures under equal conditions
    - Improve training efficiency and reasoning ability
    - Support high-resolution image understanding

Qualitative Examples

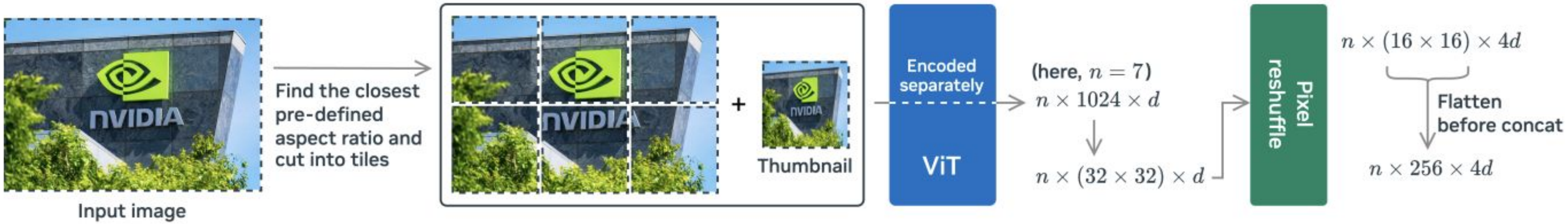
- Demonstrates NVLM's performance on diverse vision-language: meme interpretation, hardware comparison, scene understanding, handwritten code



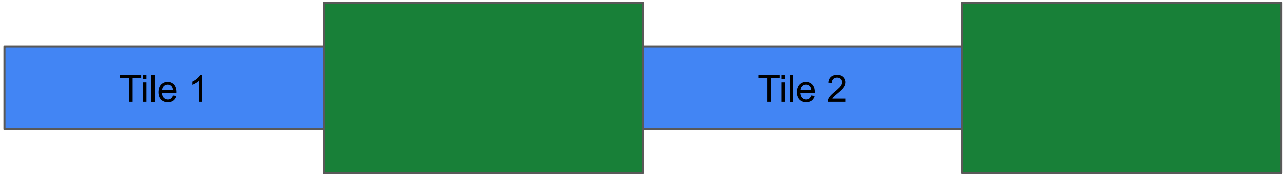
Handling High-Resolution Images Efficiently

- High-res images (e.g. 1024x1024) -> huge token count -> expensive compute
- Standard vision encoder (ViT) trained on small resolutions (224x224)
- **Solution: Dynamic Tiling + Tile Tagging**
  - Split large images into tiles (e.g., 224x224)
  - Feed each tile into vision encoder independently
  - Add 1-D tile tags (e.g., “Tile 1”, “Tile 2”) to preserve spatial context
- Reduces memory usage, improves OCR and visual reasoning

Tile Tagging



- Each tile: 448 x 448
- Patch size by ViT = 14 x 14
- 448/14 = 32 -> 32 x 32 = 1024 patches/tile
- n x 1024 x d -> n x (32 x 32) x d
- n x (32 x 32) x d -> n x (16 x 16) x 4d -> n x 256 x 4d



Nina Chinnam (fhs9af)

Presentation Outline

- ❖ Multimodal Large Language Models
- ❖ Architectures
- ❖ NVLM-1.0
- ❖ NVLM: Models and Training Methods
- ❖ Training Data
- ❖ Results

NVLM: Models and Training Methods

NVLM Models Overview

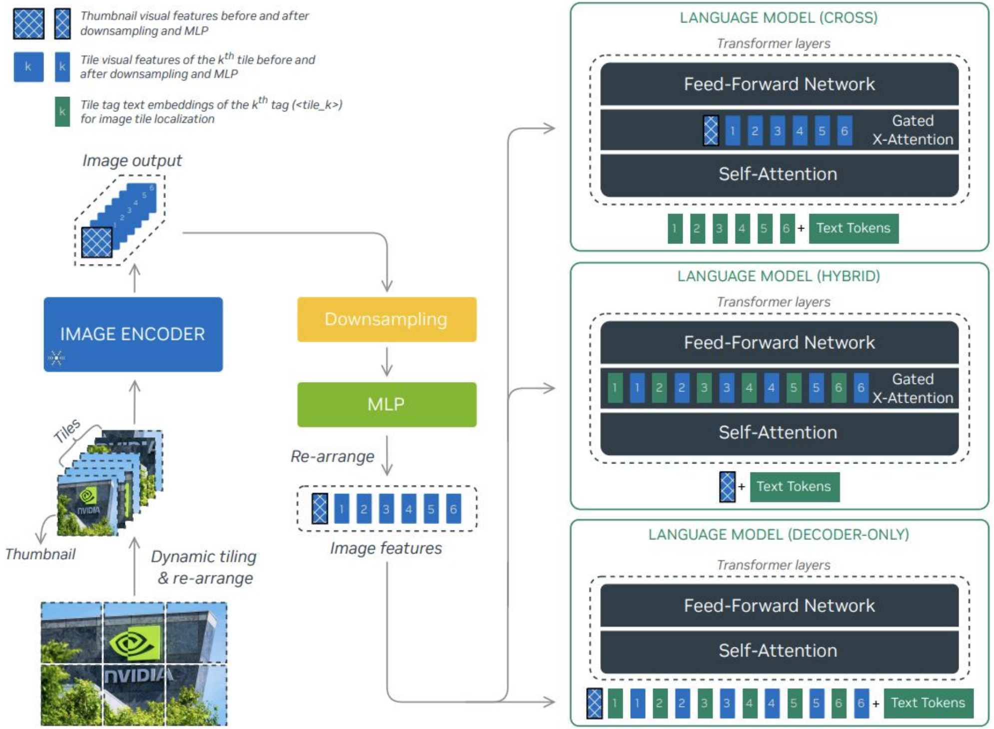


Figure 3: NVLM-1.0 offers three architectural options: the cross-attention-based NVLM-X (top), the hybrid NVLM-H (middle), and the decoder-only NVLM-D (bottom). The dynamic high-resolution vision pathway is shared by all three models. However, different architectures process the image features from thumbnails and regular local tiles in distinct ways.

NVLM-D: Decoder-Only Model

Architecture

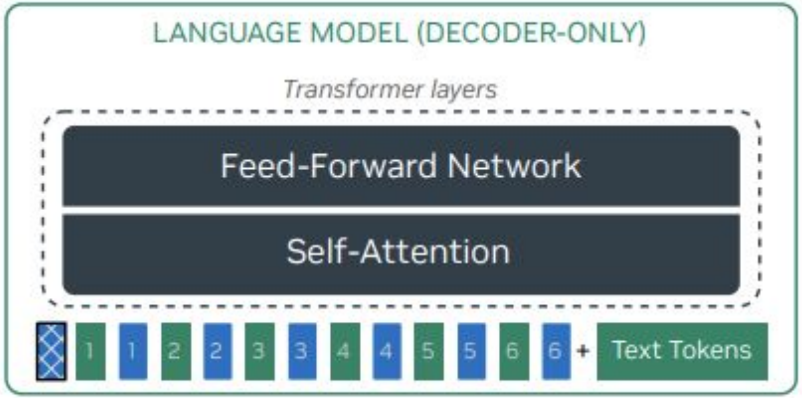
- All Tokens (text + image) go in the decoder
- 2-layer MLP

Training Process

1. **Pretrain** MLP with frozen LLM and encoder
2. **SFT**: Fine-tune LLM + MLP; encoder remains frozen

Tile Tagging

- 1-D tile tags (<tile_1>)



NVLM-X: Cross-Attention Model

Architecture

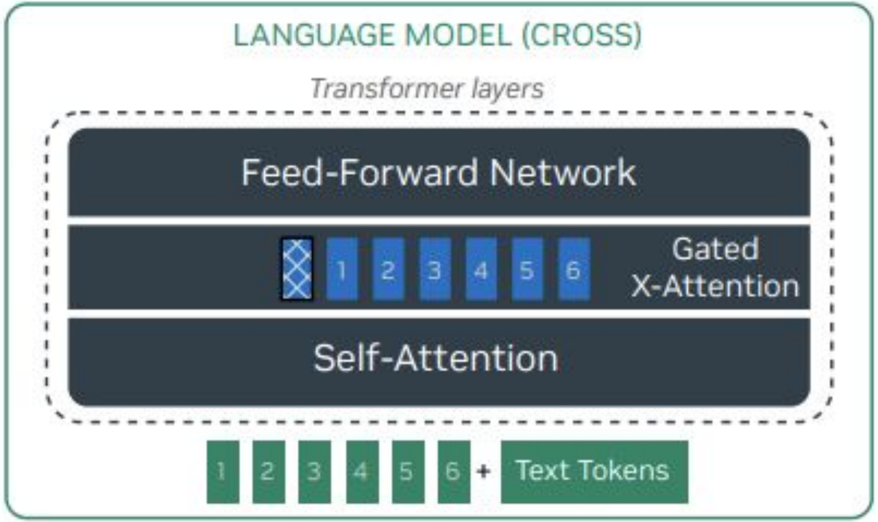
- Gated X-attention layers for visual tokens
- Visual features accessed by layer

Training Process

1. **Pretrain**: X-attn layers with frozen LLM + encoder
2. **SFT**: Unfreeze LLM (better task generalization)

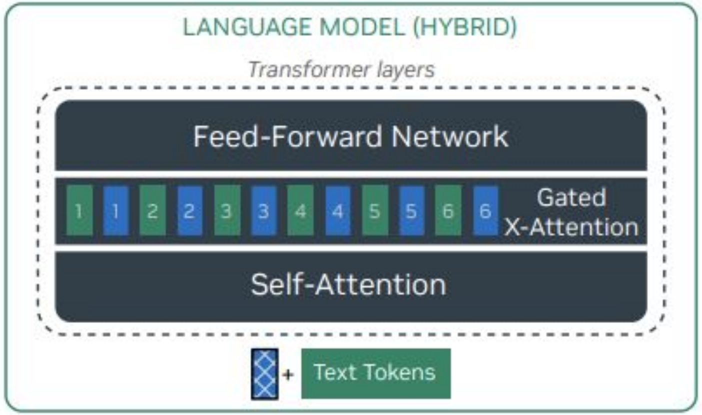
Tile Tagging

- 1-D tile tags (<tile_1>), with X-attention masks



NVLM-H: Hybrid Model

Goal: Combine NVLM-D and NVLM-X



Architecture

- Thumbnail tile: Decoder
- Regular tiles: X-attention layers

Training Process: Same as others

Tile Tagging: Same as others

Comparison of Models

Feature	NVLM-D	NVLM-X	NVLM-H
Visual Integration	Unified in decoder	Modular via X-attn	Hybrid
Input Sequence	Text + Tiles + Thumbnail	Text	Text + Thumbnail
Gated X-Attention	None	All Images	Tile Detail
Efficiency	Slowest	Fastest	Balanced
Sequence Length	Longest	Shortest	Medium
Training Complexity	Simple	Moderate	Most Complex
Best Use Case	OCR, documents	Efficiency-focused	General-Purpose

Training Data

Multimodal Pretraining Data

Goal: Learn vision-language alignment with captioning, OCR, math, and layout data

- Use high quality datasets
- Prioritize quality over scale
- **Ablation result**: curated pretraining data leads to +3.3 point gain

Table 4: Datasets used by NVLM-1.0 at the pretraining stage.

Task	Dataset
Captioning	COCO [72], CC3M [127], SBU [114], LAION-115M (sanitized) [123; 66]
VQA (natural image)	VQAv2 [38], Visual Genome [59]
Chart	DVQA [51]
Document	Docmatix [90]
OCR / Scene-Text	OCR-VQA [98], COCO-Text [144], TextOCR [132], ReCTs [170], RRC-ArT [22], RRC-LSVT [134] RCTW [128], synthdog-en [57], pdfa-eng-wds [117]
Math	CLEVR-Math [73]

Table 5: An ablation study comparing the use of our pretraining data in Table 4 with the pretraining data from LLaVA-1.5 [77] using decoder-only NVLM-D with Yi-34B as the backbone LLM. Both models are trained for 20K iterations with batch size 128 without checkpoint selection to ensure a straightforward comparison.

Pretraining data	MMMU (val)	MathVista	AI2D (test)	ChartQA	DocVQA	TextVQA	OCRBench
LLaVA-1.5 data [77]	51.8	48.9	80.5	80.3	85.2	78.9	760
Our pretraining data	52.0	53.8	82.1	81.1	87.4	79.9	806

Multimodal Supervised Fine-Tuning

Goal: Teach model to follow instructions and solve diverse multimodal tasks

- Covers 8+ task types
  - VQA, captioning, OCR, math, scientific diagrams, charts, tables
- Used GPT-4o to refine responses

ChatML Template Example Utilized in SFT

```
<|im_start|>system
Follow the user's instruction and answer questions.<|im_end|>
<|im_start|>user
<Image><image></Image>
What activity are the people engaged in on the green field?<|im_end|>
<|im_start|>assistant
The people are engaged in playing a game of frisbee on the lush green field.<|im_end|>
<|im_start|>user
Is there a person wearing a fairy outfit in the image? If so, what are they doing?<|im_end|>
<|im_start|>assistant
Yes, there is a person wearing a fairy outfit in the image. They are walking in the field, likely
participating in the fun activities along with the other people.<|im_end|>
<|im_start|>user
What can you say about the spectator's location in relation to the houses?<|im_end|>
<|im_start|>assistant
The spectators are watching a match being played in a location close to the houses, possibly
indicating that the field is within a residential area or park.<|im_end|>
```

Figure 12: An example of the ChatML template used in SFT. The emerald colored <image> tag indicates where to insert visual features. The blue colored text represents the ground truth associated with loss.

Text-Only Supervised Fine-Tuning

Goal: Prevent text degradation and enhance coding/math/text reasoning

- Datasets include ShareGPT, OrcaMath, WizardCoder, CodeFeedback
- Used GPT-4o to refine responses
- Manual filtering

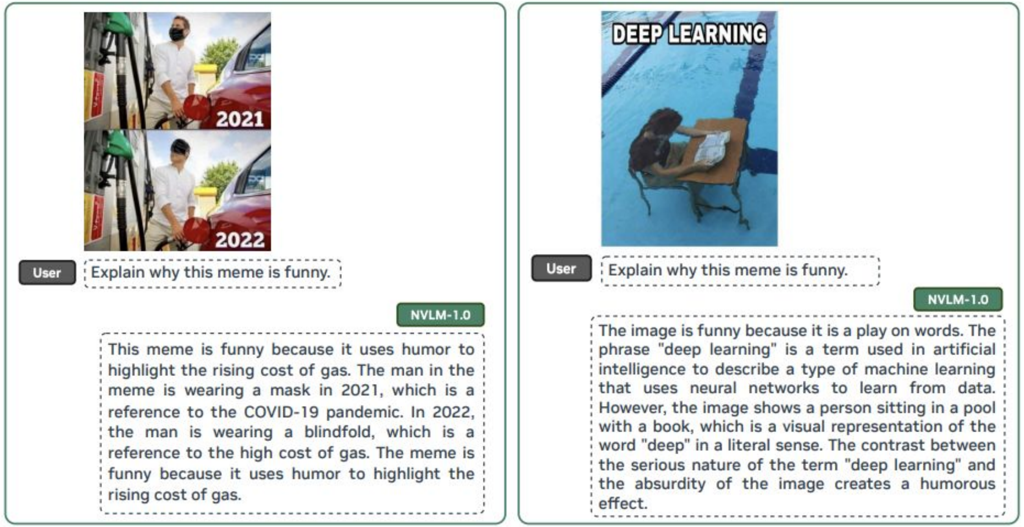


Figure 7: More examples of NVLM-1.0-D 72B model's ability to understand memes, which is a challenging task that requires an understanding of humor and knowledge of important social trends, context, or events.

Results

Key Results Summary

Tasks	MMMU test / val	MathVista testmini	VQAv2 test-dev	AI2D test / *no_mask*	TextVQA val	ChartQA test	DocVQA test	Real-WorldQA	OCR-Bench	**Text-only Avg. 4**
Proprietary										
GPT-4V [107]	**56.1** / 56.8	49.9	77.2	78.2	78.0	78.5	88.4	61.4	645	-
GPT-4-Turbo [106]	- / 63.1	58.1	-	89.4	-	78.1	87.2	-	678	-
GPT-4o [108]	- / **69.1**	63.8	-	94.2	-	85.7	92.8	-	736	-
Claude 3 Sonnet [5]	- / 53.1	47.9	-	88.7	-	81.1	89.5	51.9	646	-
Claude 3 Opus [5]	- / 59.4	50.5	-	88.1	-	80.8	89.3	49.8	694	-
Claude 3.5 Sonnet [6]	- / 68.3	67.7	-	**94.7**	-	**90.8**	**95.2**	-	**788**	-
Gemini Pro 1.0 [35]	- / 47.9	45.2	71.2	73.9	74.6	74.1	88.1	-	659	-
Gemini Ultra 1.0 [35]	- / 59.4	53.0	77.8	79.5	**82.3**	80.8	90.9	-	-	-
Gemini Pro 1.5 [36]	- / 58.5	52.1	80.2	80.3	73.5	81.3	86.5	67.5	-	-
Gemini Pro 1.5 (Aug 2024)	- / 62.2	63.9	**80.2**	94.4	78.7	87.2	93.1	**70.4**	754	-
Grok-1.5V [153]	- / 53.6	52.8	-	88.3	78.1	76.1	85.6	68.7	-	-
Grok-2 [154]	- / 66.1	**69.0**	-	-	-	-	93.6	-	-	-
Others										
QWen-VL-MAX	46.8 / 51.4	51.0	78.8	79.3	79.5	79.8	93.1	-	723	-
Adept Fuyu-Heavy [3]	- / 48.3	-	77.8	81.2	-	75.4	-	-	-	-
Open-access										
LLaVA-Next 34B [80]	44.7 / 51.1	46.5	-	-	69.5	-	-	-	574	-
VILA-1.5 40B [71]	46.9 / 51.9	-	84.3	-	-	-	-	-	-	− 6.9
Cambrian-1 34B [139]	- / 49.7	53.2	-	79.7	76.7	75.6	75.5	67.8	600	-
LLaVA-OneVision 72B [65]	- / 56.8	67.5	-	85.6	-	83.7	91.3	-	-	− 6.3
InternVL-1.2 40B [19]	- / 51.6	47.7	-	79.0	72.5	68.0	57.7	67.5	569	-
InternVL-1.5 26B [18]	- / 45.2	53.5	-	80.7	80.6	83.8	90.9	66.0	724	-
InternVL-2 40B [111]	- / 53.9	63.7	-	87.1	83.0	86.2	93.9	71.8	837	-
InternVL-2-Llama3-76B	- / 55.2	65.5	-	87.6 / 94.8	**84.4**	**88.4**	94.1	**72.2**	839	− 6.7
*InternVL-2-Pro [111]	- / 58.9	66.3	-	87.3 / **96.0**	-	87.1	**95.1**	-	837	-
*Llama 3-V 70B [32]	- / 60.6	-	79.1	93.0	83.4	83.2	92.2	-	-	0
*Llama 3-V 405B [32]	- / 64.5	-	80.2	94.1	84.8	85.8	92.6	-	-	0
NVLM-D 1.0 72B	**54.6 / 59.7**	65.2	**85.4**	85.2 / 94.2	82.1	86.0	92.6	69.7	**853**	+ 4.3
NVLM-X 1.0 72B	53.6 / 57.4	64.6	85.2	84.2 / 93.6	80.2	82.9	82.9	66.1	828	+ 2.5
NVLM-H 1.0 72B	53.0 / **60.2**	**66.6**	85.2	83.8 / 93.3	80.3	83.3	83.1	66.0	831	+ 2.7

Model	Key Strengths
D	OCR, document QA
X	Efficient, high-res handling
H	Reasoning + multimodal general

NVLM Specialties

- Maintains language performance post multimodal training
- Outperforms GPT-4o and Claude 3.5 on key benchmarks
- Models for different priorities
  - D = best OCR
  - X = fastest
  - H = most balanced
- Open contributions

UNIVERSITY of VIRGINIA

LLMs Meet Multimodal Generation and Editing: A Survey

Presented by:
Anisha Patrikar (gjq2yf) and Yagnik Panguluri (yye7pm)

Yagnik Panguluri (yye7pm)

Presentation Outline

- Why Multimodal Generation Matters
- Scope and Contribution
- Foundation of Generative Models
- Aligning Modalities
- Modalities with LLMs
- From Text to Image
- Image Edition with LLMs
- How LLMs Improve Image Generation

Why Multimodal Generation Matters

Human interaction with the world is inherently multimodal: vision, language, audio, and spatial perception. To simulate or generate real-world experiences, AI must process and generate multiple modalities, not just text.

- **Large Language Models (LLMs)** like GPT-4 and LLaMA have revolutionized text generation and understanding
- There's a growing shift toward extending LLMs beyond text to unify generation across **images, videos, 3D content, and audio**
- **Example:** Sora by OpenAI - a recent foundation model capable of generating realistic videos - yet it lacks multimodal comprehension or output in other domains like 3D or audio

Prompt: "Give me an image of two UVA students giving a presentation outside Rice Hall (UVA)

Advancements in diffusion models, alignment models (e.g., CLIP, T5), and training on massive datasets have enabled breakthroughs in multimodal generative AI

Scope and Contribution of the Survey

Scope

Focused on language-guided multimodal generation and editing

Images

Videos

Audio

3D Content

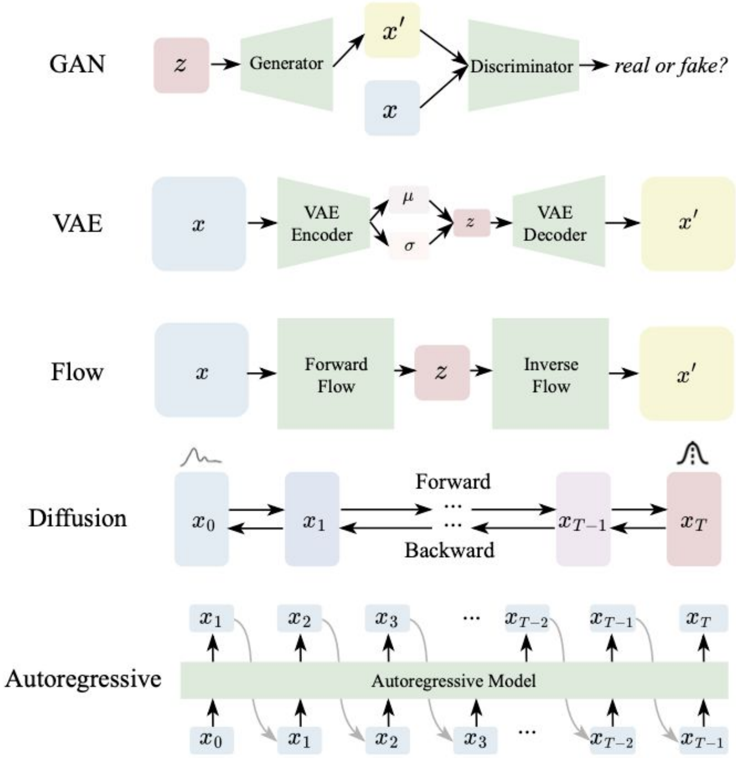
Emphasis on open-domain generation and LLM-based methods

Contribution

- **First comprehensive survey** on LLMs applied to multimodal generation and editing (beyond understanding)
- **Categorizes methods** based on LLM roles (e.g., planner, evaluator, labeler, backbone)
- **Compares pre-LLM vs. post-LLM** generative techniques
- **Explores AI safety**, emergent applications, and future prospects of LLM-driven generative systems

Foundations of Generative Models

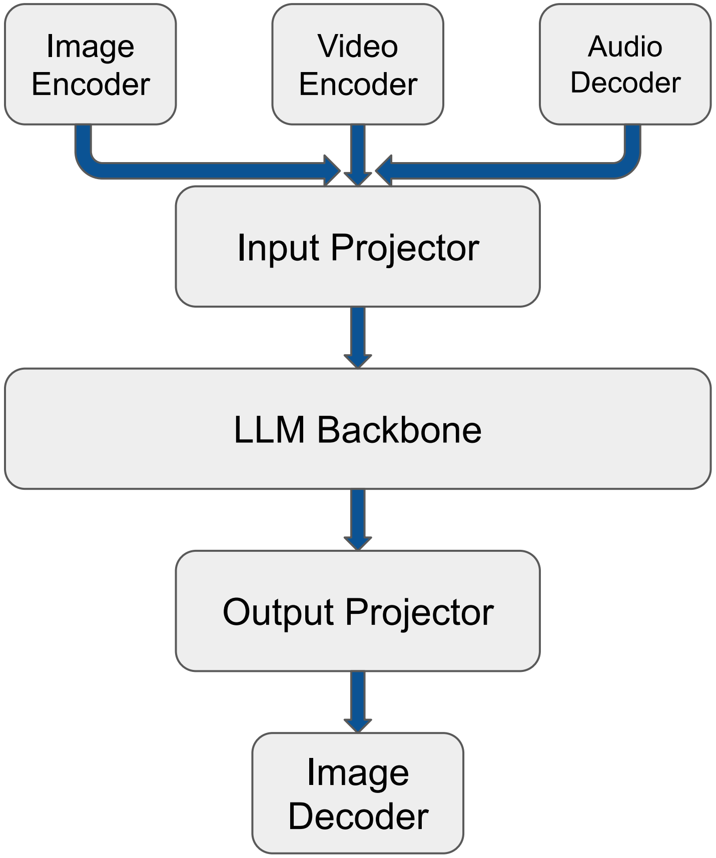
Generative models are those that can learn to generate data (images, audio, etc.) resembling a target distribution



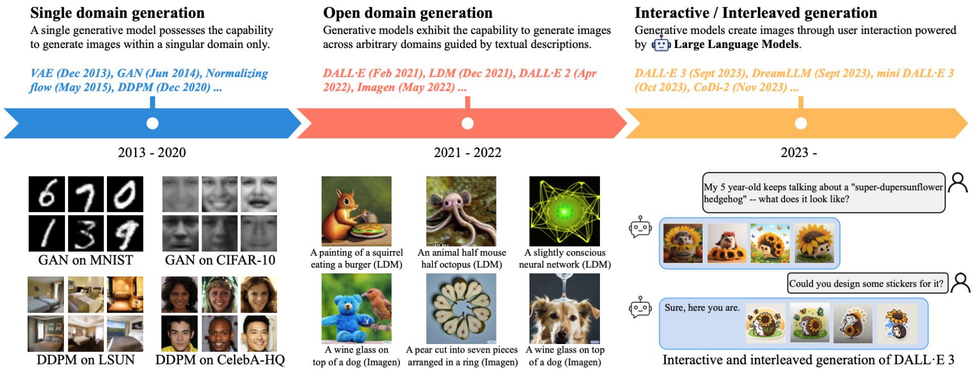
- **Generative Adversarial Networks (GANs)**
  - Two-player game: Generator vs. Discriminator
  - Strength: sharp images; Weakness: blurry outputs
- **Variational Autoencoders (VAEs)**
  - Encoder-decoder architecture
  - Strength: latent space structure; Weakness: blurry outputs
- **Flow-based Models**
  - Invertible transformations with exact likelihood
  - Strength: exact sampling; Weakness: less expensive
- **Diffusion Models**
  - Add noise -> learn to denoise
  - Strength: stability and high-quality outputs
- **Autoregressive Models**
  - Predict next token/frame given the past
  - Strength: good for text/audio sequences; Weakness: slow generation

Architecture of Multimodal LLMs (MLLMs)

- **What are MLLMs?**
  - LLMs extend to handle text, image, video, and audio
  - Use modality encoders/decoders and token projectors
  - Enable generation + editing across modalities
- **Core Components**
  - LLM Backbone: Central reasoning engine (GPT4, LLaMa)
  - Modality Encoders/Decoders: Specialized for vision/audio/etc.
  - Input/Output Projectors: Bridge modality tokens with LLM tokens
- **Example**
  - A user gives a prompt like “Edit this image to make it sunny.”



From Text to Image - Evolution of T2I



Early Image Generation (2013-2020)

- Focused on single-domain outputs (faces, bedrooms, etc.)
- Powered by GANs, VAEs, and early diffusion models
- Limited to fixed categories and styles

Open Domain Generation (2021-2022)

- Introduction of text-guided generation (e.g., DALL-E, Latent Diffusion Models)
- Leveraged CLIP-style alignment for better prompt-image correction
- Enabled diverse, creative outputs from natural language

LLM-Powered Interactive Generation (2023-Present)

- LLMs improve prompt synthesis, layout planning, dialog-based generation
- Enable interactive, multi-round image creation (e.g. DALL-E, DreamLLM, SEED-LLaMA)
- Use MLLMs to unify image and text generation

Image Editing with LLMs

Prompt-to-Edit
Edit the image to change the car to a convertible

→

Instruction-based
Replace the car with a convertible. Also, add a picnic basket on the grass

LLMs enable instruction following: edit images based on natural language commands

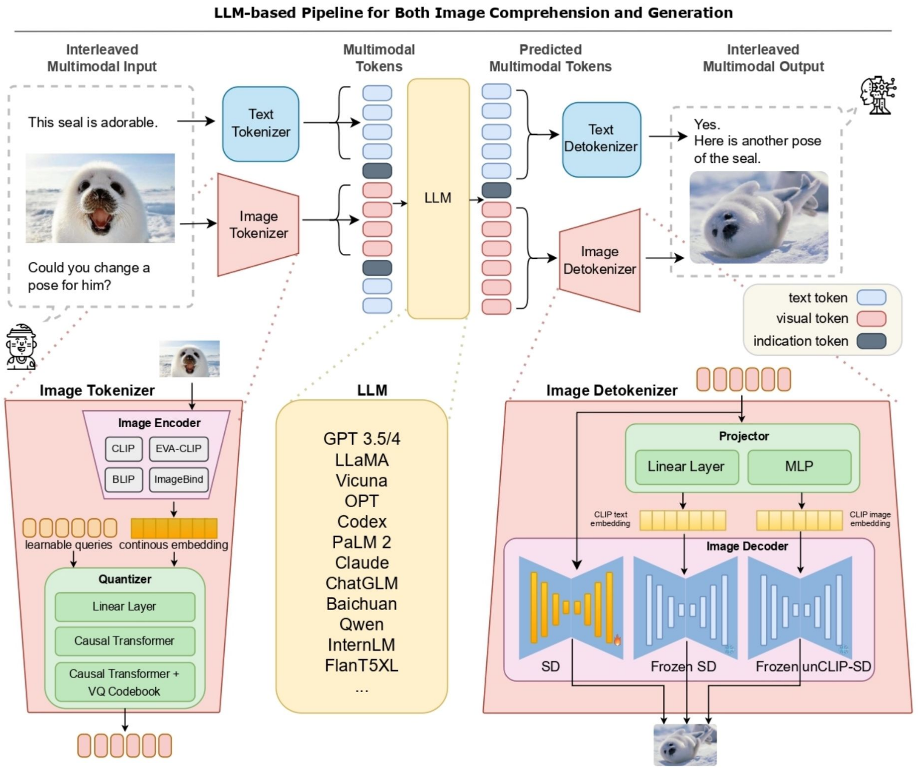
Support multi-turn interactions, detailed requests, and contextual reasoning

Benefits

- Better semantic understanding
- Fine-grained control over regions and objects
- More human-aligned editing workflows

How LLMs Improve Image Generation

1. **Layout Planning**
   a. LLMs reason about spatial relationships
   b. Generate structured layouts
2. **Prompt Synthesis and Refinement**
   a. Generate or enhance detailed and expressive prompts
   b. Improve alignment with user intent and model quality
3. **Image Quality Evaluation**
   a. LLMs assess semantic alignment and aesthetic quality
   b. Used to score outputs or fine-tune generation models
4. **Iterative Generation and Feedback**
   a. LLMs suggest layout or prompt changes based on model output
   b. Enable interactive refinement loops



Anisha Patrikar (gjq2yf)

Presentation Outline

- ❖ Generating and Editing Videos with LLMs
- ❖ 3D Generation and Editing with Language
- ❖ LLMs for Audio, Music, and Speech
- ❖ Rise of Multimodal Generative Agents
- ❖ Generative AI Safety Challenges
- ❖ Future Prospects
- ❖ Summary and Takeaways

Generating Videos with LLMs

Multimodal LLMs for Video Generation:

- **VideoPoet:** Generates consistent videos using autoregressive transformers and multimodal data
- **MAGVIT-v2:** Tokenizes videos to improve compression and generation quality

Video Layout Planning with LLMs:

- **VideoDirectorGPT:** Plans video scenes by generating bounding boxes for object placement
- **FlowZero:** Creates dynamic scene syntax, enhancing motion and spatial consistency

Temporal Prompt Generation:

- **DirecT2V:** Converts a single prompt into frame-by-frame instructions for coherent storytelling
- **VideoDrafter:** Develops multi-scene scripts for logically connected videos

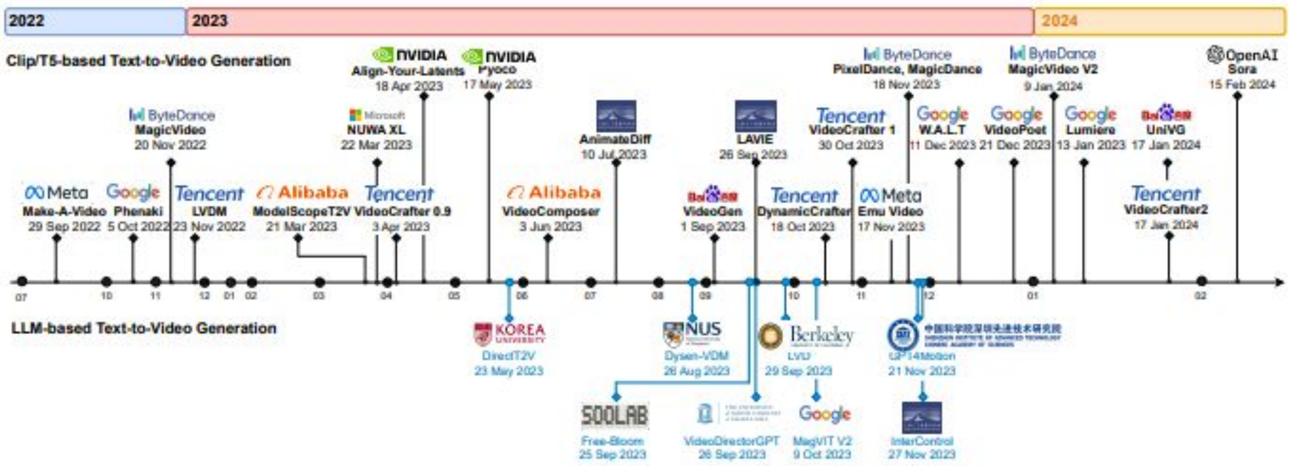


Fig. 6: Milestone works of Clip/T5-based and LLM-based language-guided video generation.

Editing Videos with Language

CLIP/T5-Based Editing:

- **Tune-A-Video:** Uses inflated text-to-image diffusion models; struggles with temporal consistency
- **Video-P2P & FateZero:** Enhance consistency via advanced inversion techniques and attention map manipulation
- **Pix2Video:** Improves temporal stability by editing the key frame first and propagating changes
- **Rerender-A-Video & CoDeF:** Focus on video-to-video style transfer using optical-flow regularization and deformable content fields

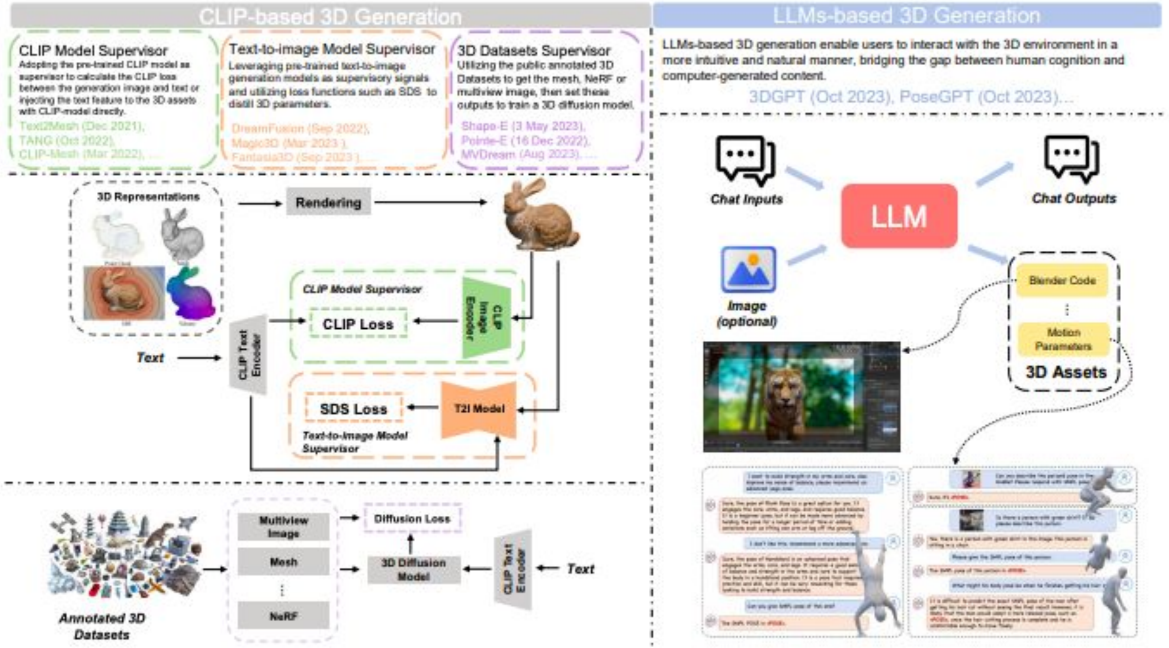
LLM-Based Editing:

- **InstructVid2Vid:** Trains on LLM-generated video-instruction pairs for complex edits like style transfer and object replacement
- **InsV2V:** Uses LLMs for diverse video-instruction pairs, enabling high-quality, coherent editing

Method	Venue	LLM	Generative Model	Training
CLIP for video editing				
Tune-A-Video [255]	ICCV 2023	-	SD	✓
Dreamix [256]	arXiv 2023	-	Imagen-video	✓
Video-P2P [257]	arXiv 2023	-	SD	✓
FateZero [258]	ICCV 2023	-	SD	✗
Pix2Video [259]	ICCV 2023	-	SD	✗
StableVideo [260]	ICCV 2023	-	SD	✗
Rerender-A-Video [261]	SIGGRAPH Asia 2023	-	SD	✓
TokenFlow [262]	ICLR 2024	-	SD	✗
CoDeF [263]	CVPR 2024	-	SD	✓
MagicEdit [264]	arXiv 2023	-	SD	✓
MagicStick [265]	arXiv 2023	-	SD	✓
LLMs for language-based video editing				
InstructV2V [266]	ICLR 2024	GPT-3	SD	✗
InstructVid2Vid [267]	arXiv 2023	GPT-3	SD	✗

3D Generation and Editing with Language

- Natural language enables intuitive 3D content creation and editing
- LLMs generate 3D scenes, assets, or motion directly from text
    - *e.g., 3D-GPT, SceneCraft, MotionGPT*
- LLMs support iterative, interactive editing without retraining
- CLIP/T5-based methods guide geometry and texture via contrastive alignment
- LLM pipelines convert text into Blender code, motion tokens, or point cloud queries
- Enables non-experts to create in AR/VR, animation, and design workflows



LLMs for Audio, Music, and Speech

- **LLMs enable generation, understanding, and editing across audio, music, and speech**
- **General audio:** classification, captioning, and sound event recognition (e.g., SALMONN, AudioGPT, LTU)
- **Music:** generation and interpretation of melody, harmony, lyrics, and rhythm (e.g., MusicGen, MusicLM, ChatMusician)
- **Speech:** improved recognition, transcription, and TTS with contextual awareness (e.g., SpeechGPT, AudioPaLM, VALL-E)
- **LLMs serve as backbones, conditioners, agents, or labelers across tasks**
- **Unifies multimodal processing across sound, language, and interaction**

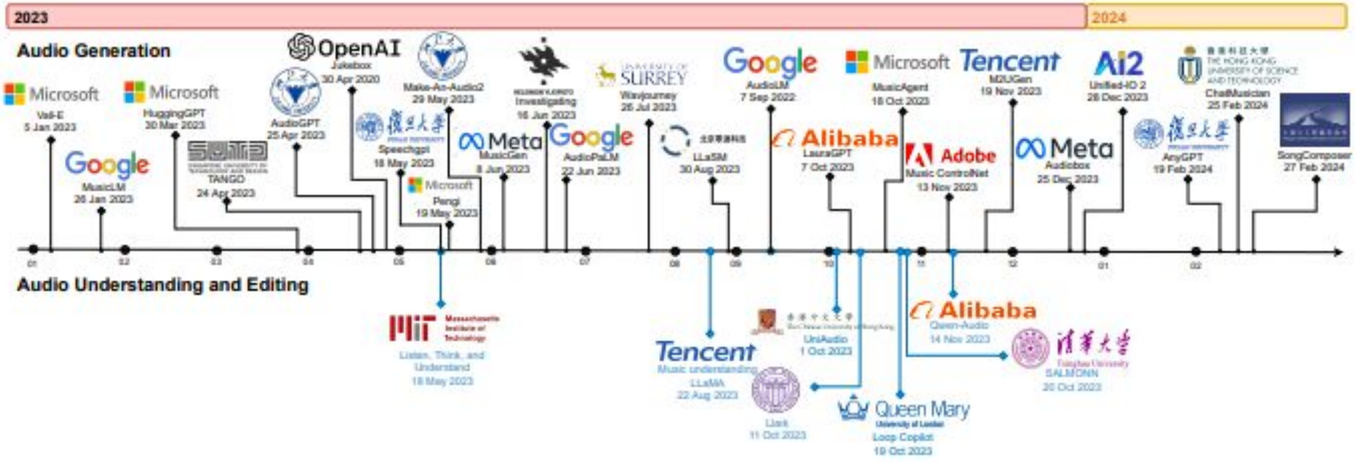


Fig. 8: Milestone works of LLMs-based audio research, including audio generation, understanding, and editing.

Rise of Multimodal Generative Agents

- **LLMs now act as controllers for multimodal generation and editing**
- **Agents use external tools to handle images, videos, audio, and speech**
- **Support natural language prompts for planning, execution, and response**
- **Two major approaches:**
  - *Training-free (prompt-based)*
  - *Instruction-tuned (finetuned on task datasets)*
- **Enable complex, interactive tasks like dubbing videos, generating music, and editing scenes**
- **Challenges remain in tool orchestration and generalization**

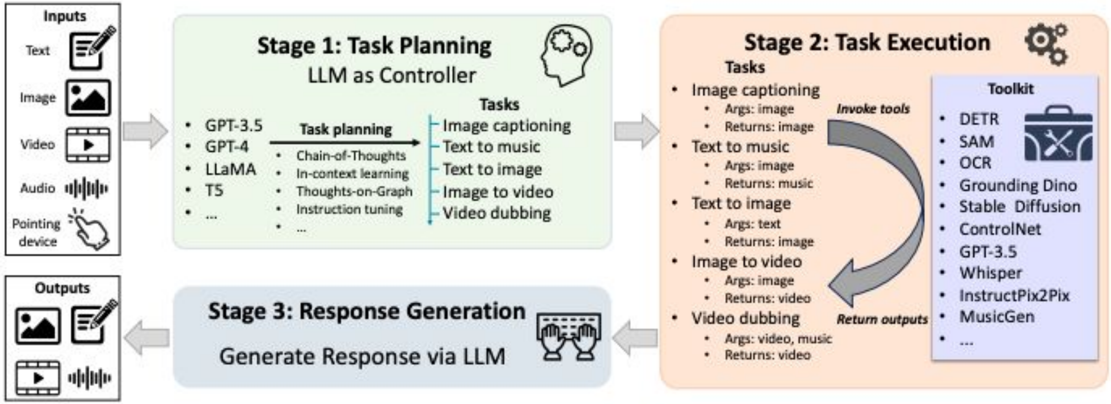


Fig. 10: The pipeline of tool-augmented multimodal agents.

Generative AI Safety Challenges

- **Risks include harmful content, misinformation, and copyright violations**
- **Attack vectors**: adversarial prompts, optimization attacks, data poisoning
- **Defenses**: prompt filtering, alignment with human values, latent safety checks
- **Emerging tools**: watermarking, data attribution for copyright tracing
- **Evaluation datasets** benchmark model safety across modalities

TABLE 12: Overview of generative AI safety across various modalities and methods. The term "Adv." denotes "adversarial attack".

Name	Media	Type	Method	Venue
Wallace et al. [442]	T	Attack	Adv.	EMNLP 2019
Fu et al. [443]	T	Attack	Adv.	arXiv 2023
Image hijacks [444]	I	Attack	Adv.	arXiv 2023
Jones et al. [446]	T	Attack	Adv.	ICML 2023
Wu et al. [452]	T + I	Attack	Prompt	arXiv 2024
Xie et al. [453]	T	Attack	Prompt	NMI 2023
Liu et al. [454]	T	Attack	Prompt	arXiv 2023
Carlini et al. [456]	T	Attack	Data	arXiv 2023
Jia et al. [457]	T	Attack	Data	EMNLP 2017
Latent Guard [461]	T+I	Defense	Detection	arXiv 2023
Van et al. [458]	T+I	Defense	Detection	arXiv 2023
Wei et al. [459]	T	Defense	Prompt	arXiv 2023
Smoothllm [460]	T	Defense	Prompt	arXiv 2023
Rafailov et al. [463]	T	Defense	Alignment	arXiv 2023
Raft [466]	T+I	Defense	Alignment	TMLR 2023
Wodajo et al. [471]	V	Defense	Detection	arXiv 2023
Safetybench [480]	T	Dataset	-	arXiv 2023
GOAT-Bench [481]	T	Dataset	-	arXiv 2024
ToViLaG [482]	T+I	Dataset	-	EMNLP 2023
Figstep [483]	T+I	Dataset	-	arXiv 2023
Liu et al. [484]	T+I	Dataset	-	arXiv 2023

Looking Forward

Unified Multimodal Models	Human-aligned AI	Efficient Models	Improved Safety	Emerging Applications
Will support seamless understanding and generation across text, image, video, audio, and 3D	Via scalable preference tuning, better instruction-following, and long-term memory	With smaller footprints, faster inference, and on-device capabilities	Through dynamic tool use, context-aware filtering, and robust watermarking	Robotics, AR/VR, education, creative industries

Summary and Takeaways

LLMs have transformed multimodal generation and editing
→ Enabling intuitive creation and manipulation of images, videos, audio, 3D content, and more.

LLMs serve flexible roles across modalities
→ Backbone, planner, labeler, agent, or conditioner depending on the task.

Strong advancements in key areas:

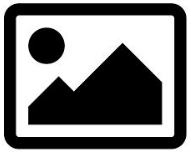
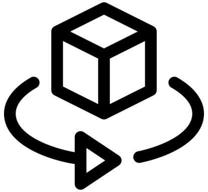
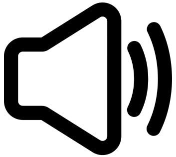


Image: Improved prompt synthesis, layout planning, interactive editing

Video: Coherent scene generation, temporal consistency, language-based editing



3D: Text-to-shape, procedural modeling, motion generation



Audio/Music/Speech: Sound classification, music composition, realistic TTS

UNIVERSITY of VIRGINIA

A Survey on Speech Large Language Models

Presented by:
Aditya Kakkar (zjq5mr) and Aryan Sawhney (ryd2fx)

UNIVERSITY of VIRGINIA

Aditya Kakkar (zjq5mr)

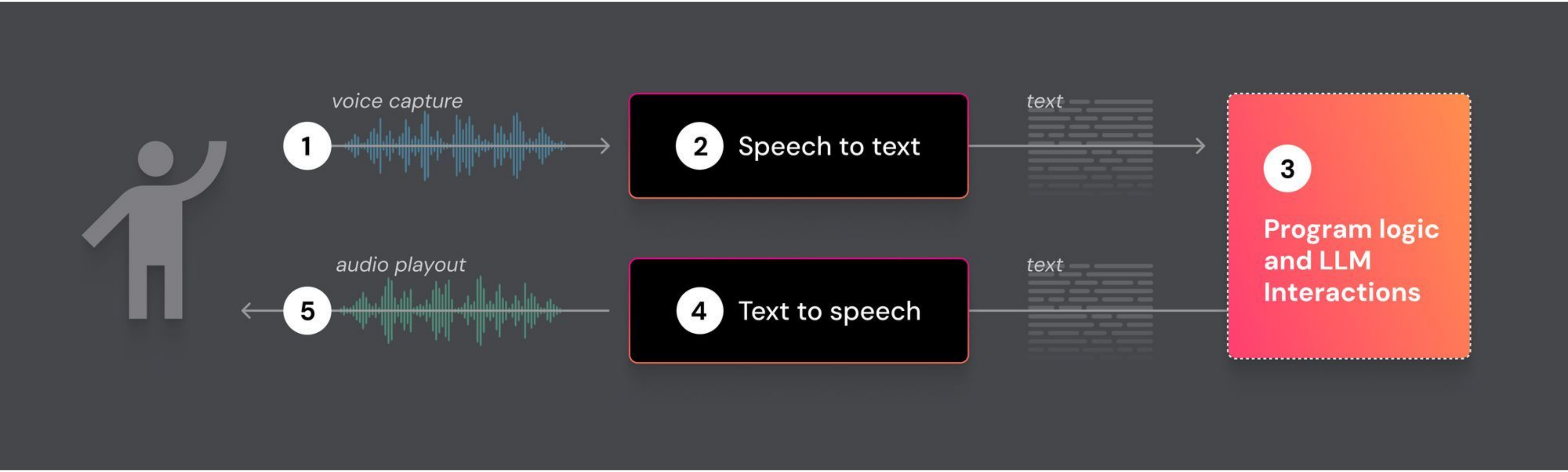
Presentation Outline

- ❖ Introduction
- ❖ Background & Motivation
- ❖ Evolution of Speech LLMs Architectures
- ❖ Advancements of Speech LLMs in Key Tasks and Challenges
- ❖ Speech LLM Input-Output Modes
- ❖ Model structure

Presentation Outline

- Introduction
- Background & Motivation
- Evolution of Speech LLMs Architectures
- Advancements of Speech LLMs in Key Tasks and Challenges
- Speech LLM Input-Output Modes
- Model structure

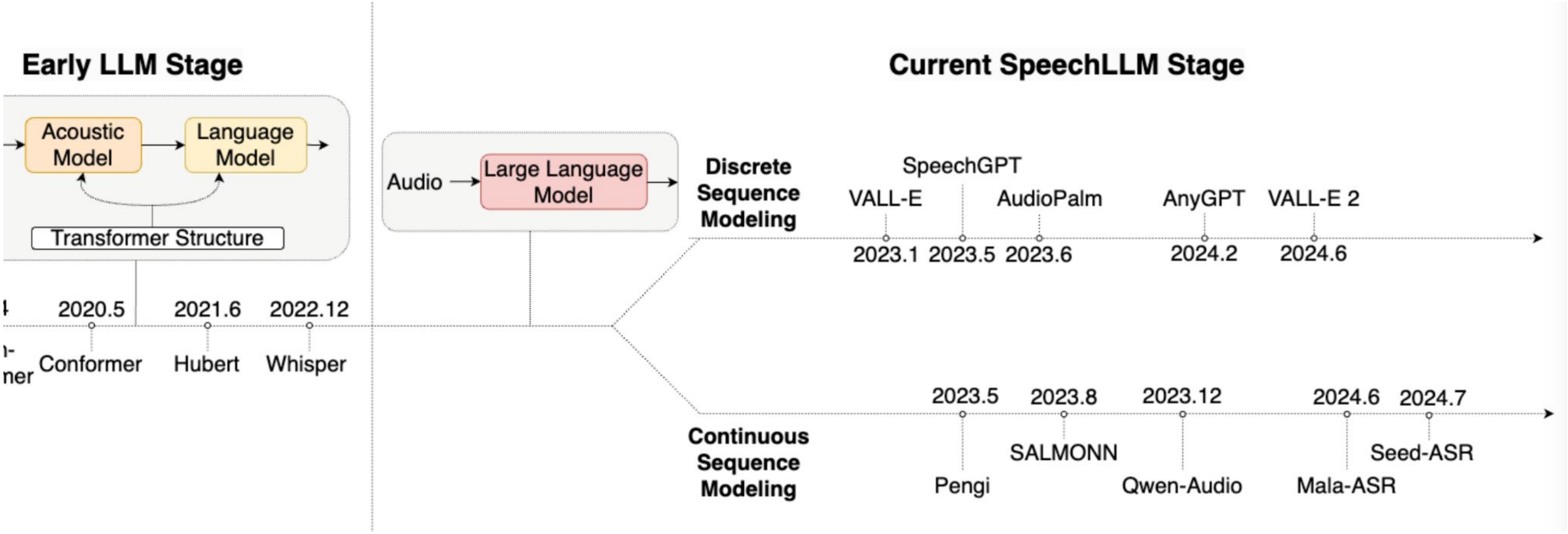
Speech Large Language Models



Evolution of Speech LLMs Architectures

Year	Model	Key Capability	Example Use Case
2018	Transformer	End-to-end speech-to-text	Transcribe spoken questions
2020	Conformer	Robust ASR with CNN + Transformer	Understand speech in noisy environments
2021	HuBERT	Self-supervised pattern learning	Learn from unlabeled audio, adapt to many tasks
2022	Whisper	Multilingual ASR + translation	Transcribe/translate from various languages
2022	SpeechT5	Multi-task: TTS, ASR, voice conversion	Read text aloud or convert voices

- **LLMs** like GPT are now being integrated into **Spoken Language Understanding (SLU)** systems, helping to solve previously difficult problems like long-form speech recognition detection.



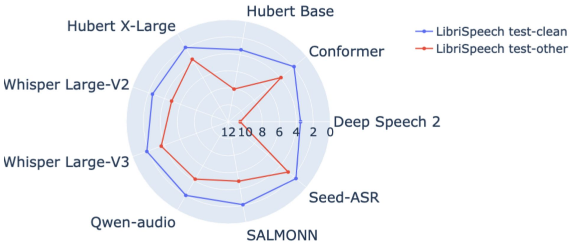
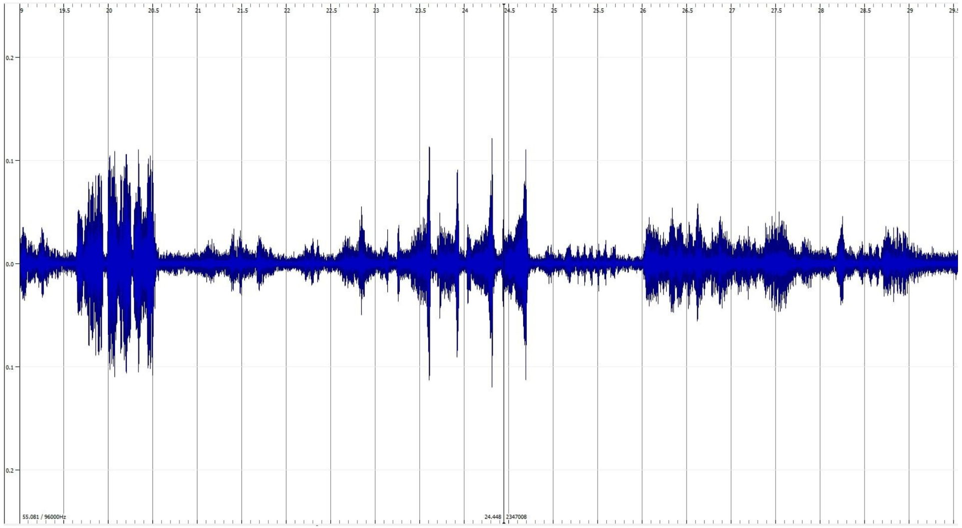
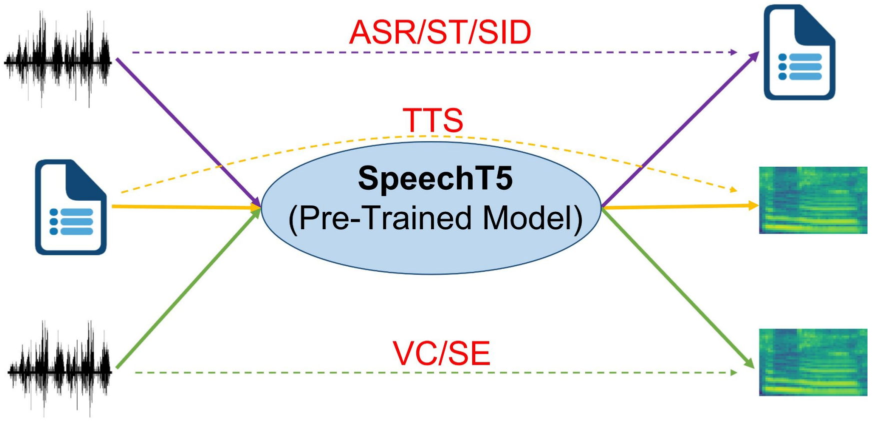


Fig. 5: Model Performance on ASR task (WER%)

SpeechT5

- SpeechT5 is trained on large datasets with aligned text and speech, and it learns It learns patterns
- Text → Text Encoder → Transformer (speech features) → Speech Decoder → Spectrogram → Vocoder → Audio



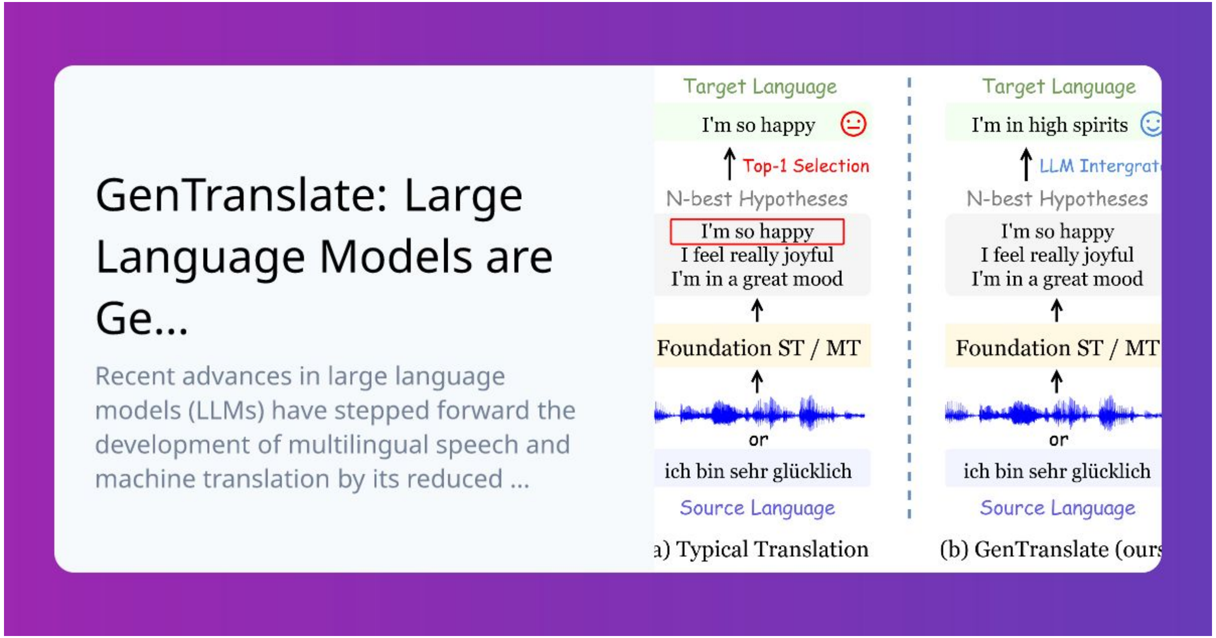


Capabilities

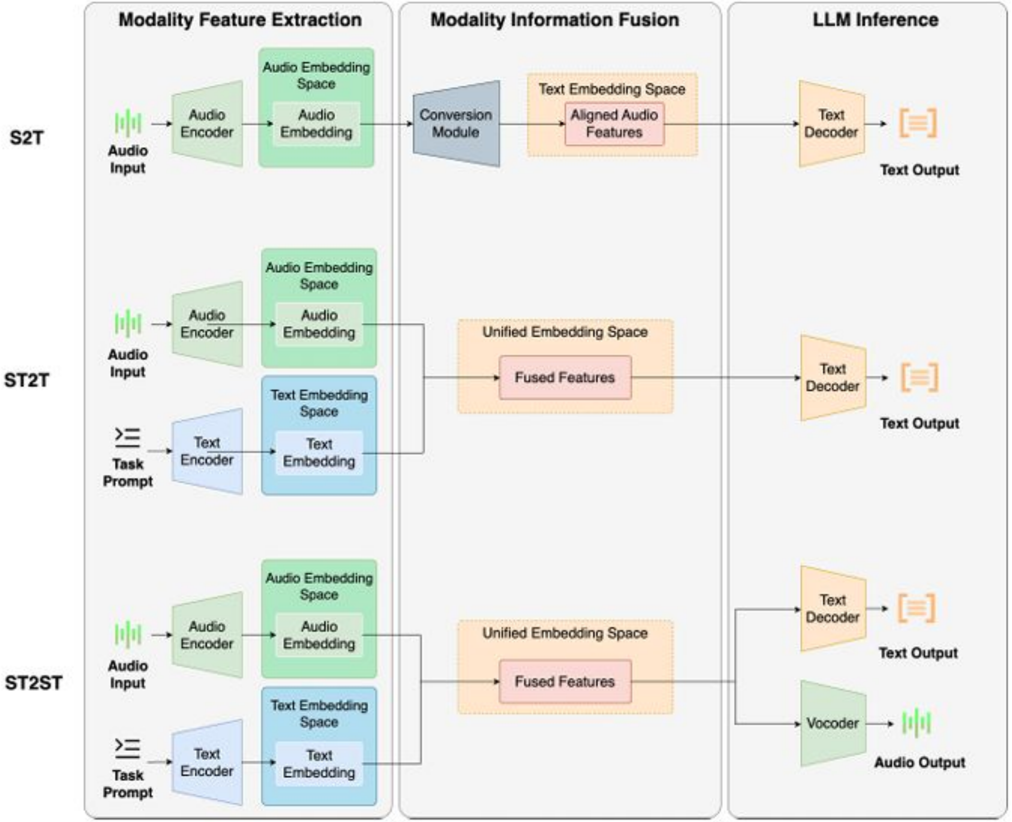
Capability	What It Does	Key Improvements	Example
ASR (Speech-to-Text)	Converts spoken language into written text	- More accurate recognition - Better noise and accent handling - Lower Word Error Rate (WER)	Conformer models outperform LSTMs on MLS dataset
Speech Translation	Translates speech from one language to another	- Fluent, real-time translation - Supports more languages	Translate Spanish audio to English live during a conference
Speaker ID & Multitask	Identifies speaker and detects sentiment along with transcription	- Performs multiple tasks at once - Improves sentiment and WER accuracy	Qwen-Audio ↓ WER by 10%, ↑ sentiment accuracy by 8%

Solution

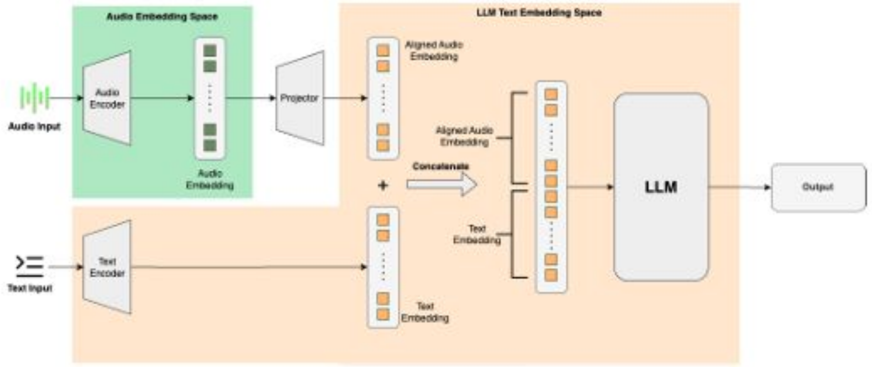
- Traditional systems often just looked at short windows of audio (like 1 second).
- LLM-based models can **analyze longer sequences** of speech.
  - Longer context window
  - self-Attention amongst tokens



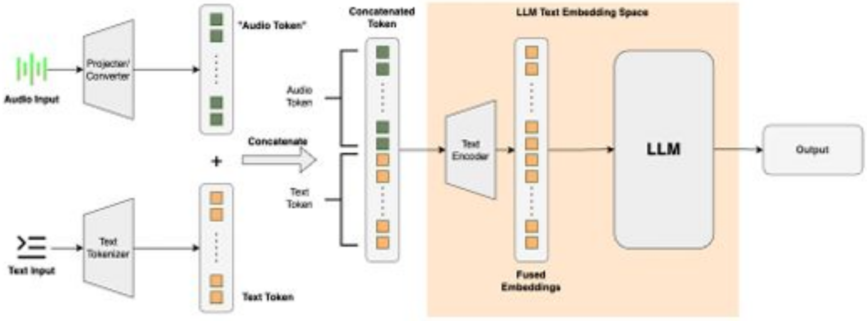
Three Input-Output Types



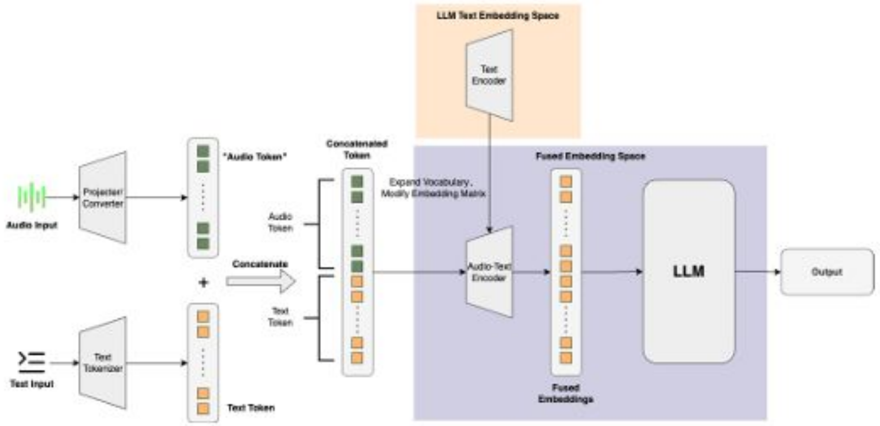
Modality Fusion



(a) Direct Projection



(b) Token Mapping

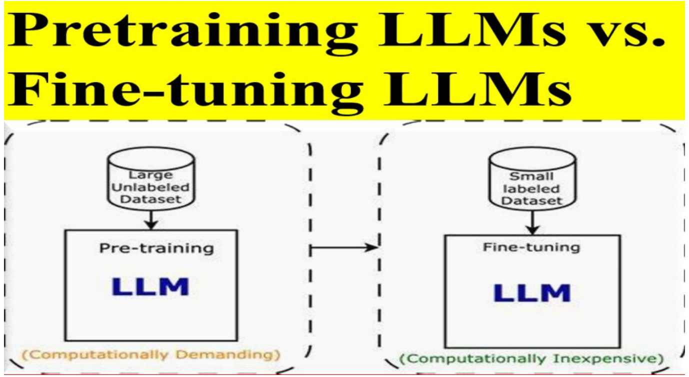


(c) Combine audio and text space

Aryan Sawhney (ryd2fx)

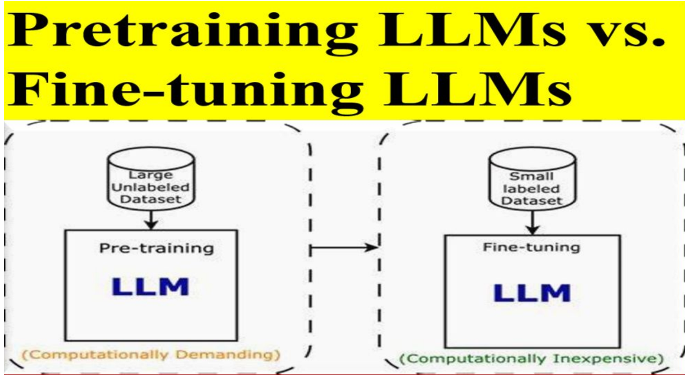
Training Strategies - Pretraining

- Pretraining involves training models on large-scale unlabeled data to enable them to learn broad, general knowledge
  - In Speech LLMs, speech encoders undergo pretraining on audio-text pairs to capture audio features
  - Common training strategies, including self-supervised learning (SSL)
- Some researchers attempt to re-pretrain speech encoders, optimizing audio feature extraction capabilities
  - Re-training of multimodal large models is necessary



Training Strategies - Supervised Fine-Tuning

- Supervised fine-tuning is a common approach where high-quality labeled data from downstream task datasets is used to train the model
  - Used to achieve alignment between the speech encoder and the LLM, and to enhance performance on specific tasks
- Common training methods include fine-tuning connectors, fine-tuning the encoder, and LLMs, such as using methods like LoRA



Training Strategies - Reinforcement Learning

- Reinforcement learning is a framework where an agent learns to maximize cumulative rewards by interacting with an environment and adjusting actions based on feedback
    - Used to optimize LLM in the desired direction while maintaining diversity in its outputs
- Two commonly used RL algorithms in this context are:
    - Proximal Policy Optimization (PPO) - a policy gradient method that optimizes performance while restricting updates (clipping) to stay close to the current policy
    - Direct Policy Optimization (DPO) - directly maximizes expected rewards using a task-specific reward function, without clipping, making it suitable for settings with clear reward signals

Performance - Automatic Speech Recognition

- Assessed Word Error Rate (WER) tested on the clean and other test set of LibriSpeech dataset

Stage	Models	test-clean	test-other
Pre-LLM Stage	Deep Speech 2	3.5	10.6
Early LLM Stage	Conformer	1.9	3.9
	HuBERT Base	3.4	8.1
	HuBERT X-Large	1.9	3.5
	Whisper Large-V2	2.5	4.9
	Whisper Large-V3	1.8	3.6
Current Speech LLM Stage	Qwen-audio	2.0	4.2
	SALMONN	2.1	4.9
	Seed-ASR	1.6	2.8

Table 1: Model Performance on ASR task (WER%)

- Traditional ASR (e.g., Deep Speech 2) had 3.5% WER on clean and 10.6% on other
- Transformer-based models (Conformer, HuBERT, Whisper) lowered clean WER to ~ 2% on clean and ~4% on other
- Latest Speech LLM models (e.g., Seed-ASR) further reduced WER to 1.6% (clean) and 2.8% (other)

Performance - Speech Translation

- Used BLEU score to evaluate each model's two-way Chinese and English translation performance in speech and machine translation tasks

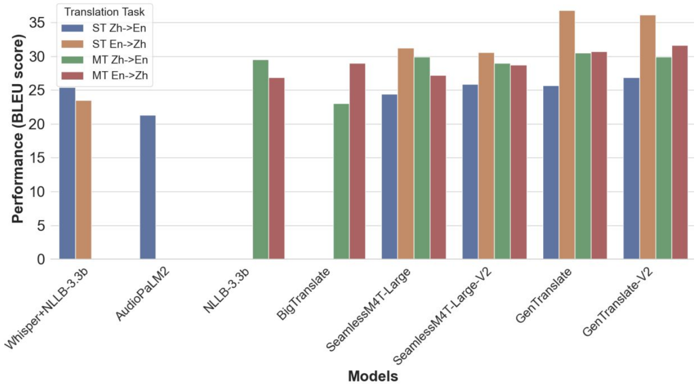


Fig. 6: Model Performance on speech translation and machine translation task (BLEU Score)

- NLLB now supports translation for over 200 languages
- BigTranslate, tuned with LLaMA, delivers quality comparable to leading systems like ChatGPT and Google Translate
- Whisper's cascaded ASR+MT method leverages massive web-scale data for strong speech translation results
- GenTranslate outperforms previous state-of-the-art models like SeamlessM4T.

Performance - Multi-tasking and Cross-tasking Abilities

- Pengi uses a unified text-generation approach to handle both open-ended tasks (like audio captioning and question answering) and closed tasks (such as sound event classification), achieving state-of-the-art results
- SALMONN integrates diverse audio inputs—including speech and music—to excel in emergent tasks like audio storytelling and cross-modal reasoning
- Qwen-Audio scales its pre-training across 30+ tasks (e.g., scene classification, emotion recognition), demonstrating robust multi-task performance with minimal task interference

Challenges

LLM Dormancy: Situation where a Speech LLM becomes unresponsive or fails to effectively process new or unseen text prompts when audio and text inputs are combined

High Cost: LLMs incur high usage costs due to slow inference times and substantial GPU memory demands from their complex architectures and many parameters

LLM Dormancy Cause: Audio features projected into text space have larger embeddings, causing them to overpower text that convey instructions or context during processing

Input: Can you divorce life from art?(A man speaker with chinese accent)			
Ability Test	**Prompt**	**Output**	
		SLAM-LLM	SALMONN
ASR	transcribe speech to english text	Can you divorce life from art?	can you divorce life from art?
Speech Translate	transcribe the speech to text and translate it to chinese.	Can you divorce life from art?	你能从艺术中离开吗?
Speech Translate	transcribe the speech to text and translate it to Japanese.		アートから生活を破棄することができますか？
Speaker Verify	which scene will this talk occur?		The speaker is male and has an American accent. He speaks with confidence and clarity.
Speaker Verify	Is the speaker a female?		Yes, the speaker is a female.
Dialogue Ability	Tell me the reason.		I'm sorry, I'm not sure what you're asking. Could you please provide more context or clarify your question?
Dialogue Ability	Based on the speech, can you help me determine the answer?		Yes, I can divorce life from art.
Dialogue Ability	can you answer the speech and give reasons?		Can you divorce life from art
Context Inference	Actually this is a speech a man asked me. I don't have answer. And someone tells me to say yes. But I don't know why. Can you explain it?		Yes, I can explain it. The man asked you if you could divorce life from art. This means that you can separate the concept of life from the artistic representation of it. For example, a painting of a landscape could be seen as a representation of life, but not necessarily a reflection of it. So, the answer to the man's question is yes, you can divorce life from art.

Table 2: Prompt Tests conducted on Slam-ASR and SALMONN; Text highlighted in red indicates errors or incorrect outputs, while text in black represents reasonable or correct responses.

Future Exploration

- Improve modality alignment by adding audio tokens while managing data imbalance and training costs
- Align modalities by normalizing or discretizing audio features before mapping to text
- Use reinforcement learning to boost multi-task performance and reduce repetition
- Use LLMs to coordinate tasks in systems like dialogue, translation, and conferencing, leveraging their strong reasoning abilities
- Enhance long speech recognition and keyword detection with better segmentation and context-rich prompts

Questions?