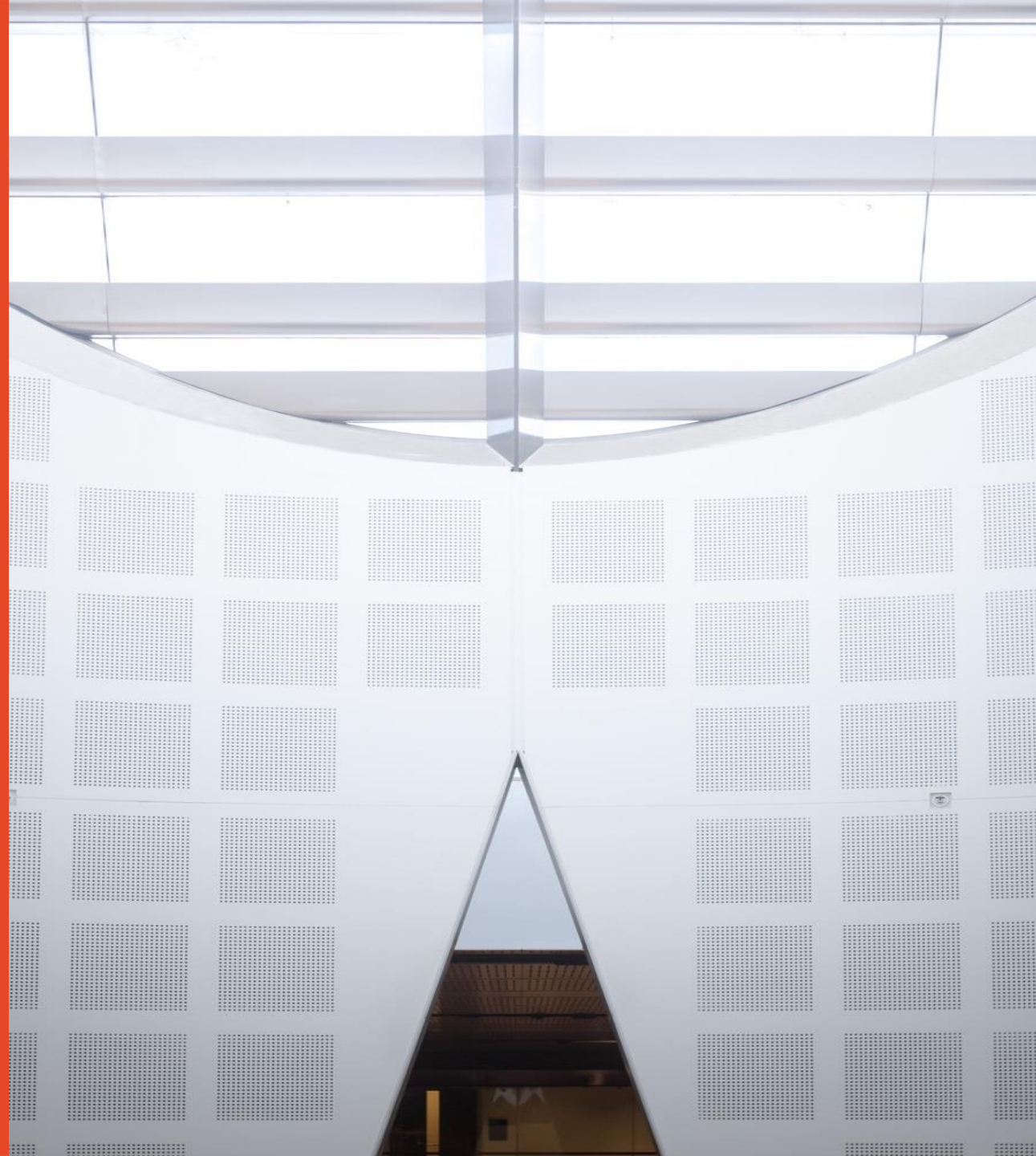
An Overview of Multi-Modal LLM

Jinxu Lin

jinxu.lin@sydney.edu.au

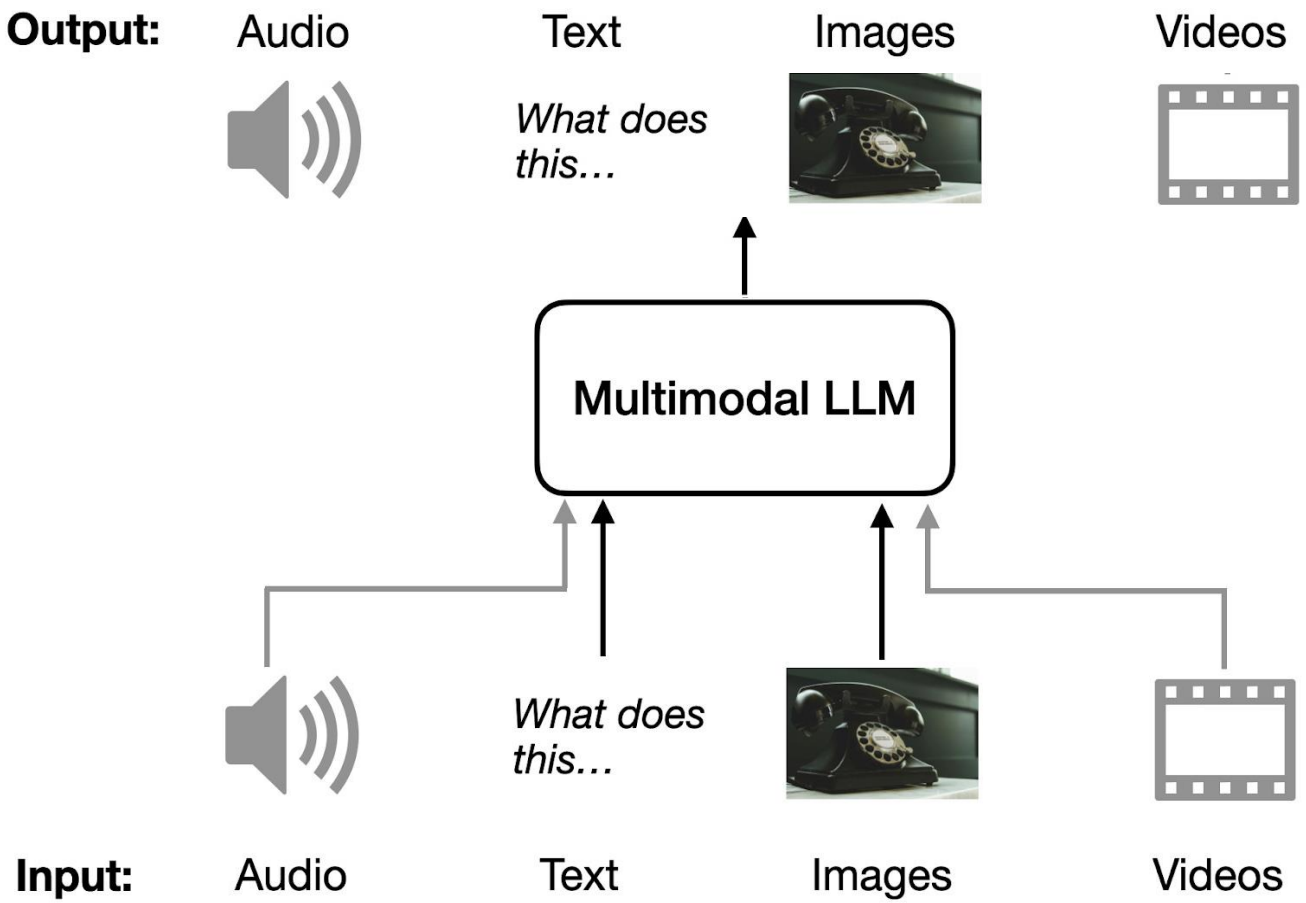
School of Computer Science

THE UNIVERSITY OF SYDNEY

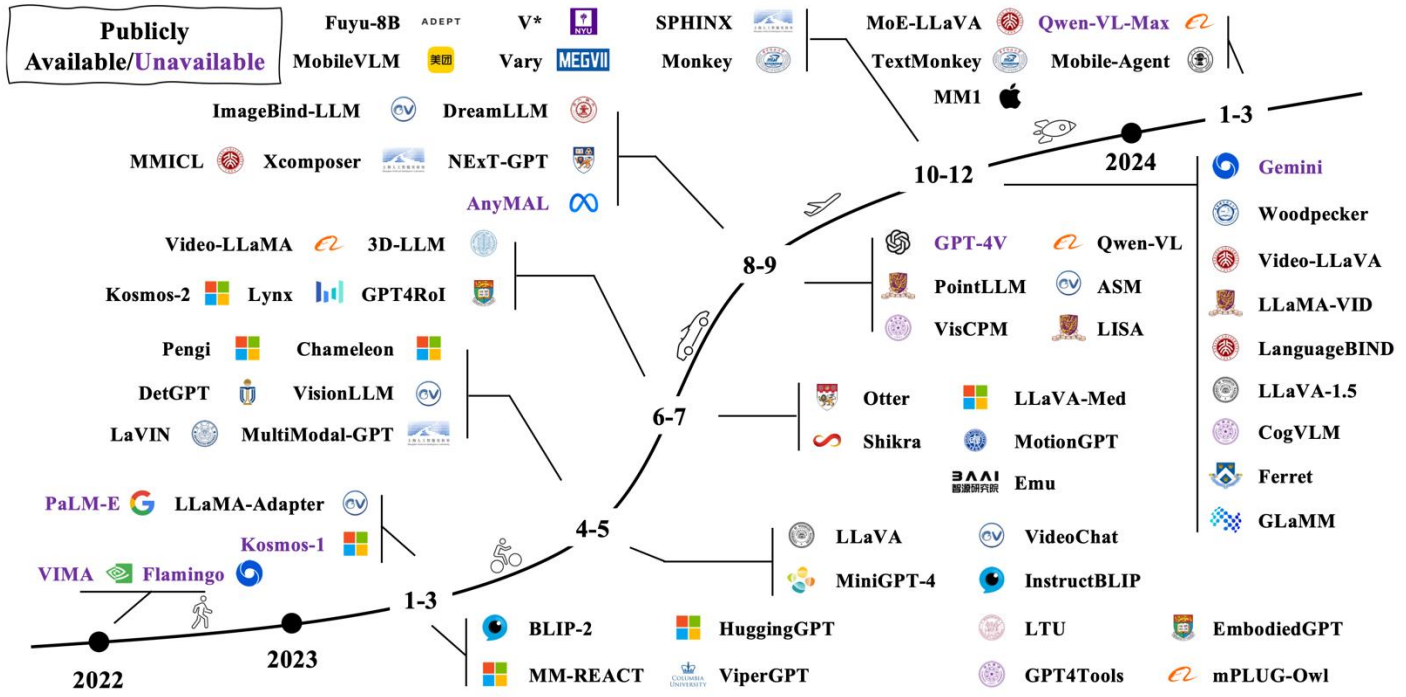


Multi-Modal Large Language Models (MLLMs)

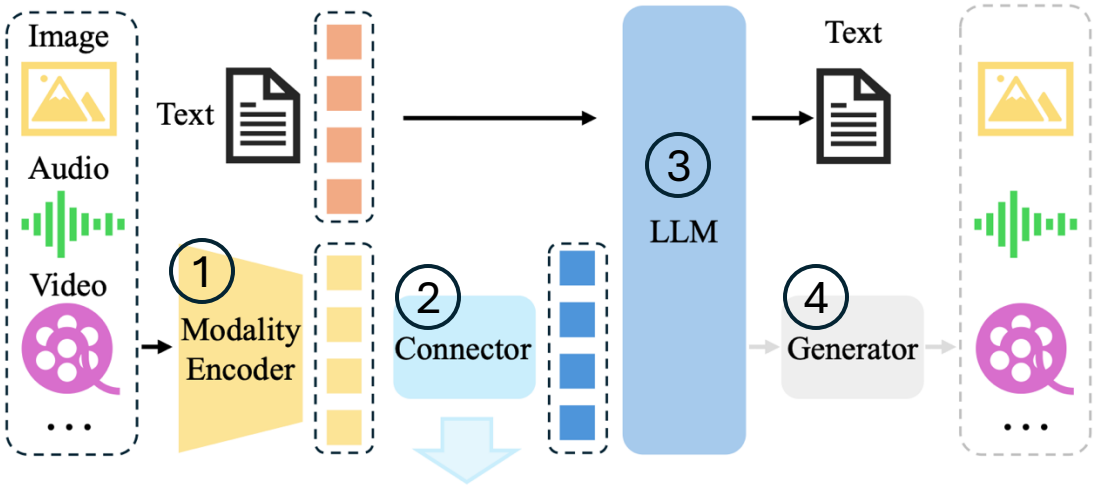
- LLMs: Text Only
- MLLMs: Enabling LLMs to comprehend and generate content across diverse modalities, including:
  - Image
  - Video
  - Audio
  - …



A timeline of representative MLLMs



MLLMs Architecture



① Encoder: Processes continuous data such as images, audio, or video and outputs features.

② Connector: Transforms these features to enable better understanding by the LLM.

③ LLM: Generates data.

④ Generator (Optional): Produces modality-specific outputs beyond text (for LLMs limited to text generation).

Pretrained Encoder

TABLE 1: A summary of commonly used image encoders.

Variants	Pretraining Corpus	Resolution	Samples (B)	Parameter Size (M)
OpenCLIP-ConvNext-L [46]	LAION-2B	320	29	197.4
CLIP-ViT-L/14 [13]	OpenAI's WIT	224/336	13	304.0
EVA-CLIP-ViT-G/14 [47]	LAION-2B,COYO-700M	224	11	1000.0
OpenCLIP-ViT-G/14 [46]	LAION-2B	224	34	1012.7
OpenCLIP-ViT-bigG/14 [46]	LAION-2B	224	34	1844.9

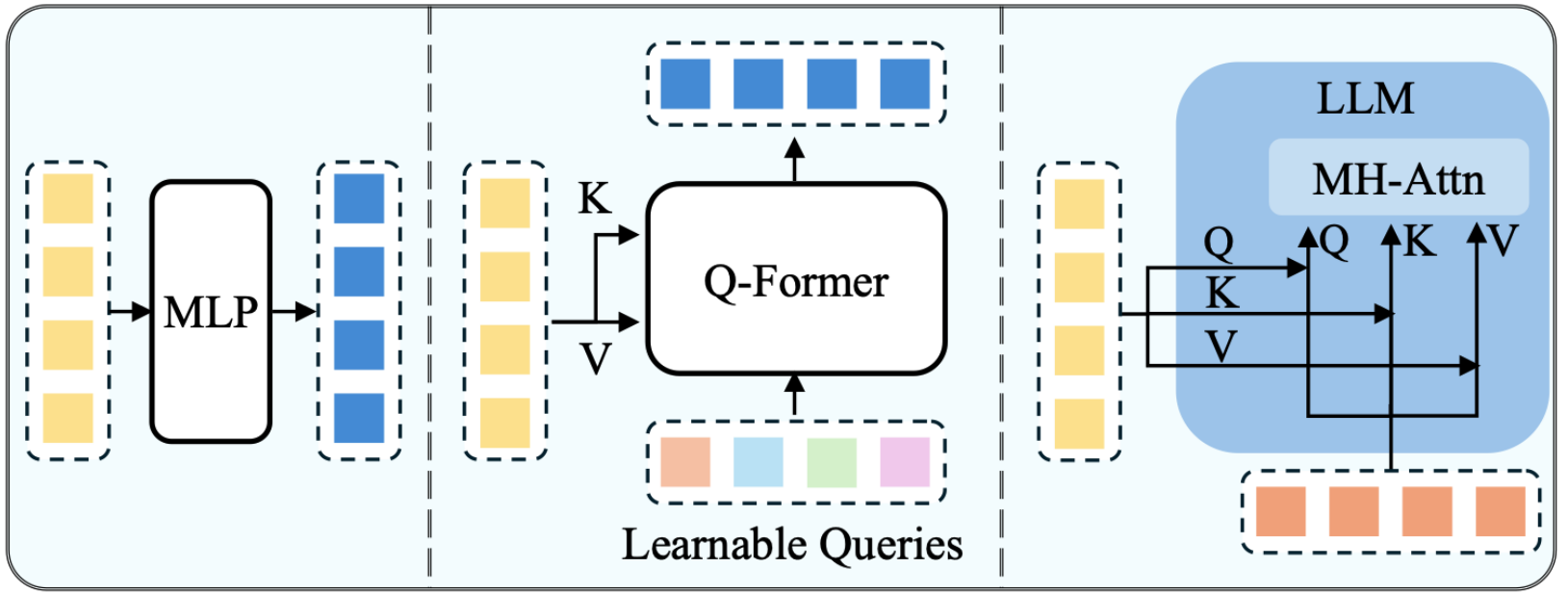
Pretrained LLM

TABLE 2: A summary of commonly used open-sourced LLMs. en, zh, fr, and de stand for English, Chinese, French, and German, respectively.

Model	Release Date	Pretrain Data Scale	Parameter Size (B)	Language Support	Architecture
Flan-T5-XL/XXL [56]	Oct-2022	-	3/ 11	en, fr, de	Encoder-Decoder
LLaMA [5]	Feb-2023	1.4T tokens	7/ 13/ 33/ 65	en	Causal Decoder
Vicuna [4]	Mar-2023	1.4T tokens	7/ 13/ 33	en	Causal Decoder
LLaMA-2 [57]	Jul-2023	2T tokens	7/ 13/ 70	en	Causal Decoder
Qwen [58]	Sep-2023	3T tokens	1.8 / 7/ 14/ 72	en, zh	Causal Decoder

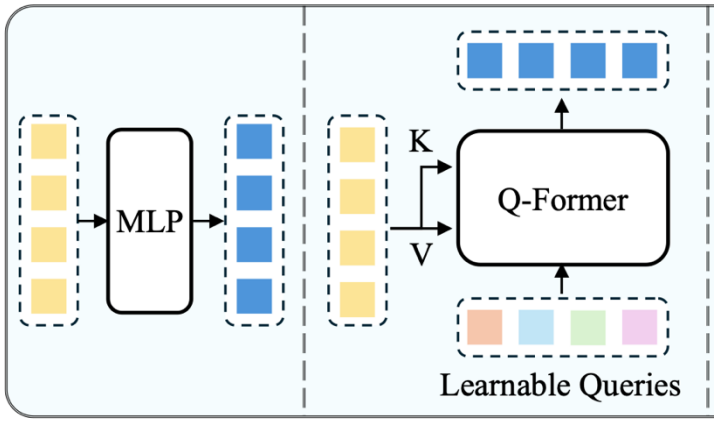
Connector

- **Connector**: transform data from other modalities into inputs that the LLM can understand.



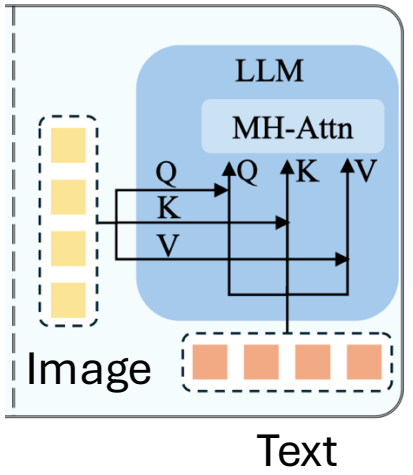
Connector

- Token-level fusion connector:
  - Features extracted from encoders are **transformed into token representations**.
  - These tokens are then **concatenated with text tokens** and processed together within the LLM.
- Example implementations:
  - Applying an MLP to project visual features into a compatible space, as done in **LLaVA**.
  - Using a **Q-Former** to compress visual features into a smaller set of representation tokens.



Connector

- Feature-level Fusion Connector
  - In feature-level fusion, **raw features from different modalities** are combined at a deeper level for richer interaction.
  - This often involves adding **extra modules** like cross-attention layers to facilitate interaction between text and visual features.
- Example implementations:
  - **Flamingo**: Adds cross-attention layers between frozen Transformer layers of an LLM to fuse visual and language features.
  - **CogVLM**: Integrates visual expert modules for dual interaction between vision and language features.
  - **LLaMA-Adapter**: Uses learnable prompts embedded with visual knowledge to align with text features.



Connector

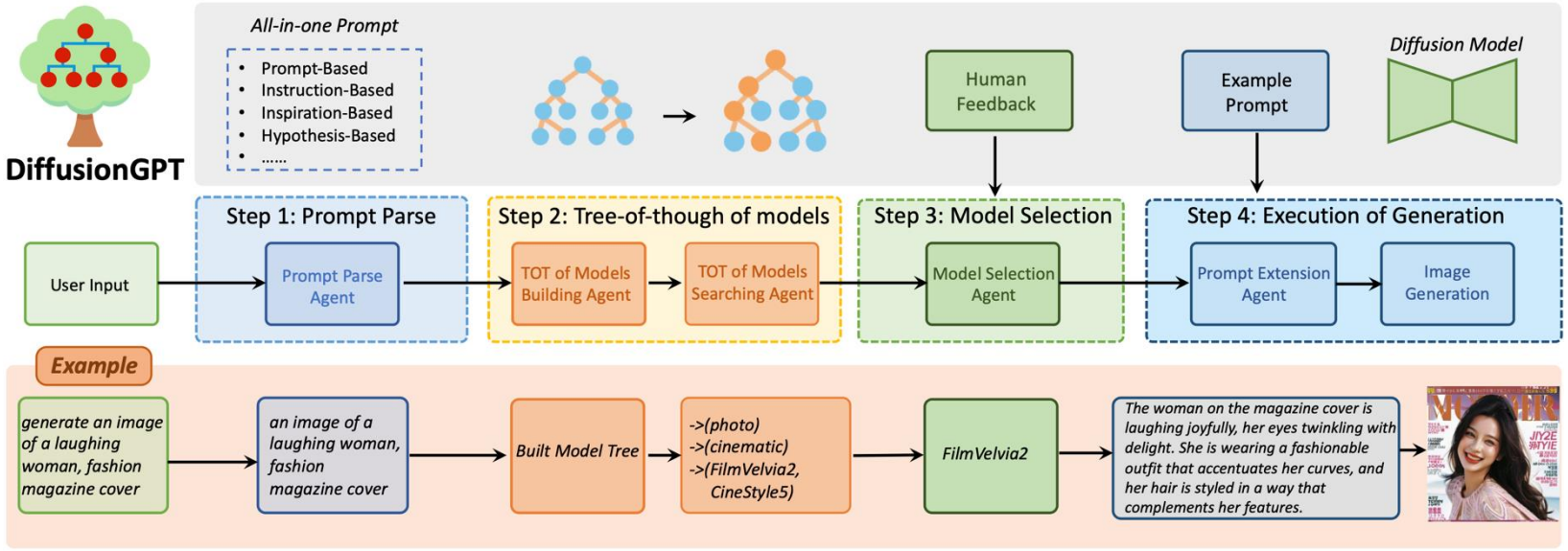
- Expert Model as Connector
  - Expert models convert raw multimodal data into a format (language/text) that LLMs can easily understand.
- Example implementations:
  - An image captioning model translates images into descriptive sentences.
  - A speech recognition model converts audio inputs into text.
  - **VideoChat-Text**: Uses pre-trained vision models to extract visual details (e.g., actions or objects in a video). Enriches these details with textual descriptions generated by a speech recognition model.

Using MLLMs to generate Multi-Modal Data

- Using LLM as conditioner
  - Diffusion GPT
- Using LLM as generator
  - Visual Autoregressive:
    - SEED, SEED-X, DreamLLM, Unified IO, Unified IO2, Chamelon, Lumina-mGPT, ANOLE
  - Visual Scale Autoregressive:
    - VAR, STAR
  - Visual Diffusion:
    - Transfusion, Show-o

Diffusion GPT (Byte Dance)

- Using LLM as conditioner
- DiffusionGPT utilizes various expert models (LLMs) to **refine** and **enhance** the input prompt.
- The output from the LLMs is then used as conditional input for different downstream generative models, such as *'FilmVelvia2'*.



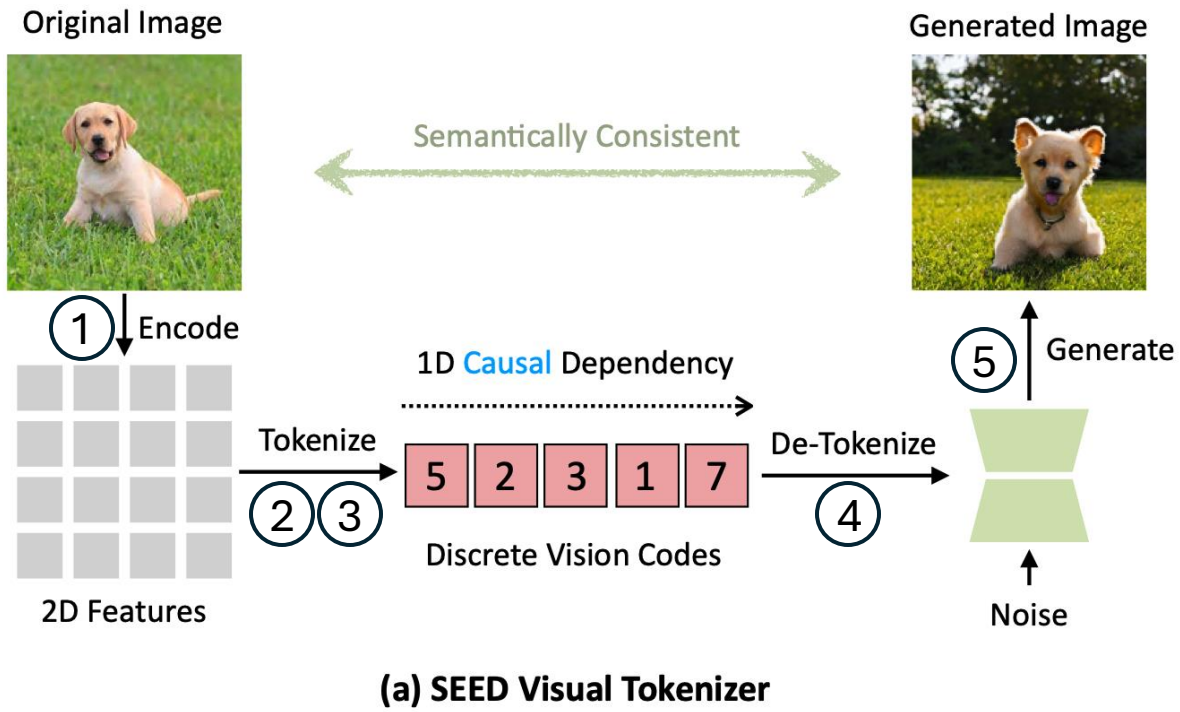
Visual Autoregressive Generation in MLLMs

- MLLMs generate images using a visual autoregressive approach, similar to how LLMs generate text through next-token prediction.
- Key steps include:
  - **Visual Vocabulary**: Image features are discretized into a visual vocabulary (codebook) using VQ-GAN, transforming continuous features into discrete representations.
  - **Unified Learning**: The unified learning process for text and images allows consistent training across modalities.
  - **Next-Token Prediction**: Cross-entropy (CE) loss is used to predict the probability of the next token in the vocabulary.

SEED (Tencent)

- Issues of Visual Autoregressive Generation:
- Images are continuous data, LLMs can only process discrete data
  - Image: RGB value from 0-255
  - Text: Vocabulary
- SEED: Using an Image Tokenizer to discrete Image, unified Text and Image

SEED (Tencent)



(a) SEED Visual Tokenizer

① ViT Encoder (pre-trained): BLIP-2

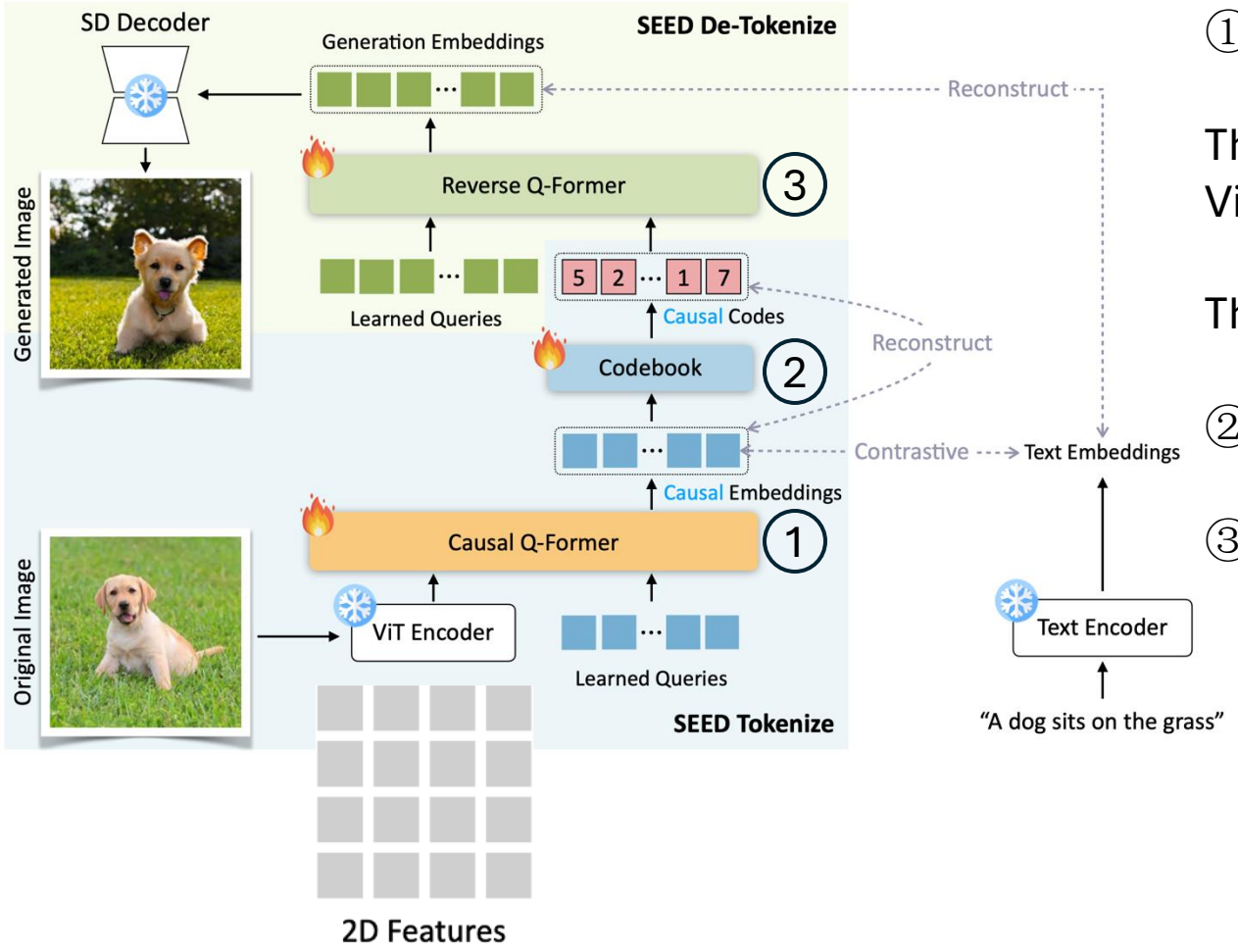
② Causal Q-Former: Converts 2D image features into 1D features with causal dependency by ViT .

③ VQ Codebook: Discretizes the 1D features before LLM.

④ Reverse Q-Former: Mapping the output of LLM to the condition of Stable Diffusion.

⑤ Unet Decoder (pre-trained): Stable Diffusion

SEED (Tencent)



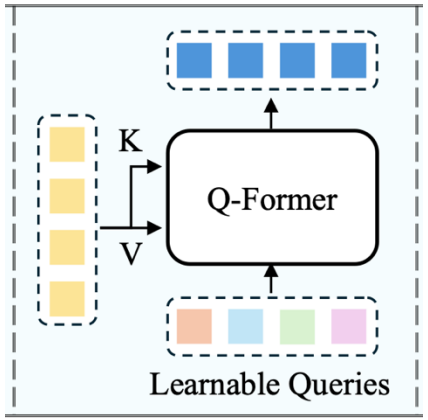
① Training the Causal Q-Former

The input consists of image features extracted by the Vision Encoder and a set of learnable 1D queries.

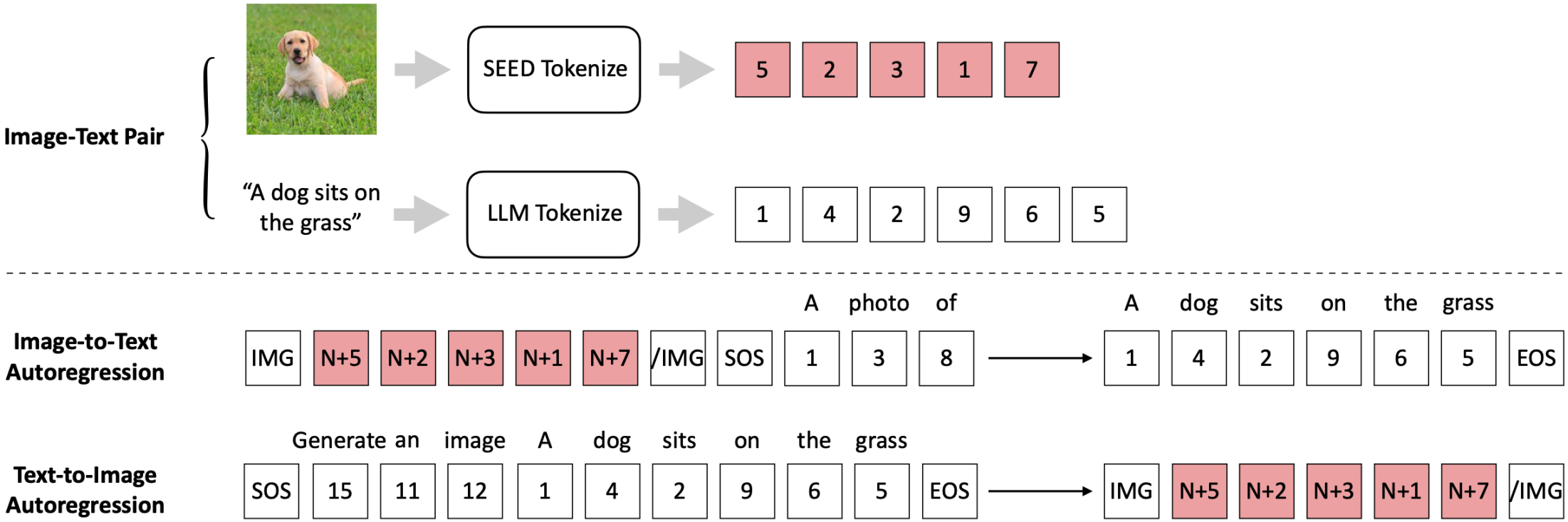
The output queries contain a series of image features.

② Training VQ Codebook

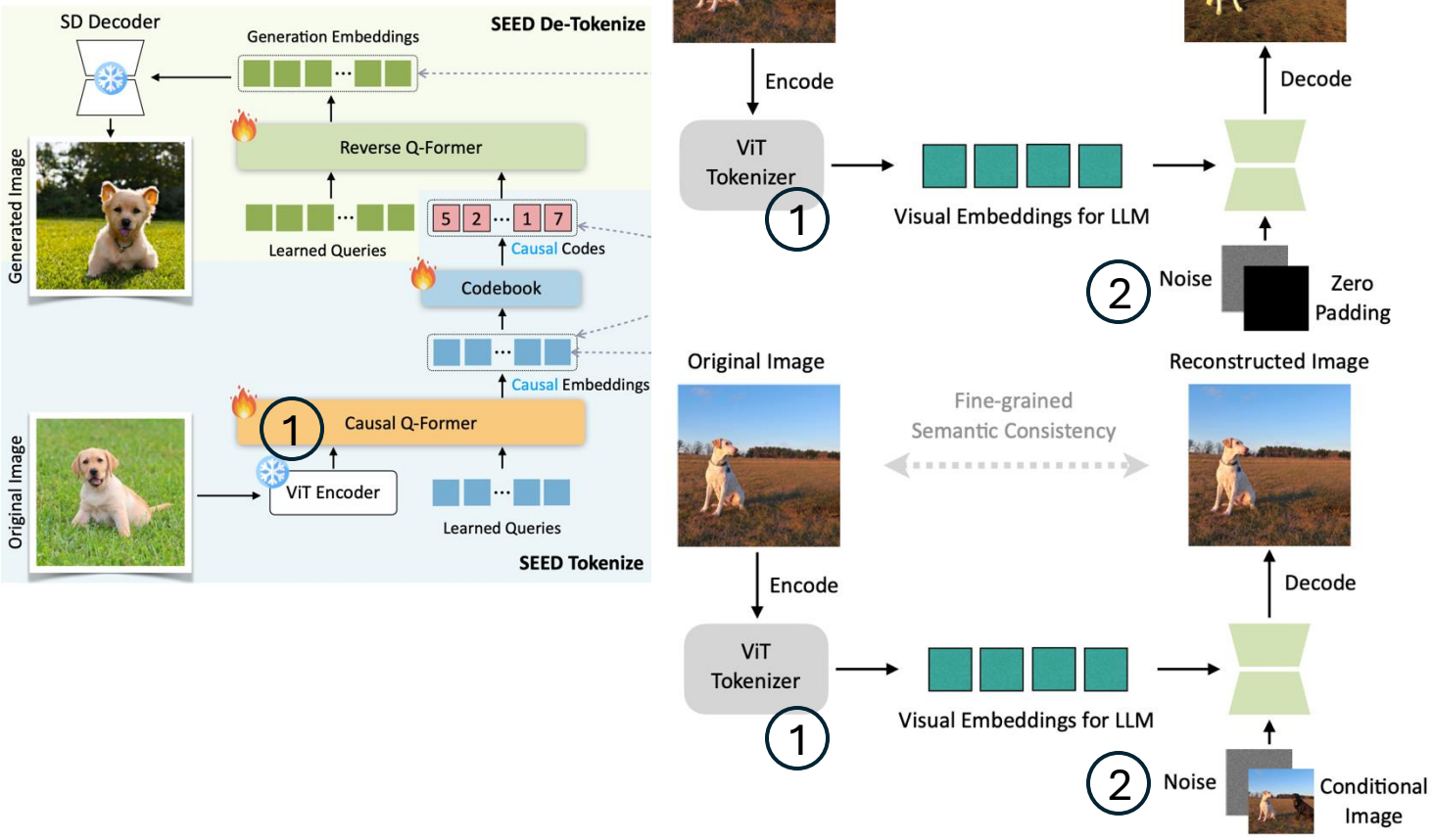
③ Training Reverse Q-Former



SEED (Tencent)



SEED-X (Tencent)

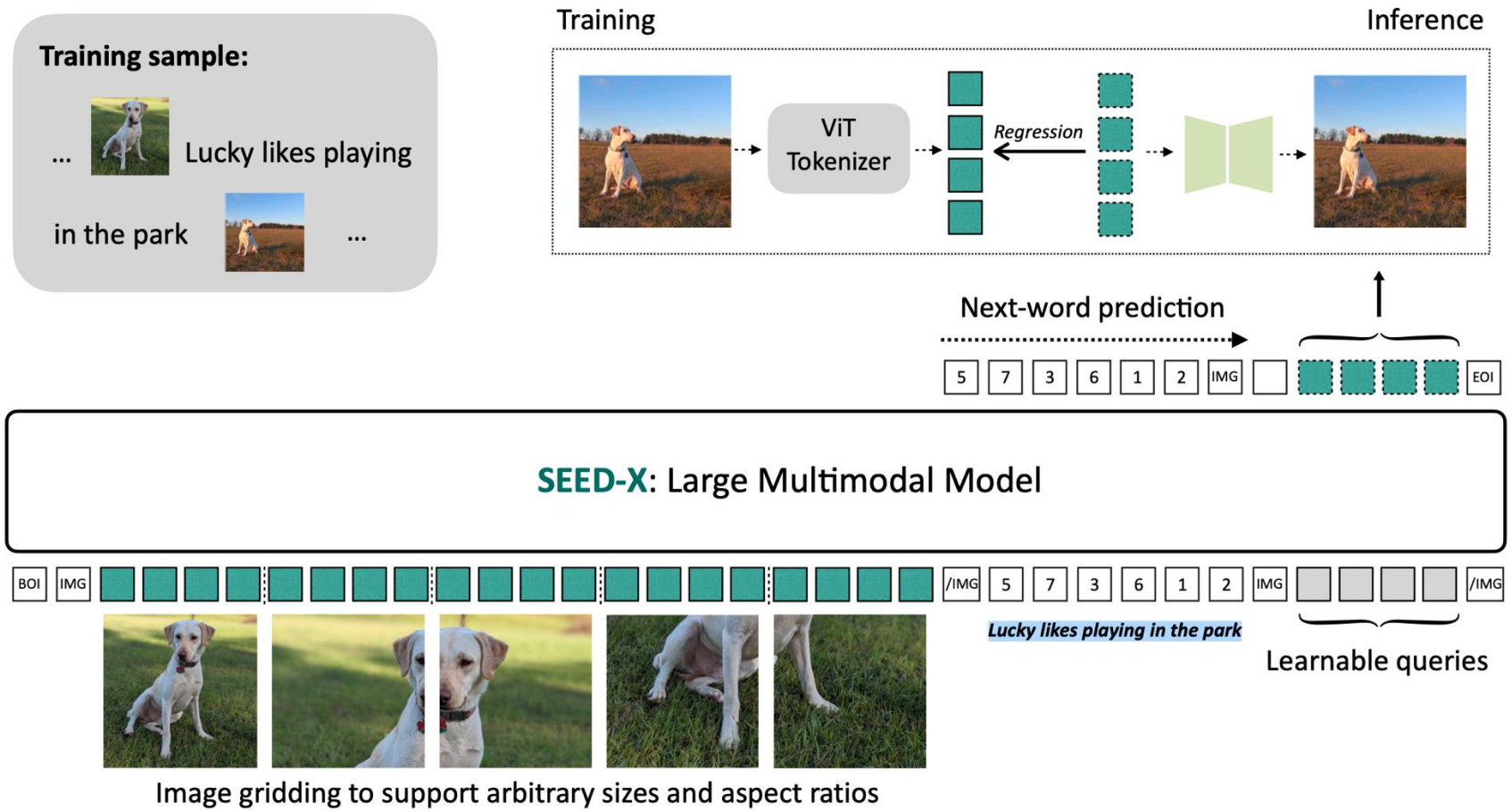


Differences:

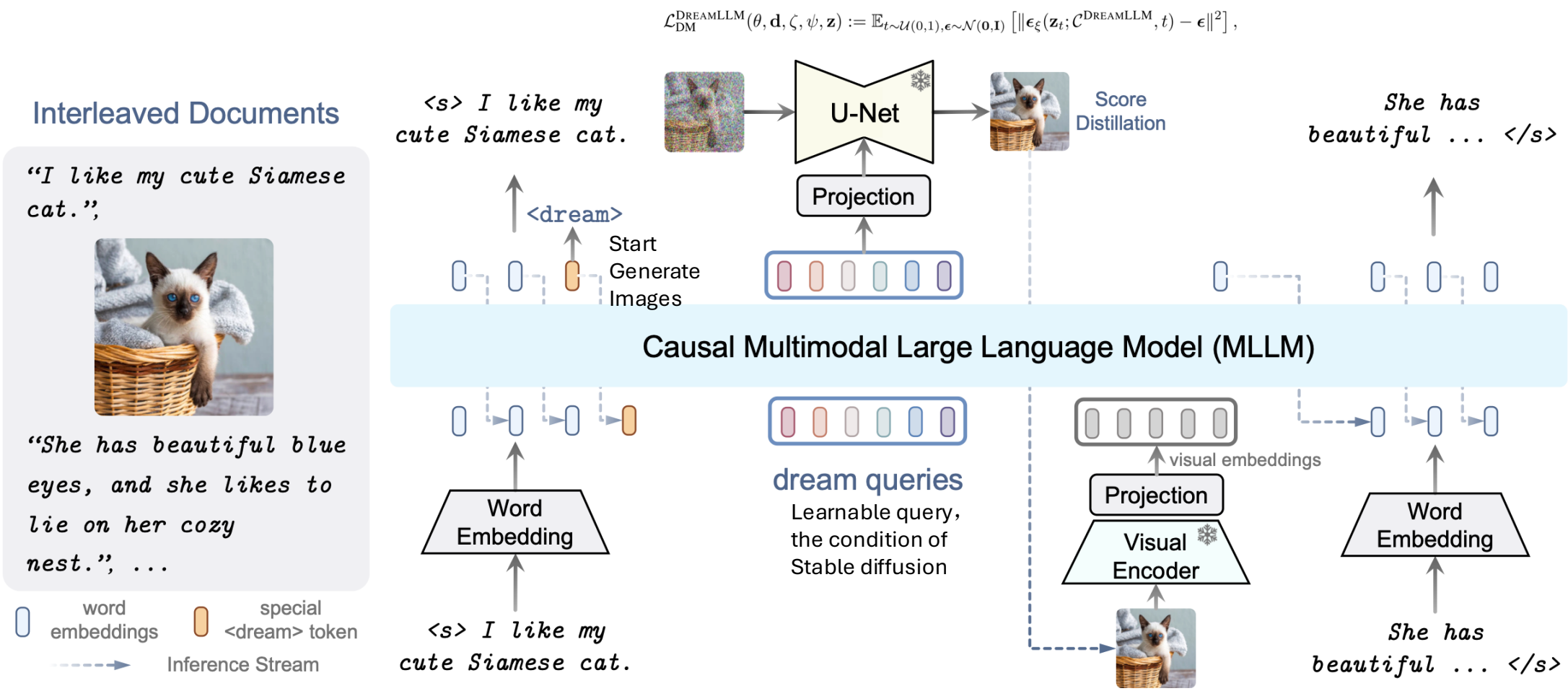
① In SEED, the output of the ViT-encoder is continuous data, needs the Q-Former to be transferred into discrete data.
In SEED-X, use Vit-tokenizaer to directly encode Image as discrete data.

② Pre-training the visual tokenizer is achieved by employing a two-stage, multi-grained condition injection strategy.

SEED-X (Tencent)

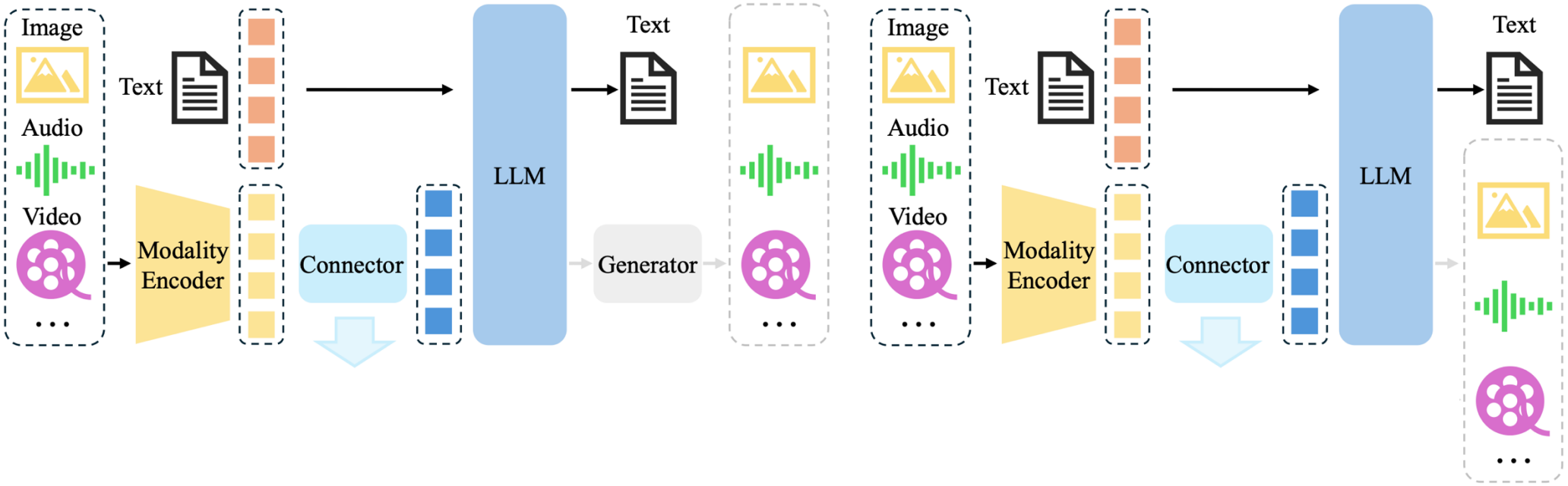


DreamLLM (Tsinghua)



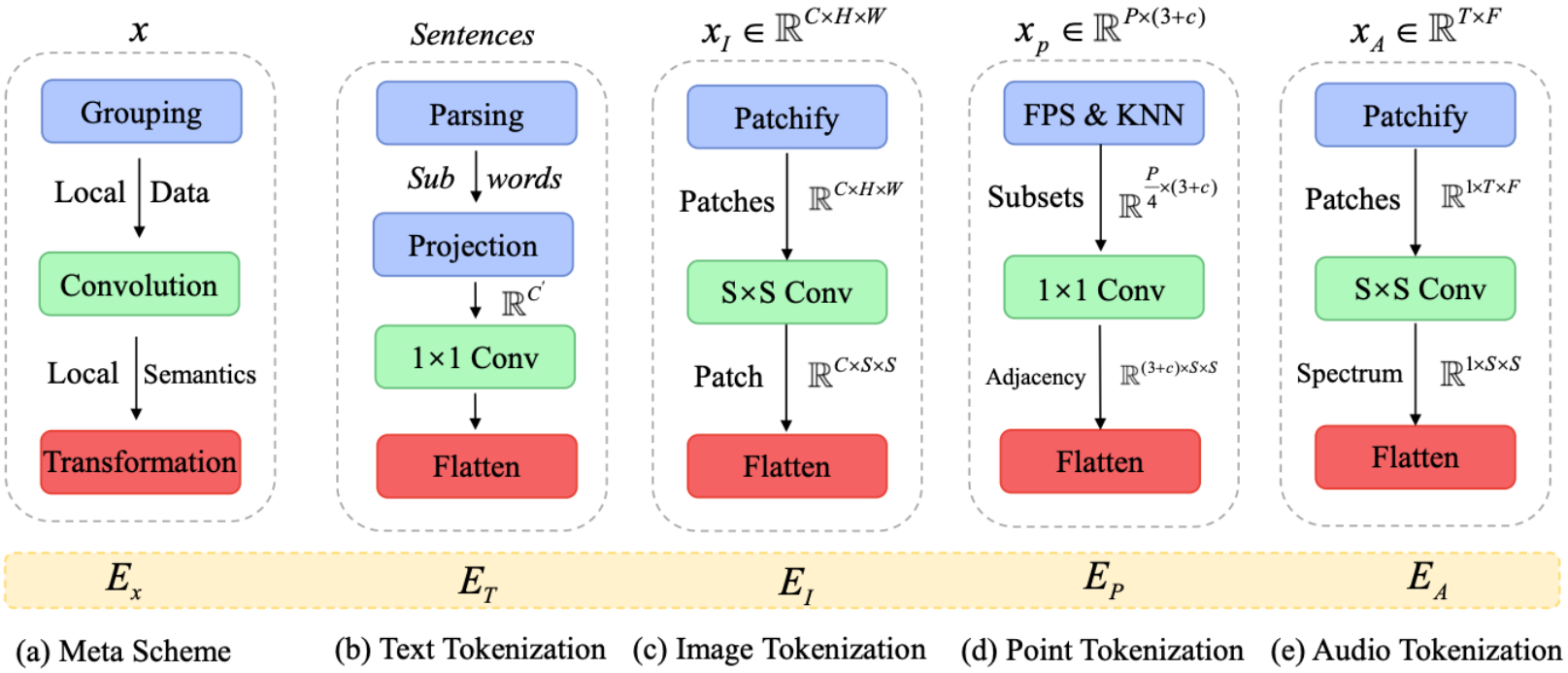
Generation by Large Multimodal Models

- SEED, SEED-X and Dream LLM need downstream generator
- Large multimodal models (LMMs) generate images without extra generator, natively by LLM itself



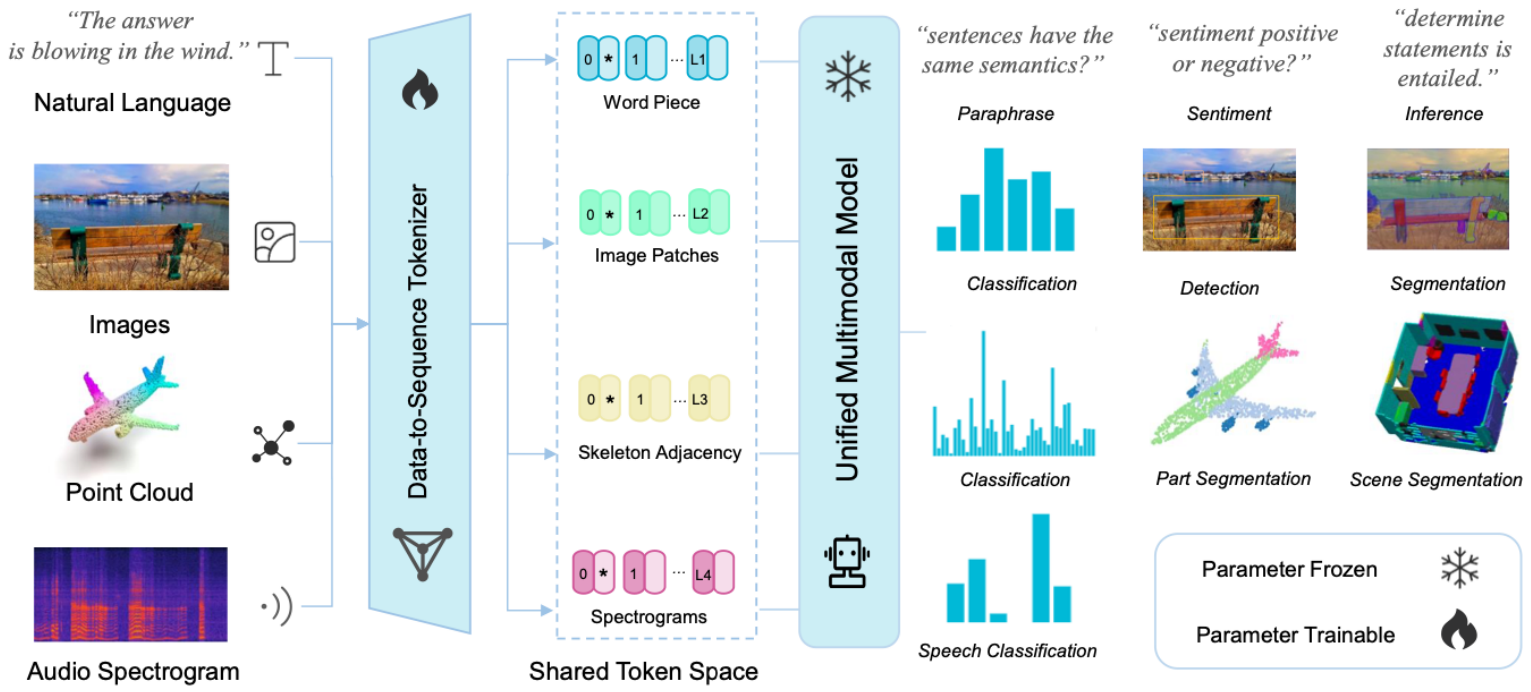
Meta-Transformer (MM Lab)

- Each Modal has its own tokenizer



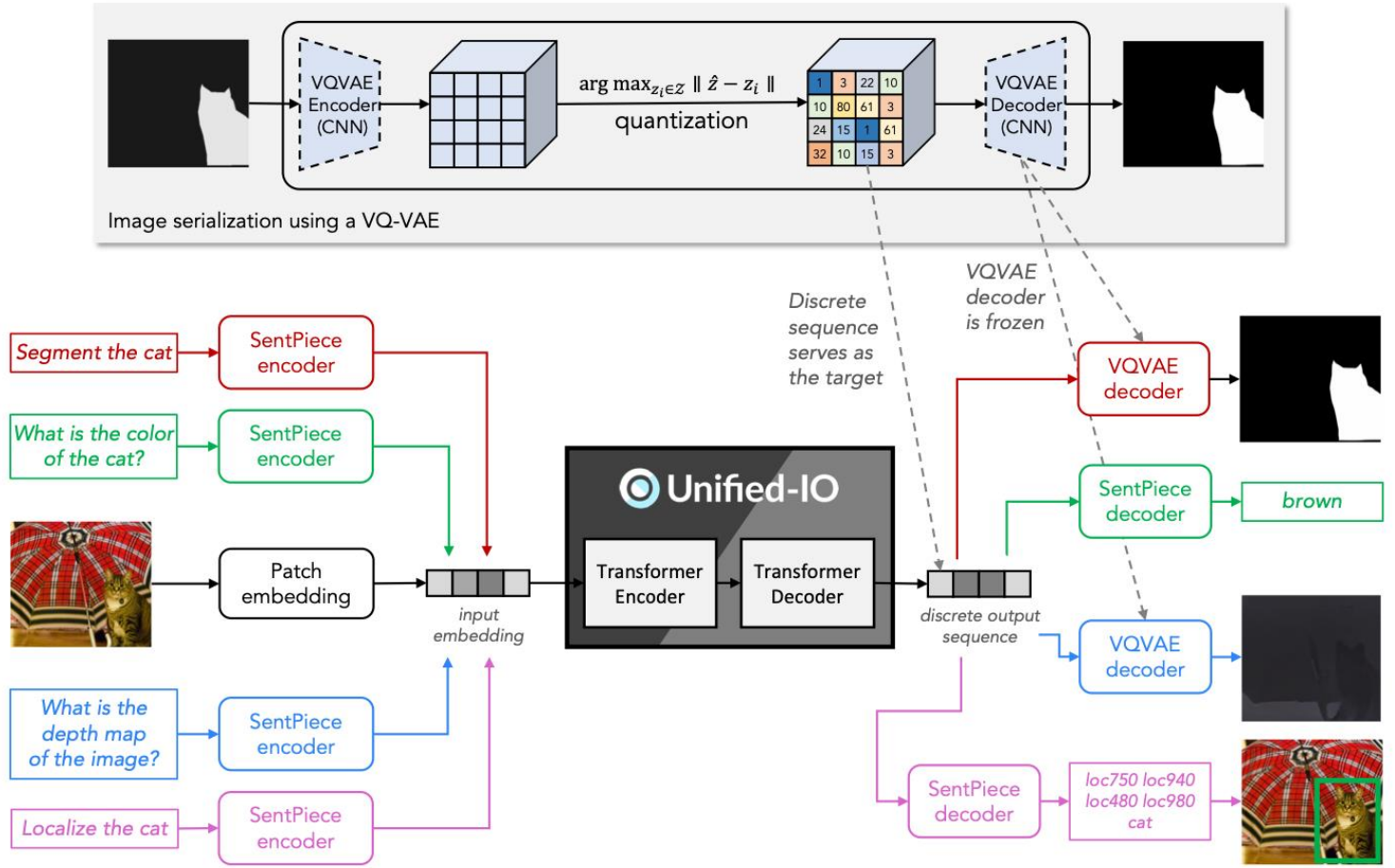
Meta-Transformer (MM Lab)

- All Modals share one Transformer
- Each downstream task has its own head (MLP, decoder…)

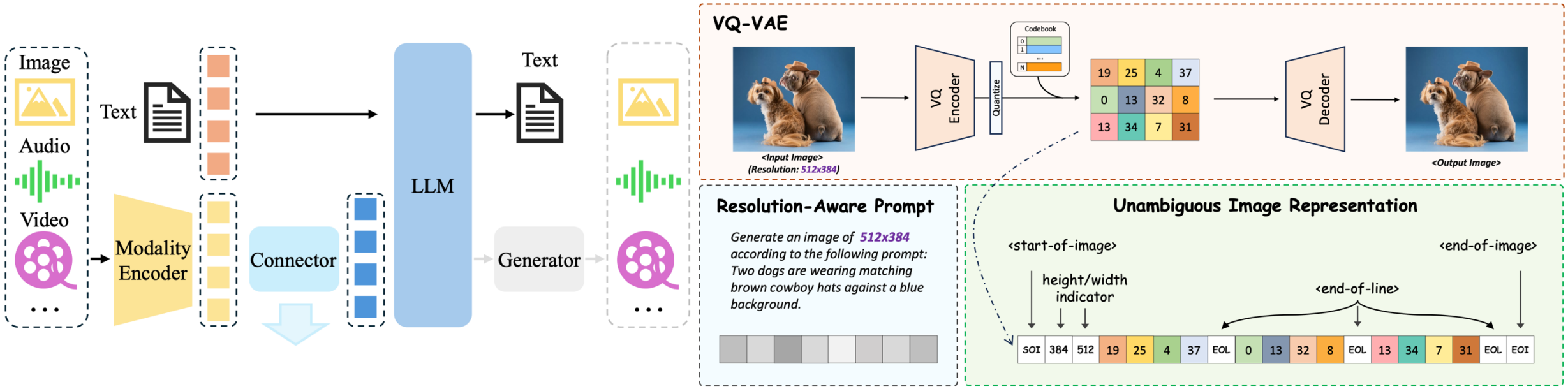


Unified IO (Allen Institute for AI)

- Meta-Transformer
  - Task-based head
- Unified IO:
  - Modal-based head

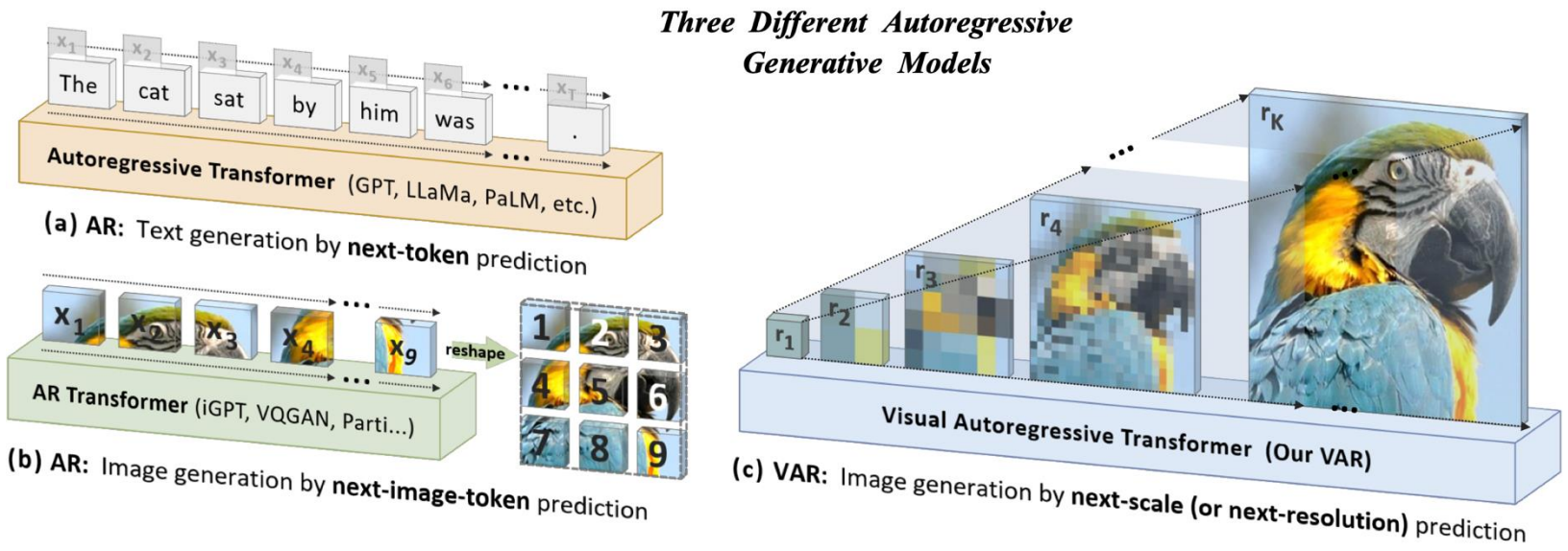


Lumina-mGPT (Shanghai AI Lab)

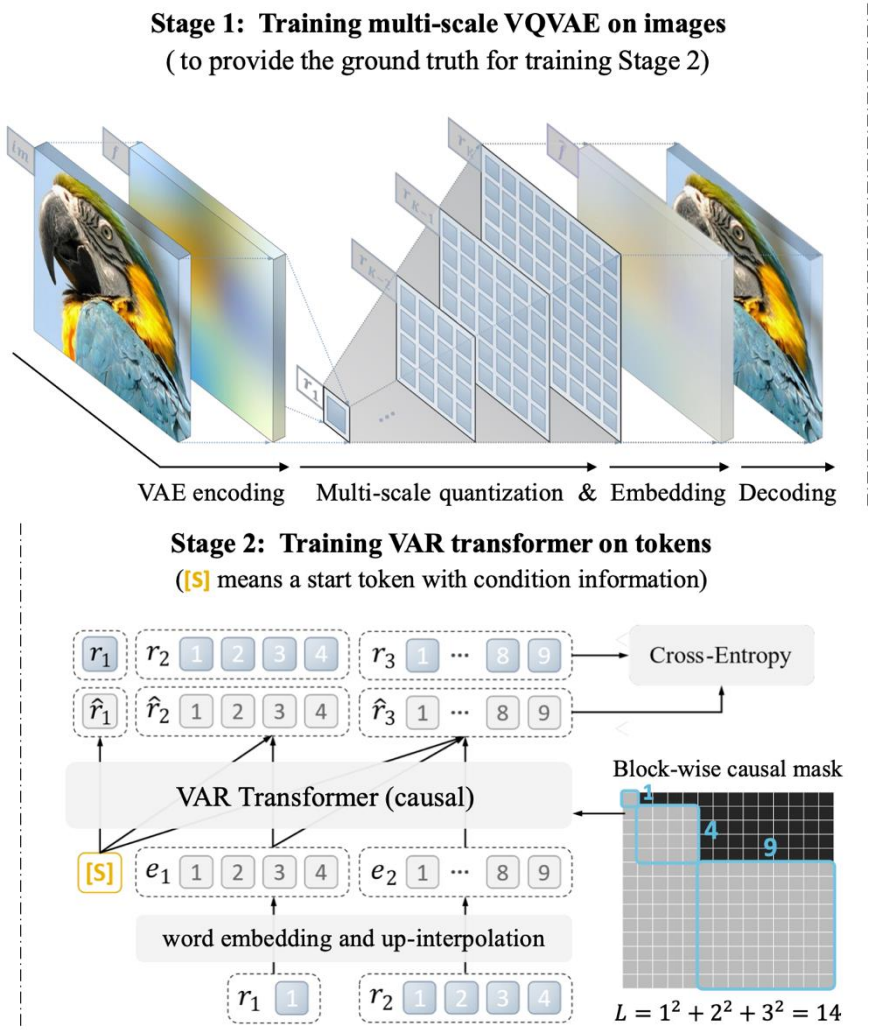


Visual Scale Autoregressive

- **Visual Autoregressive**: Flattens 2D images into a 1D "sentence" and predicts the next tokens or words sequentially.
- **Visual Scale Autoregressive**: Generates entire images at different resolutions in a progressive manner.



VAR (Byte Dance)



VAR uses k VQ-VAE, each corresponding to a different resolution.

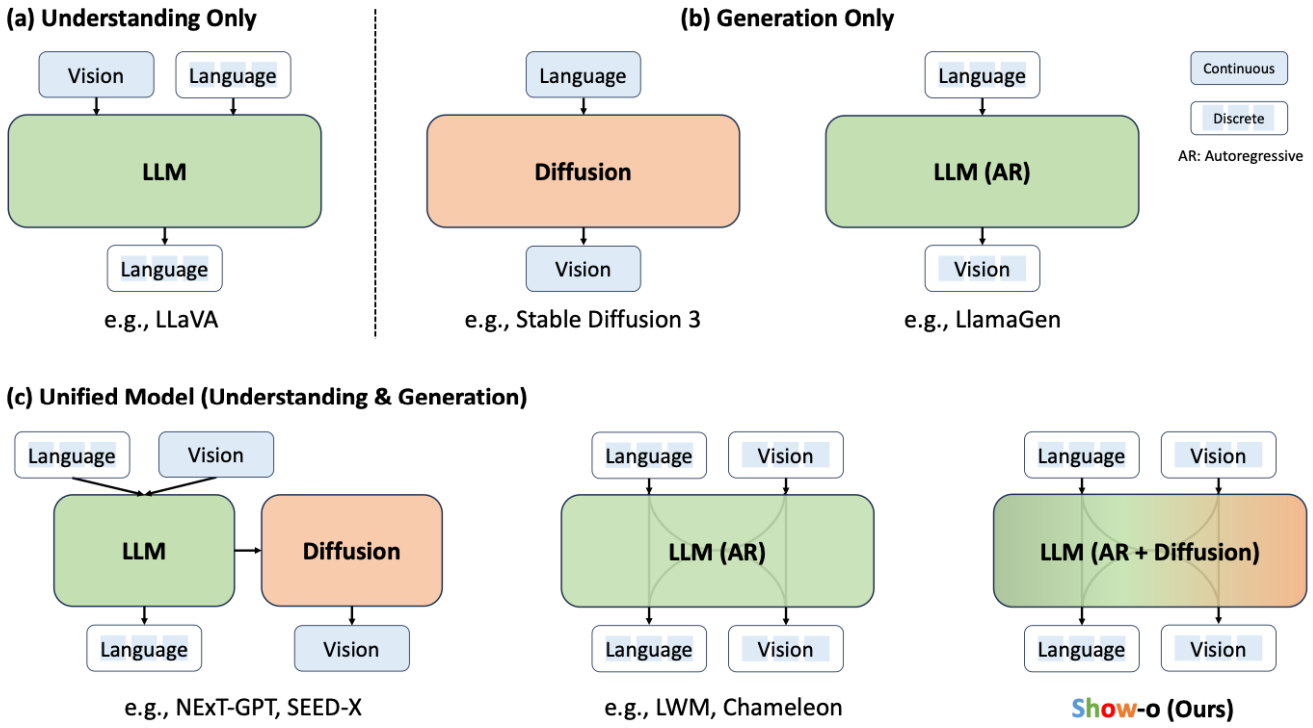
After tokenization, an image is represented by k token maps, with each token map representing a specific resolution.

In VAR, the first k−1 token maps are fed into a Transformer to predict the k-th token map.

All tokens within the k-th token map are predicted simultaneously in parallel.

Visual Diffusion

- Combining AR and Diffusion:
- Based on current technology, Diffusion models are faster (they do not require step-by-step predictions like AR models)
- Representative works: Show-O and Transfusion



Transfusion (Meta)

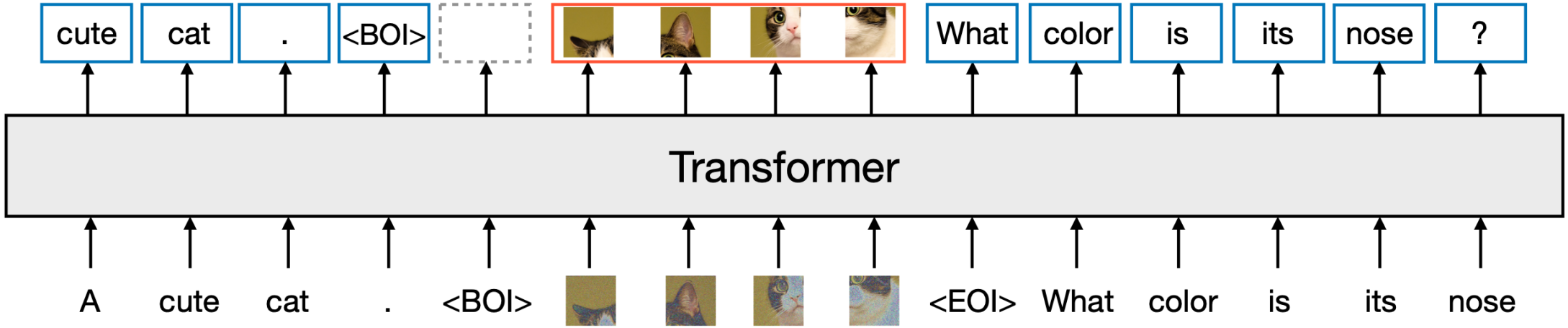
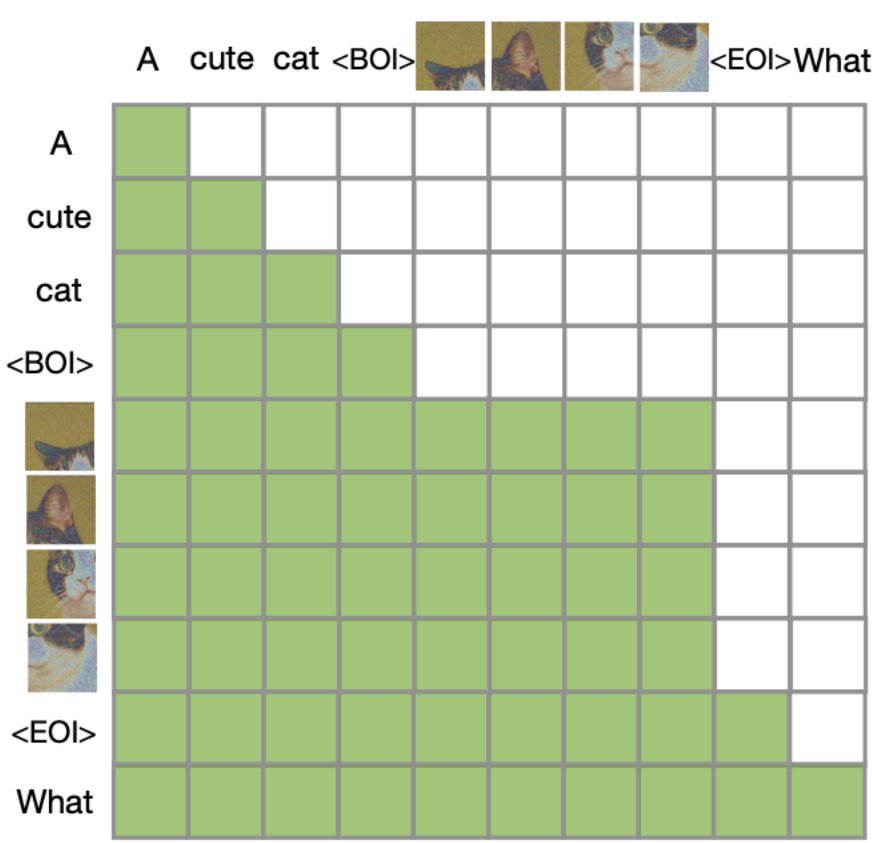


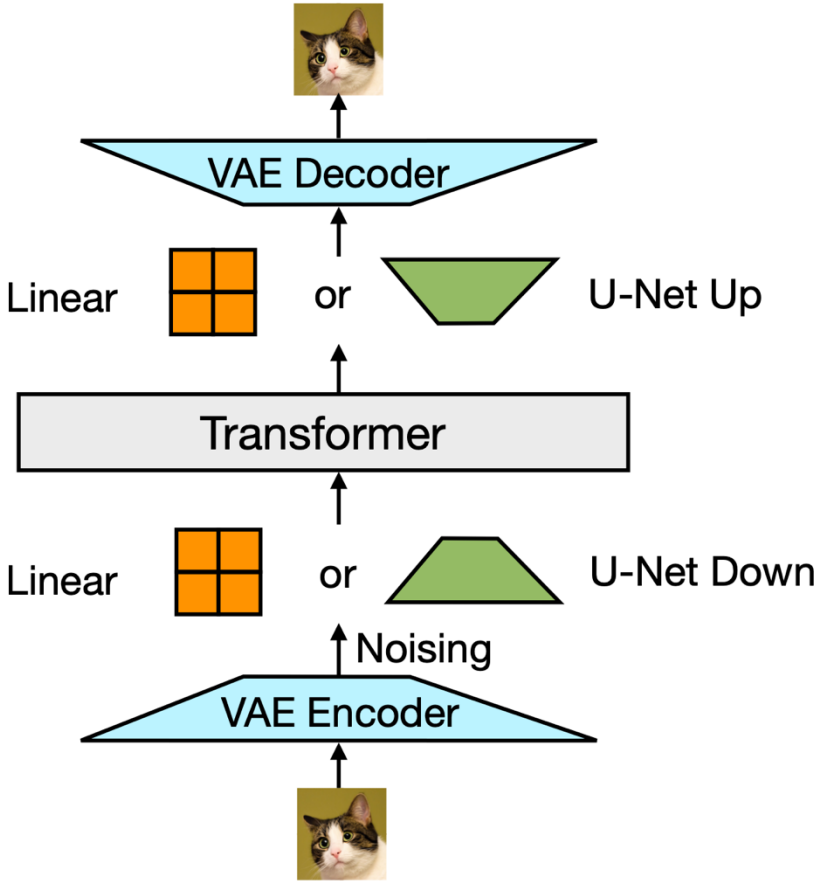
Figure 1: A high-level illustration of Transfusion. A single transformer perceives, processes, and produces data of every modality. Discrete (text) tokens are processed autoregressively and trained on the next token prediction objective. Continuous (image) vectors are processed together in parallel and trained on the diffusion objective. Marker BOI and EOI tokens separate the modalities.

Different Loss for Different Modal

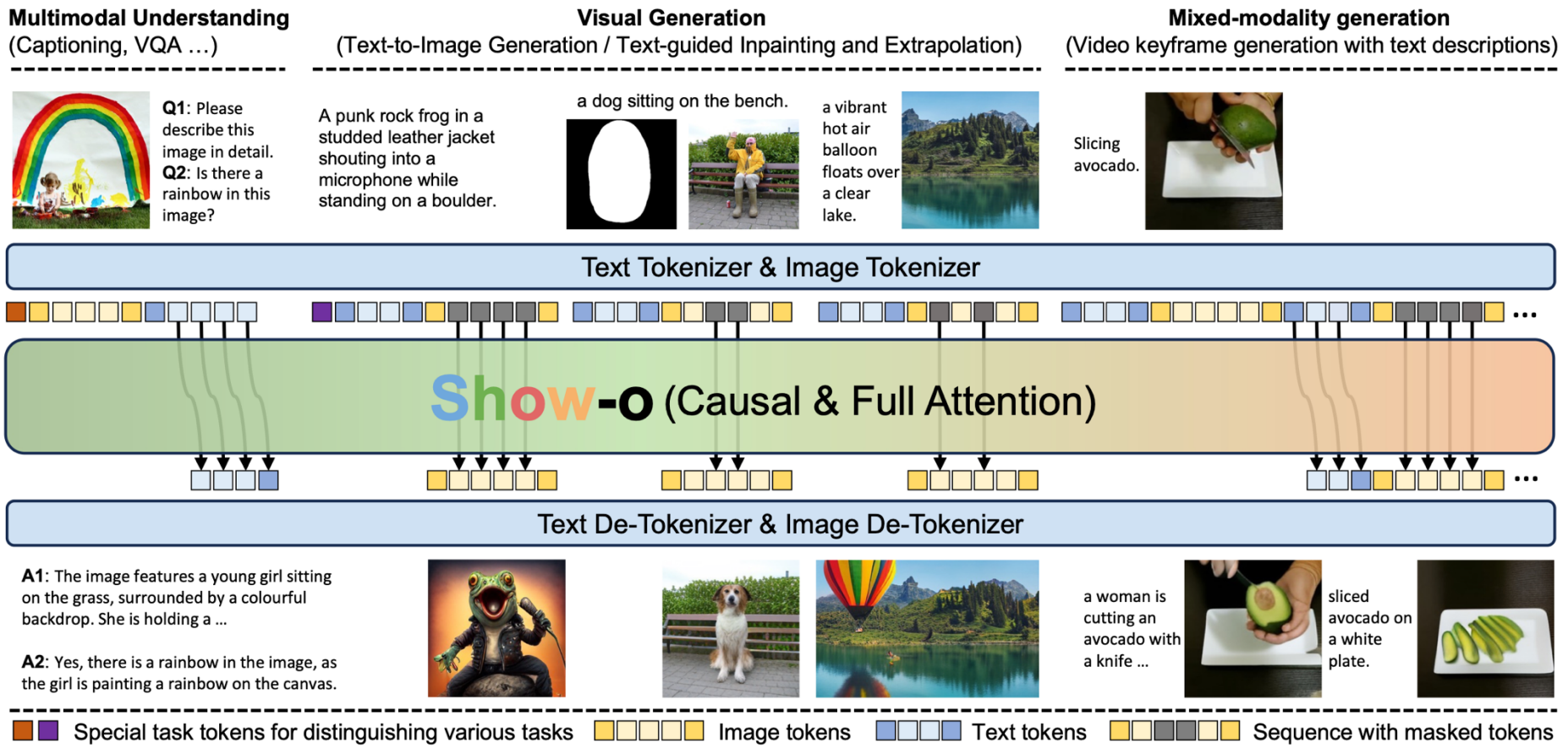
$$\mathcal{L}_{\text{Transfusion}} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{DDPM}}$$

Transfusion (Meta)



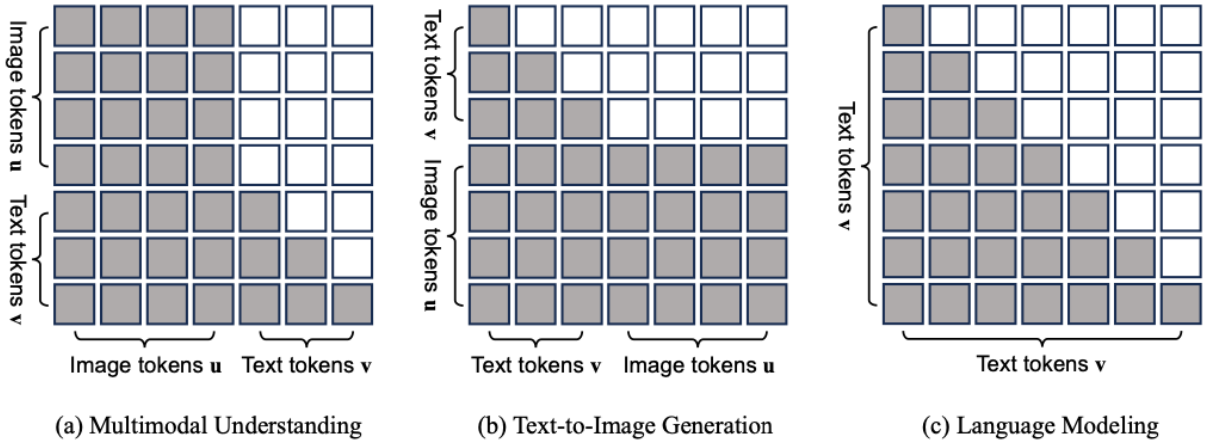


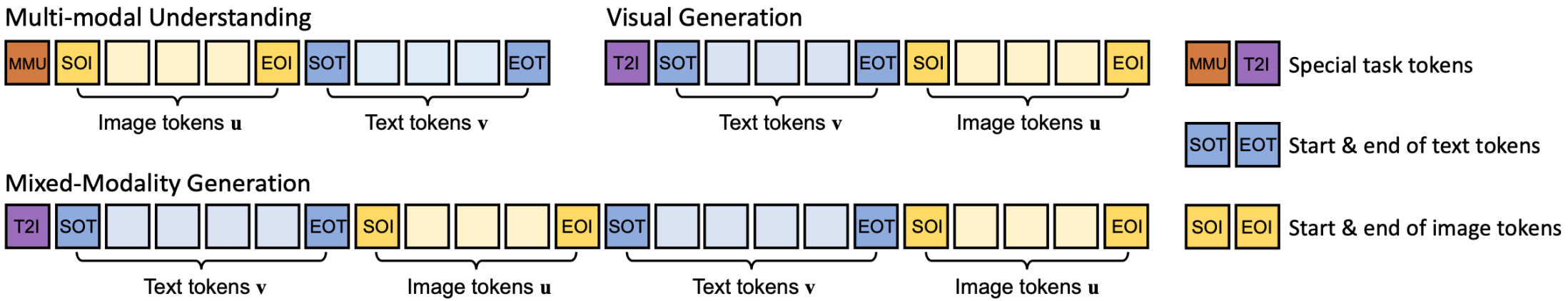
SHOW-O (Byte Dance)



SHOW-O (Byte Dance)

Show-o divides tasks into **understanding** and **generation**, and equips each of the two types of tasks with a predefined **task indicator**, namely *[MMU]* and *[T2I]*



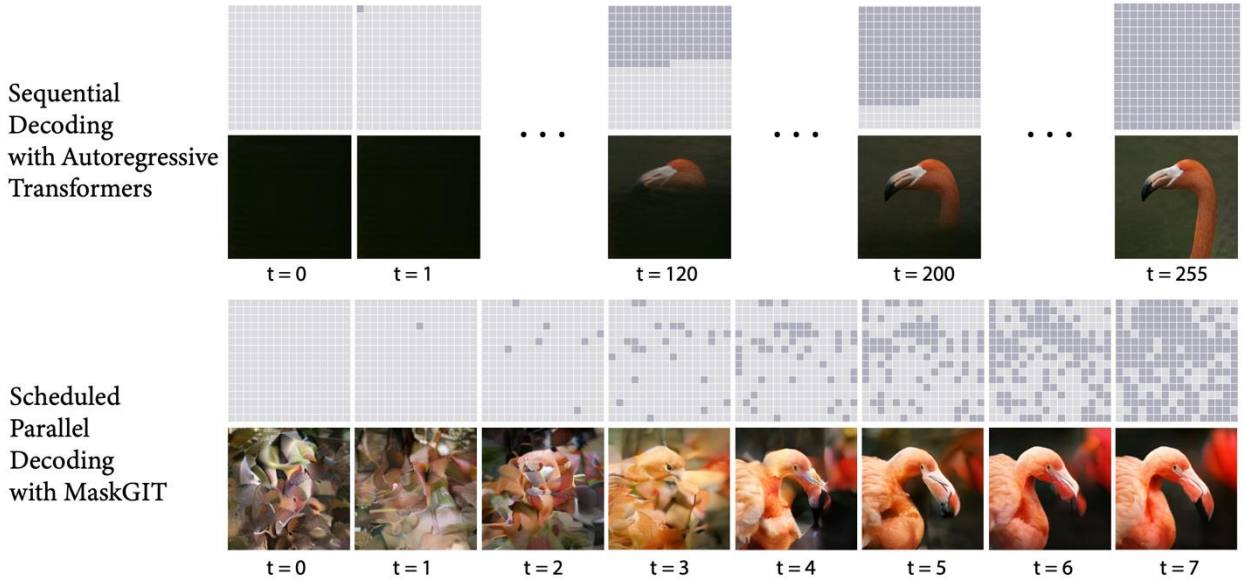


SHOW-O (Byte Dance)

- NTP (Next Token Prediction): For Text
- MTP (Mask Token Prediction): For Image
  - In training, model randomly mask the visual token and predict during training.
  - In inference, The model first generates all the tokens and then iteratively refines the image based on the previous generation.

$$\mathcal{L} = \mathcal{L}_{\text{MTP}} + \alpha \mathcal{L}_{\text{NTP}},$$

$$\mathcal{L}_{\text{MTP}} = \sum_j \log p(u_j | u_*, u_2, \cdots, u_*, u_M, v_1, \cdots, v_N; \Theta).$$



Thank You for Watching!
Q & A