Probabilistic generative models

Multimodal Large Language Models (M-LLMs)

https://tinyurl.com/5y3rkcvr

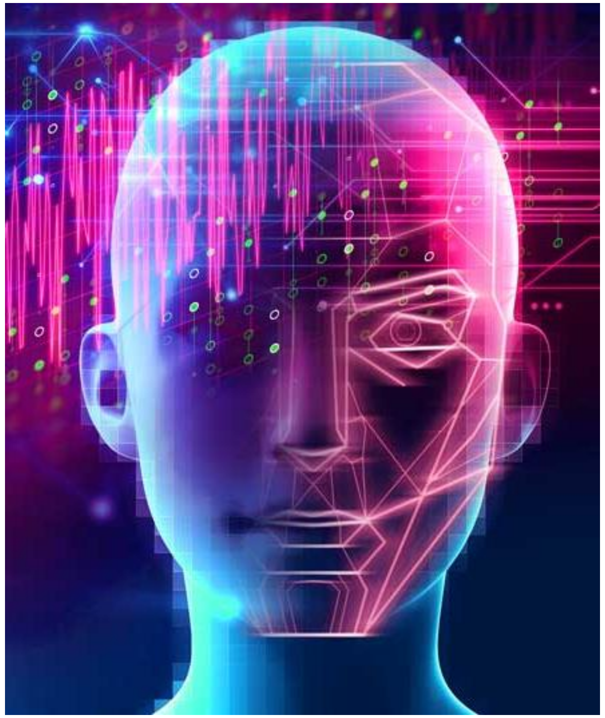
Dossier : https://tinyurl.com/hzwx77ss

Objectives of the Session

- Understand the fundamental principles of Multimodal Large Language Models (M-LLMs).
- Explore the neural network architectures suited for multimodal integration.
- Apply M-LLMs to concrete tasks involving multiple data types.

Introduction to Multimodal Large Language Models

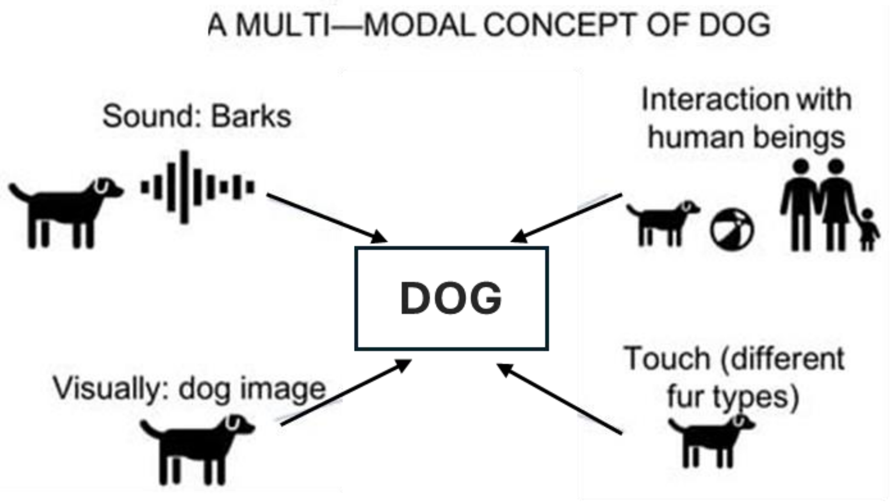
- **Definition and Importance of M-LLMs**
  - **Introduction to Multimodality**
    - **Text**
      - Natural language processing and comprehension.
      - Examples: sentiment analysis, text generation.
    - **Images**
      - Computer vision and image understanding.
      - Examples: object detection, image classification.
    - **Audio**
      - Speech recognition and sound analysis.
      - Examples: voice commands, audio transcription.

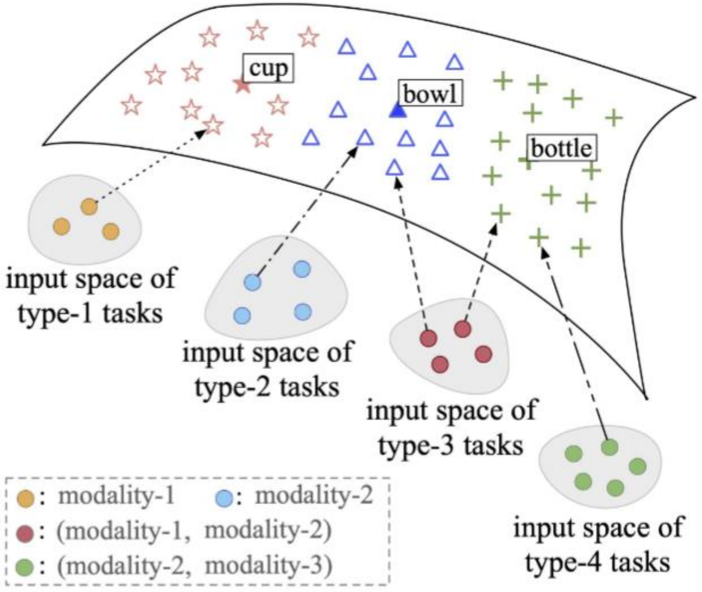


.

Introduction to Multimodal Large Language Models

- **Definition and Importance of M-LLMs**
  - **Importance in holistic understanding of context and data.**
    - **Integrating Multiple Modalities**
      - Provides a more comprehensive understanding of contexts by combining insights from various forms of data.
      - Enhances the model's ability to interpret and generate complex, context-rich information.
    - **Permit Real-World Applications**
      - Enhanced user experiences
      - Improved accuracy in decision-making systems across industries.



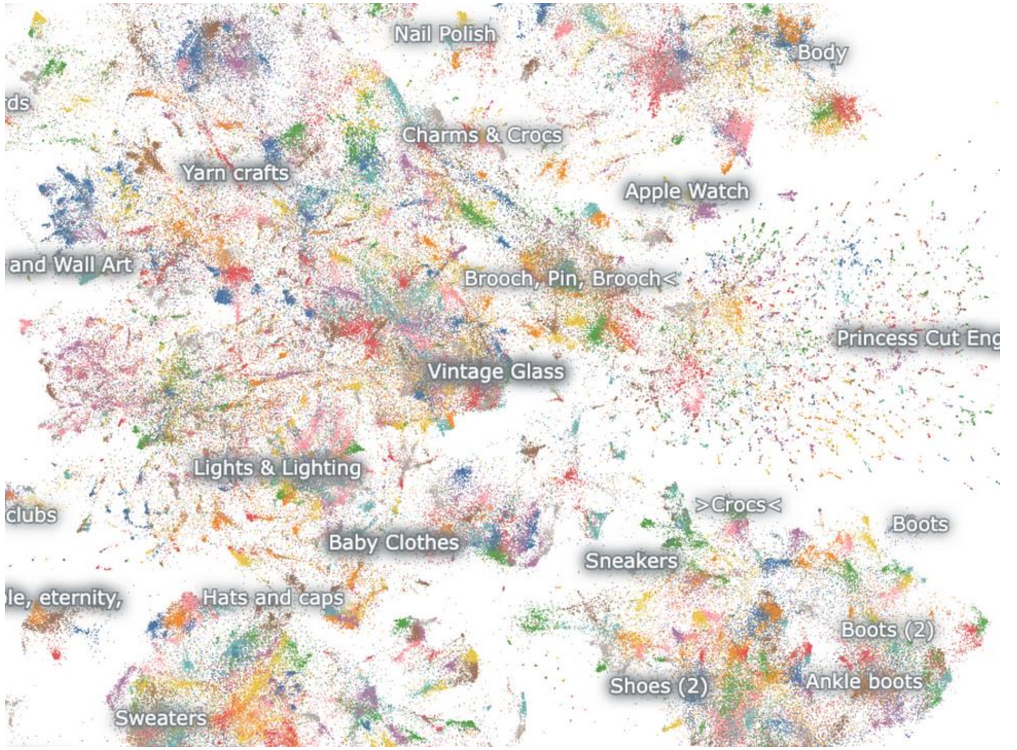


Introduction to Multimodal Large Language Models

- **Applications of M-LLMs**
  - **Image Captioning**
    - Generating descriptive text from images.
    - Applications: Assistive technologies, content creation.
  - **Voice Synthesis**
    - Creating natural-sounding speech from text input.
    - Applications: Virtual assistants, audiobooks.
  - **Video Analysis**
    - Understanding and annotating video content.
    - Applications: Security, autonomous vehicles.

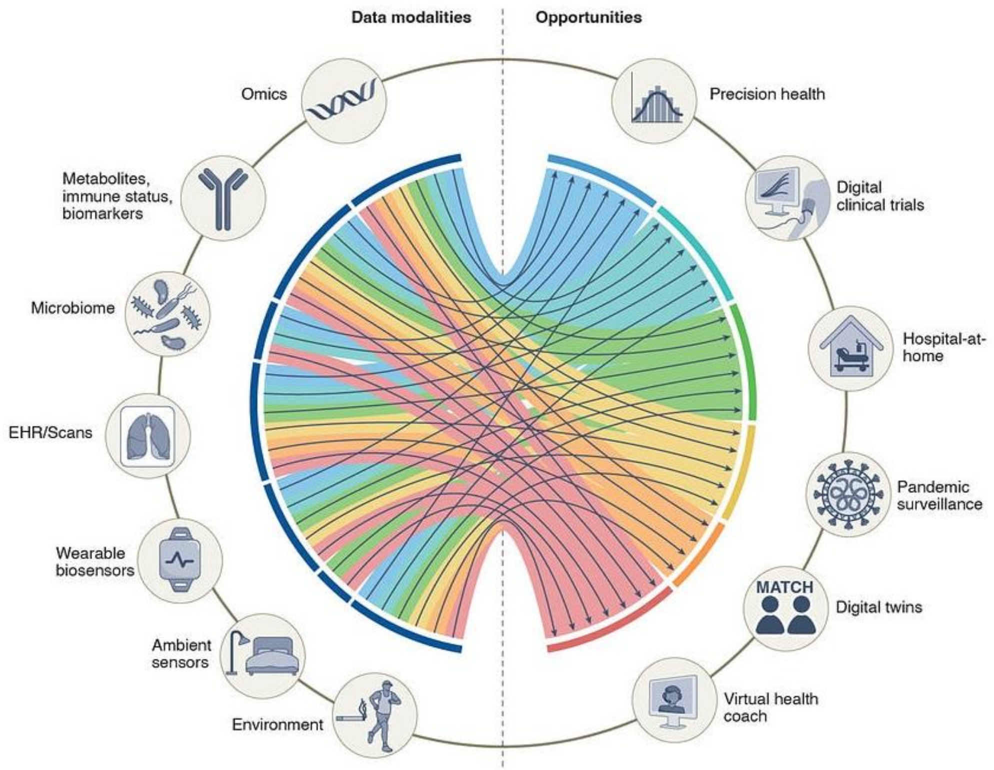
Introduction to Multimodal Large Language Models

- **Successful Case Studies**
  - **OpenAI CLIP**
    - Combines vision and language understanding.
    - Capable of zero-shot classification of images based on textual descriptions.
  - **Google Brain**
    - Advances in multimodal learning and applications.
    - Integration of text, images, and video data to improve search and annotation systems.
  - **OpenAI GPT-4o**
    - Next-generation model leveraging large-scale multimodal data.
    - Enhanced capabilities in language understanding, image recognition, and context integration.



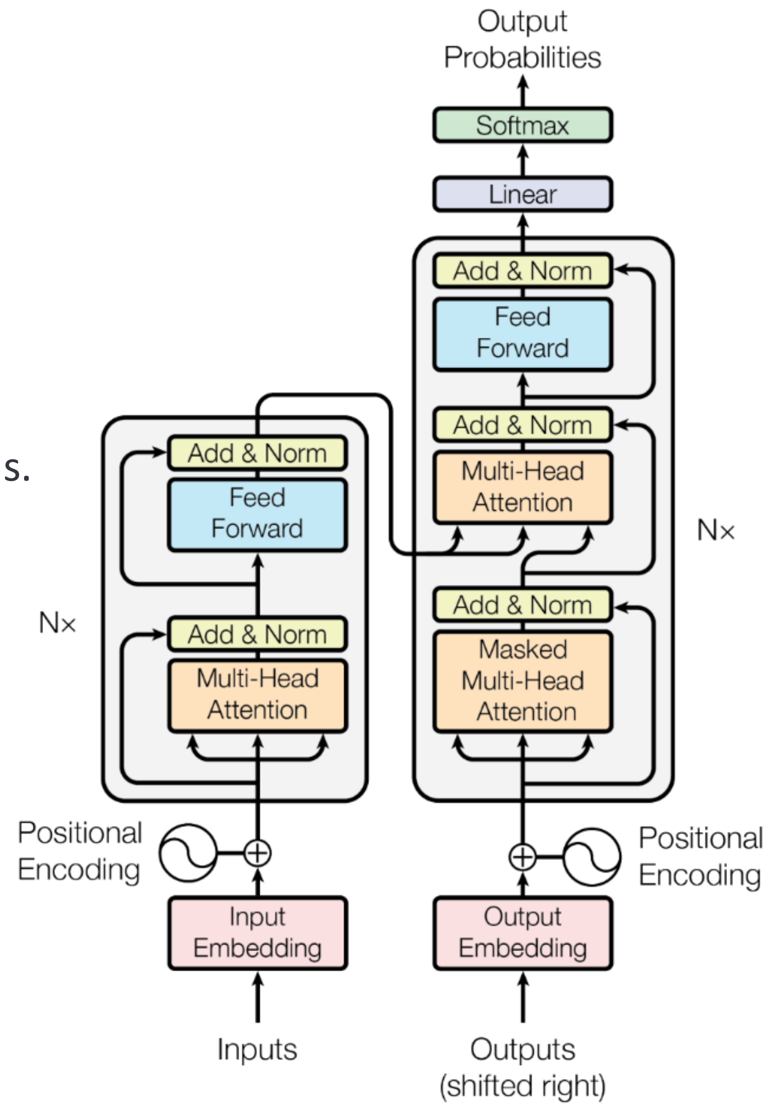
Introduction to Multimodal Large Language Models

- **Impact on Various Industries**
  - **Healthcare**
    - Multimodal diagnostics combining medical images, patient records, and genetic data.
  - **Entertainment**
    - AI-generated multimodal content, enhanced media experiences.
  - **E-commerce**
    - Improved product recommendations by integrating images, descriptions, and reviews.



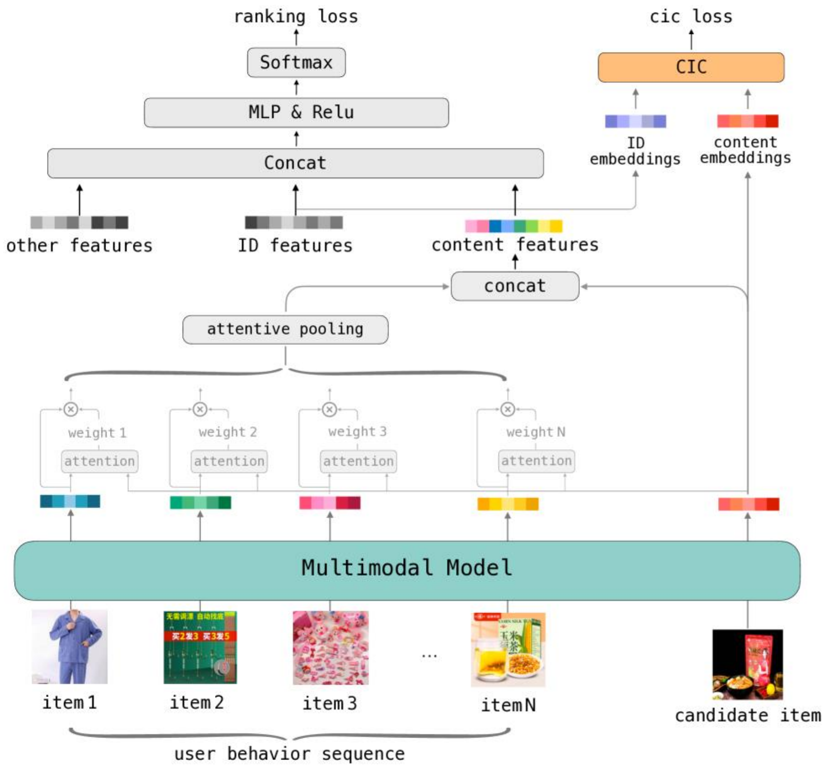
Architectures for Multimodal Integration

- **Neural Network Architectures for M-LLMs**
  - **Introduction to Basic Architectures**
    - **Convolutional Neural Networks (CNNs)**
      - Specialized for processing grid-like data such as images.
      - Features: Local connectivity, weight sharing, and pooling operations.
      - Applications: Image classification, object detection.
    - **Recurrent Neural Networks (RNNs)**
      - Designed for sequence data processing such as text and audio.
      - Features: Temporal dynamics, memory through hidden states.
      - Applications: Language modeling, speech recognition.
    - **Transformers**
      - Utilizes self-attention mechanisms for all types of data.
      - Features: Parallel processing, scalability, contextual learning.
      - Applications: Text generation (e.g., GPT), multimodal tasks.



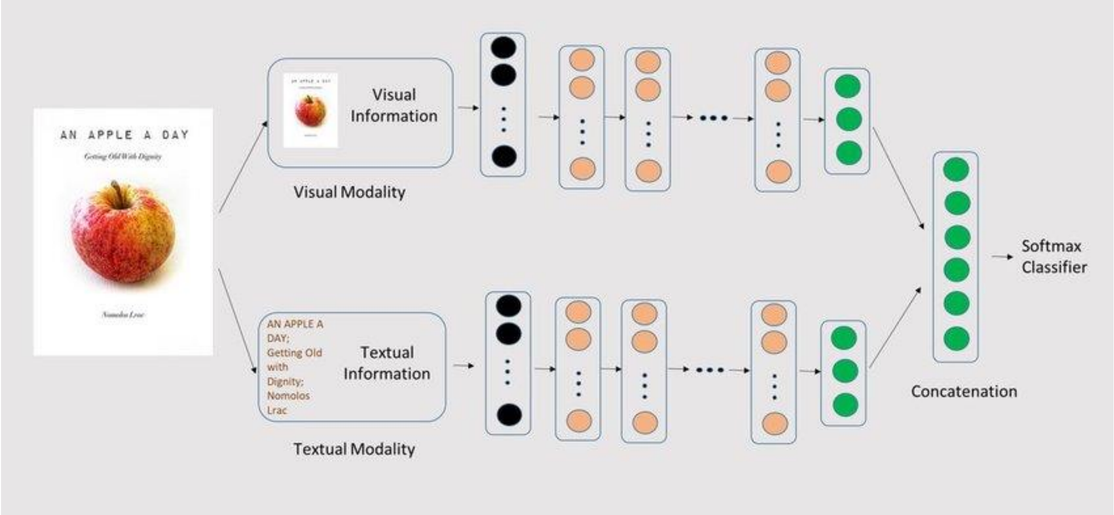
Architectures for Multimodal Integration

- **Neural Network Architectures for M-LLMs**
  - **Adaptation for Multimodal Processing**
    - **Combining Multiple Architectures**
      - Integration of CNNs for visual data and RNNs/Transformers for sequential data.
      - Example: Visual input processed by CNNs, converted into sequential tokens for Transformer processing.
    - **Feature Fusion Techniques**
      - Techniques such as concatenation, attention layers, and cross-modal interactions to merge insights from different data types.
      - Example: Attention layers that weight the importance of different modalities in context-sensitive tasks.
    - **End-to-End Multimodal Models**
      - Unified models that seamlessly process and integrate various data types.
      - Example: Transformers capable of directly ingesting text, images, and audio, enabling comprehensive contextual understanding.



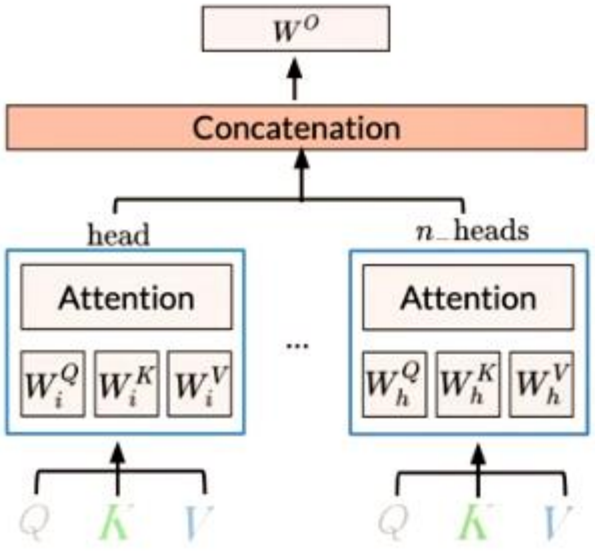
Architectures for Multimodal Integration

- **Multimodal Fusion**
  - **Feature Fusion Techniques**
    - **Concatenation**
      - Simple but effective technique where features from different modalities are combined to form a single feature vector.
      - Example: Combining text embeddings from a Transformer with image embeddings from a CNN.



Architectures for Multimodal Integration

- **Multimodal Fusion**
  - **Feature Fusion Techniques**
    - **Multi-Head Attention**
      - An advanced technique that assigns different weights to different parts of the input based on their relevance in the context.
      - Allows the model to focus on multiple aspects of the data, improving interpretability and performance.
      - Example: Attention heads that simultaneously process visual features and textual context.



Multi-Head Attention Formula

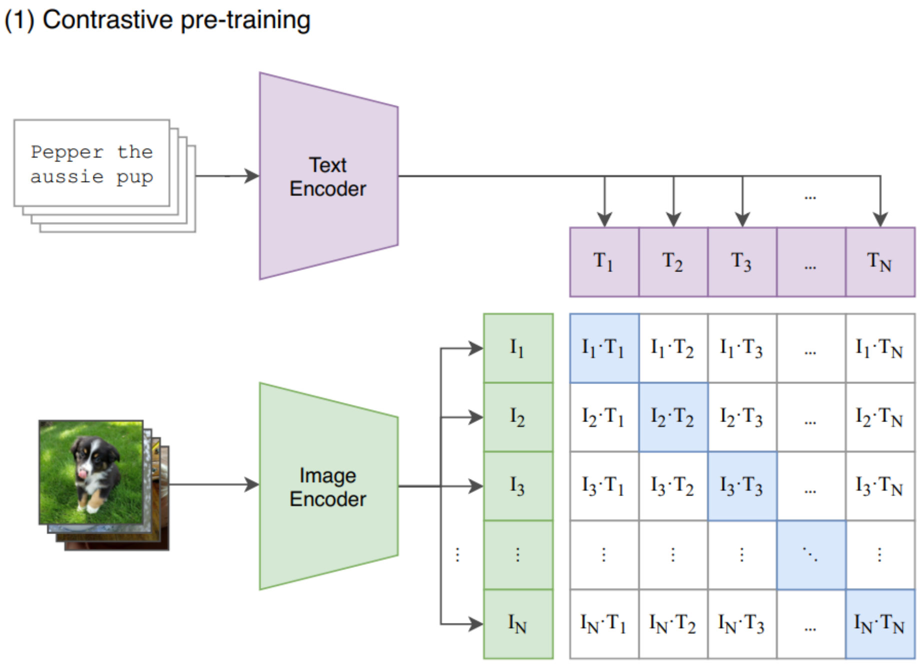
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(h_1, \ldots, h_h)W^0$$

$$\text{where } h_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Each head h_i is the attention function of **Query, Key and Value** with trainable parameters (W_i^Q, W_i^K, W_i^V)

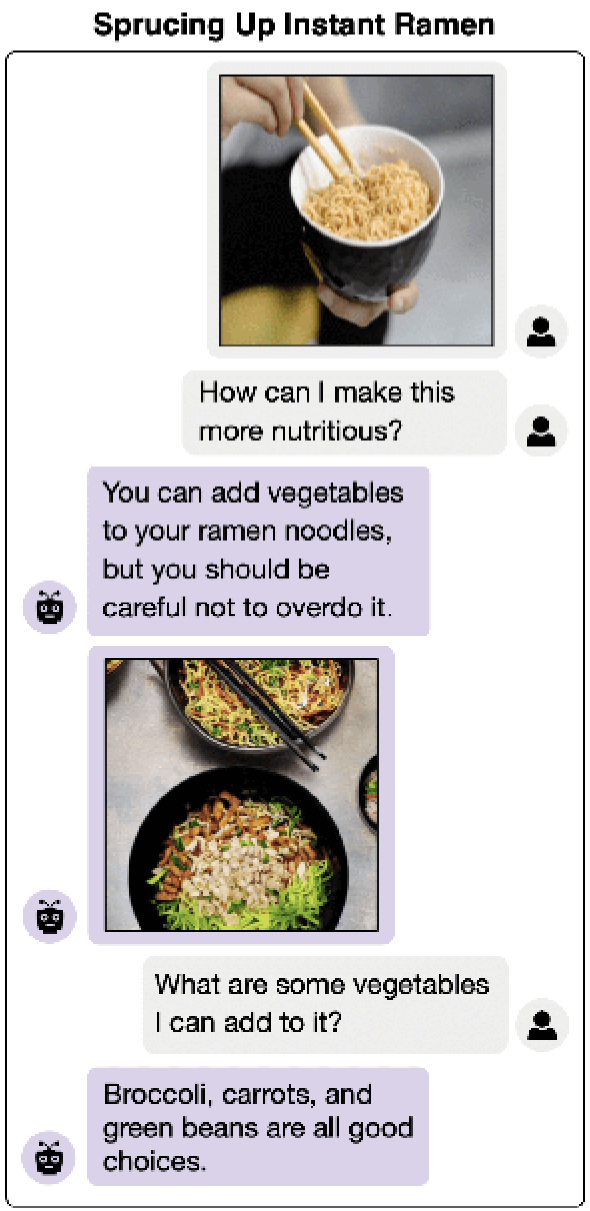
Architectures for Multimodal Integration

- **Practical Example**
  - In-depth study of a real M-LLM architecture : OpenAI CLIP
    - **Overview**
      - CLIP stands for Contrastive Language–Image Pre-training.
      - Developed by OpenAI to understand visual and linguistic concepts simultaneously.
    - **Key Features**
      - **Zero-Shot Learning**:
        - Capable of recognizing objects in images based on textual descriptions without additional training.
      - **Multimodal Learning**:
        - Integrates visual and textual data to create a unified understanding.
    - **Architecture Components**
      - **Image Encoder (CNN/ResNet)**
        - Converts images into feature vectors.
      - **Text Encoder (Transformer)**
        - Converts text descriptions into embeddings.
      - **Contrastive Learning Objective**
        - Aligns images and corresponding text embeddings in a shared multimodal space.



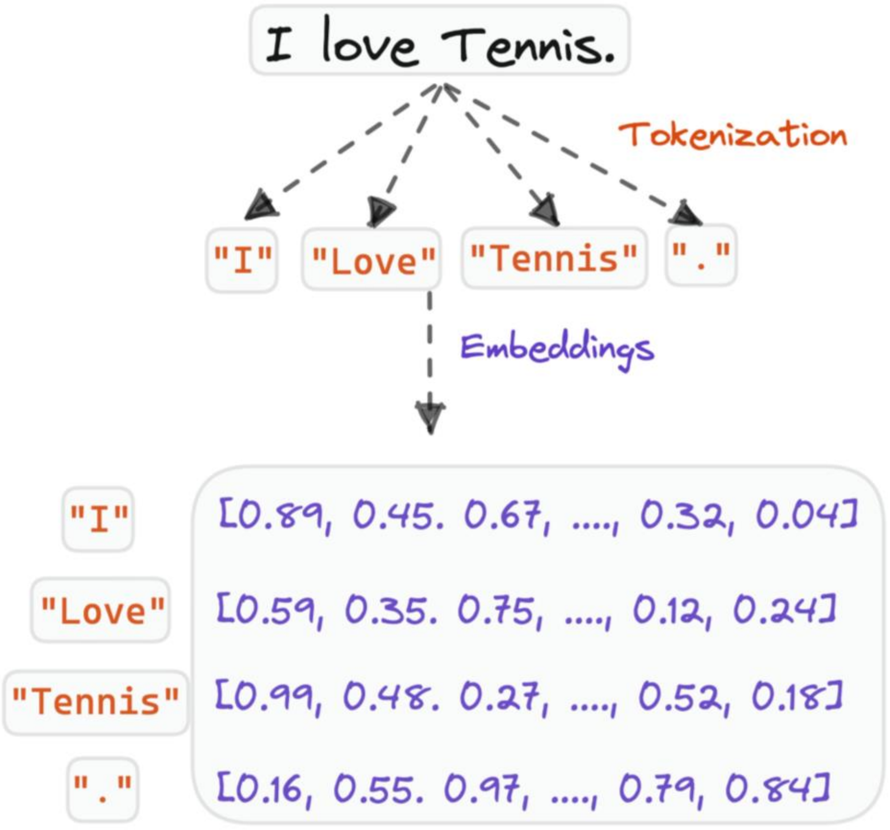
Architectures for Multimodal Integration

- **Practical Example**
  - **Practical Benefits**
    - **Enhanced Search Capability**
      - Type a description to find relevant images.
    - **Content Generation**
      - Automatically generate captions for images.
    - **Cross-Modal Understanding**
      - Powerful applications in AI-driven multimedia analysis and generation.



Training and Optimizing M-LLMs

- **Preprocessing Multimodal Data**
  - Preprocessing techniques for each modality
    - **Text Data**
      - **Tokenization**: Breaking down text into words, subwords, or characters.
        - Tools: NLTK, Spacy, BERT Tokenizer.
        - Example: "The quick brown fox" -> ["The", "quick", "brown", "fox"]
      - **Embedding**: Converting tokens into vectors.
        - Methods: Word2Vec, GloVe, BERT embeddings.
        - Purpose: Capture semantic meaning of text.
      - **Cleaning**: Removing unwanted characters, stop words, and noise.
        - Techniques: Lowercasing, removing punctuation, stemming/lemmatization.



Training and Optimizing M-LLMs

- **Preprocessing Multimodal Data**
  - Preprocessing techniques for each modality
    - **Image Data**

•**Resizing**: Scaling images to a consistent size.

•Tools: OpenCV, PIL.

•Example: Reshape all images to 224x224 pixels for model input.

•**Normalization**: Scaling pixel values to a standard range (e.g., [0, 1] or [-1, 1]).

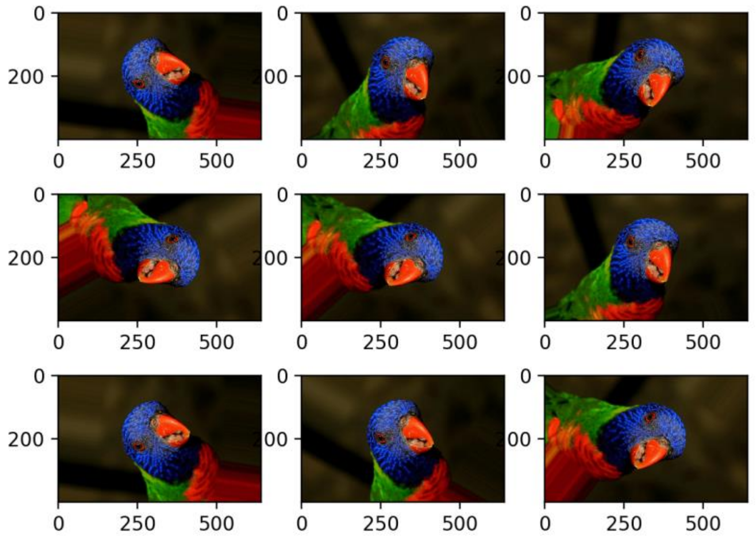
•Purpose: Ensure consistency across different images.

•Technique: (pixel_value - mean) / standard_deviation

•**Data Augmentation**: Enhancing dataset diversity with transformations.

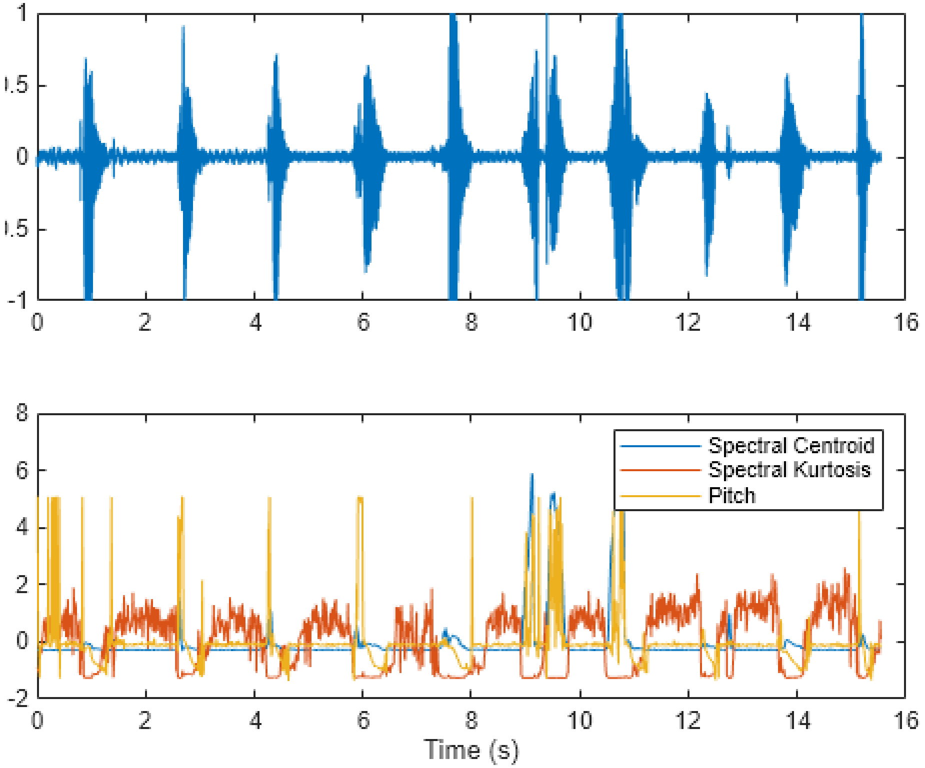
•Methods: Rotation, flipping, cropping, color adjustments.

•Tools: TensorFlow’s ImageDataGenerator, Albumentations.



Training and Optimizing M-LLMs

- **Preprocessing Multimodal Data**
  - Preprocessing techniques for each modality
    - **Audio Data**
      - **Resampling**: Standardizing audio sample rates.
        - Tools: Librosa, PyDub.
        - Example: Resample all audio clips to 16kHz.
      - **Feature Extraction**: Converting raw audio to feature representation.
        - Methods: Mel-frequency cepstral coefficients (MFCCs), spectrograms.
        - Purpose: Capture essential audio characteristics for the model.
      - **Noise Reduction**: Filtering out background noise.
        - Techniques: Spectral subtraction, wavelet denoising.
        - Tools: Audacity, noise-reduction libraries in Python.



Training and Optimizing M-LLMs

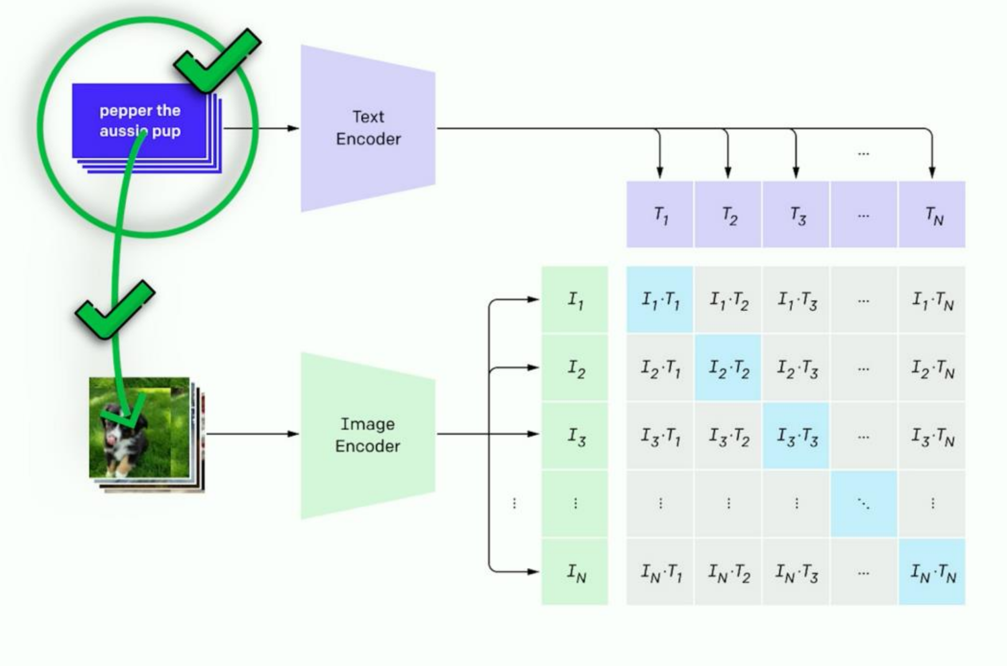
- **Preprocessing Multimodal Data**
  - Data normalization and standardization.
    - **Importance of Normalization and Standardization**
      - Consistent Data Range: Helps in achieving uniformity across different datasets.
      - Improved Model Performance: Models train faster and more effectively on normalized and standardized data.
      - Reduced Bias and Variance: Mitigates the effect of outliers and scales features more appropriately.
    - **Normalization**
      - Definition: Scaling data to a range of [0, 1] or [-1, 1].
      - Applications: Ideal for image data and scenarios where feature scales vary drastically.
    - **Standardization**
      - Definition: Scaling data to zero mean and unit variance.
      - Applications: Suitable for many machine learning models which assume data is centered around zero.
    - **Practical Example**
      - Image Data Normalization:
        - Transforming pixel values: (pixel_value / 255.0) to scale between 0 and 1.
      - Text Data Standardization:
        - Standardizing word embeddings: Center vectors by subtracting the mean embedding.

$$\text{Normalized Value} = \frac{(X - \min)}{(\max - \min)}$$

$$\text{Standardized Value} = \frac{(X - \mu)}{\sigma}$$

Training and Optimizing M-LLMs

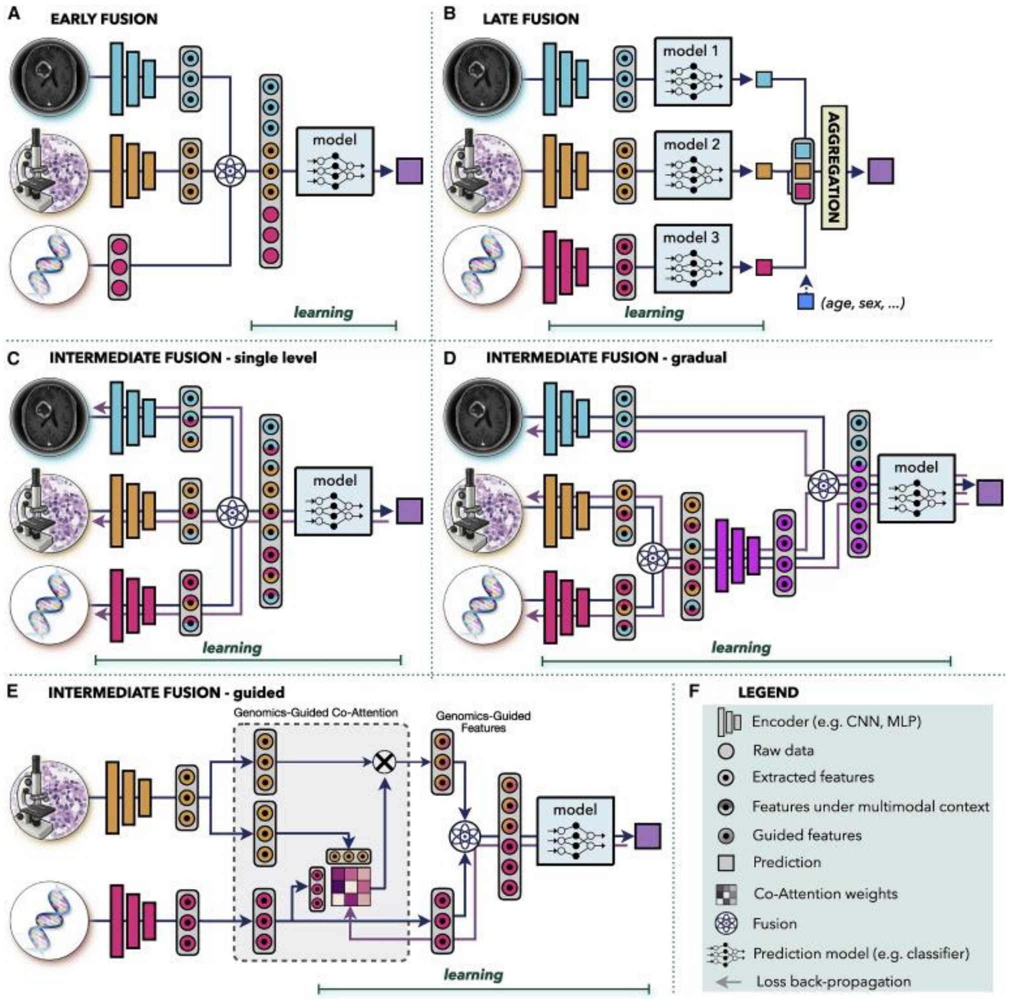
- **Training Strategies**
  - Methods for preparing multimodal datasets.
    - **Data Collection:** Ensuring diversity and relevance in text, image, and audio sources.
    - **Data Annotation:** Techniques for labeling multimodal data (manual, semi-supervised, automated).
    - **Data Synchronization:** Aligning data across different modalities (e.g., timestamps for video and audio tracks).
    - **Dataset Splitting:** Considerations for training, validation, and test sets in multimodal contexts.
  -



[Clip model]

Training and Optimizing M-LLMs

- **Training Strategies**
  - Multimodal Data Integration Techniques
    - **Early Integration:** Combining features at an early stage (data level).
    - **Intermediate Integration:** Fusion occurs at the feature extraction level.
    - **Late Integration:** Features are blended after independent processing.
    - **Hybrid Approaches:** Combining two or more integration strategies for robustness.



Training and Optimizing M-LLMs

- **Training Strategies**
  - Loss (objective) strategies suited for M-LLMs.
    - **Cross-entropy Loss:** Common for classification tasks in multimodal setups.
    - **Contrastive Loss:** Useful for tasks where models learn from pairs of similar and dissimilar data points.
    - **Triplet Loss:** Extends contrastive loss by comparing an anchor to both positive and negative examples.
    - **Custom Loss Functions:** Designing loss functions that can handle the peculiarities of multimodal data (e.g., balancing the influence of different modalities).

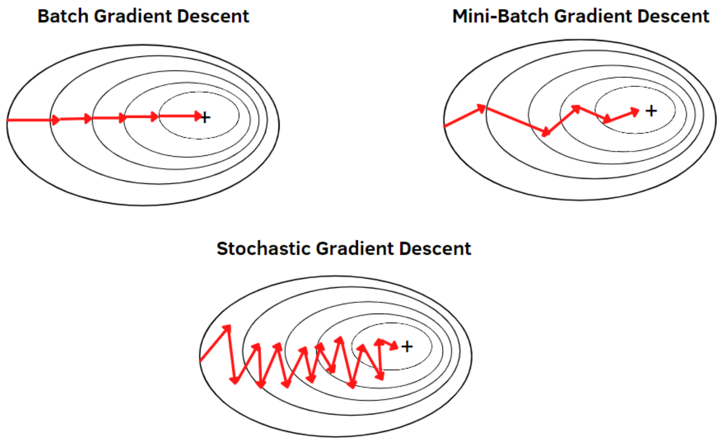
$$H(P^* | P) = -\sum_i \underbrace{P^*(i)}_{\text{TRUE CLASS DISTRIBUTION}} \log \underbrace{P(i)}_{\text{PREDICTED CLASS DISTRIBUTION}}$$

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

Training and Optimizing M-LLMs

- **Training Strategies**
  - Optimization Techniques for M-LLMs
    - **Gradient Descent Variants:** Adaptive techniques like Adam, RMSprop for handling multimodal data efficiently.
    - **Regularization:** Methods like dropout, L2 regularization to prevent overfitting on multimodal data.
    - **Learning Rate Schedules:** Importance of adaptive learning rates in training M-LLMs.
    - **Early Stopping:** Monitoring validation loss to prevent overtraining.

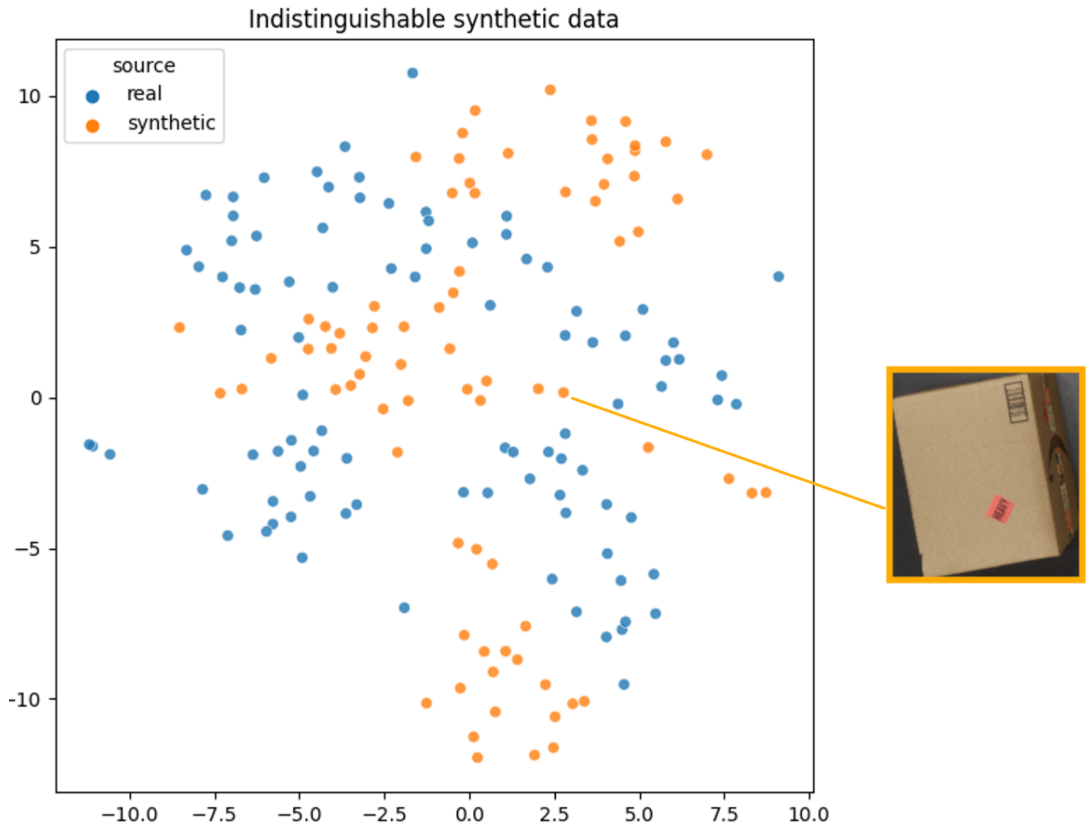


Training and Optimizing M-LLMs

- **Challenges and Solutions**
  - Common issues:
    - Brief introduction to the complexity of multimodal data training.
    - **Data Imbalance:** Different quantities or quality of data across modalities.
    - **Data Corruption:** Errors in data collection or transfer that affect model performance.
    - **Noise:** Unwanted alterations in data (e.g., background noise in audio data, visual artifacts in image data).
    - **Missing Modalities:** Occasionally, one or more modalities may be missing or incomplete.

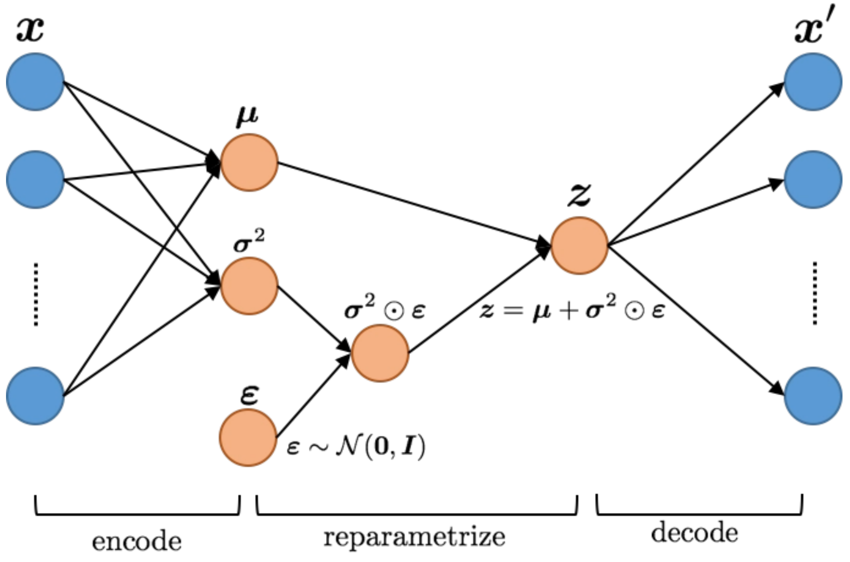
Training and Optimizing M-LLMs

- **Challenges and Solutions**
  - Addressing Data Imbalance in M-LLMs.
    - **Resampling Techniques:** Oversampling minority modalities or undersampling majority modalities.
    - **Synthetic Data Generation:** Using techniques like SMOTE for modalities where data is scarce.
    - **Weighted Loss Functions:** Assigning higher weights to underrepresented modalities during training.



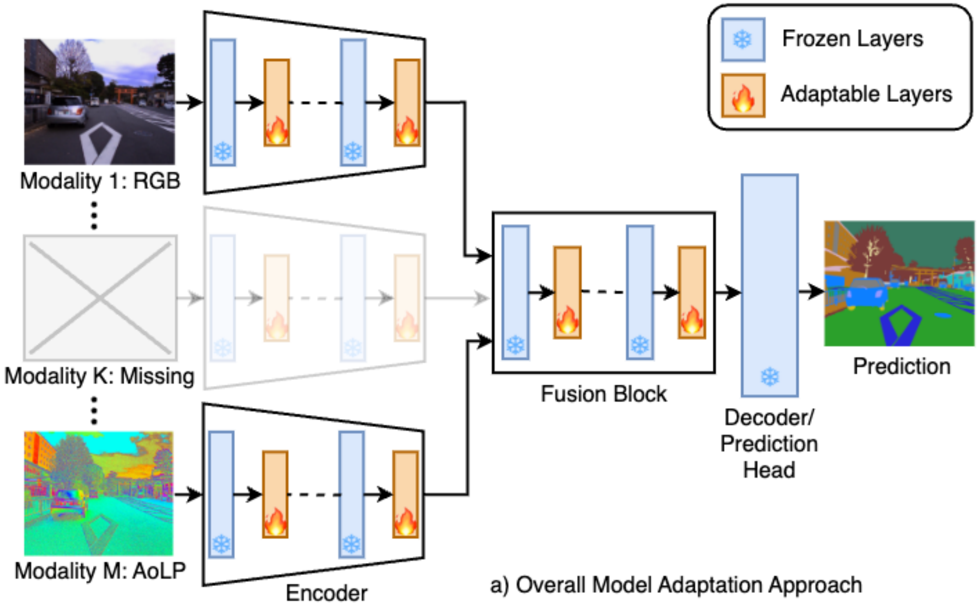
Training and Optimizing M-LLMs

- **Challenges and Solutions**
  - Mitigating Data Corruption and Noise
    - **Data Cleaning Techniques:** Identifying and correcting corrupt data entries.
    - **Noise Filtering:** Applying filters or preprocessing techniques to reduce noise without losing critical data (e.g., Fourier transforms for audio).
    - **Robust Training Approaches:** Training models to be less sensitive to noise and corruption (e.g., using noise-injection during training).



Training and Optimizing M-LLMs

- **Challenges and Solutions**
  - Strategies for Missing Modalities in M-LLMs
    - **Imputation Techniques:** Estimating missing modalities using statistical methods or predictive models.
    - **Flexible Architectures:** Designing models that can handle occasional missing data without performance degradation.
    - **Dynamic Adjustments:** Online learning techniques to adapt to data with varying modal availability.



Practical Application: Implementing a Mini M-LLM Project

- **Mini-project: Text-Image Integration**
  - Overview of the Text-Image Integration Mini-Project
    - Project Goal: To develop a basic M-LLM that integrates text and image data to perform tasks such as image captioning or textual description-based image retrieval.
    - Expected Outcome: Understand the process of M-LLM creation from data handling to model evaluation.

.

Practical Application: Implementing a Mini M-LLM Project

- **Mini-project: Text-Image Integration**
  - Data Handling for Text-Image Integration
    - **Data Loading:**
      - Sources for text and image data.
      - Using data loaders in TensorFlow/Keras or PyTorch.
    - **Preprocessing Techniques:**
      - Image preprocessing: resizing, normalization.
      - Text preprocessing: tokenization, vectorization.
    - Importance of aligning text and image data pairs accurately.

.

Practical Application: Implementing a Mini M-LLM Project

- **Mini-project: Text-Image Integration**
  - Constructing the Model
    - Architecture Overview:
      - Image Branch: Utilize Convolutional Neural Networks (CNNs).
      - Text Branch: Employ Recurrent Neural Networks (RNNs) or Transformers.
      - Fusion Technique: Concatenation of last hidden layers from both branches.
    - Configuring the model in TensorFlow/Keras or PyTorch:
      - Code snippets for model architecture.
      - Tips for effective merging of modalities.

Practical Application: Implementing a Mini M-LLM Project

- **Mini-project: Text-Image Integration**
  - Model Training and Optimization
    - Setting up the training loop: defining epochs, batch size, and learning rate.
    - Selection of loss function and optimizers suited for multimodal learning.
    - Monitoring training progress through callbacks or custom logging metrics.
  - .

Practical Application: Implementing a Mini M-LLM Project

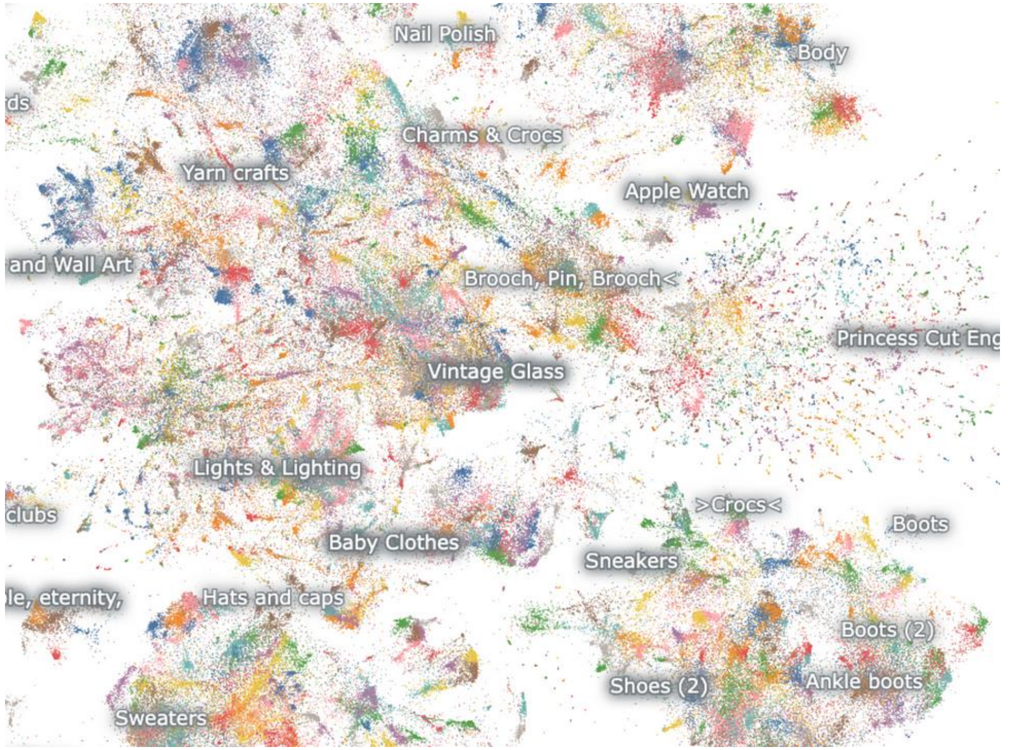
- **Mini-project: Text-Image Integration**
  - Model Evaluation and Result Visualization
    - Evaluation Metrics:
      - Accuracy, precision, and recall for alignment assessment.
      - Custom metrics (if any) tailored for specific project goals.
    - Visualization Techniques:
      - Plotting loss and accuracy curves.
      - Visualization of image inputs with corresponding textual outputs.
    - Discussion on results interpretation and potential areas of improvement.

Discussion and Q&A

- **Critical Analysis of M-LLMs**
  - Critical Analysis of Multimodal Language Models
    - Introduction to the critical analysis section.
    - Objective: To appraise the capabilities, recognize the limitations, and ponder on the ethical concerns of M-LLMs.
  - .

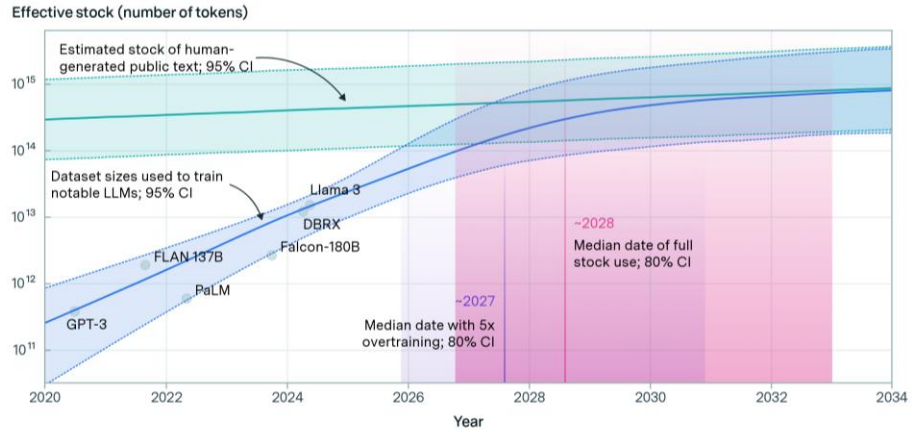
Discussion and Q&A

- **Critical Analysis of M-LLMs**
  - Potential Advantages of M-LLMs
    - **Enhanced Accuracy and Performance:** Improved results in complex tasks involving multiple data types.
    - **Better Context Understanding:** Ability to integrate contextual information from multiple sources leading to richer interpretations.
    - **Wider Applicability:** Extensive use across various industries including healthcare, automotive, entertainment, and more.
    - **Innovative Applications:** Potential for new applications such as emotion recognition, advanced human-computer interaction, etc.



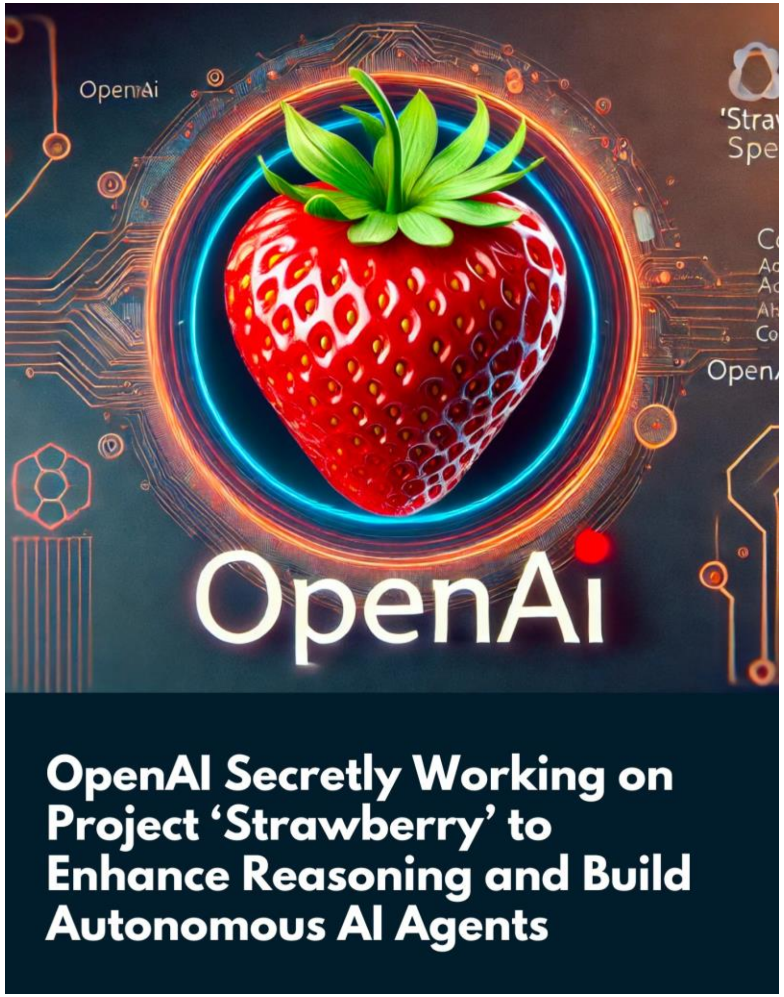
Discussion and Q&A

- **Critical Analysis of M-LLMs**
  - Challenges and Limitations
    - **Complexity in Training and Maintenance:** High computational cost and technical complexity.
    - **Data Requirements:** Massive amounts of diverse, annotated multimodal data needed.
    - **Integration Challenges:** Effective fusion of different modal types remains technically demanding.
    - **Generalization Issues:** Difficulty in generalizing the learning across different tasks or datasets.



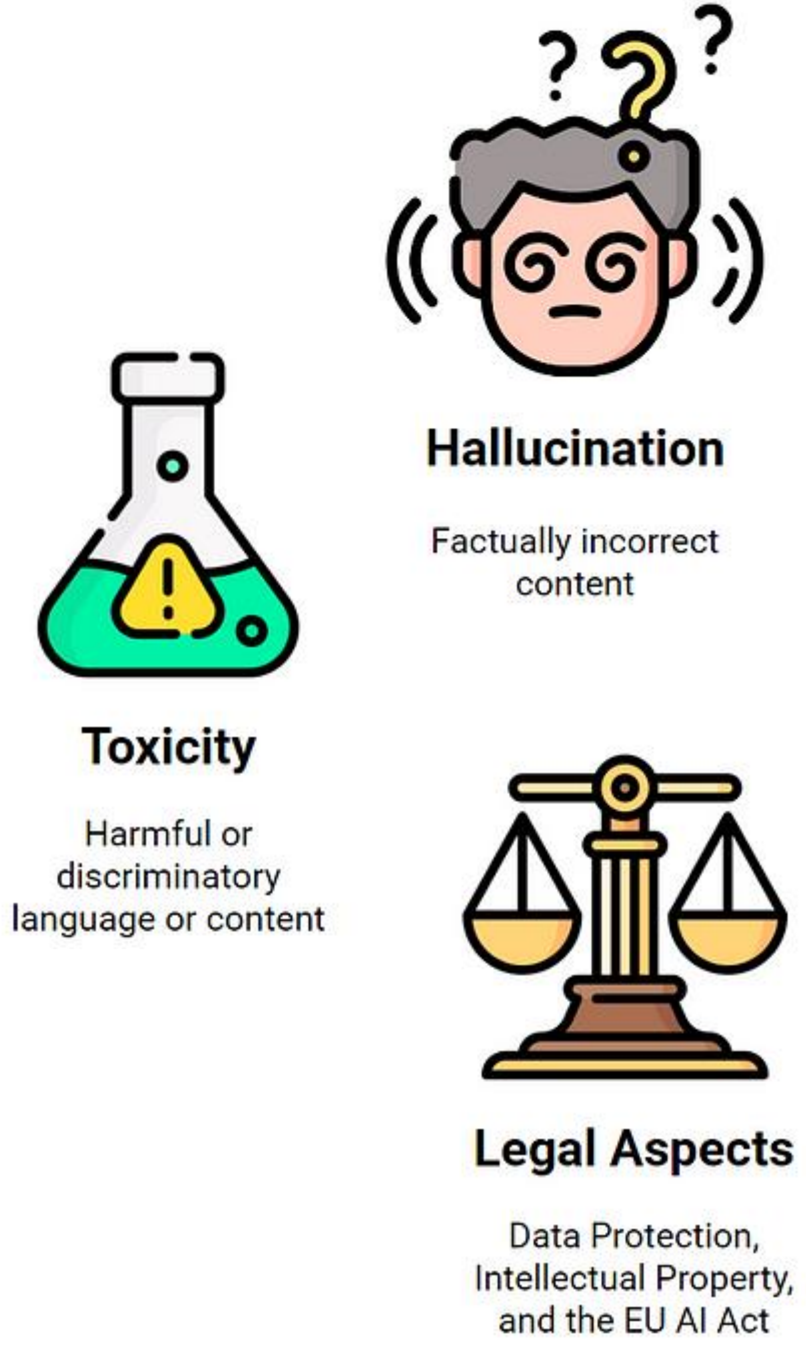
Discussion and Q&A

- **Critical Analysis of M-LLMs**
  - Looking Ahead: The Future of M-LLMs
    - **Advancements in Algorithms:** Innovations that may overcome current limitations.
    - **Increased Efficiency:** Research focused on reducing computational demands and simplifying architectures.
    - **Expansion of Use Cases:** Exploration into less traditional fields and novel applications.
    - **Improving Generalizability:** Techniques that might boost the model's ability to generalize better across diverse scenarios.



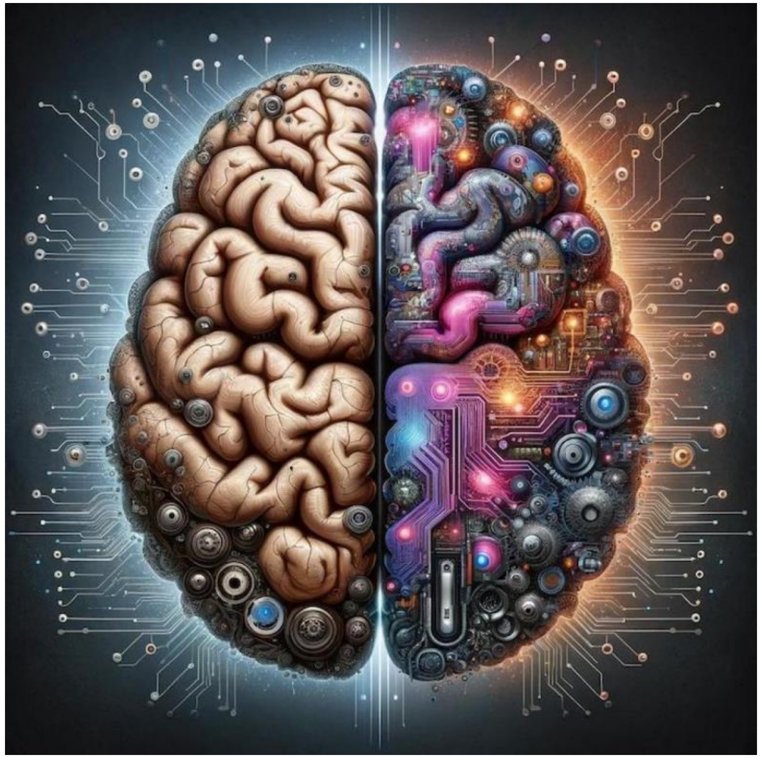
Discussion and Q&A

- **Critical Analysis of M-LLMs**
  - Ethical Considerations in Using M-LLMs
    - **Bias and Fairness:** Risks of inheriting or amplifying biases present in training data.
    - **Privacy Concerns:** Challenges relating to gathering and handling multimodal data responsibly.
    - **Transparency and Explainability:** Difficulty in understanding decision-making processes of complex models.
    - **Accountability:** Ensuring responsible use and preventing misuse of technology in sensitive applications.



Discussion and Q&A

- Emerging Research and Future Prospects in Multimodal Language Models
  - **Unexplored Modalities:**
    - **Exploration into Additional Senses:** Research aimed at integrating less common modalities like smell or tactile sensations into M-LLMs.
    - **Multisensory Integration:** Challenges and opportunities in creating true multisensory human-computer interaction models.
  - **Advanced Fusion Techniques:**
    - **Dynamic Adaptive Fusion:** Developing methodologies that adaptively select fusion techniques based on the task and data characteristics.
    - **Context-Aware Fusion:** Models that dynamically adjust their processing based on contextual understanding of the environment or situation.



Discussion and Q&A

- Emerging Research and Future Prospects in Multimodal Language Models
  - **Robustness and Generality:**
    - **Cross-Domain Functionality:** Enhancing the ability of M-LLMs to function effectively across varying domains without extensive retraining.
    - **Anti-Fragile Systems:** Systems that not only resist but also grow from noise, errors, and attacks.

Discussion and Q&A

- Emerging Research and Future Prospects in Multimodal Language Models
  - **Ethical AI and Bias Mitigation:**
    - **Debiasing Techniques:** Innovative approaches to automatically detect and mitigate bias in training data and model outputs.
    - **Transparent AI:** Efforts towards making multimodal models more explainable and understandable to users.
  - **Efficiency and Green AI:**
    - **Model Compression:** Strategies for reducing the footprint of M-LLMs to make them more energy-efficient and viable for deployment on edge devices.
    - **Energy-Efficient Training Techniques:** Research focused on developing training protocols that consume less energy.