based on the highest TF-IDF similarities, differing from typical negative sampling strategies:

$$\mathcal{L}_{\text{ITM-hn}} = -\sum_{(x,y)\in\text{Pos}} \log p(\text{aligned}|x,y) - \sum_{(x',y')\in\text{Hard Neg}} \log p(\text{unaligned}|x',y') \tag{9.9}$$

Including hard negatives, identified by high TF-IDF similarity scores, makes learning more challenging and effective, as the model must discern between closely related but unaligned pairs.

9.3.3.3 Masked Language Modeling

Masked language modeling (MLM) is a prevalent objective in pre-training frameworks, where researchers typically mask and fill input words randomly using special tokens. This method leverages the context from surrounding words and associated image regions to predict the masked words. In SIMVLM, as developed by Wang et al. (2021), this approach is combined with prefix language modeling (PrefixLM). PrefixLM applies bidirectional attention to a prefix sequence and autoregressive factorization for the subsequent tokens. In this context, words are denoted as $w = \{x_1, \ldots, x_K\}$ and image regions as $v = \{v_1, \ldots, v_T\}$. For MLM, a certain percentage $p\%$ of input words, represented as x_m, are masked at randomly generated indices m. The objective is to predict these masked words using the unmasked words $x_{\neg m}$ and all image regions v, by minimizing the negative log-likelihood:

$$\mathcal{L}_{MLM}(\theta) = -\mathbb{E}_{(x,v)} \log P_\theta(x_m|x_{\neg m}, v), \tag{9.10}$$

where θ are the trainable parameters.

In addition to MLM, PrefixLM in SIMVLM is another strategy for pre-training vision-language representation. This technique focuses on predicting the continuation of a text sequence given a prefix, formalized as:

$$\mathcal{L}_{PrefixLM}(\theta) = -\mathbb{E}_{x\sim D} \log P_\theta(\mathbf{x}_{\geq T_p}|\mathbf{x}_{<T_p}), \tag{9.11}$$

where $\mathbf{x}$ is the text sequence, D represents the pre-training data, and T_p is the length of the prefix sequence of tokens.

9.3.3.4 Masked Object Classification

This technique involves selectively masking portions of visual images, typically by setting their values to zero and then utilizing the labels predicted by an object detector as ground truth for these masked regions.

The methodology behind MOC is somewhat analogous to the masked language modeling (MLM) approach in NLP. In MOC, specific image regions are masked by altering their visual features with a certain probability $p\%$. The primary objective is to predict the object category for these masked image regions accurately, denoted as v_i^m. This process entails passing the encoder output of the masked image regions v_i^m through a fully connected (FC) layer, which computes the scores for T object classes (Li et al., 2020a). These scores are then transformed into a normalized distribution $g_\theta(v_i^m)$ via a softmax function. The MOC objective is formally expressed as:

$$\mathcal{L}_{\text{MOC}}(\theta) = -\mathbb{E}_{(w,v)}\left[\sum_{i=1}^{M} \text{CE}(c(v_m^i), g_\theta(v_m^i))\right] \tag{9.12}$$

where $c(v_m^i)$ represents the ground-truth label for the masked image region, and CE denotes the cross-entropy loss function. Here, θ signifies the parameters of the model, and the expectation $\mathbb{E}$ is over the distribution of words w and visual features v. The MOC objective, therefore, focuses on enhancing the model's ability to infer and classify objects in partially observed or occluded visual contexts, reinforcing its understanding of visual information.

9.3.3.5 Image-Text Matching (ITM)

The ITM process is integral in developing models that can understand and relate visual content to corresponding textual descriptions. A crucial aspect of ITM involves generating negative training data, typically associating negative sentences with each image and vice versa. The objective is to enhance the model's discriminative capability in distinguishing between correctly matched image-text and mismatched pairs.

In the context of ITM, each image-text pair (v, t) is associated with a ground truth label y, indicating whether the pair is correctly matched (positive) or not (negative). The optimization of ITM is conducted using a binary classification loss function, which assesses the model's ability to predict these alignments accurately. The loss function for ITM, denoted as $L_{\text{ITM}}(\theta)$, is mathematically formulated as:

$$\mathcal{L}_{\text{ITM}}(\theta) = -\mathbb{E}_{(v,t)}\left[y \log s_\theta(v, t) + (1 - y) \log(1 - s_\theta(v, t))\right] \tag{9.13}$$

where $s_\theta(v, t)$ represents the image-text similarity score computed by the model with parameters θ. The expectation $\mathbb{E}_{(v,t)}$ is taken over the distribution of image-text pairs. This loss function effectively measures the model's proficiency in identifying correct and incorrect alignments, thus refining its understanding of the complex relationships between visual and textual modalities.

9.3.3.6 Image-Text Generation

Image-text Generation (ITG) is an essential component of vision-language-related pre-training tasks. It focuses on training a model to generate text based on a given image, leveraging aligned image-text pairs. For instance, Xu et al. (2021) trained the E2E-VLP model using the ITG objective. The ITG objective is formulated as follows:

$$\mathcal{L}_{ITG} = -\sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log \prod_{t=1}^{n} P(y_t|y_{<t}, x) \tag{9.14}$$

Here, $\mathcal{X}$ represents the visual sequence with context, and $\mathcal{Y}$ is the set of generated text. The variable n indicates the length of tokens in the text y. This objective aims to maximize the probability of correctly generating the sequence of text tokens y_t based on the preceding tokens $y_{<t}$ and the visual input x.

9.3.3.7 Video-Subtitle Matching (VSM)

Video-subtitle matching (VSM) in video-text pre-training, as exemplified in HERO, focuses on two key alignment targets: local and global alignment (Li et al., 2020b). Score functions quantify the alignment between video and subtitle content, with separate scores for local and global alignment. The loss functions, however, are designed to optimize the model by minimizing the difference between these alignment scores for correctly matched video-subtitle pairs (positive pairs) and maximizing it for incorrectly matched pairs (negative pairs).

In HERO's VSM implementation, two alignment targets are considered: local and global.

Score Functions

- Local Alignment Score Function:

$$S_{\text{local}}(s_q, \mathbf{v}) = \mathbf{V}^{\text{temp}}\mathbf{q} \in \mathbb{R}^{N_v}$$

- Global Alignment Score Function:

$$S_{\text{global}}(s_q, \mathbf{v}) = \max\left(\frac{\mathbf{V}_{\text{temp}}}{||\mathbf{V}_{\text{temp}}||} \cdot \frac{\mathbf{q}}{||\mathbf{q}||}\right)$$

Loss Functions

- Hinge loss for positive and negative query-video pairs:

$$\mathcal{L}_h(S_{\text{pos}}, S_{\text{neg}}) = \max(0, \delta + S_{\text{pos}} - S_{\text{neg}})$$

- Local alignment loss:

$$\mathcal{L}_{\text{local}} = -\mathbb{E}_D\left[\log(\mathbf{p}_{\text{st}}[y_{\text{st}}] + \log(\mathbf{p}_{\text{ed}}[y_{\text{ed}}])\right]$$

- Global alignment loss:

$$\mathcal{L}_{\text{global}} = -\mathbb{E}_D\left[(\mathcal{L}_h(S_{\text{global}}(s_q, \mathbf{v}), S_{\text{global}}(\hat{s}_q, \mathbf{v})) + \mathcal{L}_h(S_{\text{global}}(s_q, \mathbf{v}), S_{\text{global}}(s_q, \hat{\mathbf{v}})))\right]$$

- Combined VSM loss:

$$\mathcal{L}_{\text{VSM}} = \lambda_1 \mathcal{L}_{\text{local}} + \lambda_2 \mathcal{L}_{\text{global}}$$

In this model, s_q represents the sampled query from all subtitle sentences, $\mathbf{v}$ is the entire video clip, and $\mathbf{V}_{\text{temp}} \in \mathbb{R}^{N_v \times d}$ is the final visual frame representation generated by a temporal Transformer. The query vector $\mathbf{q} \in \mathbb{R}^d$, start and end indices $y_{\text{st}}, y_{\text{ed}} \in \{1, \ldots, N_v\}$, and the probability vectors $\mathbf{p}_{\text{st}}, \mathbf{p}_{\text{ed}} \in \mathbb{R}^{N_v}$ are derived from the scores. The hinge loss function $\mathcal{L}_h$ is used for both positive and negative query-video pairs, where $(s_q, \mathbf{v})$ is a positive pair and $(s_q, \widehat{\mathbf{v}}), (\widehat{s}_q, \mathbf{v})$ are negative pairs. The margin hyper-parameter δ and balancing factors λ_1, λ_2 are key components of this framework.

9.3.3.8 Frame Order Modeling (FOM)

Frame order modeling (FOM) is conceptualized as a classification challenge within the HERO model's context, focusing on accurately predicting the chronological order of a given set of video frames (Li et al., 2020b). The primary goal of FOM is to determine the original sequence of timestamps for a subset of frames extracted from a video, thereby testing the model's understanding of temporal dynamics and narrative flow in video content.

The FOM objective is formulated as a loss function, mathematically expressed as:

$$\mathcal{L}_{\text{FOM}} = -\mathbb{E}\left[\sum_{i=1}^{R} \log \mathbf{P}[r_i, t_i]\right] \tag{9.15}$$

where:

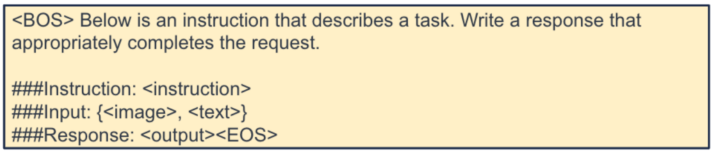
- R denotes the total number of frames that have been reordered and is subject to classification.
- i represents the index within the reordered set, ranging from 1 to R.
- t_i symbolizes the true timestamp position of the i^{th} frame within the video, which spans from 1 to N_v, where N_v is the total number of frames in the video.
- r_i is the index corresponding to the reordered position of the i^{th} frame.
- $\mathbf{P}$ is a probability matrix of dimensions $N_v \times N_v$, where each element $P[r_i, t_i]$ indicates the model's predicted probability that the frame at reordered position r_i corresponds to timestamp t_i.

9.3.4 MMLLM Tuning and Enhancements

Following the pre-training phase, MMLLMs can be further enhanced to improve their adaptability, reasoning, and task generalization capabilities. This enhancement is achieved through various methodologies, three of which are presented here: *multimodal instruction tuning* (MM-IT), which refines models to follow instructions for a broad spectrum of tasks; *multimodal in-context learning* (MM-ICL), which enables models to apply preexisting knowledge to new tasks presented within input prompts; and the *multimodal chain-of-thoughts* (MM-COT) approach, which enables more transparent and logical reasoning by the model in solving complex problems.

9.3.4.1 Multimodal Instruction Tuning

> Instruction tuning (IT) diverges from the data-heavy demands of traditional supervised fine-tuning and the limited improvements of prompting methods in few-shot scenarios by aiming to generalize task performance beyond initial training data (Sect. 4.2). Building on this, *multimodal instruction tuning* (MM-IT) adapts IT principles to enhance LLMs through fine-tuning multimodal datasets structured around instructional tasks (Liu et al., 2024; Zhao et al., 2023; Zhu et al., 2023). This approach empowers LLMs to handle new tasks by interpreting instructions efficiently, markedly boosting zero-shot learning abilities across various modalities.



```
<BOS> Below is an instruction that describes a task. Write a response that
appropriately completes the request.

###Instruction: <instruction>
###Input: {<image>, <text>}
###Response: <output><EOS>
```

Fig. 9.4: Multimodal instruction tuning template for visual question answering task.

A multimodal instruction sample is represented as a triplet, $(\mathcal{I}, \mathcal{M}, \mathcal{R})$, encapsulating the instruction, multimodal input, and the ground truth response, respectively. The model's task, governed by parameters θ, is to predict the answer based on both the instruction and the multimodal input:

$$\mathcal{A} = f(\mathcal{I}, \mathcal{M}; \theta) \tag{9.16}$$

Here, $\mathcal{A}$ signifies the predicted answer. The training objective often adheres to the original auto-regressive objective, compelling the MMLLM to predict the subsequent response token. This objective is mathematically expressed as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(\mathcal{R}_i | \mathcal{I}, \mathcal{R}_{<i}; \theta) \tag{9.17}$$

where N denotes the length of the ground-truth response, highlighting the model's aim to accurately generate the next token in the response sequence based on the preceding context and instruction. Fig. 9.4 presents a sample template for a visual question answering task, and Table 9.3 presents a selection of the most commonly used datasets for multimodal instruction tuning

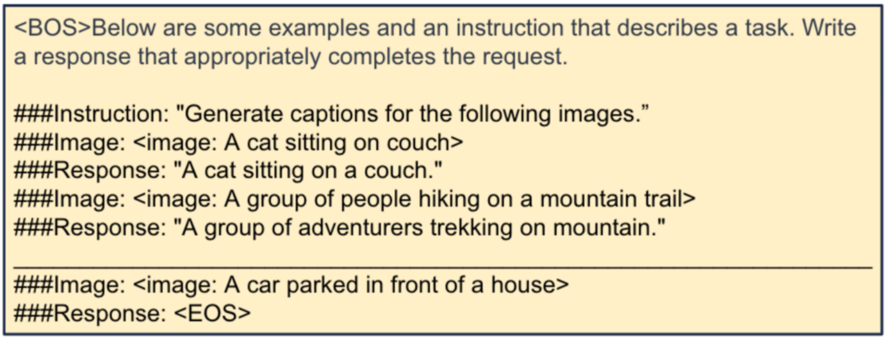
Table 9.3: Multimodal Instruction Tuning Datasets. In the table, the symbols represent the transition from input to output modalities, where I->O denotes Input to Output, T for Text, I for Image, V for Video, A for Audio, B for Bounding box, and 3D for Point Cloud.

Dataset Name	**I->O**	**Size (#Instances)**
MiniGPT-4'sIT	I+T->T	5K
StableLLaVA	I+T->T	126K
LLaVA'sIT	I+T->T	150K
SVIT	I+T->T	3.2M
LLaVAR	I+T->T	174K
ShareGPT4V	I+T->T	-
DRESS'sIT	I+T->T	-
VideoChat'sIT	V+T->T	11K
Video-ChatGPT'sIT	V+T->T	100K
Video-LLaMA'sIT	I/V+T->T	171K
InstructBLIP'sIT	I/V+T->T	~1.6M
X-InstructBLIP'sIT	I/V/A/3D+T->T	~1.8M
MIMIC-IT	I/V+T->T	2.8M
PandaGPT'sIT	I+T->T	160K
MGVLID	I+B+T->T	-
M3IT	I/V/B+T->T	2.4M
LAMM	I+3D+T->T	196K
BuboGPT'sIT	(I+A)/A+T->T	9K
T2M	T->I/V/A+T	14.7K
MosIT	I+V+A+T->I+V+A+T	5K

9.3.4.2 Multimodal In-context Learning

In-context learning (ICL) equips LLMs to understand and perform tasks by learning from a few examples, often with instructions. This method is distinct from traditional supervised learning which requires extensive data. This method enables LLMs to handle novel tasks without additional training. Unlike instruction tuning, which fine-tunes models on instructional datasets, ICL leverages the model's pre-trained capabilities to adapt to new tasks during inference, bypassing the need for further model updates.

As the concept of ICL extends into the multimodal domain, it evolves into *multimodal in-context learning* (MM-ICL), enriching the learning process with diverse modalities (Gupta and Kembhavi, 2023). MM-ICL incorporates a demonstration set alongside the original sample at the inference stage, enhancing the learning context with multiple in-context examples.



<BOS>Below are some examples and an instruction that describes a task. Write a response that appropriately completes the request.

###Instruction: "Generate captions for the following images."
###Image: <image: A cat sitting on couch>
###Response: "A cat sitting on a couch."
###Image: <image: A group of people hiking on a mountain trail>
###Response: "A group of adventurers trekking on mountain."

###Image: <image: A car parked in front of a house>
###Response: <EOS>

Fig. 9.5: Multimodal In-context Learning for Caption Generation task

Fig. 9.5 depicts an example of MM-ICL for caption generation with two examples. The structure of these examples, including their quantity, can be adjusted flexibly, acknowledging that model performance often hinges on the sequence of presented examples. We also list in Table 9.4 a few critical datasets for MM-ICL.

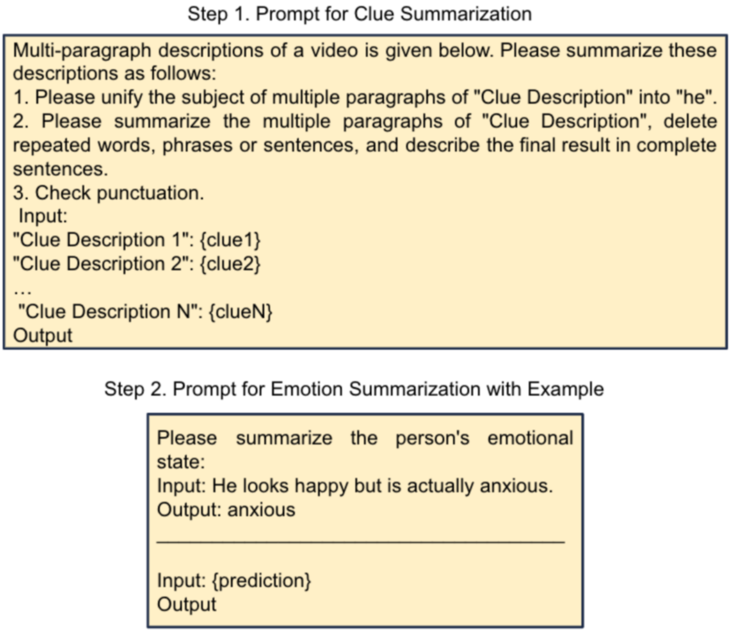
Table 9.4: Multimodal In-context Learning Dataset

Dataset	**Modality**	**Size**	**Notes**
MM-ICL	Image-Text	5.8M	Includes interleaved text-image inputs and multimodal in-context learning inputs constructed manually.
MIMIC-IT	Image-Text	2.8M	Provides multimodal instruction-response pairs to improve VLMs in perception, reasoning, and planning across multiple languages.

9.3.4.3 Multimodal Chain-of-Thought

> *Multimodal chain-of-thought* (MM-CoT) is an extension of the chain-of-thought concept in LLMs, which is recognized for its effectiveness in complex reasoning tasks. CoT involves LLMs generating the final answer and the intermediate reasoning steps, akin to human cognitive processes. MM-CoT adapts this unimodal CoT to a multimodal context, requiring initial modality bridging. This bridging can be achieved by fusing features or translating visual inputs into textual descriptions. Regarding learning paradigms, MM-CoT can be developed through fine-tuning or through training-free few/zero-shot learning, each with varying sample size requirements.

In their research, Lian et al. (2023) use ChatGPT to synthesize clues from multiple descriptions provided by human annotators into a cohesive summary, focusing on key behaviors and expressions, and then use this consolidated insight to deduce the subject's underlying emotional state accurately, as shown in Fig. 9.6.



Step 1. Prompt for Clue Summarization

Multi-paragraph descriptions of a video is given below. Please summarize these descriptions as follows:
1. Please unify the subject of multiple paragraphs of "Clue Description" into "he".
2. Please summarize the multiple paragraphs of "Clue Description", delete repeated words, phrases or sentences, and describe the final result in complete sentences.
3. Check punctuation.
Input:
"Clue Description 1": {clue1}
"Clue Description 2": {clue2}
…
"Clue Description N": {clueN}
Output

Step 2. Prompt for Emotion Summarization with Example

Please summarize the person's emotional state:
Input: He looks happy but is actually anxious.
Output: anxious

Input: {prediction}
Output

Fig. 9.6: Multimodal chain-of-thought for emotion detection through video clip annotations as clues from human

The configuration and pattern of reasoning chains in MM-CoT can be broadly classified into two types:

1. **Adaptive Configuration:** In this approach, LLMs autonomously determine the length of the reasoning chain (Wu et al., 2023a). This flexibility allows the model to adapt the reasoning process to the complexity of the task, ensuring a more tailored and potentially more accurate response. It is particularly beneficial in scenarios where the depth of reasoning required can vary significantly from one task to another.
2. **Predefined Configuration:** Contrary to the adaptive approach, the length of the reasoning chain is predetermined here (Himakunthala et al., 2023). This setup provides a consistent and uniform structure for reasoning across different tasks. While this approach might simplify the model's operation, it may limit the depth of reasoning in more complex scenarios.

Beyond the configuration, the generation pattern of the reasoning chain itself is another area in MM-CoT and provides the following choices:

1. **Infilling-Based Pattern:** This pattern involves deducing intermediate steps to logically connect the surrounding context, effectively filling the gaps in the reasoning process (Himakunthala et al., 2023). It requires the model to identify and bridge missing links in a sequence of thoughts, ensuring a coherent and logical flow of ideas. Consider a task where the model is given a sequence of images depicting a story and is asked to narrate the events. The infilling-based pattern would require the LLM to fill in the narrative gaps between the images, ensuring a coherent storyline.
2. **Predicting-Based Pattern:** In contrast, the prediction-based pattern extends the reasoning chain forward based on given conditions such as instructions or the history of previous reasoning steps (Wu et al., 2023a). This approach requires the model to understand the current context and anticipate logical continuations, synthesizing new steps in the reasoning chain. When an LLM is asked to predict the next scene in a visual story, the prediction-based pattern involves extending the narrative based on the given images and textual descriptions. This requires the model to anticipate future events or actions, building upon the existing context.

Some well-known datasets for MM-CoT reasoning are described in Table 9.5.

9.3.5 Multimodal RLHF

MMLLMs face more challenges than do LLMs trained on a single modality due to the complexity of integrating and interpreting information across diverse data types. Similar to its application in unimodal LLMs, RLHF can address numerous issues in multimodal LLMs, including incorporating human preferences and choices, integrating human feedback into descriptions, and generating responses that adhere

Table 9.5: Multimodal Chain-of-Thought Dataset

Dataset	Modality	Size	Notes
EMER	Video-Text	100	Focuses on explainable emotion-based reasoning, offering clues and summarization for reasoning tasks.
EgoCOT	Video-Text	3,670 hours	Embodied planning dataset on a large scale for embodied scenario planning.
VIP	Video-Text	3.6M, 1.5K test	Designed for Video Chain-of-Thought evaluation, featuring inference-time challenges with extensive caption data.
ScienceQA	Image-Text	21K Q-A	Multimodal, multi-choice question dataset across science and diverse domains for in-depth analysis.

to safety and ethical standards. We will highlight some of the research in the field that addresses trustworthiness and methods to incorporate human preferences and alignment.

! Practical Tips

Li et al. (2023) focused on using preference distillation to produce helpful and anchored responses in the visual context. The research introduced the VLFeedback dataset, which contains 80,000 multimodal instructions, with responses from 12 LVLMs and preference annotations from GPT-4V. The findings demonstrate that the Silkie model, refined with this dataset, significantly outperforms the base model on various benchmarks. Compared with human-annotated datasets, the dataset effectively boosts the perception and cognitive abilities of LVLMs and shows advantages in terms of scalability and broader performance improvements.

RLHF-V is an RLHF-based approach aimed at improving the trustworthiness of MMLLMs by aligning their behavior with fine-grained human feedback (Yu et al., 2023). It addresses a critical issue existing MMLLMs face: the tendency to produce hallucinated text not factually grounded in the associated images, which compromises their reliability for real-world applications, especially those with high stakes. The RLHF-V framework collects human preferences through segment-level corrections for hallucinations and applies dense, direct preference optimization based on this feedback. Through extensive experiments across five benchmarks involving both automatic and human evaluations, RLHF-V is shown to significantly enhance the trustworthiness of MMLLM behaviors while demonstrating promising data and computational efficiency.

! Practical Tips

In their study, Sun et al. (2023) presented a new alignment algorithm, "Factually Augmented RLHF", which enhances the existing reward model by integrating factual content, including image captions and accurate multichoice answers. This strategy aims to address and reduce the occurrence of reward hacking in RLHF, leading to notable improvements in model effectiveness. Additionally, this study enriches the training dataset for vision instruction tuning, which was originally generated by GPT-4, with pre-existing human-authored image-text pairs to bolster the model's general performance. By applying RLHF to a language multimodal model (LMM) for the first time, the method showed a marked improvement in performance on the LLaVA-Bench dataset, aligning closely with the results of the text-only GPT-4.

9.3.6 Output Projector

The Output Projector, denoted as $\text{OUT\_ALIGN}_{T \to X}$, transforms the signal token representations $\mathbf{S}_X$, derived from the LLM, into features $\mathbf{H}_X$ that are interpretable by the subsequent Modality Generator MG_X.

Specifically, for a given modality-text dataset $\{(\mathbf{I}_X, t)\}$, the process starts with input t being processed by the LLM to yield $\mathbf{S}_X$, which is subsequently converted into $\mathbf{H}_X$.

The primary objective is to ensure that $\mathbf{H}_X$ aligns closely with the modality generator's understanding, as defined by:

$$\underset{\text{OUT\_ALIGN}_{T \to X}}{\arg\min} \quad \mathcal{L}_{\text{mse}}(\mathbf{H}_X, \tau_X(t)), \tag{9.18}$$

where

$$\mathbf{H}_X = \text{OUT\_ALIGN}_{T \to X}(\mathbf{S}_X). \tag{9.19}$$

$\mathcal{L}_{\text{mse}}$ represents the mean squared error loss, aiming to minimize the discrepancy between the projected features $\mathbf{H}_X$, and τ_X is the textual condition encoder in MG_X. This optimization process primarily utilizes processing texts without requiring direct multimodal inputs such as audio or visual inputs X.

Practical Tips

The Output Projector is usually implemented using a Tiny Transformer or an MLLP, focusing on efficiency and adaptability.

9.3.7 Modality Generator

The Modality Generator MG_X is engineered to generate outputs across various modalities, effectively translating encoded features into multimodal content.

! Practical Tips

This component often employs SOTA latent diffusion models (LDMs) for synthesizing outputs specific to each modality, such as images, videos, and audio (Zhao et al., 2022). Commonly used implementations include Stable Diffusion for image synthesis, Zeroscope for video synthesis, and AudioLDM-2 for audio output generation (Cerspense, 2023; Liu et al., 2023; Rombach et al., 2022).

The process leverages $\mathbf{H}_X$ from the output projector as conditional inputs to guide the denoising step, which is essential for generating high-quality multimodal content.

During the training phase, the original content is first encoded into latent features z_0 using a pre-trained variational autoencoder (VAE) (Kingma and Welling, 2013). This latent representation is then perturbed with noise ϵ to produce a noisy latent feature z_t.

A pre-trained Unet (ϵ_X)is normally used for computing the conditional LDM loss $\mathcal{L}_{X-gen}$ (Ronneberger et al., 2015). Given as:

$$\mathcal{L}_{X-gen} := \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1),t} \|\epsilon - \epsilon_X(z_t, t, \mathbf{H}_X)\|_2^2 \tag{9.20}$$

$IN\_ALIGN_{X \to T}$ and $OUT\_ALIGN_{T \to X}$ are optimized by minimizing $\mathcal{L}_{X-gen}$.

9.4 Benchmarks

This section overviews selected benchmark datasets for evaluating multimodal LLMs across various modalities and tasks. Although not exhaustive, this compilation emphasizes benchmarks notable for their task diversity, modality range, and widespread application in the field. For a more detailed or comprehensive list of benchmark datasets, readers are encouraged to refer to the work of Yin et al. (2023).

1. **CMMU** is a comprehensive collection of 12, 000 multimodal questions, manually curated from college exams, quizzes, and textbooks across six fundamental disciplines: Art and Design, Business, Science, Health and Medicine, Humanities and Social Science, and Tech and Engineering (Zhang et al., 2024b). This diversity mirrors that of its counterpart, MMMU, which extends across 30 distinct subjects. The dataset is characterized by its variety, featuring 39 different

types of images—including charts, diagrams, maps, tables, music sheets, and chemical structures—to test a wide range of multimodal understanding capabilities.

2. **MMCBench** presents a detailed framework for assessing LMMs, emphasizing their resilience and self-consistency when faced with typical corruption challenges (Zhang et al., 2024c). It focuses on the interplay between text, image, and speech modalities, encompassing key generative tasks such as text-to-image, image-to-text, text-to-speech, and speech-to-text.
3. **MMVP** evaluates the visual capabilities of multimodal LLMs through VQA tasks (Tong et al., 2024). It includes a directory of 300 test images and a CSV file with questions and correct answers.
4. **TimeIT** addresses six timestamp-related video tasks and incorporates 12 datasets from various domains (Ren et al., 2023). It focuses on time-sensitive long video understanding tasks such as dense video captioning, video grounding, video summarization, video highlight detection, step localization, and transcribed speech generation, with a total training data size of 124,861 instances.
5. **ViP-Bench** is a benchmark designed to test multimodal models on visual reasoning capabilities through 303 image-question pairs derived from MM-Vet, MM-Bench, and Visual Genome (Cai et al., 2023). It aims to evaluate models on six key aspects of visual understanding at the region level: recognition, OCR, knowledge, math, object relationship reasoning, and language generation. The benchmark employs GPT-4 for grading multimodal model responses from 0 to 10, offering a quantitative comparison tool.
6. **M3DBench** compiles more than 320K pairs of 3D multimodal instruction-following data, including over 138K instructions with unique multimodal prompts (Cai et al., 2023). It utilizes existing datasets and instructions generated by LLMs for diverse 3D tasks. The dataset spans object detection to question answering, with instructions and responses tailored to each task. High data quality is ensured by filtering out irrelevant responses through pattern matching, making M3DBench a robust dataset for 3D instruction-following evaluations.
7. **Video-Bench** introduces a comprehensive benchmark for assessing Video LLMs (Ning et al., 2023). This benchmark encompasses ten carefully designed tasks that gauge Video-LLMs' proficiency in video-specific understanding, leveraging prior knowledge for question-answering, and skills in comprehension and decision-making and has a size of approximate 15,033.
8. **Bingo** classifies instances of model failures and successes in multimodal understanding, comprising 190 instances of failures contrasted with 131 instances of success (Cui et al., 2023). Each instance, paired with one or two questions, falls into the categories of "Interference" (image-to-image and text-to-image) and "Bias" (region, OCR, and factual). The benchmark aims to dissect the nuanced reasons behind hallucinations in responses, offering a detailed exploration of bias within GPT-4V(ision) across diverse images reflecting cultural, linguistic, and factual diversities.
9. **MMHAL-BENCH** is designed to evaluate hallucinations in multimodal models, focusing on hallucination detection with tailored evaluation metrics (Sun

et al., 2023). It features 96 image-question pairs across eight question categories and twelve object topics and was specifically constructed to test LMMs against false claims about image contents. The benchmark leverages images from the Open Images validation and test sets to avoid data leakage. It includes comprehensive object meta-categories such as "accessory," "animal," and "vehicle." Responses are evaluated using GPT-4, which assesses the presence of hallucinations by comparing LMM outputs with human-generated answers and the image content.

10. **Sparkles** leverages GPT-4 to construct a multimodal dialog dataset that simulates realistic conversations around images and text, aiming for dialogs that span a variety of real-life scenarios (Huang et al., 2023). This process uses a two-turn dialog pattern, starting with a user query about images, followed by an assistant's detailed response, and then introducing a new image for further discussion. The dataset generation emphasizes dialog demonstrations for in-context learning and candidate image descriptions for selecting relevant images, employing detailed textual descriptions to represent images due to GPT-4's text-only input capability.
11. **SciGraphQA** introduces a large-scale, synthetic, multiturn question-answer dataset for academic graphs (Li and Tajbakhsh, 2023). The dataset encompasses 295,000 samples derived from 290,000 Computer Science or Machine Learning papers from ArXiv (2010-2020). Utilizing Palm-2 generates dialogs based on graphs within these papers, incorporating titles, abstracts, relevant paragraphs, and contextual data. Each dialog averages 2.23 question-answer turns. GPT-4's evaluation of the dataset's question-answer match quality averages at 8.7/10 across a 3,000-sample test set.
12. **LAMM** introduces a comprehensive multimodal instruction tuning dataset comprising 186,000 language-image and 10,000 language-3D instruction-response pairs, utilizing images and point clouds from diverse vision tasks (Yin et al., 2024). This dataset, constructed through GPT-API and self-instruction methods, includes four types of multimodal instruction-response pairs: daily dialogs, factual knowledge dialogs, detailed descriptions of images and 3D scenes, and visual task dialogs to enhance visual task generalization. It incorporates a variety of 2D and 3D vision tasks, such as captioning, scene graph recognition, VQA, classification, detection, counting, and OCR.

9.5 State-of-the-Art MMLLMs

This section provides an overview of several SOTA MMLLMs, showcasing models that integrate various modalities into their framework. A detailed Table 9.6 is presented, which encapsulates a wide range of well-known multimodal LLMs, each mapping distinct components to the generic framework outlined earlier. Next, we delve into the specifics of three multimodal LLMs, each representing a significant leap in the complexity and capability of handling multimodal data. Starting with

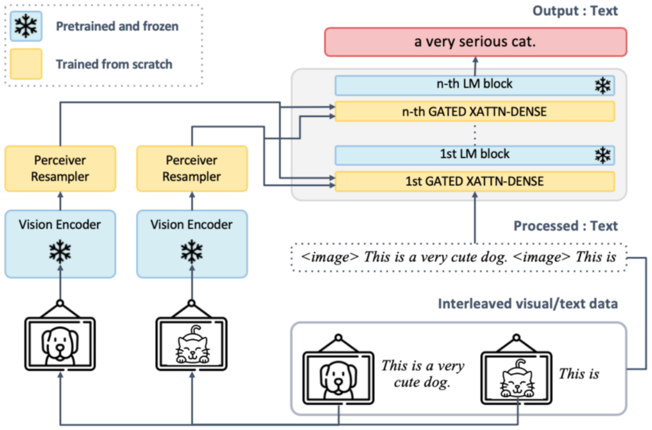


Fig. 9.7: The architecture of the Flamingo model.

Flamingo, which combines vision and language, we then discuss Video-LLaMA, which adds video and audio to text. Finally, we discuss NExT-GPT, which expands to support six different modalities, allowing any conversion between them.

9.5.1 Flamingo (Image-Video-Text)

Flamingo is a pioneering visual language model in MMLLMs, heralding advancements in few-shot learning for a broad spectrum of vision and language tasks (Alayrac et al., 2022). It distinguished itself by surpassing the fine-tuned state-of-the-art models in six of sixteen tasks, utilizing considerably less task-specific training data. The subsequent discussion delves into the architectural decisions aligned with components outlined in the preceding framework, highlighting where certain elements, such as multimodal RLHF, the output projector, and the modality generator, are absent. Notably, RLHF techniques were not implemented, and given the textual nature of the outputs, there was no necessity for output projection and generation processes.

9.5.1.1 Modality Encoder

Central to Flamingo is the integration of the Normalizer-Free ResNet (NFNet) F6 as the vision encoder, which employs contrastive learning for vision-text modalities to encode visual inputs efficiently. Flamingo adopts BERT for text encoding, diverging from the conventional use of GPT-2. The model processes embeddings from both vision and text modalities through mean pooling, subsequently projecting them into a joint embedding space to facilitate seamless modality integration.

Table 9.6: This table provides a detailed overview of various multimodal LLMs, highlighting their choices of base models, input-output modalities, modality encoders, input projectors, core LLMs, and modality generators.

Model	**I→O**	**Modality Encoder**	**Input Projector**	**LLM**	**Output Projector**	**Modality Generator**
Flamingo	I+V+T→T	I/V: NFNet-F6	Cross-attention	Chinchilla-1.4B/7B/70B	-	-
BLIP-2	I+T→T	I: CLIP/Eva-CLIP ViT@224	Q-Former w/ Linear Projector	Flan-T5/OPT	-	-
LLaVA	I+T→T	I: CLIP ViT-L/14	Linear Projector	Vicuna-7B/13B	-	-
IDEFICS	I+T→T	I: OpenCLIP ViT-H/14	Cross-attention	LLaMA-v1 7B/65B	-	-
MiniGPT-4	I+T→T	I: Eva-CLIP ViT-G/14	Q-Former w/ Linear Projector	Vicuna-13B	-	-
X-LLM	I+V+A+T→T	I/V: ViT-G; A: C-Former	Q-Former w/ Linear Projector	ChatGLM-6B	-	-
VideoChat	V+T→T	V: ViT-G	Q-Former w/ Linear Projector	Vicuna	-	-
InstructBLIP	I+V+T→T	I/V: ViT-G/14@224	Q-Former w/ Linear Projector	Flan-T5/Vicuna	-	-
Video-LLaMA	I+V+A+T→T	I/V: EVA-CLIP ViT-G/14; A: ImageBind	Q-Former w/ Linear Projector	Vicuna/LLaMA	-	-

Continued on next page

Table 9.6 – *Continued from previous page*

Model	**I→O**	**Modality Encoder**	**Input Projector**	**LLM**	**Output Projector**	**Modality Generator**
BuboGPT	I+A+T→T	I:CLIP/Eva-CLIPViT; A:ImageBind	Q-Former w/ Linear Projector	Vicuna (Frozen)	-	-
Qwen-VL-(Chat)	I+T→T	I:ViT@448 initialized from OpenClip's ViT-bigG	Cross-attention	Qwen-7B (PT: Frozen; IT: PEFT)	-	-
Palm-E	I+3D+T->T	I:ViT, 3D:OSRT	Affine Transformations	PaLM (PT: Frozen; co-training)	-	-
MACAW-LLM	I+V+A+T→T	I/V: CLIP; A:Whisper	Linear Projector	Llama-7B	-	-
NExT-GPT	I+V+A+T→I+V+A+T	I/V/A:ImageBind	Linear Projector	Vicuna-7B (PEFT)	Tiny Transformer	I:StableDiffusion; V:Zeroscope; A:AudioLDM
MiniGPT-5	I+T→I+T	I:Eva-CLIP ViT-G/14	Q-Former w/ Linear Projector	Vicuna-7B (PEFT)	Tiny Transformer w/ MLP	I:StableDiffusion-2
LLaVA-1.5	I+T→T	I:CLIP ViT-L@336	MLP	Vicuna-v1.5-7B/13B (PT: Frozen; IT: PEFT)	-	-
X-InstructBLIP	I+V+A+3D+T→T	I/V:Eva-CLIPViT-G/14; A:BEATs; 3D:ULIP-2	Q-Former w/ Linear Projector	Vicuna-v1.1-7B/13B (Frozen)	-	-
CoDi-2	I+V+A+T→I+V+A+T	I/V/A:ImageBind	MLP	Llama-2-Chat-7B (PT: Frozen; IT: PEFT)	MLP	I:StableDiffusion-2.1; V:Zeroscope-v2; A:AudioLDM-2

9.5.1.2 Input Projector

Flamingo's ability to handle visual inputs, including images and videos, necessitates addressing the variability in feature outputs. This is achieved through the perceiver resampler component, which standardizes outputs to a consistent 64 visual tokens, as shown in Fig. 9.7. The modality alignment between language and visual modalities is achieved by incorporating cross-attention (GATED XATTN-DENSE) layers among the preexisting frozen language model layers, enhancing the attention mechanism toward visual tokens during text token generation.

9.5.1.3 Pre-training: Core LLMs, Datasets and Task-Specific Objectives

The foundation of Flamingo is built upon the Chinchilla language model by freezing nine of the pre-trained Chinchilla LM layers. The training regimen spans four distinct datasets: M3W (Interleaved image-text), ALIGN (Image-text pairs), LTIP (Image-text pairs), and VTP (Video-text pairs). This approach enables Flamingo to predict subsequent text tokens y by considering both preceding text and visual tokens, quantified as:

$$p(y|x) = \prod_{\ell=1}^{L} p(y_\ell|y_{<\ell}, x_{\leq \ell}). \tag{9.21}$$

The training loss function is defined as a weighted sum of the expected negative log-likelihoods of the generated text across the datasets, where λ_m signifies the training weight for the m-th dataset:

$$\sum_{m=1}^{M} \lambda_m \mathbb{E}_{(x,y)\sim \mathcal{D}_m} \left[-\sum_{\ell=1}^{L} \log p(y_\ell|y_{<\ell}, x_{\leq \ell}) \right], \tag{9.22}$$

where $\mathcal{D}_m$ and λ_m represent the m-th dataset and its associated weighting, respectively.

9.5.1.4 MMLLM Tuning and Enhancements

The Flamingo models exhibit exceptional performance in in-context learning, outclassing state-of-the-art models fine-tuned for specific tasks despite relying on a singular set of model weights and a limited number of 32 task-specific examples – a thousand times fewer task-specific training examples than existing state-of-the-art approaches. The analysis presents support examples as pairs of images or videos (visual inputs) with corresponding text (expected responses or task-specific information, such as questions) to predict responses for new visual queries. The default prompts use are "Output: output" for tasks excluding question-answering, and

“Question: question Answer: answer” for question-answering or visual dialog tasks.

9.5.2 Video-LLaMA (Image-Video-Audio-Text)

Zhang et al. (2023) introduced *Video-LLaMA*, a multimodal framework designed to augment LLMs with the ability to comprehend visual and auditory elements in videos. Unlike prior initiatives that have enabled LLMs to process visual or audio signals, Video-LLaMA takes a comprehensive approach by incorporating cross-modal training leveraging frozen pre-trained visual and audio encoders alongside frozen LLMs. The framework is distinctive for its focus on video comprehension, addressing two key challenges: capturing the temporal dynamics within visual scenes and effectively merging audio-visual information. The experiments that were conducted reveal Video-LLaMA’s remarkable ability to facilitate audio and video-grounded dialogs, underscoring its viability as an advanced prototype for audio-visual AI assistants. The following section explores the architectural choices corresponding to the components presented in the prior framework, identifying the absence of specific elements, again including multimodal RLHF, the output projector, and the modality generator. It is important to note that RLHF methodologies were not applied, and the requirement for output projection and generation was not needed because the output was only the text.

9.5.2.1 Modality Encoder

For the encoding of visual inputs, the branch leverages a frozen visual encoder with a ViT G/14 model from EVA-CLIP and a BLIP-2 Q-former to process video frames, as shown in Fig. 9.8. Each frame is transformed into a set of image embedding vectors, resulting in a sequence of frame representations $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$, where $\mathbf{v}_i \in \mathbb{R}^{K_f \times d_f}$ denotes the d_f-dimensional image embeddings for the i-th frame.

The pre-trained Imagebind is used as the audio encoder to address the auditory component of videos (Girdhar et al., 2023). The videos are uniformly sampled as M segments of 2-second audio clips. Each of these clips is then transformed into spectrograms utilizing 128 Mel spectrogram bins, effectively capturing the audio’s spectral features. The audio encoder processes these spectrograms, converting each into a dense vector representation. As a result, the compiled audio representation for a given video is denoted as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M]$, where each $\mathbf{a}_i$ represents the encoded feature vector of the i-th audio segment.

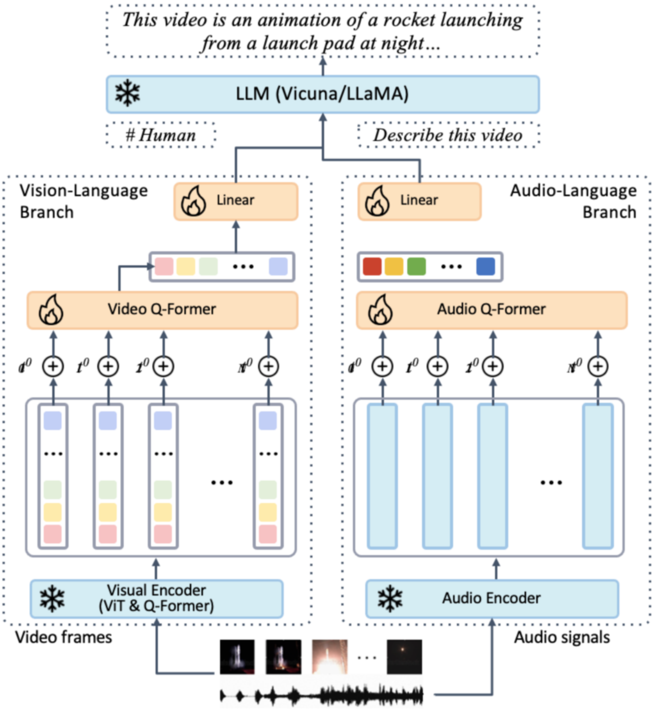


Fig. 9.8: The architecture of Video-LLaMA.

9.5.2.2 Input Projector

In both the video and audio branches, the Q-Former combined with a linear projection is used to align the encoded modalities with textual data.

For the vision-language branch, position embeddings are incorporated to imbue these representations with temporal context. This is because the frame representations, $\mathbf{v}_i$, are derived from the frozen image encoder and thus lack inherent temporal information. Next, the position-encoded frame representations are introduced into the Video Q-former. The purpose is to fuse the frame-level representations into a consolidated video representation, achieving a set of k_V video embedding vectors, each of dimension d_v. Consequently, this yields a comprehensive video representation $\hat{\mathbf{v}} \in \mathbb{R}^{k_V \times d_v}$, effectively capturing both the visual and the temporal dynamics of the video content. A linear layer is introduced to transform the video embedding vectors into video query vectors to align the video representations with the input requirements of the LLMs. These vectors match the dimensionality of the LLM's text embeddings, ensuring video and textual data compatibility. During the forward pass, video query vectors are concatenated with text embeddings, serving as a video soft prompt. This concatenation effectively guides the frozen LLMs to generate text outputs conditioned on the video content, thereby integrating video information into the multimodal understanding process.

Similar to the vision-language branch, a position embedding layer is applied to incorporate temporal information into these audio segments in the audio-language branch. This addition ensures that temporal dynamics, which are critical for understanding the sequence and evolution of sounds within the video, are captured. Following this temporal encoding, the audio Q-former is used to fuse the features of different audio segments into a unified audio representation. Mirroring the vision-language branch, a linear layer is employed to map the comprehensive audio representation into the embedding space of the LLMs.

9.5.2.3 Pre-training: Core LLMs, Datasets and Task-Specific Objectives

Video-LLaMA leverages Vicuna-7B, as the core LLM for its multimodal understanding and generation capabilities.

Video-LLaMA's pre-training process utilizes the Webvid-2M dataset, a collection of short videos accompanied by textual descriptions from stock footage websites, to train its vision-language branch. This dataset and the CC595k image caption dataset derived from CC3M and refined by Liu et al. (2024) form the basis for a video-to-text generation task during pre-training. The audio-language branch in Video-LLaMA utilizes the ImageBind audio encoder, which is inherently aligned across multiple modalities hence no pre-training is required.

9.5.2.4 MMLLM Tuning and Enhancements

Following its pre-training phase for the Video-Language branch, Video-LLaMA demonstrated proficiency in generating content based on video information. However, its ability to adhere to specific instructions showed a need for enhancement, and instruction-based fine-tuning was performed. The datasets employed for this purpose included:

1. A collection of 150K image-based instructions from the LLaVA dataset.
2. A set of 3K image-based instructions sourced from MiniGPT-4.
3. An assembly of 11K video-based instructions from VideoChat.

For the audio tuning process in Video-LLaMA, the approach addresses the challenge posed by the scarcity of audio-text data by incorporating the vision-text datasets mentioned above into the training regimen. This strategy enables these components to learn the alignment between the common embedding space produced by the ImageBind encoder and the embedding space of the LLMs.

9.5.3 NExT-GPT (Any-to-Any)

NExT-GPT is a general-purpose, multimodal LLM that integrates a large language model with multimodal adaptors and diffusion decoders, allowing it to handle and generate text, images, videos, and audio content (Wu et al., 2023c). It is fine-tuned on a small number of parameters, making training cost-effective and expanding to new modalities straightforward. The system also features modality-switching instruction tuning and a high-quality dataset for improved cross-modal understanding and generation, demonstrating the feasibility of creating a unified AI agent for diverse modalities.

9.5.3.1 Modality Encoder

NExT-GPT employs ImageBind as a universal encoder across all modalities, diverging from the traditional approach of modality-specific encoders used in many previous studies. ImageBind demonstrated the capability to forge a joint embedding space encompassing multiple modalities, eliminating the need to train on data representing every possible modality combination and showing state-of-the-art results.

9.5.3.2 Input Projector

NExT-GPT utilizes a linear projection layer (4 million parameters) to transform the outputs through ImageBind into language-like representations, thus aligning all modalities in a format that the LLM can readily understand and process.

9.5.3.3 Pre-training: Core LLMs, Datasets and Task-Specific Objectives

NExT-GPT uses Vicuna2, an open-source text-based LLM widely adopted in existing multimodal LLMs, as its core LLM.

In stage 1 of pre-training NExT-GPT, everything in the pipeline – the encoders the process inputs, the decoders that process outputs, and the LLM – are kept frozen, and only the input alignment through the linear projection layer is adapted through backpropagation. This training strategy aims to align various modalities – images, audio, or videos – with their corresponding textual descriptions (captions) using specific datasets for each modality. The CC3M dataset, comprising 3 million image-caption pairs, is employed for image modality alignment. Video modality alignment utilizes the WebVid-10M dataset, which contains 10.7 million video-caption pairs from diverse content sourced from stock footage websites, totaling 52,000 hours of video. AudioCaps provides a foundation for the audio modality with around 46,000 pairs of audio clips and human-written text, curated through crowdsourcing on the AudioSet dataset. This method trains NExT-GPT to generate captions that match the input modality against a benchmark "gold" caption.

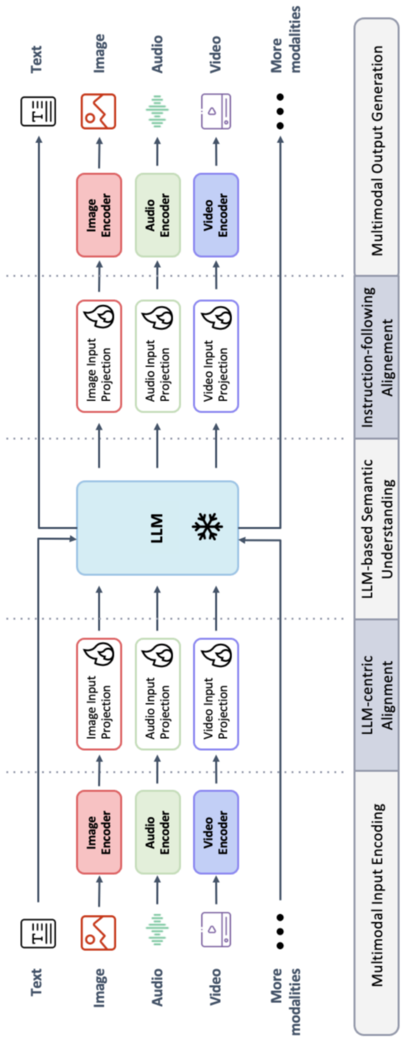


Fig. 9.9: The architecture of NExT-GPT.

During stage 2 of training, the focus is on the output projection layers. The ImageBind, LLM, and input projection layers are kept frozen, and the training employs the same datasets used in the initial stage: the CC3M dataset for images, the WebVid-10M dataset for videos, and the AudioCaps dataset for audio.

9.5.3.4 MMLLM Tuning and Enhancements

Stage 3 of NExT-GPT's training uses multimodal instruction tuning, a process designed to refine the model's ability to understand and execute complex instructions across different modalities. In this phase, the core LLM (Vicuna2) is fine-tuned using the LoRA technique, and both the input and output projection layers are fine-tuned during this stage, but without altering the encoders or decoders. The following datasets are used:

1. **"Text+X" to "Text" Data** Here, "X" stands for any nontextual modality included in NExT-GPT (i.e., image, video, or audio). The process involves feeding the model inputs that combine textual information with one of these modalities, to generate textual responses that accurately reflect the combined input. The datasets used include LLaVA-Instruct-150K (vision-language), miniGPT-4 image description dataset, and Videochat video instruction dataset.
2. **"Text" to "Text+X" Data** This dataset is used to generate not only textual outputs but also multimodal content, referred to as "Text to Text+X". A dataset for text-to-multimodal (T2M) data was created, utilizing a collection of "X-caption" pairs from existing corpora and benchmarks such as Conceptual Captions, MS COCO (Microsoft Common Objects in Context), AudioCaps, and more. By employing templates and GPT-4, varied textual instructions that include these captions are produced, forming a dataset that supports the generation of both textual and multimodal outputs from the text prompts.
3. **MosIT Data** In crafting NExT-GPT, a key innovation was the development of a specialized dataset named Modality-switching Instruction Tuning (MosIT) to refine the model's instruction-following capabilities across different modalities. Recognizing the shortfall in existing datasets, which did not fully capture the complexity of real-life interactions between users and AI across different formats, the creators of NExT-GPT identified a need for a more sophisticated approach. To ensure that the dataset included a rich variety of multimodal content the team sourced materials from external resources, including YouTube for videos, and various AI-generated content (AIGC) tools such as Stable-XL and Midjourney for creating images and audio clips. Each dialog within the MosIT dataset consists of 3-7 turns, with the human-AI exchanges designed to shift modalities between inputs and outputs, resulting in a dataset of 5,000 dialogs.

9.5.3.5 Output Projector

The output projector in NExT-GPT translates tokens generated by the LLM into formats suitable for modality-specific decoders. To accomplish this, NExT-GPT employs TinyTransformer (31 million parameters), which is dedicated to handling the conversion for each specific modality. The training of these output projectors occurs during the second and third stages of the overall training process.

9.5.3.6 Modality Generator

The final step in NExT-GPT involves creating outputs for different modalities with specialized decoders. This begins when the system receives multimodal signals and instructions from the LLM, which are then converted by Transformer-based layers into formats that the decoders can process. For this purpose, NExT-GPT uses leading diffusion models tailored for each modality: Stable Diffusion for images, Zeroscope for videos, and AudioLDM for audio. These models are integrated into the system as conditioned diffusion models, and fed with the transformed signal representations to generate the final content in the specified modality.

9.6 Tutorial: Fine-Tuning Multimodal Image-to-Text LLMs

9.6.1 Overview

Having discussed the theoretical underpinnings of MMLLMs in detail, we can now test the behavior of a “Text+X” to “Text” model. For this demonstration, we choose images to represent the the modality “X”. Image/text-to-image models are useful for detecting specific properties of images, categorizing the events occurring in the images, and generating automated captions, among other tasks. In this tutorial, we test the out-of-the-box capabilities of a SOTA MMLLM on image labeling and captioning and explore ways to improve performance with fine-tuning and few-shot prompting.

We set up two experiments with the same dataset to accomplish this goal. First, we will ask the model to identify which sport is in each image, both in a zero-shot framework and in a fine-tuned framework, and compare the results. Second, we will ask the model to write simple captions of what is occurring within the images, comparing zero-shot, few-shot, and fine-tuning modes.

Goals:

- Successfully set up and prompt the IDEFICS 9-billion parameter model with arbitrary text and images.
- Generate zero-shot predictions for the 100SIC test set and try to improve performance with QLoRA fine-tuning.
- Generate zero-shot captions for the 100SIC test set and compare them to fine-tuned and in-context learning captions.

Please note that this is a condensed version of the tutorial. The full version is available at https://github.com/springer-llms-deep-dive/llms-deep-dive-tutorials.

9.6.2 Experimental Design

There are many MMLLM to select from, so to narrow our choices we consider models small enough to be QLoRA-tuned in a Google Colab notebook and which are already integrated with Huggingface so that we can easily take advantage of their PEFT and fine-tuning routines. With these considerations, we choose as our model the 9 billion parameter variant of IDEFICS (Image-aware Decoder Enhanced à la Flamingo with Interleaved Cross-attentionS), an open-source text and image-to text LLM modeled on Flamingo (Laurençon et al., 2023). The model takes arbitrarily interleaved text and images as input and outputs a textual response.

The dataset we choose for this experiment is the 100 Sports Image Classification dataset (100SIC) hosted at Kaggle[1]. This set includes many small photos labeled by sport for 100 different sports. It consists of approximately 13,000 training images and 500 test and validation images. For caption fine-tuning, we supplement this dataset with a subset of the flickr30k dataset (Young et al., 2014), a 30,000+ item catalog of image and caption pairs. We used the subset extracted by Shin Thant[2], who identified flickr30k images of sports.

9.6.3 Results and Analysis

9.6.3.1 Predicting the Sport

We start by loading the model. IDEFICS is too large to predict with and tune on a single moderate GPU effectively, so we will use BitsAndBytes to quantize to 4-bit and fine-tune in the QLoRA paradigm. For sport classification, we adopt the following prompt template:

```
<image>
Question: What sport is in this image?
Answer:
```

Listing 9.1: Sport classification prompt

We use this to generate predictions for every image in the test set and compare the output against the label assigned by the compilers of the dataset:

```
- Zero-shot results:
  - 212 / 500 correct
```

Listing 9.2: Zero-shot test set predictions

It thus guessed the correct name for the sport on approximately 42% of the images. Note that we have done a simple exact-match evaluation, so if the model guesses

[1] https://www.kaggle.com/datasets/gpiosenka/sports-classification/data

[2] https://github.com/ShinThant3010/Captioning-on-Sport-Images

Table 9.7: Three cherry-picked examples demonstrating three themes of relative classification performance in increasing rareness in the zero-shot vs. fine-tuned approach. For bobsled, the zero-shot model correctly identifies the sport in most cases but does not know which name it should use. For chuckwagon racing, the model is unfamiliar with this obscure sport and guesses other types of equestrian competitions. For tug of war, fine-tuning has actually degraded the model's predictive power – this would likely improve with additional fine-tuning.

Index	**bobsled**		**chuckwagon racing**		**tug of war**	
	ZS	FT	ZS	FT	ZS	FT
1	bobsledding	bobsled	rodeo	chuckwagon racing	hurling	oxen pulling
2	the u	bobsled	horse-drawn carriage racing	chuckwagon racing	rugby	tug of war
3	bobsleigh	bobsled	calgary stampede rodeo	chuckwagon racing	tug of war	log rolling
4	bobsledding	bobsled	horseback riding	chuckwagon racing	tug of war	log rolling
5	bobsled	bobsled	chariot racing	chuckwagon racing	tug of war	axe throwing

another acceptable name for a sport, it will be considered a missed prediction. We can improve our predictions by fine-tuning the model with the training set. The Llama base model is too large for full fine-tuning, so we employ a QLoRA tuning approach similar to that discussed in the tutorial in Sect. 4.6. Selecting 10 training examples per sport as our train set and adopting the same template as in the zero-shot example to create QA pairs for fine-tuning, we fine-tune the model and again predict on the test set:

```
- Fine-tune results:
  - 419 / 500 correct
```

Listing 9.3: Fine-tune test set predictions

This shows major improvement, moving from 42% to 84% correct. We highlight a few interesting examples in Table 9.7 to demonstrate the details of this improvement.

9.6.3.2 Captioning Photos

A second common use of image-to-text models is generating automated captions. In this section, we test the capabilities of IDEFICS for this task. As before, we can use a simple zero-shot prompt template to query the model to generate a caption. For this exercise, we use an image of a Wake Forest quarterback in a black jersey throwing a pass in a game of American football and the following prompt template:

```
<image>
```

```
Question: What is a caption for this photo?
Answer:
```

Listing 9.4: Sport captioning prompt

When using zero-shot prompting, we get the following response:

```
Question: What is a caption for this photo? Answer: Aaron Murray,
     Georgia Bulldogs quarterback, throws a pass during the first
     half of the Chick-fil-A Bowl NCAA college football game
    against the Nebraska Cornhuskers
```

Listing 9.5: Sport captioning zero-shot

While this is a quarterback throwing a pass, every other piece of information in this response is false. It is not Aaron Murray nor a Georgia Bulldog, and this is not the Chick-fil-A bowl nor a game against Nebraska. All of this information was hallucinated, but notably the final two false facts are not even items that could be determined based on the image alone. Ideally we would like our captions to be straightforward descriptions of the image, downplaying specific identifying information that is not plainly visible in the photograph.

An inexpensive way to improve the model captioning is with in-context examples. For this approach, we pass several examples of training images along with hand-written captions before the target image that we are generating for. With this approach, we get the following output.

```
<image1> Question: What is a caption for this photo? Answer: A
    man prepares to throw an ax at a target.
<image2> Question: What is a caption for this photo? Answer: A
    woman rolls a bowling ball down a bowling alley.
...
<image5> Question: What is a caption for this photo? Answer:
A man in a white jersey throws a football.
```

Listing 9.6: Sport captioning in-context prompt

Under few-shot conditions, the model has generated "`A man in a white jersey throws a football.`" This is a slight mistake as the jersey color is black, but the model has formatted the caption according to our preferences and not hallucinated extraneous information like the identity of the player or their opponent. This is a promising avenue with some improvements.

A more expensive approach is to use the sports image/caption pair subset of the flickr30k dataset to fine-tune the model. We use the same QLoRA approach described above and fine-tune the base IDEFICS model with roughly 1600 samples using the same template from the zero-shot example. Once the training is complete, we can generate a caption for our test figure again.

```
A football player in a black uniform is throwing a football.
```

Listing 9.7: Sport captioning fine-tuning output

This response is both concise, similar to the few-shot response, and accurate to the photo. We generate captions for twenty test images using all three approaches as a final comparison, and qualitatively grade the responses by hand, considering both accuracy and style. The final results are:

```
- Zero-shot results:
  - 7 / 20 acceptable
- In-context results:
  - 11 / 20 acceptable
- Fine-tuning results:
  - 14 / 20 acceptable
```

Listing 9.8: Test set captioning results

9.6.4 Conclusion

Moderately sized text/image-to-text MMLLMs show considerable zero-shot capabilities but are greatly improved with fit-to-task fine-tuning. We have shown how utilizing PEFT can greatly improve image classification and open-ended captioning capabilities, even with little optimization and standard parameter choices. Production applications would clearly benefit from additional care in selecting tuning parameters, training set properties, and the MMLLM architecture itself, but only a small amount of effort is required to create a moderately well-functioning image classifier from available open-source software.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Lalit Bahl, Peter Brown, Peter De Souza, and Robert Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *ICASSP'86. IEEE international conference on acoustics, speech, and signal processing*, volume 11, pages 49–52. IEEE, 1986.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. *arXiv preprint arXiv:2312.00784*, 2023.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer, 2005.

Cerspense. Zeroscope: Diffusion-based text-to-video synthesis, 2023.

Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*, 2023.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

Huyen Chip. Multimodality and large multimodal models (lmms), 2023. URL https://huyenchip.com/2023/10/10/multimodal.html.

A Chowdhery, S Narang, J Devlin, M Bosma, G Mishra, A Roberts, P Barham, HW Chung, and C Sutton. S. gehrmannet al.,"palm: Scalinglanguage modeling with pathways,". *arXiv preprint arXiv:2204.02311*, 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Chenhang Cui, Yiyang Zhou, Xinyu Yang, Shirley Wu, Linjun Zhang, James Zou, and Huaxiu Yao. Holistic analysis of hallucination in gpt-4v (ision): Bias and interference challenges. *arXiv preprint arXiv:2311.03287*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023.

Vaishnavi Himakunthala, Andy Ouyang, Daniel Rose, Ryan He, Alex Mei, Yujie Lu, Chinmay Sonar, Michael Saxon, and William Yang Wang. Let's think frame by frame: Evaluating video chain of thought with video infilling and prediction. *arXiv preprint arXiv:2305.13903*, 2023.

Geoffrey E Hinton and Russ R Salakhutdinov. A better way to pretrain deep boltzmann machines. *Advances in Neural Information Processing Systems*, 25, 2012.

Jordan Hoffmann et al. Training compute-optimal large language models, 2022.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Yupan Huang, Zaiqiao Meng, Fangyu Liu, Yixuan Su, Nigel Collier, and Yutong Lu. Sparkles: Unlocking chats across multiple images for multimodal instruction-following models. *arXiv preprint arXiv:2308.16463*, 2023.

Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, et al. Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

COL Stephen A LaRocca, John J Morgan, and Sherri M Bellinger. On the path to 2x learning: Exploring the possibilities of advanced speech recognition. *Calico Journal*, pages 295–310, 1999.

Hugo Laurençon et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11336–11344, 2020a.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, and Lingpeng Kong. Silkie: Preference distillation for large visual language models. *arXiv preprint arXiv:2312.10665*, 2023.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020b.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021.

Shengzhi Li and Nima Tajbakhsh. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*, 2023.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020c.

Zheng Lian, Licai Sun, Mingyu Xu, Haiyang Sun, Ke Xu, Zhuofan Wen, Shun Chen, Bin Liu, and Jianhua Tao. Explainable multimodal emotion reasoning. *arXiv preprint arXiv:2306.15401*, 2023.

Junyang Lin, An Yang, Yichang Zhang, Jie Liu, Jingren Zhou, and Hongxia Yang. Interbert: Vision-and-language interaction for multi-modal pretraining. *arXiv preprint arXiv:2003.13198*, 2020.

Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *arXiv preprint arXiv:2311.16103*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. *arXiv preprint arXiv:2312.02051*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

Shin'ichi Satoh and Takeo Kanade. Name-it: Association of face and name in video. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 368–373. IEEE, 1997.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction. *arXiv preprint arXiv:2201.02184*, 2022.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2022.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Gokhan Tur, Andreas Stolcke, Lynn Voss, Stanley Peters, Dilek Hakkani-Tur, John Dowding, Benoit Favre, Raquel Fernández, Matthew Frampton, Mike Frandsen, et al. The calo meeting assistant system. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1601–1611, 2010.

Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard, and Alex Pentland. Social signal processing: state-of-the-art and future perspectives of an emerging domain. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 1061–1070, 2008.

Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pretrained models: A comprehensive survey. *Machine Intelligence Research*, pages 1–36, 2023.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023a.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and S Yu Philip. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023b.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023c.

Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2e-vlp: end-to-end vision-language pre-training enhanced by visual learning. *arXiv preprint arXiv:2106.01804*, 2021.

Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Xiaoshui Huang, Zhiyong Wang, Lu Sheng, Lei Bai, et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. *arXiv preprint arXiv:2312.00849*, 2023.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Xunlin Zhan, Yangxin Wu, Xiao Dong, Yunchao Wei, Minlong Lu, Yichi Zhang, Hang Xu, and Xiaodan Liang. Product1m: Towards weakly supervised instance-level product retrieval via cross-modal pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11782–11791, 2021.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024a.

Ge Zhang, Xinrun Du, Bei Chen, Yiming Liang, Tongxu Luo, Tianyu Zheng, Kang Zhu, Yuyang Cheng, Chunpu Xu, Shuyue Guo, et al. Cmmmu: A chinese massive multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2401.11944*, 2024b.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Jiawei Zhang, Tianyu Pang, Chao Du, Yi Ren, Bo Li, and Min Lin. Benchmarking large multimodal models against common corruptions. *arXiv preprint arXiv:2401.11943*, 2024c.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *Advances in Neural Information Processing Systems*, 35:3609–3623, 2022.

Zijia Zhao, Longteng Guo, Tongtian Yue, Sihan Chen, Shuai Shao, Xinxin Zhu, Zehuan Yuan, and Jing Liu. Chatbridge: Bridging modalities with large language model as a language catalyst. *arXiv preprint arXiv:2305.16103*, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Chapter 10
LLMs: Evolution and New Frontiers

Abstract This concluding chapter provides an overview of the evolution of LLMs, emphasizing significant trends and developments. It explores the shift toward synthetic data to sustain model scaling, the expansion of context windows enhancing interpretative capabilities, the progression of training techniques that streamline efficiency and depth of knowledge transfer, and the transition from traditional Transformer architectures to alternative approaches such as state space models, which offer improved scalability and efficiency. Further discussion highlights the trends of smaller models, technology democratization, and domain-specific models, illustrating a movement toward more customized, accessible, and industry-specific AI solutions. Finally, the chapter delves into the frontiers of LLM technologies and their use in agent-based applications and search engines, which are increasingly replacing traditional technologies.

10.1 Introduction

The evolution of large language models encompasses significant architectural advancements, training techniques, and application trends. Innovations in model architecture and training efficiency have propelled LLMs to new heights, enabling them to handle more complex and extensive tasks. The shift toward synthetic data and larger context windows exemplifies the ongoing efforts to enhance model capabilities and performance. Emerging trends such as small language models, democratization through open-source initiatives, and domain-specific language models highlight the diverse applications and accessibility of LLMs. Additionally, new frontiers in LLM agents and enhanced search capabilities are setting new standards for complex task execution and information retrieval, further expanding the potential of LLMs in various fields.

U. Kamath et al., *Large Language Models: A Deep Dive*,
https://doi.org/10.1007/978-3-031-65647-7_10

10.2 LLM Evolution

10.2.1 Synthetic Data

As AI models increase in size and exhaust readily available high-quality internet data, there is a pressing need to shift toward synthetic data to sustain model development and achieve the necessary scaling. This trend assumes that increasing data quantities will enhance model performance, particularly for complex, rare tasks. While some argue that synthetic data may not advance state-of-the-art models because it mirrors existing data distributions, others believe that their diversity could improve models.

Anthropic leverages synthetic data extensively in its AI models, notably Claude 2.1, to enhance robustness by accurately refusing questions it cannot answer. Their approach, Constitutional AI (CAI), uses synthetic data in two primary ways: critiquing responses based on a set of ethical principles and generating pairwise preference data to train models using RLHF, a process known as RLAIF, as discussed in Chapter 5. CAI's dual approach—-principled instruction correction and principle-following RLHF—-has proven effective, allowing Anthropic to excel in synthetic data utilization and model training despite its relatively small team (Bai et al., 2022).

Models such as Alpaca and Vicuna utilize synthetic data for supervised fine-tuning of Llama models, enhancing performance within the 7-13B parameter range (Peng et al., 2023; Taori et al., 2023). Current trends include the use of methods such as Self-Instruct, where an LLM generates diverse instructional data from seed instructions. However, efforts are still in the initial stages to explore methods to enrich data diversity. In contrast, some still use low-quality internet prompts repurposed as training instructions by models such as GPT-4.

Synthetic preference datasets such as UltraFeedback collect user-generated prompts and model completions for RLHF training (Cui et al., 2023). Teknium1 has been actively employing synthetic instructions to train models such as OpenHermes on Mistral (Gallego, 2024). Meanwhile, Intel's recent LLM, Neural-Chat-v3-1, uses the DPO model to incorporate synthetic preferences. Berkeley's Starling model utilizes Nectar, a GPT-4-labeled ranking dataset. It aggregates prompts and scores from various models such as GPT-4, GPT-3.5-instruct, GPT-3.5-turbo, Mistral-7B-Instruct, and Llama-2-7B, resulting in a total of 3.8 million pairwise comparisons. Starling has achieved state-of-the-art performance on MT Bench 7b, although concerns about data contamination have been noted (Zhu et al., 2023a). Quality-Diversity through AI Feedback (QDAIF) employs evolutionary algorithms to boost data diversity (Bradley et al., 2023). Evol-instruct uses a rule-based system to generate diverse, high-quality instructions with feedback from GPT-4 (Xu et al., 2023).

10.2.2 Larger Context Windows

The context window of an LLM acts as a lens, providing perspective and functioning as short-term memory, and is useful for generation-based and conversation-based tasks. Larger context windows enhance an LLM's ability to learn from prompts by allowing for the input of more extensive and detailed examples, which results in more accurate and relevant responses. Additionally, a substantial context window enhances the model's ability to understand and connect information across distant parts of the text, which is especially beneficial for tasks requiring detailed document summarization, question-answering, and chatbot conversations, where larger context windows help maintain coherence over longer interactions.

The evolution of GPT models has shown substantial increases in context window size. Starting from a 2,000-token limit with GPT-3, the capacity expanded to 4096 tokens in the initial GPT-4 model. This was extended to 32768 tokens in the GPT-4–32k variant. The latest model, GPT-4 Turbo, now supports up to 128000 tokens, representing a 32x improvement over the initial GPT-4 and a 4x increase from GPT-4–32k, enhancing its ability to analyze and interpret extensive text data. Claude by Anthropic supports a 9,000 token context, and its successor, Claude 2, significantly extends this capacity to 100,000 tokens, allowing it to process documents up to 75,000 words in a single prompt. Meta AI's Llama family of models also supports more than 100,000 tokens.

Rotary Position Embeddings (RoPE) enhance Transformer models by embedding token positions directly into the model (Su et al., 2024). This technique involves rotating the position embeddings relative to each token's sequence position, facilitating consistent token position identification as the context window increases. Positional Skip-wise Training (PoSE) focuses on efficient context window extension for LLMs through a novel training technique that skips positions in a controlled manner, improving the handling of extended contexts in training and inference phases (Zhu et al., 2023b). LongRoPE extends LLM context windows to more than 2 million tokens, pushing the boundaries of current context management technologies and utilizing advanced rotational embeddings to handle extremely long inputs effectively (Ding et al., 2024).

Munkhdalai et al. (2024) introduce a method for scaling LLMs to handle extremely long inputs using a new attention technique called Infini-attention. Their approach integrates compressive memory with local and long-term linear attention mechanisms, demonstrating success in handling up to 1 million tokens for context retrieval and 500,000 tokens for book summarization tasks.

10.2.3 Training Speedups

This section discusses various techniques developed to enhance the efficiency of Transformer models. Despite their significant improvements in sequence modeling tasks, Transformer models suffer from high computational and memory costs due to their quadratic complexity with respect to sequence length. Innovations such as parameter sharing, pruning, mixed-precision, and micro-batching have addressed these challenges, enabling more practical and widespread adoption of Transformer technology (Fournier et al., 2023).

Techniques such as *gradient checkpointing* involve selectively storing activations during the forward pass, which are then recomputed during the backward pass to save memory. This trade-off between memory and computational overhead allows scaling up the number of layers without linearly increasing memory use. The *parameter sharing* approach reduces the number of trainable parameters by reusing the same parameters across different parts of the network. Techniques such as *pruning* enhance model efficiency by removing less important weights after training. It can be applied in a structured manner, affecting components such as layers or attention heads, or unstructured, targeting individual weights. Pruning helps build smaller, faster models that are better optimized for modern computational hardware.

To increase the training speed and decrease the memory consumption of deep learning models, modern GPUs and TPUs utilize mixed-precision techniques. They perform computations in half-precision (16 bits) while maintaining a master copy of weights in single-precision for numerical stability. NVIDIA's Automatic Mixed-Precision simplifies integration with frameworks like TensorFlow, PyTorch, and MXNet. GPipe facilitates model scaling and performance improvement by allowing large models to be distributed across multiple processing units through an innovative micro-batching technique. This method splits mini-batches into smaller micro-batches, enabling parallel processing and reducing memory demands during forward and backward operations. This strategy allows for significant scaling in model size proportional to the number of accelerators used, enhancing training throughput without sacrificing computational efficiency.

10.2.4 Multi-Token Generation

Traditional LLMs using conventional next-token prediction are resource intensive and often fail to capture long-term dependencies effectively. Meta's research presents a novel approach to training LLMs through multi-token prediction. This method diverges from traditional next-token prediction by forecasting several future tokens simultaneously, enhancing both efficiency and performance (Gloeckle et al., 2024). This technique triples inference speed and increases sample efficiency, particularly in larger models and coding tasks. Meta's 13-billion-parameter model demonstrated a 12% and 17% improvement in problem-solving capabilities on the HumanEval and MBPP benchmarks, respectively.

The approach relies on a shared model trunk that processes input sequences into a latent representation, with multiple output heads designed to predict different future tokens independently. This structure allows for parallel token predictions without increasing computational demands during training. During inference, the model uses the trained output heads to generate multiple tokens simultaneously, further speeding up the process and reducing latency.

10.2.5 Knowledge Distillation

Knowledge distillation (KD) involves transferring insights from a large, sophisticated model (the teacher) to a smaller, more efficient model (the student). Given the significant computational demands and resource constraints of large-scale models, this process has become crucial for practical deployment. With the rise of LLMs such as GPT-4 and Gemini, the focus of knowledge distillation has evolved from simply reducing model size or mimicking outputs to a more intricate transfer of deep-seated knowledge.

This shift is primarily due to the rich and nuanced understanding these LLMs have developed, which cannot be fully captured by traditional compression methods such as pruning or quantization. Instead, the contemporary approach in LLM-based knowledge distillation leverages carefully crafted prompts to extract specific knowledge or capabilities. These prompts tap into the LLM's expertise across various domains, including natural language processing, reasoning, and problem solving. This strategy allows for more targeted and dynamic knowledge transfer, focusing on particular skills or areas of interest.

Moreover, the current phase of knowledge distillation extends beyond simple output replication. It aims to transfer more abstract qualities such as reasoning patterns, preference alignment, and ethical values. Modern techniques involve teaching the student model to emulate the teacher's thought processes and decision-making patterns. This is often achieved through chain-of-thought prompting, which trains the student model to understand and replicate the teacher's reasoning process, enhancing cognitive capabilities across complex tasks.

In their survey, Xu et al. (2024) categorize the exploration of KD into three primary facets: KD algorithms, skill distillation, and verticalization distillation, each encompassing a variety of methodologies and subtopics.

KD algorithms focus on the foundational techniques of knowledge distillation, detailing how knowledge is constructed from teacher models and integrated into student models. It covers labeling, expansion, curation, feature understanding, feedback mechanisms, and self-knowledge generation. Additionally, it discusses various learning approaches, including supervised fine-tuning, divergence minimization, reinforcement learning, and rank optimization to facilitate effective knowledge transfer, enabling open-source models to match or exceed the capabilities of proprietary models.

Skill distillation addresses enhancing specific competencies through KD, including context following, instruction adherence, retrieval-augmented generation, alignment in thinking patterns, persona/preference modeling, and value alignment. It also explores NLP task specialization, such as natural language understanding and generation, information retrieval, recommendation systems, text generation evaluation, and code generation. Furthermore, this segment investigates how KD improves LLMs' ability to handle multi-modal inputs, enhancing their functionality across different contexts.

Verticalization distillation evaluates the application of KD across specialized fields such as law, healthcare, finance, and science, illustrating how KD adapts LLMs to specific industry needs. This highlights the transformative impact of KD techniques on domain-specific AI solutions, and it underscores their versatility and effectiveness in meeting the varied demands of different industries within the AI and machine learning ecosystem.

10.2.6 Post-Attention Architectures

State space models (SSMs) have emerged as a focal point in the evolution of deep learning technologies, particularly in addressing the limitations of traditional neural network architectures such as CNNs, RNNs, GNNs, and even Transformers. These models represent dynamic systems through state variables initially drawn from control theory and computational neuroscience. The Mamba model enhances computational efficiency, achieving 5x faster inference and linear scalability compared to Transformers. It features input-adaptive SSMs for better content reasoning, significantly outperforming same-sized Transformers and matching those twice its size in language, audio, and genomics tasks (Gu and Dao, 2023).

In language modeling, researchers have explored applications such as the Gated State Space (GSS) method for long-range language modeling, which offers substantial speed improvements and reduced computational overhead (Mehta et al., 2022). The Structured State Space sequence model (S4) introduces a new, more efficient parameterization for state space models, achieving significant computational savings and strong performance across benchmarks. S4 matches or surpasses previous models in tasks such as sequential CIFAR-10 and image/language modeling, performs generation 60× faster, and sets new records in the Long Range Arena benchmark, effectively handling sequences up to 16,000 in length (Gu et al., 2021).

10.3 LLM Trends

10.3.1 Small Language Models

LLMs have been central to advancements in numerous fields, yet the substantial computational resources required for these models have generally limited their use to well-resourced organizations. Increasingly, researchers are working to replicate the capabilities of large models in much smaller packages. *Small Language Models* (SLMs) are scaled-down versions of LLMs. They possess far fewer parameters—

ranging from millions to billions–than the hundreds of billions or trillions found in LLMs. The smaller size of SLMs offers several benefits:

1. **Efficiency**: SLMs consume less power and need less memory, making them suitable for deployment on smaller devices as in the case of edge computing. This capability facilitates practical applications, such as on-device chatbots and personal mobile assistants, that can operate directly from a user's device.
2. **Accessibility**: The reduced resource demands of SLMs make them more attainable for a wider spectrum of developers and organizations. This broad accessibility helps democratize artificial intelligence, enabling even small teams and independent researchers to leverage the capabilities of language models without the need for substantial infrastructure.
3. **Customization**: SLMs are simpler to adapt to specific domains and tasks, making it possible to develop specialized models that are precisely tailored to specific needs. This customization can lead to improved performance and greater accuracy in niche applications.
4. **Enhanced Security and Privacy**: A notable advantage of SLMs is their potential for improved security and privacy. Their manageable size allows for deployment on-premises or within private cloud environments, which minimizes the risk of data breaches. This feature is particularly valuable in industries that handle sensitive information, such as finance and healthcare, where maintaining control over data is crucial.

Here are some popular small language models currently making waves in the industry, although this is by no means an exhaustive list.

1. **Llama**: The Llama-3 model is an open-access, 2.7 billion-parameter tool proficient in handling nuanced language tasks, translation, and dialog generation (Touvron et al., 2023).
2. **Phi-2**: Developed by Microsoft, this model utilizes 2.7 billion parameters to achieve exceptional performance in mathematical reasoning, common sense evaluations, and logical tasks (Javaheripi et al., 2023). Phi-2 employs synthetic data for training, competing with, and sometimes surpassing, models ten times its size in tasks such as reading comprehension and text summarization.
3. **Mistral 7B**: A robust model with 7.3 billion parameters, Mistral 7B surpasses the performance of previous Llama models and approaches the capabilities of specialized code models (Jiang et al., 2023a). It integrates advanced techniques like grouped-query attention for faster processing and sliding window attention to manage longer text sequences efficiently.
4. **Gemma 2B and Gemma 7B**: These variants, both pre-trained and instruction-tuned, excel in text-based tasks, outperforming comparable open models in 11 out of 18 evaluations (Team et al., 2024). The development of Gemma models also emphasizes safety and responsibility, ensuring their reliability in practical applications.
5. **Vicuna-13B**: This open-source conversational model, based on the Llama-13B framework, is enhanced by fine-tuning on user-shared conversations. Initial evaluations, with GPT-4 as the benchmark, indicate that Vicuna-13B delivers quality

surpassing 90% of that seen in models such as ChatGPT and Google Bard. It outperforms other models such as Llama and Alpaca in the majority of tests (Peng et al., 2023).

10.3.2 Democratization

Recent months have seen transformative changes in LLMs, fueled largely by the expanding influence of the open-source community. The essence of open source—-marked by its commitment to collaborative development, transparency, and free access-—has profoundly impacted the progress of LLMs. LLMs' open-source initiatives encompass various resources, including pre-training data, models and architectures, instruction-tuning datasets, alignment-tuning datasets, and even hardware.

Petals addresses the challenges of researchers who lack access to the high-end hardware necessary for leveraging LLMs such as BLOOM-176B and OPT-175B (Borzunov et al., 2022). Petals enables collaborative inference and fine-tuning of these large models by pooling resources from those who want to share their GPU cycles. It provides a solution faster than RAM offloading for interactive applications, with the ability to run inference on consumer GPUs at approximately one step per second.

Hugging Face's ZeroGPU initiative uses Nvidia A100 GPUs to provide shared, on-demand GPU access via their Spaces app, aiming to democratize access to computational resources and reduce costs for smaller organizations.

Various datasets related to pre-training, instruction tuning, alignment tuning, and more, are continuously made available to the community. Contributors regularly release open-source datasets online, and initiatives such as LLMDataHub and Open LLM Datasets are instrumental in centralizing these resources. This central repository simplifies access and utilization for developers and researchers engaged in LLM development.

OpenLLM enables developers to operate any open-source LLM, such as Llama-2 or Mistral, through OpenAI-compatible API endpoints both locally and in the cloud (Pham et al., 2023). This platform supports a wide range of LLMs, facilitates seamless API transitions for applications, and offers optimized serving for high-performance and simplified cloud deployment using BentoML.

While open-source LLMs are discussed extensively in Chapter 8, readers seeking the latest developments can refer to the Hugging Face leaderboard at HuggingFace for ongoing updates and rankings.

10.3.3 Domain-Specific Language Models

Domain-Specific Language Models (DSLMs) address the limitations of general purpose models by specializing in particular industries or fields. These models are finely tuned with domain-specific data and terminology, making them ideal for complex and regulated environments where precision is essential. This targeted approach ensures that DSLMs provide accurate and contextually appropriate responses, reducing the likelihood of errors and "hallucinations" that general-purpose models may produce when faced with specialized content.

DSLMs are particularly beneficial for professionals such as lawyers, medical providers, and financial analysts who rely on precise and reliable information. By focusing on a narrower scope and incorporating industry-specific jargon, these models are designed to effectively handle the specific workflows and processes of their designated fields. As enterprises increasingly recognize the value of tailored AI solutions, it is projected that by 2027, more than half of the generative AI models employed by businesses will be domain specific, serving distinct industrial or functional needs.

In the legal field, SaulLM-7B, developed by Equall.ai, is a prime example of employing legal-specific pre-training and fine-tuning to address the complexities of legal language, significantly improving task performance in legal applications (Colombo et al., 2024). In healthcare, models such as GatorTron, Codex-Med, Galactica, and Flan-PaLM have been developed to address the nuances of medical data and clinical information, pushing the boundaries of what AI can achieve in diagnosing and managing patient care (Singhal et al., 2023; Taylor et al., 2022; Yang et al., 2022, 2023). Similarly, the finance sector has seen advancements with models such as BloombergGPT and FinBERT, trained on extensive financial data to enhance tasks such as risk management and financial analysis (Liu et al., 2021; Wu et al., 2023).

10.4 New Frontiers

10.4.1 LLM Agents

LLM agents represent a framework for leveraging LLM capabilities to accomplish highly complex and sophisticated tasks. These agents are modular programs that can read in a user request, reason through the steps required to complete it, create and allocate sub-tasks to various modules, and synthesize the results into a satisfactory output. The key feature of this framework is a blend of traditional computing logic and system tools with LLMs prompted to use them intelligently.

For example, consider the task of conducting market research on different available headphones and choosing a few top options based on pricing, features, and user reviews. ChatGPT alone cannot accomplish this goal, as it can only access information based on data seen during pre-training. An LLM agent, on the other hand, can read this request, decide it needs to search the internet for information, construct the relevant search queries, execute the searches, download the resulting pages, process the information, and return a series of suggestions.

Although there are different flavors, agents generally have a few common modules, graphically illustrated in Fig. 10.1 and listed here:

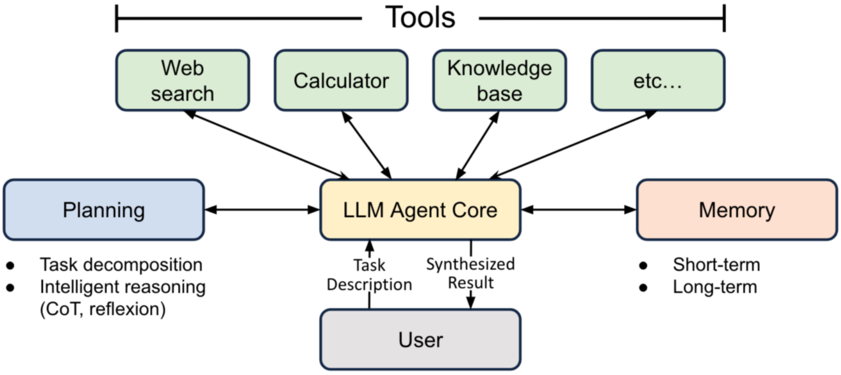


Fig. 10.1: Visualization of the high-level modules in an LLM agent. The core module takes in a user task, accesses relevant information from the memory module, allocates sub-task decomposition to the planning module, and uses the available tools to accomplish the sub-tasks. Finally, the core synthesizes the information to respond to the task and returns the result to the user.

- **Core:** This is the central module that defines the characteristics of the agent, ingests user instructions, and allocates tasks to other modules. This is accomplished by a lengthy and highly specific prompt template that instructs the LLM on how to execute these tasks.
- **Planning:** This module determines the series of steps needed to accomplish the overall task. Using reasoning approaches like Chain-of-Thought (Sect. 4.5.4) and iterative improvement algorithms like Reflexion (Shinn et al., 2023), it develops and refines a plan consisting of a sequence of sub-tasks which can be executed by the various functions of the agent.
- **Tools:** A series of tools available to the agent that go beyond the standard capabilities of LLMs. The possibilities for this section are endless but may consist of web search commands, code compilers, calculators, and API calls of any sort.
- **Knowledge:** A knowledge base that can be queried by the agent if necessary. This could be a RAG system similar to those described in Chapter 7, or a structured database that can be queried through calls (e.g. SQL) that can be generated by the language model.
- **Memory:** This module contains a record of information derived from interactions between the user and agent, which can be reviewed if deemed necessary for a given task. Sometimes, it is divided between short-term memory, which has granular details of all interactions of the current session, and long-term memory, which is a more curated list of relevant information learned over the course of many interactions.

These agents can be carefully crafted for specific tasks such as scientific writing (Ziems et al., 2024), playing video games (Wang et al., 2023), manipulating robots (Michael Ahn, 2022), and more. Researchers have also developed generalist agents that will attempt any task given by the user. An early example is *AutoGPT*[1], which closely follows the layout in Fig. 10.1–it takes in a user command, uses crafted prompt templates in the core to establish a workflow, engages in chain-of-thought reasoning and self-criticism to generate a plan, and leverages memory modules and tools to accomplish the goal. Notably, this model accepts no user feedback on its plan, autonomously attempting the entire task-solving process. Another popular agent base is *BabyAGI*[2], which is similar in big picture layout to AutoGPT but iterates on its plan after every task instead of executing a decided-on string of tasks.

! Practical Tips

Many agents are built on the backs of open-source packages designed to handle complex LLM frameworks. With popular examples such as LangChain and LlamaIndex (Sect. 8.6.1), these packages implement many functions for calling LLMs, integrations for common tools, a suite of prompt templates for many use cases, and web-hosting features. BabyAGI, in particular, uses LangChain integrations in its workflow, and the symbiosis goes both directions–LangChain has integrated AutoGPT and BabyAGI into their product, allowing agent systems seamless access to the different LLMs, vector indices, and tools already implemented by LangChain.

10.4.2 LLM-Enhanced Search

Another frontier of LLM applications is enhancing search capabilities in different contexts. LLM-powered search has the potential to improve web and document search algorithms in a few different ways, which fall into a few broad categories:

- Improving top search results
- Query engineering
- Reasoning from search results

Improving top search results

Traditional search methods rely on keyword matches against a search query, but there is a major limitation to this approach – it cannot return text on a subject similar to the search query but without the exact keywords. Word vectors have expanded the search

[1] `https://github.com/Significant-Gravitas/AutoGPT`

[2] `https://github.com/yoheinakajima/babyagi`

range of individual terms, producing matches on semantically similar terms. Sentence embeddings with language models go a step further, matching longer phrases with semantic similarity. As embedding models improve, there is promise that the scope of search retrieval will sharpen. This is accomplished with reranking (Sect. 7.4.3.1). Reranking involves collecting several top results using an efficient search algorithm and dynamically reranking them with a slower but more powerful ranking algorithm. In the context of web search, a traditional keyword-based internet search may return hundreds of thousands of matches with a less sophisticated top ranking, and the top-k results can be reranked based on semantic similarity between the query and the contents of the web page. Such techniques have been the industry standard in web searching since the advent of Transformer models but have gained greater capabilities in the era of powerful LLM-based agents that can precisely parse the meaning of human language.

Improving queries

LLMs also offer the possibility of improving search querying. We have discussed certain of these approaches in the context of RAG (Chapter 7). These include:

- Query re-writing: Training a model to take in a human search query and refashion it into a form more likely to return relevant searches.
- Query-to-document: Generating synthetic documents of a form similar to the desired search result to create a closer match.
- Query-to-SQL: Using an LLM to convert human language queries against a structured database into a code-based call (see Listing 7.4.1.1 in Sect. 7.4.1.1).

Forward-Looking Active REtrieval augmented generation (FLARE) is an additional technique for extracting more relevant information for a search (Jiang et al., 2023b). In FLARE, an LLM generates successive queries off the back of the original query, imagining new contexts with potentially relevant information, executing those searches and incorporating the new results.

Reasoning from search results

Another way that LLMs revolutionize search can be understood by analogy to retrieval-augmented generation (Chapter 7). Semantic searching of documents is already a step in the RAG pipeline, and instead of using the resulting documents to answer the original query, a RAG-powered search engine will simply return the most semantically similar documents. This technique is viable for single-document or web domain searches and represents an improvement over traditional keyword-based searches, which struggle to detect similar, but not identical, subjects to the query.

The RAG-style search-and-describe approach is also useful for web searches. Instead of simply returning top hits to a query, an LLM-powered search engine can return top semantically similar matches from pre-indexed webpages, extract the information from the pages, and use them as context to directly answer the query. The company `You.com` created a web portal with similar functionality. Using a web-based query page similar to other chatbots, `You.com` is deeply integrated with a