applied sciences

MDPI

Review

From Large Language Models to Large Multimodal Models: A Literature Review

Dawei Huang [1], Chuan Yan [2], Qing Li [3,*] and Xiaojiang Peng [3,*]

[1] College of Applied Science, Shenzhen University, Shenzhen 518052, China; huangdawei2023@email.szu.edu.cn
[2] Department of Computer Science, George Mason University, Fairfax, VA 22030, USA; cyan3@gmu.edu
[3] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
* Correspondence: liqing@sztu.edu.cn (Q.L.); pengxiaojiang@sztu.edu.cn (X.P.)

Abstract: With the deepening of research on Large Language Models (LLMs), significant progress has been made in recent years on the development of Large Multimodal Models (LMMs), which are gradually moving toward Artificial General Intelligence. This paper aims to summarize the recent progress from LLMs to LMMs in a comprehensive and unified way. First, we start with LLMs and outline various conceptual frameworks and key techniques. Then, we focus on the architectural components, training strategies, fine-tuning guidance, and prompt engineering of LMMs, and present a taxonomy of the latest vision–language LMMs. Finally, we provide a summary of both LLMs and LMMs from a unified perspective, make an analysis of the development status of large-scale models in the view of globalization, and offer potential research directions for large-scale models.

Keywords: large language models (LLMs); large multimodal models (LMMs); artificial intelligence

Citation: Huang, D.; Yan, C.; Li, Q.; Peng, X. From Large Language Models to Large Multimodal Models: A Literature Review. *Appl. Sci.* **2024**, *14*, 5068. https://doi.org/10.3390/app14125068

Academic Editor: Douglas O'Shaughnessy

Received: 26 April 2024
Revised: 28 May 2024
Accepted: 6 June 2024
Published: 11 June 2024

1. Introduction

In recent years, Large Language Models (LLMs) such as BERT [1], the GPT series [2–5], PaLM series [6,7], LLaMA series [8,9], and PanGu series [10,11] have continuously developed and matured, demonstrating powerful abilities in text understanding and generation across various tasks. Concurrently, cross-modal models in the Computer Vision community, such as CLIP [12] and Stable Diffusion [13,14], have emerged, achieving new heights in image understanding and generation tasks. Moreover, Large Multimodal Models (LMMs) that evolved from LLM foundations have made significant strides and breakthroughs, gradually forming the embryonic shape of general-purpose Artificial General Intelligence (AGI).

Despite the existence of numerous reviews focusing on either LLMs or LMMs, as yet there is no comprehensive and systematic review that covers the entire evolutionary process from LLMs to LMMs. Overviews of LLMs [15,16] focus on systematically presenting the background, core concepts, architectures, training strategies, application scenarios, datasets, evaluation benchmarks, and existing challenges, and outline their overall evolutionary trajectory, but fail to address the connection between LLMs and LMMs. On the other hand, the review of LMMs in [17] provides a thorough introduction to their recent progress, including design architectures and future directions, listing approximately 120 state-of-the-art model examples; however, it neglects to delve into the training techniques pertinent to LMMs. Surveys in [18], conversely, delve into LMM-related training techniques and prompt engineering, yet fall short in offering a macro-level analysis of LMMs. While these existing reviews contribute valuable insights, they fail to adequately integrate and comparatively analyze the commonalities and differences in the technological pathways of LLMs and LMMs. Considering that LMMs are fundamentally constructed by incorporating additional modal processing components into the foundation of LLMs, both share many similarities in their architectural design and key techniques, making it particularly crucial to introduce LLMs and LMMs under a unified perspective.

Furthermore, current review works overlook the comparative analysis of the development status of large models across different regions in view of globalization. The current landscape shows a disparity in that certain enterprises and academic institutions in particular regions enjoy monopolistic advantages in large model development, while many third-world countries either lack developed large models or recognition for their models within the academic community. Thus, based on publicly disclosed commercial large models and academic research outcomes, conducting a comparative analysis of the development situations of LLMs and LMMs worldwide is essential for assessing the balance of large model technology advancements.

This article begins with an in-depth discussion of key LLM techniques in Section 2 and an introduction to several influential LLMs in Section 2.6. Then we move on to LMMs, presenting a overview of the relevant architectural components, training strategies, instruction tuning, and prompt engineering in Section 3. In addition, we focus on the currently most popular visual language domain and conduct a taxonomy of the latest 66 visual language LMMs in Section 3.5, in which we analyze the different types of vision language LMMs. Finally, we offer a unified view by conducting a review and analysis of both LLMs and LMMs in Section 4, contrasting the balance and fairness of large model development across different global regions and summarizing potential future technical trends for large models in Section 6. We aim to provide a more profound unified viewpoint to grasp the connections between the expansion from LLMs to LMMs and support work on large model-related research endeavors.

2. Large Language Models

In this section, we conduct a systematic survey of LLMs, reviewing their architectural designs, pretraining, and fine-tuning techniques, where we categorize fine-tuning into full fine-tuning and parameter-efficient fine-tuning. Subsequently, we summarize the mainstream prompt engineering approaches for LLMs. Additionally, in Section 2.6 we present an exhaustive list of ten representative pretrained LLMs; we provide a summary of these models in Table 1.

2.1. LLM Architectures

Encoder–Decoder: This architecture, originating from RNN [19] and LSTM [20], was first instantiated in LLMs using the transformer concept [21]. Colloquially referred to as a sequence-to-sequence framework, it usually consists of encoder and decoder modules. The encoder module encapsulates the salient features and semantic attributes from the input text, while the decoder module sequentially generates output text sequences based on the information conveyed by the encoder.

Encoder-Only: This architecture focuses on the encoder side. It typically acquires contextual language representations using a bidirectional self-attention mechanism, and is primarily intended for scenarios involving one-way tasks that require only input processing, such as text categorization and sentiment analysis. Representative models of this class include BERT [1], RoBERTa [22], and ALBERT [23].

Decoder-Only: This architecture concerns prediction from the succeeding output token within a sequence. It can be bifurcated into two variants: Casual Decoder andPrefix Decoder (Non-Causal Decoder), with the distinction lying in the attention mechanism employed. Casual decoder exclusively relies on previous tokens for prediction of the next tokens, whereas prefix decoder does not strictly depend on already-produced tokens, instead employing bidirectional attention [24]. Th decoder-only architectural framework is particularly apt for text generation tasks, and constitutes the prevalent choice among current large language models, as exemplified by the GPT series [2–4].

2.2. LLM Pretraining

Pretraining is one of the most important steps in the training of large language models. It refers to the initial self-supervised training of the model on a large corpus. This process

involves the design of specific pretraining goals that allow the model to learn generic feature representations or latent structures from unlabeled data. Common pretraining goals include the following three main strategies:

Autoregressive Language Modeling (ALM): ALM models are trained to predict each subsequent token based on the sequence of preceding tokens. This pretraining task is typically employed in decoder-only architectures, such as the GPT series [2–4].

Prefix Language Modeling (PLM): PLM models are trained to forecast subsequent text based on a partial prefix within the input text. This prefix is often randomly selected. This pretraining objective is pertinent to both encoder–decoder and prefix-based decoder-only architectures. An example is UniLM [25].

Masked Language Modeling (MLM): MLM operates by randomly masking certain tokens within the input sequence using a designated mask token. The model is then required to infer the identities of these masked tokens using only the unmasked contextual information. This strategy is instrumental in BERT-like models [1,24]. MLM enables the model to learn from bidirectional context, as it simultaneously considers both preceding and succeeding contexts within a sequence. In contrast to ALM and PLM, MLM provides the model with a more comprehensive understanding of context while maintaining parallelization efficiency.

2.3. LLM Full Fine-Tuning

Full fine-tuning refers to the process by which a large language model is adapted to specific downstream tasks. This involves updating all parameters of the pretrained model using task-specific data [1]. This constitutes a concrete manifestation of transfer learning within LLMs. For instance, GPT [2] initially undergoes pretraining on a vast corpus and subsequently undergoes full fine-tuning across twelve diverse NLP downstream tasks, such as natural language inference, question answering, and semantic similarity. Following this fine-tuning phase, the GPT variant achieves state-of-the-art performance on nine of these targeted tasks.

Full fine-tuning currently represents the most prevalent approach for large language models to tackle downstream NLP tasks. However, a significant limitation of full fine-tuning lies in the escalating scale of LLMs. As the number of parameters within these models continues to grow, unaffordable computational consumption results.

2.4. LLM Parameter-Efficient Fine-Tuning

Distinct from full fine-tuning, parameter-efficient fine-tuning strategies aim to achieve the adaptation of pretrained models with a minimum of parameter updates and computational resources. Prevailing methods for parameter-efficient tuning include:

Adapter Tuning [26]: Adapter-based approaches propose the insertion of compact learnable modules, referred to as adapters. This tuning can be inserted within every layer or a subset of layers in the model. Owing to lower dimensionality, adapters enable fine-tuning that exclusively targets these newly added parameters while preserving the original model weights, which ensures the feasibility of fine-tuning and maintains computational efficiency.

Low-Rank Adaptation (LoRA) [27]: The core idea behind LoRA is to perform updates through low-rank approximations of the model's original weight matrices. This necessitates learning only two smaller matrices that effectively modify the entire model behavior, thereby significantly reducing the number of required parameters while maintaining performance comparable to full fine-tuning.

Quantized Low-Rank Adaptation (QLoRA) [28]: Building upon LoRA [27], QLoRA integrates quantization. It introduces quantization operations to further compress the space of trainable parameters. QLoRA is particularly advantageous in scenarios where deep learning models are deployed and updated in resource-constrained environments, such as edge devices, offering substantial benefits in terms of efficiency under strict limitations.

2.5. LLM Prompt-Engineering

Next, we summarize the techniques in prompt engineering, which treat the prompt as a learnable parameter without updating the parameters of pretrained models. By optimizing only a minuscule number of parameters, prompt engineering can improve the performance of pretrained models in various downstream tasks while approaching the efficacy of full fine-tuning.

Prefix Tuning [29]: Incorporating concepts from prompting and in-context learning [4,30], prefix tuning optimizes a set of continuous task-specific vectors, denoted as "prefixes". For decoder-only architectures, distinct prefix vectors are prepended to the Key (K) and Value (V) matrices at each layer of the decoder's attention mechanism. Conversely, for encoder–decoder structures, separate prefix vectors are appended to the Query (Q) and Key (K) matrices preceding each layer of the attention mechanism in both the encoder and decoder parts. Unlike full fine-tuning, prefix tuning freezes all parameters of the pretrained language model and focuses solely on optimizing the small subset of prefixed parameters that have been introduced.

P-Tuning [31]: P-tuning maps discrete prompts into trainable continuous prompt embeddings, it utilizes LSTM and MLP to construct a prompt encoder. Distinct from prefix tuning [29], P-tuning inserts prompt tokens at arbitrary positions within the input layer; these are consecutively transformed into hidden states via the prompt encoder and jointly trained alongside the input embeddings. Nonetheless, in order to realize enhanced performance outcomes it is imperative to undertake the optimization of prompt embedding in concert with the overall model tuning process.

Prompt Tuning [32]: Prompt tuning can be regarded as a simplified version of P-tuning [31] and Prefix Tuning [29], as illustrated in Figure 1. Prompt tuning concatenates a sequence of prompts with the input sequence to form the model's input. The embedded prompts (P_e) and input embeddings (X_e) collectively constitute a parameter matrix $[P_e : X_e]$ which is processed by the model, wherein only the parameters of the prompts P_e are updated; the original weights of the pretrained model remain unaltered. This method exhibits heightened parameter efficiency, and is becoming increasingly competitive as model parameter counts grow. It is able to attain performance comparable to full fine-tuning even when model parameters exceed billions. Through prompt tuning, a single pretrained model can be efficiently repurposed for multiple downstream tasks merely by training different prompt parameters for each respective task.

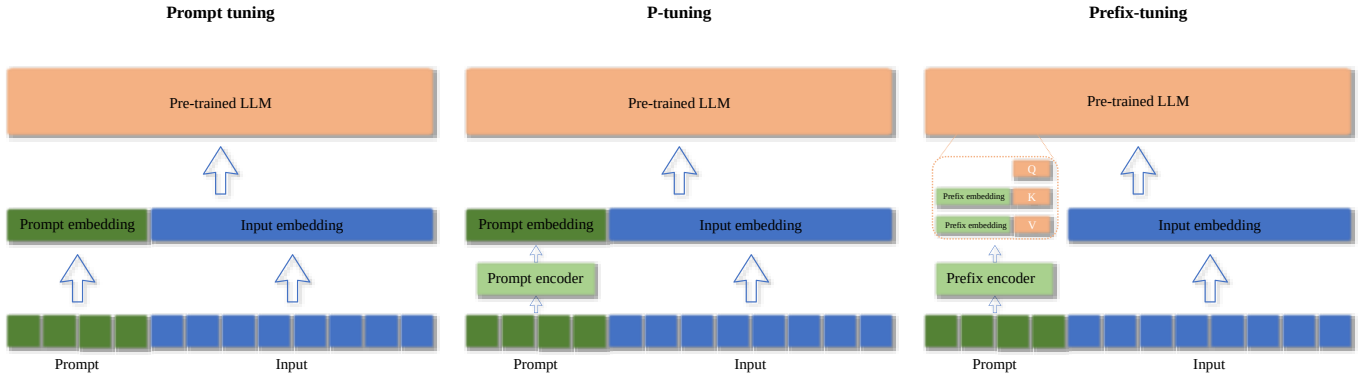


Figure 1. Schematic representation of the technical architectures for prompt tuning, P-tuning, and prefix tuning.

In-Context Learning: The few-shot performance demonstrated by GPT-3 [4] highlights the potential of in-context learning (ICL), enabling language models to master downstream tasks without supplementary model tuning by relying solely on a handful of exemplary demonstrations. The survey by [33] decomposes in-context learning into two distinct stages, namely, training and inference. During the training stage, the model acquires the capability for in-context learning from the pretraining objectives. At the inference

stage, the manifestation of the model's in-context learning prowess is showcased through the strategic design of examples and the selection of appropriate evaluation mechanisms.

Chain-of-Thought: The essence of the chain-of-thought (CoT) concept [34] lies in mimicking the human cognitive process for tackling complex issues by presenting the model with a modest collection of exemplars. Solutions are broken down into a progression of intermediate reasoning steps articulated in natural language while clearly delineating the logical trajectory from inquiry to resolution. Within the GSM8K [35] mathematics problem benchmark, the PaLM 540B model utilizing chain-of-thought prompting achieved a substantial leap beyond fine-tuned GPT-3 and other previous best performers. The pivotal merit of chain-of-thought prompting, juxtaposed with standard prompting and fine-tuning, is its dual nature; it obviates the reliance on extra data while concurrently augmenting the model's aptitude for mathematical logical reasoning.

2.6. Representative Pretrained LLMs

Herein, we enumerate ten influential pretrained large language models (LLMs) in the field of natural language processing (NLP) in the order of their respective model or paper publication timelines (shown in Figure 2). These models serve as the foundational backbone for the evolution of future large multimodal models (LMMs).

Transformer (2017): Transformer [21], proposed by a Google team in 2017, represents an advancement built around an encoder–decoder framework. It is particularly notable for its introduction of the attention mechanism, the mathematical formulation of which is detailed as Equation (1).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

Moreover, the incorporation of a multi-head attention mechanism into the transformer architecture allows the resulting models to simultaneously employ several independent self-attention heads, thereby facilitating the learning of a multitude of diverse attentional subspaces in parallel. A thorough explanation of these mechanisms can be found in [21].

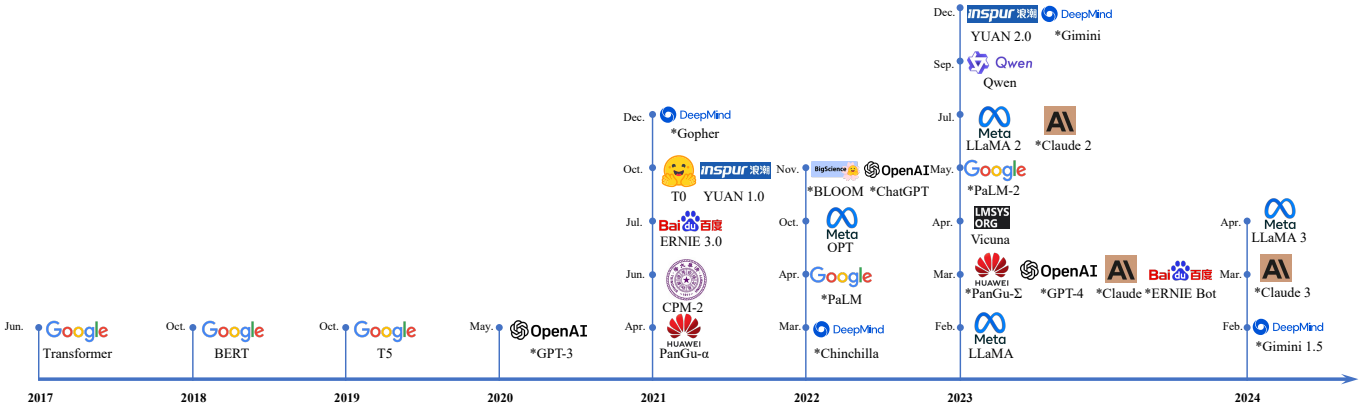


Figure 2. Timeline of pretrained and fine-tuned LLM releases. The sign * on the left side of the model name denotes that they remain closed-source.

T5 (2019): The T5 [24] model architecture adopts an encoder–decoder framework, incorporating multiple layers of transformer modules with a maximum of 11 billion parameters. T5 is characterized as a general model that can be fine-tuned across a diverse array of downstream NLP tasks through the use of a unified "text-to-text" transfer learning paradigm.

As depicted in Figure 3, the T5 model refines BERT-like pretraining strategies at a granular level. Empirical findings indicate that employing (Replace Corrupt Spans) as

an unsupervised pretraining objective yields optimal outcomes. Furthermore, the paper experimentally examines the performance of various model variants based on different self-attention mechanisms, ultimately concluding that the encoder–decoder architecture outperforms standalone language model and prefix LMs in text-to-text tasks. A comprehensive understanding of the model's specifics can be found in [24].

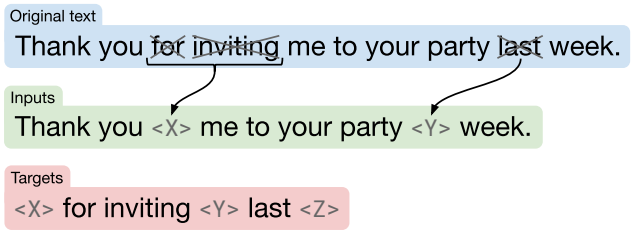
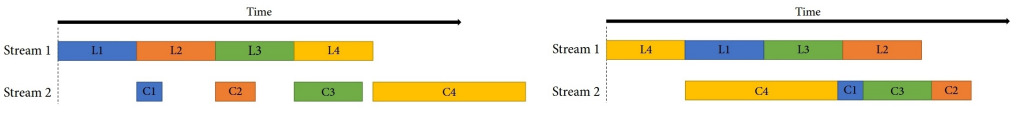


Figure 3. Pretraining objective diagram of T5 [24] using Replace Corrupt Spans. Replace Corrupt Spans means that each consecutive span of corrupted tokens is replaced by a unique sentinel token, e.g., <X>, <Y>, and <Z>.

GPT-3 (2020): GPT-3 [4] adopts the transformer decoder architecture, aligning with the overall framework of GPT-2 [3] while introducing alterations in terms of both parameter scale and training strategies. GPT-3 encompasses eight distinct model sizes, ranging from the smallest variant, GPT-3 Small, with 125 million parameters, to the largest configuration, GPT-3, with 175 billion parameters. The modifications in the training regimen specifically manifest through considerable augmentations in the hidden dimension (d_{model}) and batch size, coupled with the implementation of a relatively lower learning rate. In response to limitations observed in GPT-2's zero-shot performance for certain tasks, GPT-3 innovatively conceptualizes in-context learning, demonstrating empirically that few-shot learning can lead to improved performance without altering the model's parameter count.

CPM-2 (2021): CPM-2 [36], in contrast to its predecessor CPM [37] which was based on a decoder-only architecture, adopts the encoder–decoder framework of the conventional transformer model [21]. CPM-2 consists of two distinct versions: the standard CPM-2 variant possesses 11 billion parameters, while its Mixture-of-Experts (MoE) counterpart boasts an unprecedented scale with 198 billion parameters. A central innovation of CPM-2 lies in its cost-effective design; it employs knowledge inheritance [38] strategies to harness existing knowledge from pretrained language models to facilitate the training process of CPM-2. Furthermore, it utilizes prompt tuning [32], a method that requires updating only 0.01% of the model's parameters compared to fine-tuning that is still able to achieve performance closely comparable to that of full fine-tuning. Additionally, the model introduces an innovative memory-efficient Inference Framework for MoE Layers (INFMoE). Shown in Figure 4, it is tailored for efficient inference on TensorRT-based platforms specifically designed for the MoE variant. Moreover, the research delves into the working mechanisms of prompt tokens, with a detailed examination offered in [36].



(a) In naïve order: C2, C3, C4 must wait for loading. (b) INFMOE scheduling: all computation run without gaps.

Figure 4. INFMoE framework compared with others; L represents parameter loading, while C represents computational consumption. More details can be found in [36].

PaLM (2022): PaLM [6] is a decoder-only transformer [21] language model with variant parameter scales, including models of 8 billion, 62 billion, and 540 billion parameters. The model is pretrained on an extensive dataset amounting to 780 billion tokens, wherein half

of the data consist of social media conversations, 27% originate from webpage content, and the remaining portion encompasses books and code, among other sources.

PaLM's innovation lies in its utilization of the Pathways system [39] for large-scale pretraining, which facilitates collaborative computation across multiple pods within a TPU v4 cluster, enabling synchronous updates to the model parameters to realize highly efficient parallel processing and significantly reduce time costs. In terms of evaluation, experiments conducted on BIG-bench have demonstrated that the version of PaLM with 540 billion parameters surpasses state-of-the-art models such as GPT-3 [4], Gopher [40], and Chinchilla [41] in few-shot learning capabilities. Moreover, this model exceeds the average human performance on a majority of tasks under assessment.

OPT (2022): The OPT [42] model has undergone a full-scale replication of GPT-3 175B [4], employing the same decoder-only architecture and maintaining the same maximum parameter count of 175 billion. Importantly, the team has released the OPT model weights, code, and training logbook to the open-source community in their entirety. The pretraining corpus utilized by OPT consists entirely of publicly accessible data, amounting to approximately 180B tokens. It integrates portions of the RoBERTa [22], The Pile [43], and PushShift.io Reddit [44] corpora, undergoing data cleaning processes that include filtering non-English data and deduplication.

Comparisons between OPT and GPT-3 were carried out across sixteen datasets, encompassing zero-shot, multi-shot, and dialogue experiments. The results show that OPT's performance is broadly comparable to that of GPT-3. Additionally, the team has analyzed the limitations of OPT, such as its suboptimal response to declarative instructions and its potential for generating harmful or discriminatory content.

LLaMA (2023): LLaMA [8] adopts a decoder-only transformer [21] architecture and comprises four distinct parameter configurations, ranging from the smallest (7B) to the largest (65B). The LLaMA model structure has three principal modifications: the incorporation of pre-layer RMSNorm [45], the employment of Rotary Positional Embeddings (RoPE) [46], and the substitution of SwiGLU units for ReLU activations. Moreover, inspired by prior work, LLaMA implements techniques to reduce computational overhead, such as refraining from storing attention weights and bypassing the computation of masked key/query scores. Regarding the pretraining corpus, LLaMA relies entirely on openly accessible and publicly available datasets, with 67% of the data coming from English CommonCrawl and 15% from C4, along with additional contributions from GitHub and Wikipedia, among others.

Benchmark evaluations have shown that LLaMA-13B outperforms GPT-3 [4] across a wide range of benchmarks despite having only a tenth of GPT-3's parameter count. Meanwhile, LLaMA-65B is able to contend with the state-of-the-art models such as Chinchilla [41] and PaLM 540B [6], demonstrating competitive results on various metrics. LLaMA's characteristic efficiency in its utilization of parameters has rendered it a popular choice as a large language model (LLM) backbone for numerous downstream tasks.

PanGu-Σ (2023): Pangu-Σ [11] represents the inaugural trillion-parameter sparse language model, inheriting the decoder-only architecture and innate parameters of its predecessor, PanGu-α [10], with a peak parameter count reaching 1.085 trillion. It introduces an input mechanism that employs distinct embeddings for distinct domains, which is coupled with a two-level routing design in the Random Routed Experts (RRE) framework. The pretraining corpus of Pangu-Σ, totalling 329 billion tokens, primarily encompasses diverse data formats of bilingual Chines–English, content from [24,47,48] and code from [49,50].

During pretraining, Pangu-Σ employs an Expert Computing and Storage Separation (ECSS) strategy, harnessing heterogeneous computing to enhance training throughput by a factor of 6.3 compared to PanGu-α. In experimental evaluations, Pangu-Σ demonstrates superiority over previous state-of-the-art models across sixteen downstream NLP tasks in the Chinese domain, encompassing dialogue, question answering, and machine translation, among others. However, due to variations in the pretraining datasets employed, the

paper does not present comparative results against GPT-3 [4] or other contemporary SOTA large-scale models.

PaLM-2 (2023): PaLM-2 [7] is based on the transformer [21] architecture, altohugh the specific architectural details are not mentioned in the technical report. While the report does not reveal the exact parameter count for the PaLM-2 family, it states that the models in this series have a smaller number of parameters compared to PaLM 540B [6]. Despite this reduced parameter count, PaLM-2 demonstrates improved reasoning abilities across various tasks. The report highlights the significance of scaling laws from recent work [41]. In addition, through empirical validations, it underscores the criticality of considering data scale and model size as co-determinants of equal weight in driving performance improvements [41].

A distinguishing innovation in PaLM-2 lies in breaking away from the conventional approach used for pretraining large language models, which typically uses a single objective such as causal or masked language modeling. Building upon the achievements of UL2 [51], PaLM-2 adopts a hybrid of multiple pretraining objectives. Additionally, the pretraining dataset employed for PaLM-2 has been expanded to cover several hundred languages, thereby broadening its multilingual capacity.

LLaMA 2 (2023): The architecture of LLaMA 2 [9] adheres to that of its predecessor LLaMA [8], adopting a decoder-only paradigm with a maximum parameter count scaling up to 70 billion. Notably, LLaMA 2 introduces certain enhancements over LLaMA, including doubling the context window size from its original 2048 tokens to an extended 4096 tokens and incorporating the utilization of grouped-query attention(GQA) [52] (shown in Figure 5), which economizes cache space required for Key–Value (KV) caching while effectively sustaining model accurac.

In terms of the pretraining corpus, the LLaMA 2 ensemble of models is trained on a newly constructed publicly accessible hybrid online dataset encompassing 2 trillion tokens, thereby augmenting the training volume by 42% compared to the original LLaMA. Despite demonstrating superior performance across multiple benchmark evaluations relative to all open-source models, LLaMA 2 exhibits a discernible performance gap when juxtaposed against closed-source contemporaries such as GPT-4 [5] and PaLM-2-L [7].

Table 1. Summary of pretrained LLMs, including architectures, parameter counts, pretraining corpus sizes, objectives, providers, and notable derived fine-tuned LMs

Pretrained LMs	Architectures	Parameters	Pretraining Corpus Size	Pretraining Objectives	Model Providers	Fine-Tuned LMs
T5	encoder-decoder	11 B	1 T	MLM	Google	T0 [53]
GPT-3	decoder-only	175 B	300 B	ALM	OpenAI	ChatGPT [54] WebGPT [55]
CPM-2	encoder-decoder	198 B	2.6 T	MLM	Tsinghua	-
PaLM	decoder-only	540 B	780 B	ALM	Google	-
PaLM-2	-	<540 B	>780 B	Mixture	Google	-
LLaMA	decoder-only	65 B	1.4 T	ALM	Meta	Vicuna [56] Alpaca [57] LIMA [58]
LLaMA 2	decoder-only	70 B	2 T	ALM	Meta	LLaMA 2-CHAT [9]
OPT	decoder-only	175 B	180 B	ALM	Meta	OPT-IML [59]
PanGu-Σ	decoder-only	1.085 T	329 B	ALM	Huawei	-

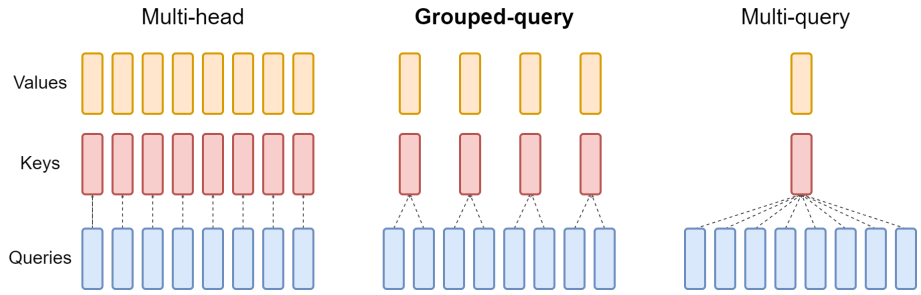


Figure 5. Comparison of attention mechanism architectures in [52]: multi-head, grouped-query, and multi-query methods. Grouped-query partitions multiple query headers into groups; each group shares the same keys and values, striking a balance between performance and computational cost.

3. Large Multimodal Models

Large Multimodal Models (LMMs) have emerged as a hot research topic following the rise of LLMs, utilizing LLMs as a central hub to handle multimodal tasks by extending from their single textual modality to encompass modalities such as images, audio, and video. In this section, we consolidate the latest research findings on LMMs, providing a comprehensive introduction to their architectural components, training strategies, instruction fine-tuning, and prompt engineering. Finally, with a particular focus on the vision–language domain, we present a taxonomy and analysis of 66 cutting-edge vision–language LMMs in Section 3.5.

3.1. LMM Architectures

This section provides an overview of five constituent components commonly found in different types of large multimodal models (LMMs), arranged in the order of the input processing flow: Multimodal Encoder→Input Modal Aligner→Pretrained LLM backbone→Output Modal Aligner→Multimodal Decoder.

Notably, not all LMMs incorporate five constituents uniformly, depending on the specific functional focus of the model; for instance, comprehension-only LMMs that exclusively generate text, such as BLIP-2 [60] and MiniGPT-4 [61], typically encompass only the first three elements, while generative-only LMMs in image or audio, exemplified by DiffusionGPT [62], encompass the latter three components and General LMMs such as Kosmos-G [63], which integrate functionalities across modalities, encompass the entire five components.

3.1.1. Multimodal Encoders

Multimodal encoders are designed to extract features from multiple input modalities, such as text, images, videos, and audio. In the context of LMMs, pretrained encoders are typically employed with their parameters frozen in order to utilize their feature extraction capabilities. Below, we present some representative encoders for different modalities.

Image Encoders: Image Encoders offer a diverse range of options, with commonly used variants including MAE [64], ViT [65], Swin Transformer [66], CLIP ViT [12], Eva-CLIP ViT [67], OpenCLIP [68], and DINOv2 [69].

Video Encoders: LMMs commonly repurpose image encoders as video encoders, applying downsampling preprocessing to retain a subset of representative frames. Subsequently, the processing follows the same workflow as for static images. This approach preserves the core visual information of the video while reducing computational demands.

Audio Encoders: Commonly employed audio encoders include CLAP [70], C-Former [71], Whisper [72], and HuBERT [73].

General Encoders: General encoders process data from various modalities, harmoniously mapping them onto a shared feature space. Prominent examples of such encoders currently include Meta-Transformer [74], ImageBind [75], and LanguageBind [76]. Among these methods, LanguageBind is particularly noteworthy for its capacity to uniformly

encode data from at least five distinct modalities: text, image/video, audio, depth, and infrared. This operation can be mathematically formalized as Equation (2):

$$\mathbf{z}_m = f_m(\mathbf{x}_m; \theta_m), \quad m \in \{\text{Text, Image, Audio, Depth, Infrared}\} \tag{2}$$

where $\mathbf{x}_m$ denotes the input of modality m, θ_m represents the parameters of the corresponding encoder f_m, and $\mathbf{z}_m$ signifies the extracted features of the input. To visually contrast these frameworks, Figure 6 compares LanguageBind with ImageBind, illustrating how LanguageBind bypasses image bridging to directly map all modalities to a linguistic domain.

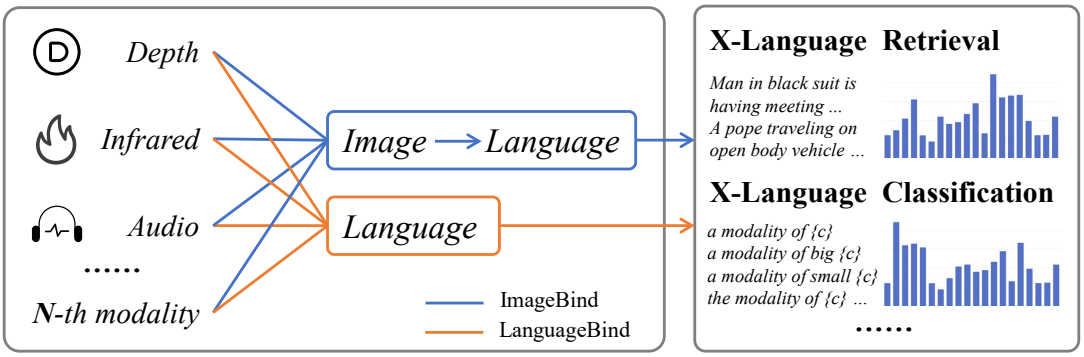


Figure 6. Comparison of LanguageBind and ImageBind. ImageBind relies on images as intermediaries, while LanguageBind directly maps all modalities to the linguistic domain, where "X" represents non-language modalities and "c" represents category. Source: [76].

3.1.2. Input Modal Aligner

The input modal aligner is dedicated to aligning feature vectors emanating from diverse modalities, as extracted by the multimodal encoder, to a text feature space and then transforming them into LLM-compatible feature representations. Notable input modal aligners employed in this context encompass Linear Mapper, Multilayer Perceptron (MLP), Querying Transformer (Q-Former) [60], Prompt-Transformer (P-Former) [77], Multiscale Querying Transformer (MQ-Former) [78], and Cross-Attention Layer.

3.1.3. Upstream LLM Backbone

LMMs use pretrained or fine-tuned LLMs as upstream LLM backbones. This constitutes the core of the LMM architecture. The upstream LLM backbone receives aligned multimodal inputs, utilizing its understanding, reasoning, and generation capabilities in textual feature space to yield text outputs or instruction tokens. These instruction tokens are responsive to user-provided prompts, and serve to steer other components towards the execution of more intricate cross-modal tasks. A selection of commonly employed upstream LLM backbones is listed in Table 1.

3.1.4. Output Modal Aligner

In contrast to the input modal aligner, the output modal aligner maps the instruction tokens produced by the upstream LLM backbone into feature representations compatible with decoding into the target non-linguistic modalities. Commonly utilized output modal alignes include Linear Aligner, Multilayer Perceptron (MLP), and encoder–decoder transformers [79,80].

3.1.5. Multimodal Decoder

A multimodal decoder refers to a component that decodes features that have been postprocessed by the output modal aligner. Its aim is to generate outputs in various target modalities. Distinct decoder options are employed for different multimodal content types: in image synthesis contexts, the Stable Diffusion series [13,14] and CM3 [81] are applied;

in video generation contexts, the Zeroscope series is employed; and for audio synthesis contexts, the AudioLDM [82,83] series is deployed.

3.2. LMM Training

The training of LMMs excludes pretrained components, including the multimodal encoder, upstream LLM backbone, and multimodal decoder, while separately training the input modal aligner and output modal aligner. The training objectives of LMMs are twofold: (1) to align multimodal inputs into a text feature space via the input modal aligner, and (2) to ensure the quality of multimodal outputs by training the output modal aligner to map instruction tokens back into generator-understandable features.

3.3. LMM Instruction Tuning

Instruction tuning, initially introduced in FLAN [84], is a training technique that involves refining a pretrained LLM on small specialized datasets comprising instructions formatted in specific ways. This technique aims to achieve the efficacy of fine-tuning and prompting [4] with a reduced dataset size and fewer parameter updates, thereby enhancing the model's capability to comprehend human instructions and improving its zero-shot performance.

The success of instruction tuning on many single-modal LLMs (e.g., GPT-3 [4], InstructGPT [85], T0 [53]) can also be extended to the multimodal domain. Figure 7 shows a comparison of instruction tuning between LMM and the original LLM. Currently, multimodal instruction tuning datasets primarily take three forms of construction: dataset adaptation, self-instruction synthesis, and a combination of both.

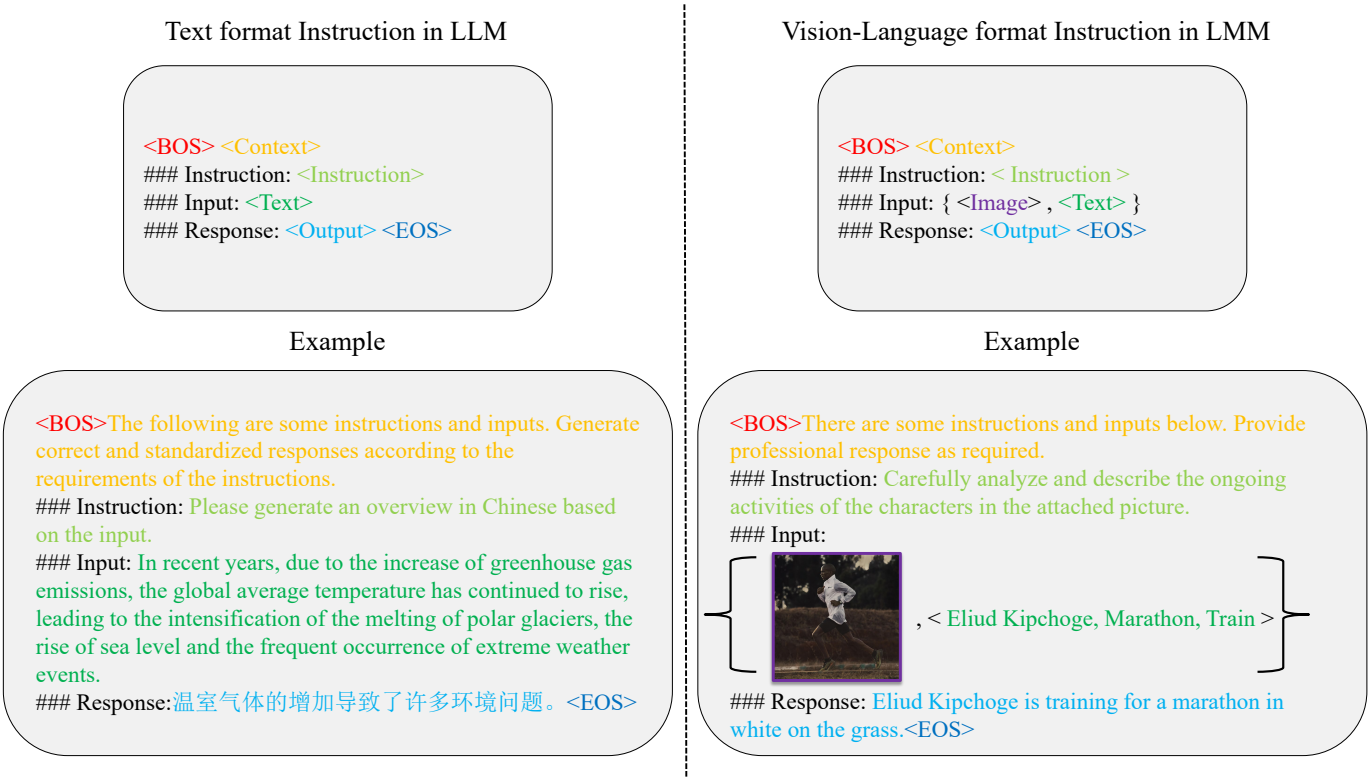


Figure 7. Comparison of instruction tuning data template between LLMs and LMMs. Here, <BOS> and <EOS> signs are tokens indicating the beginning and end of data input, respectively.

Dataset Adaptation: Dataset adaptation involves re-targeting existing large-scale annotated datasets while adjusting their format at low cost and high speed to create suitable instruction data. This approach has been adopted by models such as MiniGPT-4 [61] and

InstructBLIP. However, it is limited by its reliance on human intervention and the lack of novelty in directly adopting or superficially modifying original annotations, making it potentially insufficient for broad generalization to new scenarios.

Self-Instruction Synthesis: Self-instruction synthesis leverages the comprehension and generative capacities of LLMs, using a small set of manually annotated template samples to guide large language models such as GPT-4 [5] in restructuring existing annotated datasets for instruction data creation, adopted by models including Shikra [86], VideoChat [87], and LLaVAR [88]. It has a flexible data generation mechanism, ensuring diversity in instruction data and generalizability to real-world contexts. However, its dependence on LLMs introduces inherent hallucination issues and biases associated with these models.

3.4. LMM Prompt Engineering

In LLMs, prompt engineering [29,31–34,89] refrains from updating the model parameters and instead seeks to augment its functionality by employing prompt tokens optimizing the performance of the LLM. Within the realm of LMMs, prompt engineering includes the inheritance of in-context learning [33] and chain-of-thought (CoT) [34] derived from LLMs, evolving into the multimodal in-context learning and multimodal chain-of-thought paradigms.

Multimodal In-Context Learning LLMs leverage in-context learning (ICL) techniques, enabling them to achieve few-shot performance comparable to fine-tuning during the inference stage, relying on only a small number of demonstrations for downstream tasks [4]. In LMMs, as shown in Figure 8, in-context learning can enhance performance when extended to multimodal in-context learning through the provision of demonstrations across modalities. Typically, there are two distinct application scenarios of multimodal ICL within LMM; the first involves using a small set of demonstrations to guide an LMM in modal reasoning tasks, while the second entails teaching the LMM how to utilize external foundational tools to solve tasks through a series of exemplary steps, as presented in [18].

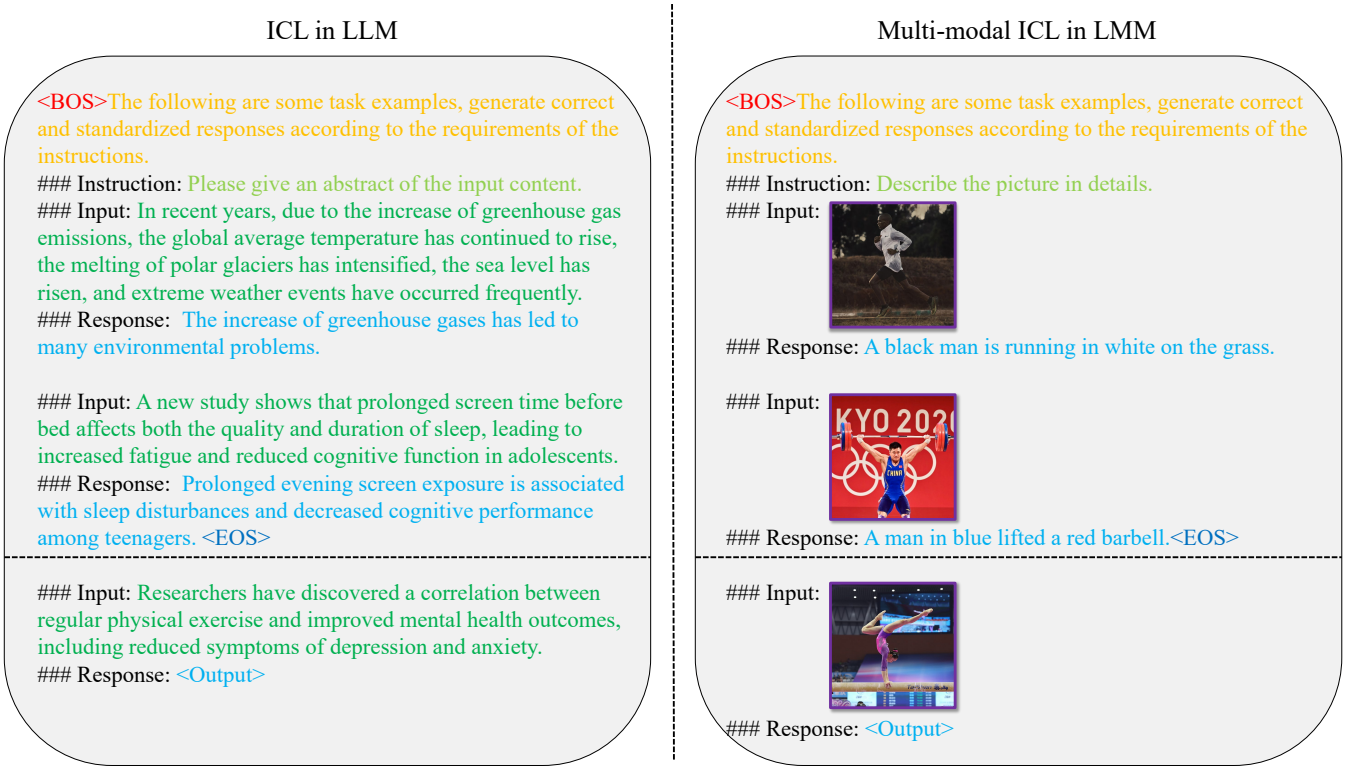


Figure 8. Comparison of in-context learning in LLMs and LMMs. Here, <BOS> and <EOS> signs are tokens indicating the beginning and end of the input demonstrations.

Multimodal Chain-of-Thought The concept of chain-of-thought (CoT) originally emerged as a technique in LLM to tackle complex reasoning tasks [34]. It involves guiding the LLM to break a complex problem into a series of subproblems and solve them iteratively, thereby significantly enhancing the performance of LLM [4,5]. CoR led to an expansion of the prompt paradigm from <Input, Output> to <Input, CoT Rationale, Output> [34].

Several works have advanced single-modal CoT to multimodal CoT (M-CoT), which can be categorized under different prompt paradigms such as zero-shot M-CoT [90], few-shot M-CoT [91], and fine-tuning M-CoT [92,93]. In zero-shot M-CoT, no examples are used to guide the LMM through the CoT process; instead, the model's reasoning ability is triggered during the inference phase with a simple textual instruction such as "*Do it step by step.*". Few-shot M-CoT resembles in-context learning, where a small number of exemplars are provided that detail the intermediate steps of reasoning. On the other hand, fine-tuning M-CoT necessitates the use of a specific dataset for fine-tuning the LMM's reasoning capabilities. Typically, the former two approaches are more applicable to larger LLMs, while the latter is often employed for smaller ones.

Regarding modal alignment, M-CoT can further be divided into translation mapping and learnable mapping. In translation mapping, non-text modal inputs are straightforwardly converted into text descriptions, effectively transmitting non-text modalities to the text modality via 'translation' before engaging in CoT reasoning (although this process inherently entails considerable loss of modality-specific information). By contrast, learnable mapping involves constructing a trainable model that integrates features from other modalities into the textual feature space, forming a joint embedding that serves as input to the LLM for CoT reasoning.

For instance, Multimodal-CoT [92] has been utilized with fine-tuning and learnable mapping to elicit CoT capabilities in T5-770M. By devising a two-stage learnable reasoning framework, T5-770M surpassed GPT-3.5 on the ScienceQA benchmark [91] while concurrently mitigating the occurrence of hallucination-induced errors inherent in LLMs.

3.5. Taxonomy of Vision–Language LMMs

Vision–language LMMs represent the most typical and fully developed branch within the current research on LMMs; hence, in this section, we focus on the latest advancements in vision–language LMMs by undertaking a taxonomy of 66 models based on their distinctive multimodal inputs and outputs. Further, aligning with the structural framework of LMMs outlined in Section 3.1, we present a detailed breakdown of each model's components and list them systematically in Tables 2–5.

3.5.1. Image+Text to Text

The "Image+Text to Text" model, also referred to as the image understanding model, constitutes both the starting point and a focal point of extensive research interest within the LMM community. Notably, it has the highest prevalence in terms of model quantity. In this section, we have selected 28 representative image understanding models as a basis for our study, as outlined in Table 2. A summary of these models' frameworks is depicted in Figure 9. Figure 10 shows an example of dialogue in an image understanding model.

Upon statistical analysis of the components employed in image understanding models from Table 2, it emerges that CLIP-ViT is currently the most widely adopted visual encoder, followed closely by Eva-CLIP-ViT. These encoders demonstrably outperform alternative types of visual encoders. This superiority may stem from the contrastive learning paradigm employed by CLIP, coupled with its 400 million high-quality image–text pairs, which together endow it with a robust capability for visual feature extraction.

Table 2. Display of architecture components across different image understanding models, ordered by initial of model name.

Model	Vision Encoder	Input Modal Aligner	Upstream LLM Backbone
BLIP-2 [60]	CLIP ViT	Q-Former + Linear Mapper	Flan-T5/OPT
BLIVA [94]	CLIP ViT	Q-Former + Linear Mapper	Vicuna-7B/Flan-T5XXL
ChatSpot [95]	CLIP ViT	Linear Mapper	Vicuna-7B/LLaMA
CogVLM [96]	Eva-2-CLIP ViT	MLP	Vicuna-v1.5-7B
DRESS [97]	Eva-CLIP ViT	Linear Mapper	Vicuna-v1.5-13B
DLP [77]	CLIP ViT	Q-Former + P-Former + Linear Mapper	OPT/Flan-T5
IDEFICS [98]	OpenCLIP	Cross Attention Layer	LLaMA
InternLM-XComposer [99]	Eva-CLIP ViT	Cross Attention Layer	Intern-LM
InternLM-XComposer2 [100]	CLIP ViT	Cross Attention Layer	Intern-LM2
Lyrics [78]	CLIP ViT	MQ-Former + Linear Mapper	Vicuna-13B
LLaVA [101]	CLIP ViT	Linear Mapper	Vicuna-7B/13B
LLaVA-1.5 [102]	CLIP ViT	MLP	Vicuna-v1.5-7B/13B
LLaVAR [88]	CLIP ViT	Linear Mapper	Vicuna-13B
MiniGPT-v2 [103]	Eva-CLIP ViT	Linear Mapper	LLaMA-2-Chat-7B
MiniGPT-4 [61]	Eva-CLIP ViT	Q-Former + Linear Mapper	Vicuna-13B
mPLUG-Owl [104]	CLIP ViT	Cross Attention Layer	LLaMA-7B
mPLUG-Owl2 [105]	ViT	Modality-Adaptive Module	LLaMA-2-7B
mPLUG-DocOwl [106]	CLIP ViT	Cross Attention Layer	LLaMA-7B
MobileVLM [107]	CLIP ViT	Lightweight Downsample Projector (LDP)	MobileLLaMA
MobileVLM V2 [108]	CLIP ViT	Lightweight Downsample Projector v2 (LDPv2)	MobileLLaMA
Otter [109]	CLIP ViT	Cross Attention Layer	LLaMA-7B
Osprey [110]	ConvNeXt-Large	MLP	Vicuna
PandaGPT [111]	ImageBind	Linear Mapper	Vicuna-13B
PaLI-X [112]	ViT	Linear Mapper	UL2-32B
Qwen-VL [113]	ViT	Cross Attention Layer	Qwen-7B
RLHF-V [114]	BEiT-3	Linear Mapper	Vicuna-v1-13B
Silkie [115]	ViT	Cross Attention Layer	Qwen-7B
VILA [116]	ViT	Linear Mapper	LLaMA-2-7B/13B

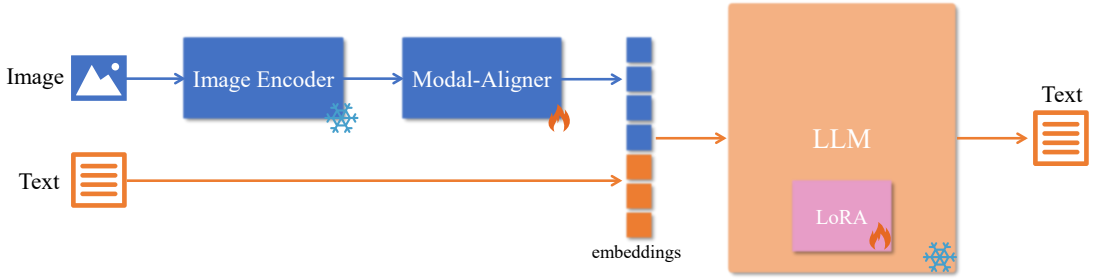


Figure 9. Common architecture of "Image+Text to Text" models.

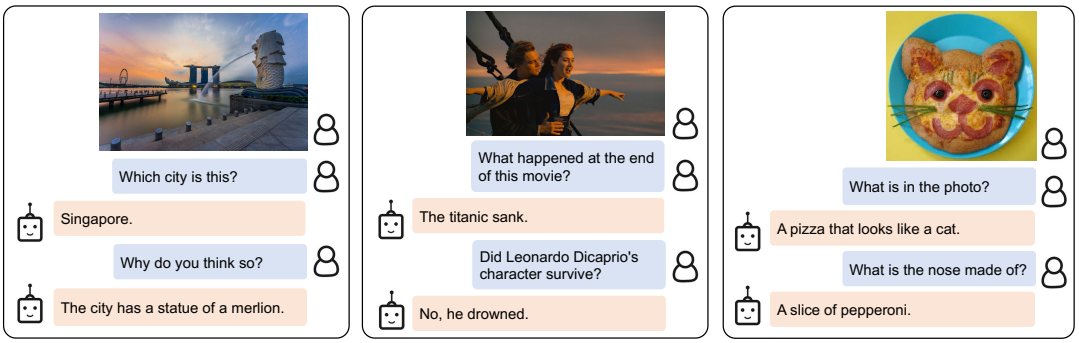


Figure 10. Examples from [60] of how BLIP-2 understands images and communicates with users.

Additionally, the linear mapper and cross-attention layer are frequently employed as modal aligners, possibly due to their structural simplicity and performance efficiency when compared to aligners such as P-former. Another approach involves employing custom modal aligner designs. For instance, mPLUGOwl2 [105] introduced a modality-adaptive module designed to accurately compute and compare the similarity between different modalities within a shared semantic space. In parallel, the MobileVLM series [107,108] uses a lightweight downsample projector (LDP), which, unlike Q-former, maintains the spatial location information of visual features while being architecturally lightweight.

Regarding upstream LLM preferences, Vicuna [56] has emerged as the mainstream selection, surpassing the LLaMA series [8,9]. Vicuna, as a derivative model fine-tuned atop the supervised data from ShareGPT.com, exhibits superior performance compared to LLaMA. This phenomenon elucidates a technical predilection within the field of image understanding models, wherein there is an inclination towards LLMs that combine light weight, low training cost, and high-quality dialogue.

3.5.2. Video+Text to Text

The quantity of "Video+Text to Text" models for video understanding is relatively low compared to image understanding models. In Table 3, we have selected nine representative models as references. Due to the inherent unity in processing both videos and images, these models also inherently possess the ability to understand images. Figure 11 presents an overview of the architectural frameworks employed by these models, while Figure 12 shows an example of dialogue in video understanding models.

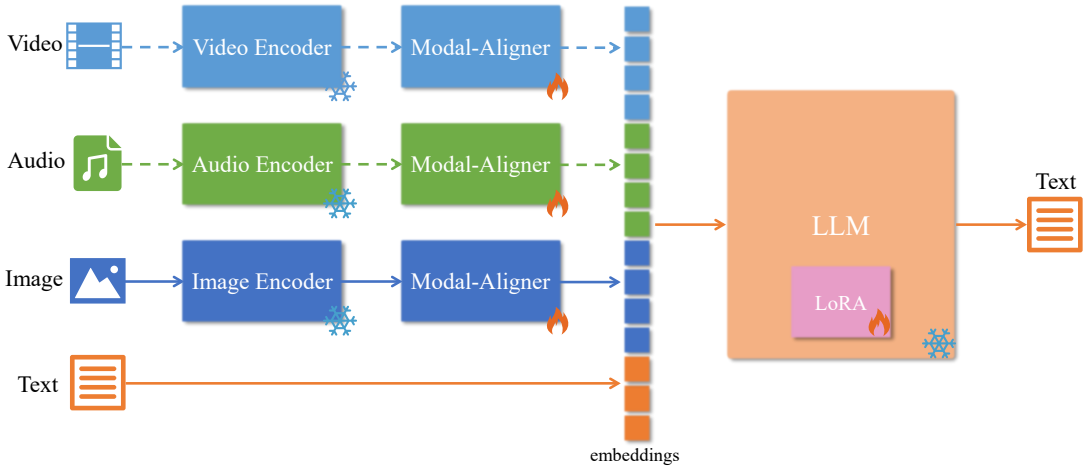


Figure 11. Common architecture of "Video+Text to Text" models.

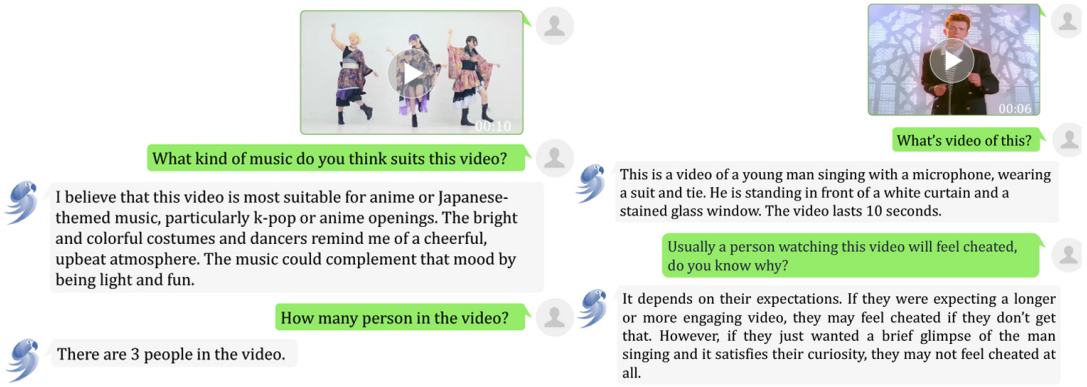


Figure 12. Example showcasing how a model understands and replies to video, taken from the VideoChat paper [87].

Table 3. Display of architectural components across different video understanding models.

Model	Vision Encoder	Input Modal Aligner	Upstream LLM Backbone
Dolphins [117]	CLIP ViT	Cross Attention Layer	LLaMA/MPT
InstructBLIP	ViT	Q-Former + Linear Mapper	Vicuna/Flan-T5
InternVL [118]	InternViT-6B	Cross-attention + QLLaMA + MLP	Vicuna-13B
Lynx [119]	Eva-CLIP ViT	Cross Attention Layer	Vicuna-7B
mPLUG-video [120]	TimeSformer [121]	Cross Attention Layer + Linear Mapper	Chinese GPT-3 1.7B [122]/2.3B [123]
VAST [124]	ViT	Cross Attention Layer	Vicuna-13B
VideoChat [87]	ViT	Q-Former + Linear Mapper	Vicuna
Video-ChatGPT [125]	CLIP ViT	Linear Mapper	Vicuna-v1.1
Video-LLaMA [126]	Eva-CLIP ViT	Q-Former + Linear Mapper	Vicuna/LLaMA

In the context of image generation models, CLIP ViT often serves as a prevalent choice for the vision encoder component, models from the Stable Diffusion series [13,14] are predominantly employed as the vision generator. On the other hand, the selection of upstream large language model (LLM) backbones exhibits a higher degree of diversity, ranging from Vicuna to LLaMA and even extending to web-based ChatGPT [54]. These models feature more varied composition and implementation methods. For example, in Visual-ChatGPT [90] and DiffusionGPT [62], the model structure is not an end-to-end design, instead integrating ChatGPT in order to refine the prompt optimization.

3.5.3. Image+Text to Text+Image

Models in the "Image+Text to Text+Image" category should not be considered as mere image generation models; instead, they are bifurcated into two principal classes, namely, image editing models and generative models. Figure 13 presents an overview workflow of "Image+Text to Text+Image" models. As depicted in Table 4, among the 23 selected models, models such as CogCoM [127], DetGPT [128], Shikra [86], and SPHINX-X [129] (which does not contain a vision generator component) consequently do not possess the inherent ability to synthesize images. However, they are endowed with the ability to perform a variety of operations on input images, encompassing extraction, annotation, and segmentation, leading to their classification as image processing models. For instance, CogCoM is capable of executing cropping and zoom-in operations to acquire detailed local visual content, and can identify textual information within images through OCR while performing reasoning based on visual input. Although image editing does not equate to image generation, it does manifest the capacity of LMMs to comprehend and manipulate visual input data.

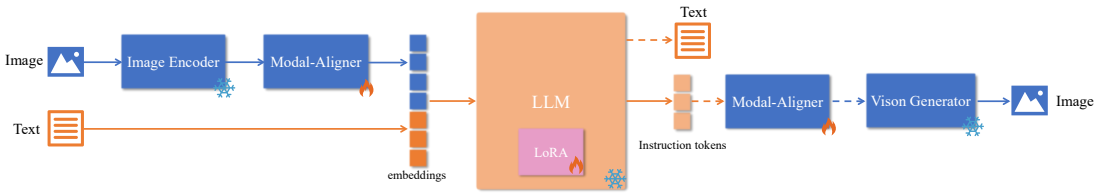


Figure 13. Common architecture of "Image+Text to Image+Text" models.

The remaining models incorporate a vision generator within their architectural composition, signifying that they are equipped for both image editing and for generating new image content or style transfer conditioned upon either textual or visual inputs. A case in point is LLaVA-Plus, which introduces 'Thought' and 'Action' fields to its input data and exists in two configurations: LLaVA-Plus (All Tools) harnesses all visual tools except semantic segmentation to enrich image features during both training and evaluation phases, whereas LLaVA-Plus (Fly) dynamically invokes and executes relevant tools for object detection, semantic segmentation, OCR, and conditional image generation according to the provided instructions. Figure 14 shows the capacities of LLaVA-Plus.

Figure 14. Examples showing how LLaVA-Plus processes and generates images.

Furthermore, a discernible trend among certain "Image+Text to Image+Text" models is their development incorporating a vision generator into the foundational architecture of prior image understanding models. Illustrative examples include Kosmos-G [63], which builds upon Kosmos-1 [130], along with LLaVA-Plus and LISA [131], both of which are constructed atop the LLaVA [101] framework.

Table 4. Display of architectural components across different "Image+Text to Image+Text" models, ordered by model name. Visual foundation models (VFMs) are a kind of pretrained model used to solve specific visual tasks, such as image recognition, image generation, etc.

Model	Vision Encoder	Input Modal Aligner	Upstream LLM Backbone	Output Modal Aligner	Vision Generator
CogCoM [127]	Eva-2-CLIP ViT	MLP	Vicuna-v1.5-7B	-	-
DreamLLM [132]	CLIP ViT	Linear Mapper	Vicuna-v1.5-7B	MLP	Stable Diffusion
DetGPT [128]	CLIP ViT	Linear Mapper	Vicuna-13B	-	-
DiffusionGPT [62]	-	-	ChatGPT	-	Stable Diffusion-1.5
Emu [133]	Eva-CLIP ViT	Causal Transformer	LLaMA-13B	MLP	Stable Diffusion-1.5
Emu-2 [134]	Eva-2-CLIP ViT	Linear Mapper	LLaMA-33B	MLP	Stable Diffusion XL

Table 4. *Cont.*

Model	Vision Encoder	Input Modal Aligner	Upstream LLM Backbone	Output Modal Aligner	Vision Generator
GILL [79]	CLIP ViT	Linear Mapper	OPT-6.7B	GILLMapper	Stable Diffusion-1.5
GLaMM [135]	CLIP ViT	MLP	Vicuna-7B/13B	MLP	Stable Diffusion XL
GPT4ROI [136]	CLIP ViT	Linear Mapper	Vicuna-7B	-	-
Kosmos-G [63]	CLIP ViT	Linear Mapper	MAGNETO	AlignerNet	Stable Diffusion-1.5
LaVIT [137]	ViT	Cross Attention Layer	LLaMA-7B	Designed-quantizer	Stable Diffusion
LISA [131]	ViT	-	LLaVA-7B/13B	-	-
LLaVA-Plus [138]	-	-	LLaVA-7B/13B	-	Stable Diffusion
MiniGPT-5 [139]	Eva-CLIP ViT	Q-Former + Linear Mapper	Vicuna-7B	Transformer + MLP	StableDiffusion-2
MM-Interleaved [80]	CLIP ViT	Cross Attention Layer	Vicuna-13B	Transformer	Stable Diffusion-2.1
PixelLM [140]	CLIP-ViT	MLP	LLaMA2-7B/13B	-	-
SEED [141]	CLIP ViT	Q-Former+Linear Mapper	OPT-2.7B	Q-Former+Linear Mapper	Stable Diffusion
Shikra [86]	CLIP ViT	Linear Mapper	Vicuna-7/13B	-	-
SPHINX-X [129]	DINOv2/CLIP-ConvNeXt	Linear Mapper	TinyLlama-1.1B/LLaMA2-13B/Mixtral-8×7B/InternLM2-7B	-	-
Visual ChatGPT [90]	Visual Foundation Models	-	ChatGPT	-	Visual Foundation Models
VL-GPT [142]	CLIP ViT	Causal Transformer	LLaMA-7B	Transformer	Stable Diffusion

According to the illustration in Table 4, it is apparent that almost all large-scale image generation models leverage variants from the Stable Diffusion series as their vision generators. This widespread adoption is attributed to the dominant performance of Stable Diffusion in the field of image synthesis, which consistently excels across a multitude of metrics. With respect to vision encoders, the majority of models opt for either CLIP ViT or Eva-CLIP ViT architectures.

In terms of upstream LLM backbone selections, the findings in image understanding diverge from those observed in this study. LLaMA has been preferred over Vicuna, primarily due to the differing landscape. LLaMA's commanding role is more preferable in this context.

3.5.4. Video+Text to Text+Video

"Video+Text to Text+Video" models can be conceptualized as video generation models, as listed in Table 5, where six representative models are presented. Among these, models such as CoDi-2 [143], ModaVerse [144], and NExT-GPT [145] exemplify 'any-to-any' large multimodal models capable of understanding or generating text, audio, image, and video content. Figure 15 provides an overview framework for these models.

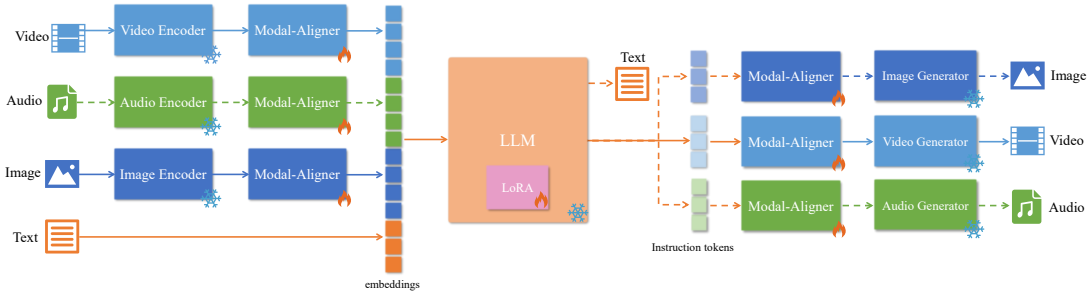


Figure 15. Common architecture of "Video+Text to Text+Video" models.

Table 5. Display of architectural components across different "Video+Text to Text+Video" models, ordered by model name. "Model Zoo" refers to various models in Huggingface, such as "damo vilab/text to video ms1.7b" [146].

Model	Vision Encoder	Input Modal Aligner	Upstream LLM Backbone	Output Modal Aligner	Video Generator
CoDi [147]	CLIP	Cross Attention Layer	-	Cross Attention Layer	designed Video LDM
CoDi-2 [143]	ImageBind	MLP	LLaMA 2-7B	MLP	Zeroscope-v2
GPT4Video [148]	CLIP ViT	Cross Attention Layer	LLaMA-7B	-	ZeroScope
HuggingGPT [149]	-	-	ChatGPT	-	Model Zoo
ModaVerse [144]	ImageBind	Linear Mapper	LLaMA-2	MLP	Videofusion
NExT-GPT [145]	ImageBind	Linear Mapper	Vicuna-7B	Transformer-31M	Zeroscope

Among the employed components, ImageBind [75] is the most commonly utilized vision encoder, whereas LLaMA 2 [9] represents the prevailing choice of upstream LLM backbone. For the video generator, the mainstream option is the ZeroScope series of open-source models from HuggingFace, which is a video generation model developed by Alibaba DAMO Academy. This model automatically generates videos consistent with user-provided text descriptions, incorporating visual elements (scenes), audio elements (music and sound effects), and subtitles. Furthermore, HuggingGPT [149] adopts the design concept of Visual-ChatGPT [125], employing ChatGPT [54] as its upstream LLM backbone to comprehend, process, and refine user prompts. In addition, it harnesses the capabilities of various existing external models within the community to facilitate multimodal input–output interactions.

At present, video generation models can only produce videos of several seconds in duration, regardless of whether they are based on LMM, Latent Diffusion Models (LDM) [147,150–152], or GANs [153,154], and often fall short in terms of both logical coherence and authenticity. Against this backdrop, OpenAI introduced Sora in February 2024, a model that, by understanding textual descriptions, can generate high-definition video content up to one minute long while adhering to the laws of physics in the real world and maintaining logical continuity. Currently, neither the technical details nor the inner workings of the Sora model have been publicly disclosed. Figure 16 presents an overview of the Sora model's architecture.

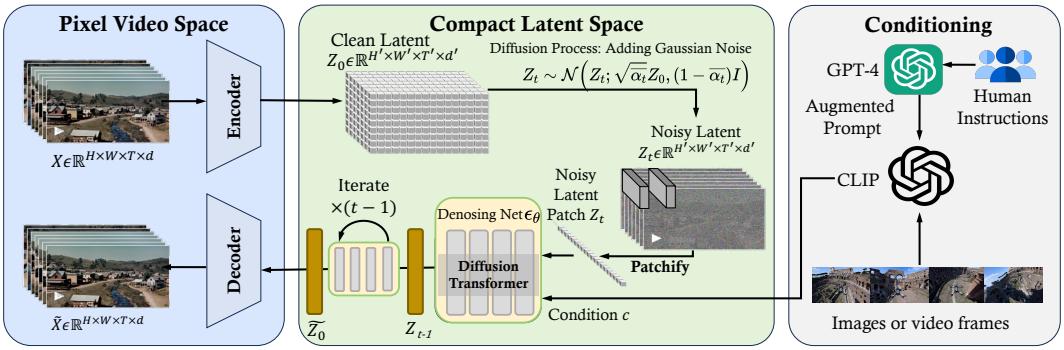


Figure 16. Overview of the Sora framework, sourced from [155].

4. A Unified Perspective of Large Models

In this section, we delve into commonalities and differences in the technical trajectories of large models from the view of architectural design, training strategies, fine-tuning techniques, and prompt engineering in both the LLM and LMM contexts. The overview frameworks of LLMs and LMMs are detailed in Figure 17.

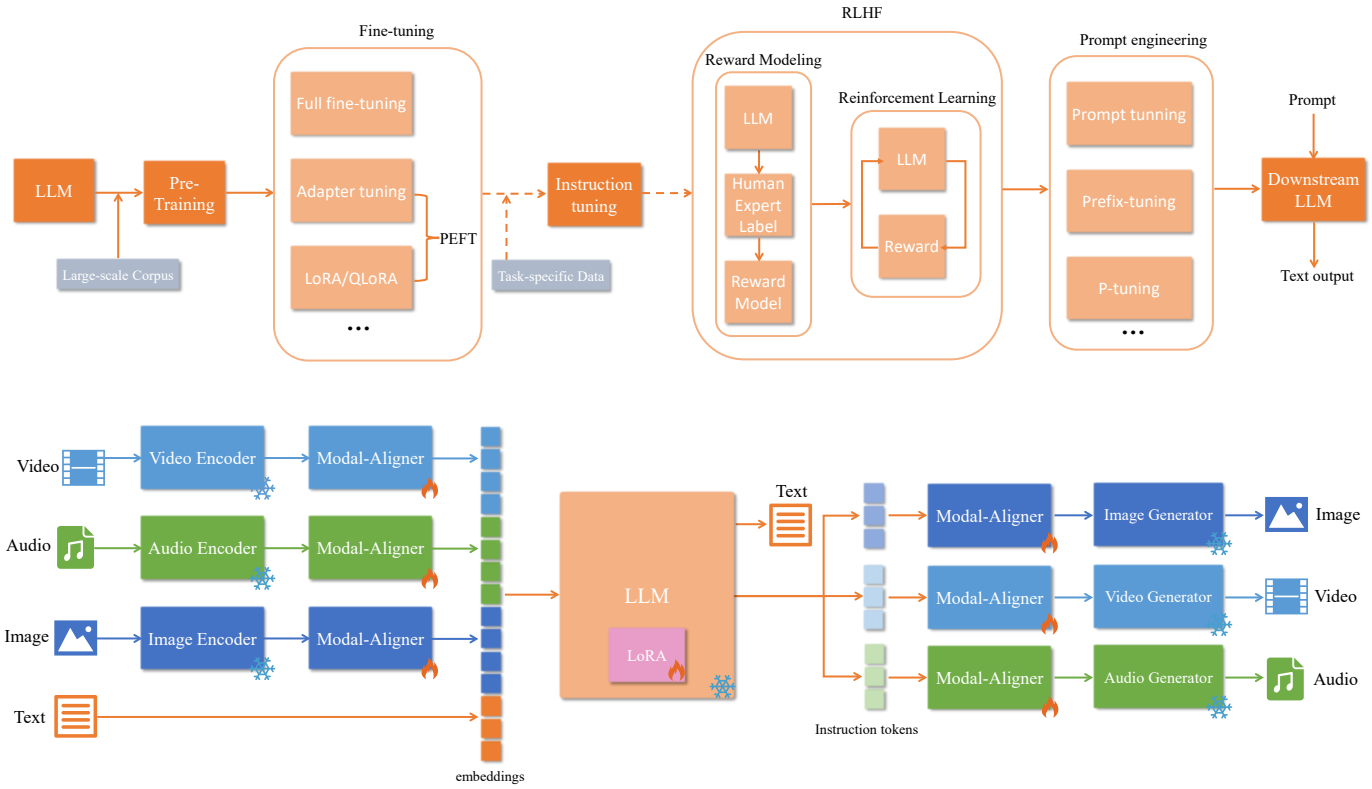


Figure 17. Comparison of LLM and LMM overview frameworks

At the architectural level, both LLMs and LMMs fundamentally draw upon the transformer architecture [21]. Within LLMs, there exist three primary transformer-based architectural configurations: encoder-decoder, encoder-only, and decoder-only. Conversely, LMMs build upon either pretrained or fine-tuned LLM backbones, with additional components potentially rooted in transformers as well; for example, vision encoders such as CLIP [12] utilize Vision Transformer [65] (ViT), vision generators such as Stable Diffusion-3 [13,14] utilize Diffusion Transformer [156] (DiT), and modal aligners can adopt smaller multilayer transformer architectures.

Regarding training strategies, LLMs place significant emphasis on pretraining, engaging in self-supervised learning over an unlabeled large-scale corpus utilizing autoregressive language modeling (ALM), prefix language modeling (PLM), or masked language modeling (MLM) objectives. This pretraining phase allows the model to learn universal patterns in text, thereby setting the upper limit of LLM capabilities. By contrast, LMMs do not train the upstream LLM backbone, instead focusing on the training the modal aligner to effectively align non-textual modalities into textual feature space, which establishes the lower bound of LMM capacities.

In terms of fine-tuning strategies, both LLMs and LMMs employ fine-tuning methodologies to adapt to new downstream tasks. LLMs utilize two categories of fine-tuning, depending on the extent of parameter updates: full fine-tuning and parameter-efficient fine-tuning (PEFT). Meanwhile, LMMs are fine-tuned to enhance the model's ability to generalize to novel tasks based on new instructions, thereby improving zero-shot performance. In addition, techniques such as LoRA can be incorporated during the fine-tuning process to control the parameter updating consumption.

Both LLMs and LMMs make use of prompt engineering to optimize model outputs through the refinement of prompts, in-context examples, or chain-of-thought (CoT) approaches. In LMMs, the application of prompt engineering expands beyond text modalities to encompass inputs from other modalities as well.

5. Comparative Analysis from the View of Globalization

In this section, we delve into the development status of large-scale models across diverse global regions, analyzing their latest achievements as well as examining the safety, limitations, and other contextual factors that shape this status. By integrating case studies along with economic and political analysis, we strive to offer a comprehensive understanding of the landscape of large-scale models from the view of globalization.

The United States (USA) holds a core and leadership position in various aspects of large model development, encompassing data, computational power, and model architecture. A suite of emblematic large models has originated within USA, including OpenAI's GPT series [2–5,85] and DALL-E series [157–159], Google's Transformer and PaLM series [6,7], DeepMind's Gemini series [160,161], and Meta (Facebook)'s LLaMA series [8,9], to name a few. Furthermore, the monopolistic edge in high-performance hardware enjoyed by the USA, embodied by NVIDIA's computing GPUs such as the A100 and H100, provides speed and efficiency advantages for training large models.

The USA's leadership in large model development is deeply rooted in its strong innovation ecosystem, backed by substantial private investments and a culture that encourages risk-taking. Socioeconomically, the presence of Silicon Valley and other tech hubs fosters a competitive environment that spurs advancements. Culturally, the emphasis on free speech and information access has enabled the creation of vast datasets.

However, this leadership is not without controversy. Models such as GPT-4 [5], while groundbreaking in the technological realm, have raised ethical concerns in the safety realm over data bias, potential misuse, and the exacerbation of existing social inequalities. Moreover, the environmental costs of these models, as well as production of the related hardware, poses additional carbon challenges due to their massive energy consumption [162,163].

China is demonstrating robust momentum in the development of large models, accompanied by a vibrant market. Robust backing from the Chinese government and scientific community has spurred intense competition among several leading enterprises, including Baidu, Alibaba, Tencent, and others seeking to innovate in the AI large model field. Chinese companies adopt an ecosystem-diverse and systematic approach, typically rolling out a series of models to create a holistic technological ecosystem. Examples include Baidu's ERNIE series [164–166] of large models and its derivative ERNIEBot, Huawei's PanGu series [10,11,49] model, Alibaba's Tongyi series, Tencent's HunYuan series. Additionally, Chinese universities actively participate in large model development and research, partner-

ing with tech companies or independently creating multiple large models, with Tsinghua University's CPM [36,37,167] series and GLM [168] being one such instance.

China's progress in AI is fueled by a combination of state-driven policies, significant investment, and a large domestic market. The socioeconomic context is characterized by a digital economy that thrives on massive data generation, which, coupled with government support, accelerates AI innovation. Politically, the central government's strategic plans, such as "Made in China 2025", prioritize AI, demonstrating a top-down commitment.

The case of China showcases how supportive state-driven policies and strong company–government partnership can drive innovation; however, this also raises questions about data privacy, given the state's involvement. While China has initiated the establishment of a relevant legal framework for data protection, ensuring in practice that all data sources and processing by international giants such as Tencent, Baidu, Alibaba, and ByteDance strictly adhere to regulatory requirements remains a complex issue.

South Korea is capitalizing on its semiconductor industry's computational strength to follow closely behind the USA in large model development. The Samsung Group's economic will to maintain technological competitiveness drives public-private partnerships and investments in AI. It has produced or invested several notable large models, such as the Naver Corporation's 2000B parameter HyperCLOVA [169], Kakao's KoGPT [170] based on GPT-3, and LG's EXAONE series LMMs.

Europe is taking a distinctive approach to large model development, emphasizing privacy protection, ethical considerations, and security. An illustration of this is the launch of the global collaboratively-built open-source BLOOM [171] by HuggingFace, supported by the French government, which aims to eliminate the secrecy and exclusivity of conventional large language models. Another example is the release of the pretrained Luminous by the German startup Aleph Alpha.

Politically, the EU's General Data Protection Regulation (GDPR) sets a global standard for data privacy, influencing how models are deployed. Culturally, a heightened awareness of historical experiences with totalitarianism and the importance of individual rights shapes Europe's cautious stance on AI.

A case study on Europe's models could highlight the practical implications of open-source collaborative efforts in fostering transparency and reducing biases.

Other countries and regions, including **Japan**, **India**, and **Southeast Asia**, generally trail in the race for large model development. Japan, despite its history in the semiconductor industry, has faced challenges with a less developed IT sector and a decline in semiconductor industries; nonetheless, it has introduced some large models through partnerships with other companies, such as the Japanese version of HyperCLOVA developed in conjunction with NAVER and its subsidiary LINE. Meanwhile, India, grappling with a complex linguistic environment and limited resources, has witnessed the advent of a handful of domestic large models, e.g., OpenHathi by Sarvam AI and Hanooman by the BharatGPT consortium.

6. On Future Directions

6.1. Less Computation, More Tokens

With the ever-increasing scale of parameters and the elongation of input token sequences, various large-scale models built upon the transformer block inevitably encounter computational efficiency issues concerning long sequences. Against this backdrop, the Mamba architecture is emerging as an innovative design, integrating the state space model (SSM) [172,173] framework with the transformer [21] architecture, thereby reducing reliance on the attention mechanism and realizing linear-time complexity in sequence modeling.

Compared to transformers of equivalent scale, Mamba exhibits heightened throughput and superior performance; thus, it is gradually becoming a central component in diverse large models. In the field of NLP, Mamba-inspired variants such as DenseMamba [174], Dual-path Mamba [175], and SPMamba [176] have been successfully employed. In CV, adaptations such as Vison Mamba (Vim) [177] (detailed in Figure 18), VMamba [178], and FusionMamba [179]

showcase the versatility of the Mamba architecture. Meanwhile, examples within the multimodal realm include VL-Mamba [180] and Motion Mamba [181], among others.

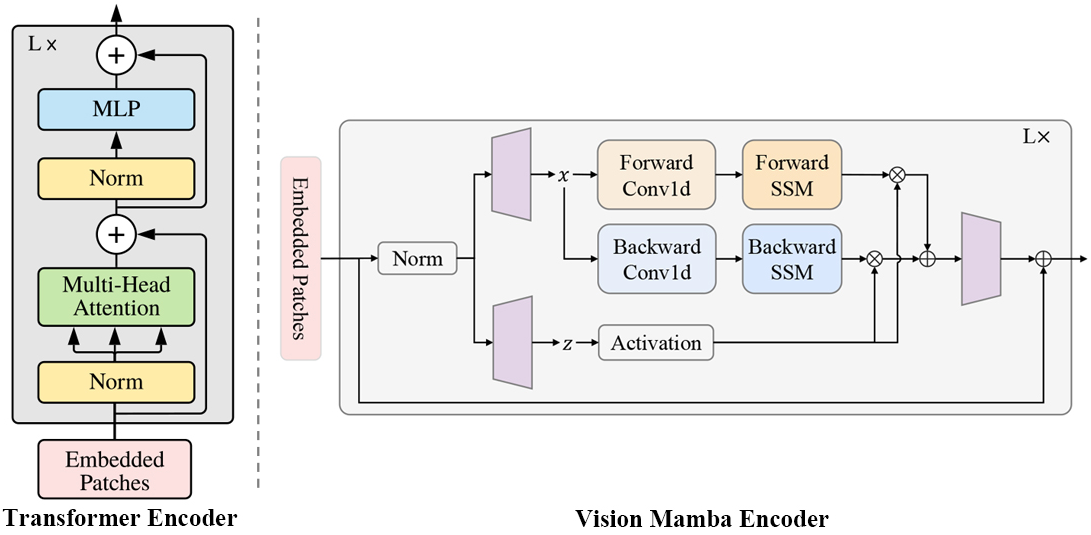


Figure 18. Comparison of ViT (Vision Transformer-based) and Vim (Mamba-based) architectures within the computer vision realm.

Hawk and Griffin [182], both novel RNN variants, have been propose by a Google. Hawk, utilizing gated linear recurrences, outperforms Mamba in downstream tasks, while Griffin, a hybrid model combining gated linear recurrence and local attention, matches Llama-2's performance with significantly reduced training data. Both models exhibit efficiency during training, lower inference latency, and higher throughput.

Looking ahead, in addition to Mamba, Hawk, and Griffin, there will be more and more new architectures challenging transformer, providing a foundational basis for the construction of better large-scale models.

6.2. Less Fine-Tuning, More Prompt Engineering

With the advancement of prompt engineering techniques, as evidenced by studies such as [29,32,33,92,93], we are currently witnessing a shift in the approach to adapting large models for downstream tasks. Traditionally, performance optimization has heavily relied on extensive parameter fine-tuning or parameter-efficient fine-tuning methods such as LoRA [27,28]; however, the costs associated with fine-tuning models with large parameter counts are notably high. Presently, a more efficient alternative is emerging wherein better results can be achieved by judicious manipulation of prompts or through the utilization of CoT. This transition significantly enhances the parameter efficiency of model adaptation, mitigating the need for exhaustive parameter adjustments while still achieving improved performance.

6.3. Fewer Parameters, More Datasets

The prevailing notion holds that the performance of large models scales proportionally with the increase in model parameters, as evidenced by studies including [1,4,24,84]. However, recent works such as [8,41] have substantiated scaling laws suggesting that even with a reduction in model parameters, equivalent performance can be attained by augmenting data diversity and employing larger-scale datasets. On the other hand, compared to comprehensive large models that can solve everything with a huge number of parameter, small models that perform well on specific tasks may be more popular. Within the realm of LMMs more specifically, there has been a discernible trend towards adopting backbone LLMs with relatively smaller parameter counts that are instead rooted in more extensive pretraining datasets.

Consequently, the anticipated trajectory of future research eschews the erstwhile emphasis on ceaselessly expanding model parameter scales, instead favoring the strategic exploitation of more varied and voluminous data resources to maximize the untapped potential of these models.

7. Conclusions

This article has systematically reviewed the evolutionary trajectory from single-modal LLMs to LMMs, encompassing the latest research developments. We have sequentially examined architectural designs, training strategies, and prompt engineering techniques, enumerated several representative LLMs, and conducted a taxonomy of recent 66 state-of-the-art visual–language LMMs. Furthermore, we have contrasted the development of large models in view of globalization and outlined three potential directions for future advancements in large model development. We hope that this comprehensive review will better equip researchers to understand how the capabilities of LLMs can be extended to LMMs and provide guidance in engaging with research endeavors involving large-scale models.

Author Contributions: Conceptualization, D.H.; methodology, D.H.; formal analysis, C.Y.; resources, X.P. and Q.L.; writing—original draft preparation, D.H.; writing—review and editing, D.H and C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (62176165), the Stable Support Projects for Shenzhen Higher Education Institutions (20220718110918001), and the Natural Science Foundation of Top Talent of SZTU (GDRC202131).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
2. Radford, A.; Narasimhan, K. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://api.semanticscholar.org/CorpusID:49313245 (accessed on 29 March 2024).
3. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
4. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
5. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774.
6. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. *arXiv* **2022**, arXiv:2204.02311.
7. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. PaLM 2 Technical Report. *arXiv* **2023**, arXiv:2305.10403.
8. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971.
9. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288.
10. Zeng, W.; Ren, X.; Su, T.; Wang, H.; Liao, Y.; Wang, Z.; Jiang, X.; Yang, Z.; Wang, K.; Zhang, X.; et al. PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation. *arXiv* **2021**, arXiv:2104.12369.
11. Ren, X.; Zhou, P.; Meng, X.; Huang, X.; Wang, Y.; Wang, W.; Li, P.; Zhang, X.; Podolskiy, A.; Arshinov, G.; et al. PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing. *arXiv* **2023**, arXiv:2303.10845.
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020.

13. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv* **2022**, arXiv:2112.10752.
14. Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; Rombach, R. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv* **2023**, arXiv:2307.01952.
15. Raiaan, M.A.K.; Mukta, M.S.H.; Fatema, K.; Fahad, N.M.; Sakib, S.; Mim, M.M.J.; Ahmad, J.; Ali, M.E.; Azam, S. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* **2024**, *12*, 26839–26874. [CrossRef]
16. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A Comprehensive Overview of Large Language Models. *arXiv* **2024**, arXiv:2307.06435.
17. Zhang, D.; Yu, Y.; Li, C.; Dong, J.; Su, D.; Chu, C.; Yu, D. MM-LLMs: Recent Advances in MultiModal Large Language Models. *arXiv* **2024**, arXiv:2401.13601.
18. Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; Chen, E. A Survey on Multimodal Large Language Models. *arXiv* **2024**, arXiv:2306.13549.
19. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* **2015**, arXiv:1506.00019.
20. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2023**, arXiv:1706.03762.
22. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
23. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2020**, arXiv:1909.11942.
24. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* **2023**, arXiv:1910.10683.
25. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified Language Model Pre-training for Natural Language Understanding and Generation. *arXiv* **2019**, arXiv:1905.03197.
26. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. *arXiv* **2019**, arXiv:1902.00751.
27. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685.
28. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv* **2023**, arXiv:2305.14314.
29. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv* **2021**, arXiv:2101.00190.
30. Schick, T.; Schütze, H. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. *arXiv* **2021**, arXiv:2009.07118.
31. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT Understands, Too. *arXiv* **2023**, arXiv:2103.10385.
32. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv* **2021**, arXiv:2104.08691.
33. Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; Li, L.; Sui, Z. A Survey on In-context Learning. *arXiv* **2023**, arXiv:2301.00234.
34. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2023**, arXiv:2201.11903.
35. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training Verifiers to Solve Math Word Problems. *arXiv* **2021**, arXiv:2110.14168.
36. Zhang, Z.; Gu, Y.; Han, X.; Chen, S.; Xiao, C.; Sun, Z.; Yao, Y.; Qi, F.; Guan, J.; Ke, P.; et al. CPM-2: Large-scale Cost-effective Pre-trained Language Models. *arXiv* **2021**, arXiv:2106.10715.
37. Zhang, Z.; Han, X.; Zhou, H.; Ke, P.; Gu, Y.; Ye, D.; Qin, Y.; Su, Y.; Ji, H.; Guan, J.; et al. CPM: A Large-scale Generative Chinese Pre-trained Language Model. *arXiv* **2020**, arXiv:2012.00413.
38. Qin, Y.; Lin, Y.; Yi, J.; Zhang, J.; Han, X.; Zhang, Z.; Su, Y.; Liu, Z.; Li, P.; Sun, M.; et al. Knowledge Inheritance for Pre-trained Language Models. *arXiv* **2022**, arXiv:2105.13880.
39. Barham, P.; Chowdhery, A.; Dean, J.; Ghemawat, S.; Hand, S.; Hurt, D.; Isard, M.; Lim, H.; Pang, R.; Roy, S.; et al. Pathways: Asynchronous Distributed Dataflow for ML. *arXiv* **2022**, arXiv:2203.12533.
40. Rae, J.W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *arXiv* **2022**, arXiv:2112.11446.
41. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. *arXiv* **2022**, arXiv:2203.15556.
42. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv* **2022**, arXiv:2205.01068.
43. Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv* **2020**, arXiv:2101.00027.

44. Baumgartner, J.; Zannettou, S.; Keegan, B.; Squire, M.; Blackburn, J. The Pushshift Reddit Dataset. *arXiv* **2020**, arXiv:2001.08435.
45. Zhang, B.; Sennrich, R. Root Mean Square Layer Normalization. *arXiv* **2019**, arXiv:1910.07467.
46. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv* **2023**, arXiv:2104.09864.
47. Xu, L.; Zhang, X.; Dong, Q. CLUECorpus2020: A Large-scale Chinese Corpus for Pre-training Language Model. *arXiv* **2020**, arXiv:2003.01355.
48. Yuan, S.; Zhao, H.; Du, Z.; Ding, M.; Liu, X.; Cen, Y.; Zou, X.; Yang, Z.; Tang, J. WuDaoCorpora: A super large-scale Chinese corpora for pre-training language models. *AI Open* **2021**, *2*, 65–68. [CrossRef]
49. Christopoulou, F.; Lampouras, G.; Gritta, M.; Zhang, G.; Guo, Y.; Li, Z.; Zhang, Q.; Xiao, M.; Shen, B.; Li, L.; et al. PanGu-Coder: Program Synthesis with Function-Level Language Modeling. *arXiv* **2022**, arXiv:2207.11280.
50. Gousios, G. The GHTorent dataset and tool suite. In Proceedings of the 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, USA, 18–19 May 2013; pp. 233–236. [CrossRef]
51. Tay, Y.; Dehghani, M.; Tran, V.Q.; Garcia, X.; Wei, J.; Wang, X.; Chung, H.W.; Bahri, D.; Schuster, T.; Zheng, S.; et al. UL2: Unifying Language Learning Paradigms. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
52. Ainslie, J.; Lee-Thorp, J.; de Jong, M.; Zemlyanskiy, Y.; Lebrón, F.; Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. *arXiv* **2023**, arXiv:2305.13245.
53. Sanh, V.; Webson, A.; Raffel, C.; Bach, S.H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T.L.; Raja, A.; et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv* **2022**, arXiv:2110.08207.
54. Introducing ChatGPT. Available online: https://openai.com/blog/chatgpt (accessed on 29 March 2024).
55. Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv* **2022**, arXiv:2112.09332.
56. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality | LMSYS Org. Available online: https://lmsys.org/blog/2023-03-30-vicuna/ (accessed on 29 March 2024).
57. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford Alpaca: An Instruction-Following LLaMA Model. 2023. Available online: https://github.com/tatsu-lab/stanford_alpaca (accessed on 29 March 2024).
58. Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. LIMA: Less Is More for Alignment. *arXiv* **2023**, arXiv:2305.11206.
59. Iyer, S.; Lin, X.V.; Pasunuru, R.; Mihaylov, T.; Simig, D.; Yu, P.; Shuster, K.; Wang, T.; Liu, Q.; Koura, P.S.; et al. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *arXiv* **2023**, arXiv:2212.12017.
60. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv* **2023**, arXiv:2301.12597.
61. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv* **2023**, arXiv:2304.10592.
62. Qin, J.; Wu, J.; Chen, W.; Ren, Y.; Li, H.; Wu, H.; Xiao, X.; Wang, R.; Wen, S. DiffusionGPT: LLM-Driven Text-to-Image Generation System. *arXiv* **2024**, arXiv:2401.10061.
63. Pan, X.; Dong, L.; Huang, S.; Peng, Z.; Chen, W.; Wei, F. Kosmos-G: Generating Images in Context with Multimodal Large Language Models. *arXiv* **2023**, arXiv:2310.02992.
64. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. *arXiv* **2021**, arXiv:2111.06377.
65. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
66. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030.
67. Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; Cao, Y. EVA: Exploring the Limits of Masked Visual Representation Learning at Scale. *arXiv* **2022**, arXiv:2211.07636.
68. Cherti, M.; Beaumont, R.; Wightman, R.; Wortsman, M.; Ilharco, G.; Gordon, C.; Schuhmann, C.; Schmidt, L.; Jitsev, J. Reproducible scaling laws for contrastive language-image learning. *arXiv* **2022**, arXiv:2212.07143.
69. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv* **2024**, arXiv:2304.07193.
70. Wu, Y.; Chen, K.; Zhang, T.; Hui, Y.; Nezhurina, M.; Berg-Kirkpatrick, T.; Dubnov, S. Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation. *arXiv* **2024**, arXiv:2211.06687.
71. Chen, F.; Han, M.; Zhao, H.; Zhang, Q.; Shi, J.; Xu, S.; Xu, B. X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages. *arXiv* **2023**, arXiv:2305.04160.
72. Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv* **2022**, arXiv:2212.04356.
73. Hsu, W.N.; Bolte, B.; Tsai, Y.H.H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *arXiv* **2021**, arXiv:2106.07447.

74. Zhang, Y.; Gong, K.; Zhang, K.; Li, H.; Qiao, Y.; Ouyang, W.; Yue, X. Meta-Transformer: A Unified Framework for Multimodal Learning. *arXiv* **2023**, arXiv:2307.10802.
75. Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K.V.; Joulin, A.; Misra, I. ImageBind: One Embedding Space To Bind Them All. *arXiv* **2023**, arXiv:2305.05665.
76. Zhu, B.; Lin, B.; Ning, M.; Yan, Y.; Cui, J.; Wang, H.; Pang, Y.; Jiang, W.; Zhang, J.; Li, Z.; et al. LanguageBind: Extending Video-Language Pretraining to N-modality by Language-based Semantic Alignment. *arXiv* **2024**, arXiv:2310.01852.
77. Jian, Y.; Gao, C.; Vosoughi, S. Bootstrapping Vision-Language Learning with Decoupled Language Pre-training. *arXiv* **2023**, arXiv:2307.07063.
78. Lu, J.; Gan, R.; Zhang, D.; Wu, X.; Wu, Z.; Sun, R.; Zhang, J.; Zhang, P.; Song, Y. Lyrics: Boosting Fine-grained Language-Vision Alignment and Comprehension via Semantic-aware Visual Objects. *arXiv* **2023**, arXiv:2312.05278.
79. Koh, J.Y.; Fried, D.; Salakhutdinov, R. Generating Images with Multimodal Language Models. *arXiv* **2023**, arXiv:2305.17216.
80. Tian, C.; Zhu, X.; Xiong, Y.; Wang, W.; Chen, Z.; Wang, W.; Chen, Y.; Lu, L.; Lu, T.; Zhou, J.; et al. MM-Interleaved: Interleaved Image-Text Generative Modeling via Multi-modal Feature Synchronizer. *arXiv* **2024**, arXiv:2401.10208.
81. Aghajanyan, A.; Huang, B.; Ross, C.; Karpukhin, V.; Xu, H.; Goyal, N.; Okhonko, D.; Joshi, M.; Ghosh, G.; Lewis, M.; et al. CM3: A Causal Masked Multimodal Model of the Internet. *arXiv* **2022**, arXiv:2201.07520.
82. Liu, H.; Chen, Z.; Yuan, Y.; Mei, X.; Liu, X.; Mandic, D.; Wang, W.; Plumbley, M.D. AudioLDM: Text-to-Audio Generation with Latent Diffusion Models. *arXiv* **2023**, arXiv:2301.12503.
83. Liu, H.; Tian, Q.; Yuan, Y.; Liu, X.; Mei, X.; Kong, Q.; Wang, Y.; Wang, W.; Wang, Y.; Plumbley, M.D. AudioLDM 2: Learning Holistic Audio Generation with Self-supervised Pretraining. *arXiv* **2023**, arXiv:2308.05734.
84. Wei, J.; Bosma, M.; Zhao, V.Y.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le, Q.V. Finetuned Language Models Are Zero-Shot Learners. *arXiv* **2022**, arXiv:2109.01652.
85. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:2203.02155.
86. Chen, K.; Zhang, Z.; Zeng, W.; Zhang, R.; Zhu, F.; Zhao, R. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv* **2023**, arXiv:2306.15195.
87. Li, K.; He, Y.; Wang, Y.; Li, Y.; Wang, W.; Luo, P.; Wang, Y.; Wang, L.; Qiao, Y. VideoChat: Chat-Centric Video Understanding. *arXiv* **2024**, arXiv:2305.06355.
88. Zhang, Y.; Zhang, R.; Gu, J.; Zhou, Y.; Lipka, N.; Yang, D.; Sun, T. LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding. *arXiv* **2024**, arXiv:2306.17107.
89. Liu, X.; Ji, K.; Fu, Y.; Tam, W.L.; Du, Z.; Yang, Z.; Tang, J. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *arXiv* **2022**, arXiv:2110.07602.
90. Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; Duan, N. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. *arXiv* **2023**, arXiv:2303.04671.
91. Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.W.; Zhu, S.C.; Tafjord, O.; Clark, P.; Kalyan, A. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. *arXiv* **2022**, arXiv:2209.09513.
92. Zhang, Z.; Zhang, A.; Li, M.; Zhao, H.; Karypis, G.; Smola, A. Multimodal Chain-of-Thought Reasoning in Language Models. *arXiv* **2023**, arXiv:2302.00923.
93. Ge, J.; Luo, H.; Qian, S.; Gan, Y.; Fu, J.; Zhang, S. Chain of Thought Prompt Tuning in Vision Language Models. *arXiv* **2023**, arXiv:2304.07919.
94. Hu, W.; Xu, Y.; Li, Y.; Li, W.; Chen, Z.; Tu, Z. BLIVA: A Simple Multimodal LLM for Better Handling of Text-Rich Visual Questions. *arXiv* **2023**, arXiv:2308.09936.
95. Zhao, L.; Yu, E.; Ge, Z.; Yang, J.; Wei, H.; Zhou, H.; Sun, J.; Peng, Y.; Dong, R.; Han, C.; et al. ChatSpot: Bootstrapping Multimodal LLMs via Precise Referring Instruction Tuning. *arXiv* **2023**, arXiv:2307.09474.
96. Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; et al. CogVLM: Visual Expert for Pretrained Language Models. *arXiv* **2023**, arXiv:2311.03079v2.
97. Chen, Y.; Sikka, K.; Cogswell, M.; Ji, H.; Divakaran, A. DRESS: Instructing Large Vision-Language Models to Align and Interact with Humans via Natural Language Feedback. *arXiv* **2023**, arXiv:2311.10081.
98. Introducing IDEFICS: An Open Reproduction of State-of-the-Art Visual Langage Model. Available online: https://huggingface.co/blog/idefics (accessed on 29 March 2024).
99. Zhang, P.; Dong, X.; Wang, B.; Cao, Y.; Xu, C.; Ouyang, L.; Zhao, Z.; Duan, H.; Zhang, S.; Ding, S.; et al. InternLM-XComposer: A Vision-Language Large Model for Advanced Text-image Comprehension and Composition. *arXiv* **2023**, arXiv:2309.15112.
100. Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Wang, B.; Ouyang, L.; Wei, X.; Zhang, S.; Duan, H.; Cao, M.; et al. InternLM-XComposer2: Mastering Free-form Text-Image Composition and Comprehension in Vision-Language Large Model. *arXiv* **2024**, arXiv:2401.16420.
101. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning. *arXiv* **2023**, arXiv:2304.08485.
102. Liu, H.; Li, C.; Li, Y.; Lee, Y.J. Improved Baselines with Visual Instruction Tuning. *arXiv* **2023**, arXiv:2310.03744.
103. Chen, J.; Zhu, D.; Shen, X.; Li, X.; Liu, Z.; Zhang, P.; Krishnamoorthi, R.; Chandra, V.; Xiong, Y.; Elhoseiny, M. MiniGPT-v2: Large Language Model as a Unified Interface for Vision-Language Multi-Task Learning. *arXiv* **2023**, arXiv:2310.09478.
104. Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality. *arXiv* **2023**, arXiv:2304.14178.

105. Ye, Q.; Xu, H.; Ye, J.; Yan, M.; Hu, A.; Liu, H.; Qian, Q.; Zhang, J.; Huang, F.; Zhou, J. mPLUG-Owl2: Revolutionizing Multi-modal Large Language Model with Modality Collaboration. *arXiv* **2023**, arXiv:2311.04257.
106. Ye, J.; Hu, A.; Xu, H.; Ye, Q.; Yan, M.; Dan, Y.; Zhao, C.; Xu, G.; Li, C.; Tian, J.; et al. mPLUG-DocOwl: Modularized Multimodal Large Language Model for Document Understanding. *arXiv* **2023**, arXiv:2307.02499.
107. Chu, X.; Qiao, L.; Lin, X.; Xu, S.; Yang, Y.; Hu, Y.; Wei, F.; Zhang, X.; Zhang, B.; Wei, X.; et al. MobileVLM : A Fast, Strong and Open Vision Language Assistant for Mobile Devices. *arXiv* **2023**, arXiv:2312.16886.
108. Chu, X.; Qiao, L.; Zhang, X.; Xu, S.; Wei, F.; Yang, Y.; Sun, X.; Hu, Y.; Lin, X.; Zhang, B.; et al. MobileVLM V2: Faster and Stronger Baseline for Vision Language Model. *arXiv* **2024**, arXiv:2402.03766.
109. Li, B.; Zhang, Y.; Chen, L.; Wang, J.; Yang, J.; Liu, Z. Otter: A Multi-Modal Model with In-Context Instruction Tuning. *arXiv* **2023**, arXiv:2305.03726.
110. Yuan, Y.; Li, W.; Liu, J.; Tang, D.; Luo, X.; Qin, C.; Zhang, L.; Zhu, J. Osprey: Pixel Understanding with Visual Instruction Tuning. *arXiv* **2023**, arXiv:2312.10032.
111. Su, Y.; Lan, T.; Li, H.; Xu, J.; Wang, Y.; Cai, D. PandaGPT: One Model To Instruction-Follow Them All. *arXiv* **2023**, arXiv:2305.16355.
112. Chen, X.; Djolonga, J.; Padlewski, P.; Mustafa, B.; Changpinyo, S.; Wu, J.; Ruiz, C.R.; Goodman, S.; Wang, X.; Tay, Y.; et al. PaLI-X: On Scaling up a Multilingual Vision and Language Model. *arXiv* **2023**, arXiv:2305.18565.
113. Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; Zhou, J. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv* **2023**, arXiv:2308.12966.
114. Yu, T.; Yao, Y.; Zhang, H.; He, T.; Han, Y.; Cui, G.; Hu, J.; Liu, Z.; Zheng, H.T.; Sun, M.; et al. RLHF-V: Towards Trustworthy MLLMs via Behavior Alignment from Fine-grained Correctional Human Feedback. *arXiv* **2023**, arXiv:2312.00849.
115. Li, L.; Xie, Z.; Li, M.; Chen, S.; Wang, P.; Chen, L.; Yang, Y.; Wang, B.; Kong, L. Silkie: Preference Distillation for Large Visual Language Models. *arXiv* **2023**, arXiv:2312.10665.
116. Lin, J.; Yin, H.; Ping, W.; Lu, Y.; Molchanov, P.; Tao, A.; Mao, H.; Kautz, J.; Shoeybi, M.; Han, S. VILA: On Pre-training for Visual Language Models. *arXiv* **2024**, arXiv:2312.07533.
117. Ma, Y.; Cao, Y.; Sun, J.; Pavone, M.; Xiao, C. Dolphins: Multimodal Language Model for Driving. *arXiv* **2023**, arXiv:2312.00438.
118. Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv* **2024**, arXiv:2312.14238.
119. Zeng, Y.; Zhang, H.; Zheng, J.; Xia, J.; Wei, G.; Wei, Y.; Zhang, Y.; Kong, T. What Matters in Training a GPT4-Style Language Model with Multimodal Inputs? *arXiv* **2023**, arXiv:2307.02469.
120. Xu, H.; Ye, Q.; Wu, X.; Yan, M.; Miao, Y.; Ye, J.; Xu, G.; Hu, A.; Shi, Y.; Xu, G.; et al. Youku-mPLUG: A 10 Million Large-scale Chinese Video-Language Dataset for Pre-training and Benchmarks. *arXiv* **2023**, arXiv:2306.04362.
121. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? *arXiv* **2021**, arXiv:2102.05095.
122. for Intelligent Computing, I. Chinese GPT-3-1.3B. Available online: https://www.modelscope.cn/models/damo/nlp_gpt3_text-generation_1.3B (accessed on 19 April 2024).
123. for Intelligent Computing, I. Chinese GPT-3-2.7B. Available online: https://www.modelscope.cn/models/damo/nlp_gpt3_text-generation_2.7B (accessed on 19 April 2024).
124. Chen, S.; Li, H.; Wang, Q.; Zhao, Z.; Sun, M.; Zhu, X.; Liu, J. VAST: A Vision-Audio-Subtitle-Text Omni-Modality Foundation Model and Dataset. *arXiv* **2023**, arXiv:2305.18500.
125. Maaz, M.; Rasheed, H.; Khan, S.; Khan, F.S. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. *arXiv* **2023**, arXiv:2306.05424.
126. Zhang, H.; Li, X.; Bing, L. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. *arXiv* **2023**, arXiv:2306.02858.
127. Qi, J.; Ding, M.; Wang, W.; Bai, Y.; Lv, Q.; Hong, W.; Xu, B.; Hou, L.; Li, J.; Dong, Y.; et al. CogCoM: Train Large Vision-Language Models Diving into Details through Chain of Manipulations. *arXiv* **2024**, arXiv:2402.04236.
128. Pi, R.; Gao, J.; Diao, S.; Pan, R.; Dong, H.; Zhang, J.; Yao, L.; Han, J.; Xu, H.; Kong, L.; et al. DetGPT: Detect What You Need via Reasoning. *arXiv* **2023**, arXiv:2305.14167.
129. Gao, P.; Zhang, R.; Liu, C.; Qiu, L.; Huang, S.; Lin, W.; Zhao, S.; Geng, S.; Lin, Z.; Jin, P.; et al. SPHINX-X: Scaling Data and Parameters for a Family of Multi-modal Large Language Models. *arXiv* **2024**, arXiv:2402.05935.
130. Huang, S.; Dong, L.; Wang, W.; Hao, Y.; Singhal, S.; Ma, S.; Lv, T.; Cui, L.; Mohammed, O.K.; Patra, B.; et al. Language Is Not All You Need: Aligning Perception with Language Models. *arXiv* **2023**, arXiv:2302.14045.
131. Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; Jia, J. LISA: Reasoning Segmentation via Large Language Model. *arXiv* **2023**, arXiv:2308.00692.
132. Dong, R.; Han, C.; Peng, Y.; Qi, Z.; Ge, Z.; Yang, J.; Zhao, L.; Sun, J.; Zhou, H.; Wei, H.; et al. DreamLLM: Synergistic Multimodal Comprehension and Creation. *arXiv* **2023**, arXiv:2309.11499.
133. Sun, Q.; Yu, Q.; Cui, Y.; Zhang, F.; Zhang, X.; Wang, Y.; Gao, H.; Liu, J.; Huang, T.; Wang, X. Generative Pretraining in Multimodality. *arXiv* **2023**, arXiv:2307.05222.
134. Sun, Q.; Cui, Y.; Zhang, X.; Zhang, F.; Yu, Q.; Luo, Z.; Wang, Y.; Rao, Y.; Liu, J.; Huang, T.; et al. Generative Multimodal Models are In-Context Learners. *arXiv* **2023**, arXiv:2312.13286.

135. Rasheed, H.; Maaz, M.; Mullappilly, S.S.; Shaker, A.; Khan, S.; Cholakkal, H.; Anwer, R.M.; Xing, E.; Yang, M.H.; Khan, F.S. GLaMM: Pixel Grounding Large Multimodal Model. *arXiv* **2023**, arXiv:2311.03356.
136. Zhang, S.; Sun, P.; Chen, S.; Xiao, M.; Shao, W.; Zhang, W.; Liu, Y.; Chen, K.; Luo, P. GPT4RoI: Instruction Tuning Large Language Model on Region-of-Interest. *arXiv* **2023**, arXiv:2307.03601.
137. Jin, Y.; Xu, K.; Xu, K.; Chen, L.; Liao, C.; Tan, J.; Huang, Q.; Chen, B.; Lei, C.; Liu, A.; et al. Unified Language-Vision Pretraining in LLM with Dynamic Discrete Visual Tokenization. *arXiv* **2024**, arXiv:2309.04669.
138. Liu, S.; Cheng, H.; Liu, H.; Zhang, H.; Li, F.; Ren, T.; Zou, X.; Yang, J.; Su, H.; Zhu, J.; et al. LLaVA-Plus: Learning to Use Tools for Creating Multimodal Agents. *arXiv* **2023**, arXiv:2311.05437.
139. Zheng, K.; He, X.; Wang, X.E. MiniGPT-5: Interleaved Vision-and-Language Generation via Generative Vokens. *arXiv* **2023**, arXiv:2310.02239.
140. Ren, Z.; Huang, Z.; Wei, Y.; Zhao, Y.; Fu, D.; Feng, J.; Jin, X. PixelLM: Pixel Reasoning with Large Multimodal Model. *arXiv* **2023**, arXiv:2312.02228.
141. Ge, Y.; Ge, Y.; Zeng, Z.; Wang, X.; Shan, Y. Planting a SEED of Vision in Large Language Model. *arXiv* **2023**, arXiv:2307.08041.
142. Zhu, J.; Ding, X.; Ge, Y.; Ge, Y.; Zhao, S.; Zhao, H.; Wang, X.; Shan, Y. VL-GPT: A Generative Pre-trained Transformer for Vision and Language Understanding and Generation. *arXiv* **2023**, arXiv:2312.09251.
143. Tang, Z.; Yang, Z.; Khademi, M.; Liu, Y.; Zhu, C.; Bansal, M. CoDi-2: In-Context, Interleaved, and Interactive Any-to-Any Generation. *arXiv* 2023, arXiv:2311.18775.
144. Wang, X.; Zhuang, B.; Wu, Q. ModaVerse: Efficiently Transforming Modalities with LLMs. *arXiv* **2024**, arXiv:2401.06395.
145. Wu, S.; Fei, H.; Qu, L.; Ji, W.; Chua, T.S. NExT-GPT: Any-to-Any Multimodal LLM. *arXiv* **2023**, arXiv:2309.05519.
146. Wang, J.; Yuan, H.; Chen, D.; Zhang, Y.; Wang, X.; Zhang, S. Modelscope text-to-video technical report. *arXiv* **2023**, arXiv:2308.06571.
147. Tang, Z.; Yang, Z.; Zhu, C.; Zeng, M.; Bansal, M. Any-to-Any Generation via Composable Diffusion. *arXiv* **2023**, arXiv:2305.11846.
148. Wang, Z.; Wang, L.; Zhao, Z.; Wu, M.; Lyu, C.; Li, H.; Cai, D.; Zhou, L.; Shi, S.; Tu, Z. GPT4Video: A Unified Multimodal Large Language Model for lnstruction-Followed Understanding and Safety-Aware Generation. *arXiv* **2023**, arXiv:2311.16511.
149. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. *arXiv* **2023**, arXiv:2303.17580.
150. Blattmann, A.; Rombach, R.; Ling, H.; Dockhorn, T.; Kim, S.W.; Fidler, S.; Kreis, K. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. *arXiv* **2023**, arXiv:2304.08818.
151. Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv* **2023**, arXiv:2311.15127.
152. Girdhar, R.; Singh, M.; Brown, A.; Duval, Q.; Azadi, S.; Rambhatla, S.S.; Shah, A.; Yin, X.; Parikh, D.; Misra, I. Emu Video: Factorizing Text-to-Video Generation by Explicit Image Conditioning. *arXiv* **2023**, arXiv:2311.10709.
153. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661.
154. Brooks, T.; Hellsten, J.; Aittala, M.; Wang, T.C.; Aila, T.; Lehtinen, J.; Liu, M.Y.; Efros, A.A.; Karras, T. Generating Long Videos of Dynamic Scenes. *arXiv* **2022**, arXiv:2206.03429.
155. Liu, Y.; Zhang, K.; Li, Y.; Yan, Z.; Gao, C.; Chen, R.; Yuan, Z.; Huang, Y.; Sun, H.; Gao, J.; et al. Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models. *arXiv* **2024**, arXiv:2402.17177.
156. Peebles, W.; Xie, S. Scalable Diffusion Models with Transformers. *arXiv* **2023**, arXiv:2212.09748.
157. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. *arXiv* **2021**, arXiv:2102.12092.
158. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
159. Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; et al. Improving Image Generation with Better Captions. Available online: https://api.semanticscholar.org/CorpusID:264403242 (accessed on 19 April 2024).
160. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv* **2024**, arXiv:2312.11805.
161. Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.B.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
162. Gupta, U.; Kim, Y.G.; Lee, S.; Tse, J.; Lee, H.H.S.; Wei, G.Y.; Brooks, D.; Wu, C.J. Chasing Carbon: The Elusive Environmental Footprint of Computing. In Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Seoul, Republic of Korea, 27 February–3 March 2021; pp. 854–867. [CrossRef]
163. Patterson, D.; Gonzalez, J.; Le, Q.; Liang, C.; Munguia, L.M.; Rothchild, D.; So, D.; Texier, M.; Dean, J. Carbon Emissions and Large Neural Network Training. *arXiv* **2021**, arXiv:2104.10350.
164. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv* **2019**, arXiv:1904.09223.
165. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. *arXiv* **2019**, arXiv:1907.12412.

166. Sun, Y.; Wang, S.; Feng, S.; Ding, S.; Pang, C.; Shang, J.; Liu, J.; Chen, X.; Zhao, Y.; Lu, Y.; et al. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *arXiv* **2021**, arXiv:2107.02137.
167. Hu, J.; Yao, Y.; Wang, C.; Wang, S.; Pan, Y.; Chen, Q.; Yu, T.; Wu, H.; Zhao, Y.; Zhang, H.; et al. Large Multilingual Models Pivot Zero-Shot Multimodal Learning across Languages. *arXiv* **2024**, arXiv:2308.12038.
168. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. *arXiv* **2022**, arXiv:2103.10360.
169. Kim, B.; Kim, H.; Lee, S.W.; Lee, G.; Kwak, D.; Jeon, D.H.; Park, S.; Kim, S.; Kim, S.; Seo, D.; et al. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. *arXiv* **2021**, arXiv:2109.04650.
170. Kim, I.; Han, G.; Ham, J.; Baek, W. KoGPT: KakaoBrain Korean(hangul) Generative Pre-trained Transformer. 2021. Available online: https://github.com/kakaobrain/kogpt (accessed on 19 April 2024).
171. Workshop, B.; Scao, T.L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv* **2023**, arXiv:2211.05100.
172. Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; Ré, C. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers. *arXiv* **2021**, arXiv:2110.13985.
173. Gu, A.; Goel, K.; Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces. *arXiv* **2022**, arXiv:2111.00396.
174. He, W.; Han, K.; Tang, Y.; Wang, C.; Yang, Y.; Guo, T.; Wang, Y. DenseMamba: State Space Models with Dense Hidden Connection for Efficient Large Language Models. *arXiv* **2024**, arXiv:2403.00818.
175. Jiang, X.; Han, C.; Mesgarani, N. Dual-path Mamba: Short and Long-term Bidirectional Selective Structured State Space Models for Speech Separation. *arXiv* **2024**, arXiv:2403.18257.
176. Li, K.; Chen, G. SPMamba: State-space model is all you need in speech separation. *arXiv* **2024**, arXiv:2404.02063.
177. Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. *arXiv* **2024**, arXiv:2401.09417.
178. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Liu, Y. VMamba: Visual State Space Model. *arXiv* **2024**, arXiv:2401.10166.
179. Peng, S.; Zhu, X.; Deng, H.; Lei, Z.; Deng, L.J. FusionMamba: Efficient Image Fusion with State Space Model. *arXiv* **2024**, arXiv:2404.07932.
180. Qiao, Y.; Yu, Z.; Guo, L.; Chen, S.; Zhao, Z.; Sun, M.; Wu, Q.; Liu, J. VL-Mamba: Exploring State Space Models for Multimodal Learning. *arXiv* **2024**, arXiv:2403.13600.
181. Zhang, Z.; Liu, A.; Reid, I.; Hartley, R.; Zhuang, B.; Tang, H. Motion Mamba: Efficient and Long Sequence Motion Generation with Hierarchical and Bidirectional Selective SSM. *arXiv* **2024**, arXiv:2403.07487.
182. De, S.; Smith, S.L.; Fernando, A.; Botev, A.; Cristian-Muraru, G.; Gu, A.; Haroun, R.; Berrada, L.; Chen, Y.; Srinivasan, S.; et al. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models. *arXiv* **2024**, arXiv:2402.19427.