

# The Evolution of Multimodal Model Architectures

**Shakti N. Wadekar** Purdue University swadekar@purdue.edu  
**Abhishek Chaurasia** Chaos Industries Inc. abhi@choasinc.com  
**Aman Chadha** Stanford; Amazon\* hi@aman.ai  
**Eugenio Culurciello** Purdue University euge@purdue.edu

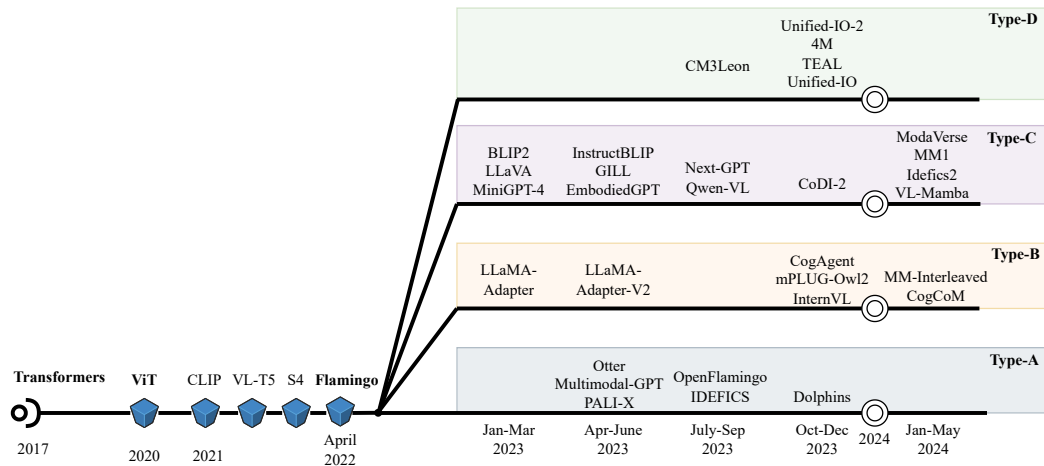


Figure 1: Development timeline of Multimodal models grouped in four proposed architecture types.

## Abstract

This work uniquely identifies and characterizes four prevalent multimodal model architectural patterns in the contemporary multimodal landscape. Systematically categorizing models by architecture type facilitates monitoring of developments in the multimodal domain. Distinct from recent survey papers that present general information on multimodal architectures, this research conducts a comprehensive exploration of architectural details and identifies four specific architectural types. The types are distinguished by their respective methodologies for integrating multimodal inputs into the deep neural network model. The first two types (Type A and B) deeply fuse multimodal inputs within the internal layers of the model, whereas the following two types (Type C and D) facilitate early fusion at the input stage. Type-A employs standard cross-attention, whereas Type-B utilizes custom-designed layers for modality fusion within the internal layers. On the other hand, Type-C utilizes modality-specific encoders, while Type-D leverages tokenizers to process the modalities at the model’s input stage. The identified architecture types aid the monitoring of any-to-any multimodal model development. Notably, Type-C and Type-D are currently favored in the construction of any-to-any multimodal models. Type-C, distinguished by its non-tokenizing multimodal model architecture, is emerging as a viable alternative to Type-D, which utilizes input-tokenizing techniques. To assist in model selection, this work highlights the advantages and disadvantages of each architecture type based on data and compute requirements, architecture complexity, scalability, simplification of adding modalities, training objectives, and any-to-any multimodal generation capability.

\*Work does not relate to position at Amazon.

# 1 Introduction

The multimodal domain of machine learning has seen significant advancements in recent years. The proliferation of models capable of processing images, audio, or video in conjunction with text (language) has notably expanded (Alayrac et al. [2022], Lu et al. [2023], Mizrahi et al. [2024], Wu et al. [2023a], Yang et al. [2024], Tang et al. [2023a]). Remarkable strides have been particularly evident in the integration of image and text modalities across diverse vision-language tasks, primarily because of the Transformer model Vaswani et al. [2017]. The Transformer model Vaswani et al. [2017], a pioneering deep neural network (NN) architecture, has spearheaded a unified framework for cross-domain learning. This singular model exhibits remarkable efficacy in comprehending and processing data from diverse domains. The introduction of the Transformer model Vaswani et al. [2017] for Natural Language Processing (NLP) in 2017 marked the inception of transformer-based model architectures. Subsequently, the introduction of the Vision Transformer (ViT) Dosovitskiy et al. [2021] and CLIP Radford et al. [2021] for the vision domain showcased the versatility of transformers in handling image-related tasks. This demonstration highlighted the transformer’s ability to learn from diverse domains, prompting a series of initiatives aimed at constructing models capable of jointly processing image and text data, leveraging the robust Transformer model architecture. Flamingo Alayrac et al. [2022], a transformer-based multimodal model that incorporates both image and text data as input, exhibited outstanding performance on vision-language tasks. These results served as a catalyst for further advancements in the multimodal domain and encouraged the integration of additional modalities Awadalla et al. [2023], Li et al. [2023a], Gong et al. [2023], Laurençon et al. [2024a], Ma et al. [2023].

Diverse methodologies have been employed in numerous research efforts dedicated to handling mixed modalities, including augmenting Large Language Models (LLMs) to create multimodal architectures Alayrac et al. [2022], Gong et al. [2023], Zhang et al. [2023a], Gao et al. [2023], Wang et al. [2023a], Ye et al. [2023a], Chen et al. [2023a], Tian et al. [2024], Lin et al. [2024], training encoder-decoder style transformers with different input modalities Mizrahi et al. [2024], Lu et al. [2022a], Lu et al. [2023], and exploring alternative approaches (Section 2). The plethora of research presents challenges in effectively monitoring the progression of model architectures and identifying emerging trends in next-generation multimodal model designs. We examine contemporary landscape of state-of-the-art multimodal models, and identify distinct multimodal model architectures based on the fusion of inputs into the deep neural networks. Primarily, we group existing multimodal architectures in four broad categories namely: Type - A, B, C, and D. In Type-A and Type-B architectures, deep fusion of input modalities is realized through the integration of inputs within the internal layers of the model, whereas in Type-C and Type-D architectures, early fusion of modalities occurs at the input stage of the model. Details of these four types are discussed in Section 3. Summary of our contributions:

- To the best of our knowledge, this is the first work that explicitly identifies the four broad architecture types: Type - A, B, C, and D. Figure 2 shows the taxonomy of multimodal model architectures. We associate state-of-the-art models with these types, and outline their advantages and disadvantages, thereby facilitating a simplified comprehension, visualization and selection of multimodal model architectures.
- Furthermore, our work also underscores the principal architectural types involved in constructing any-to-any modality multimodal models, which cannot be found in other survey works like Zhang et al. [2024a], Yin et al. [2024], Caffagni et al. [2024], Wang et al. [2023b], Wu et al. [2023b], Wu et al. [2023b], Guo et al. [2023].
- To facilitate model selection, this study highlights the advantages and disadvantages of each architecture type, considering factors such as training data and compute requirements, architecture complexity, scalability, ease of integrating modalities, and any-to-any modality capability.

## 2 Related Work

In this section, we list and discuss existing survey literature encompassing multimodal learning, multimodal data, and multimodal large language models (Table 1). Our work illuminates a spectrum of multimodal architecture types through an examination of numerous multimodal works, which is absent in other survey literature.

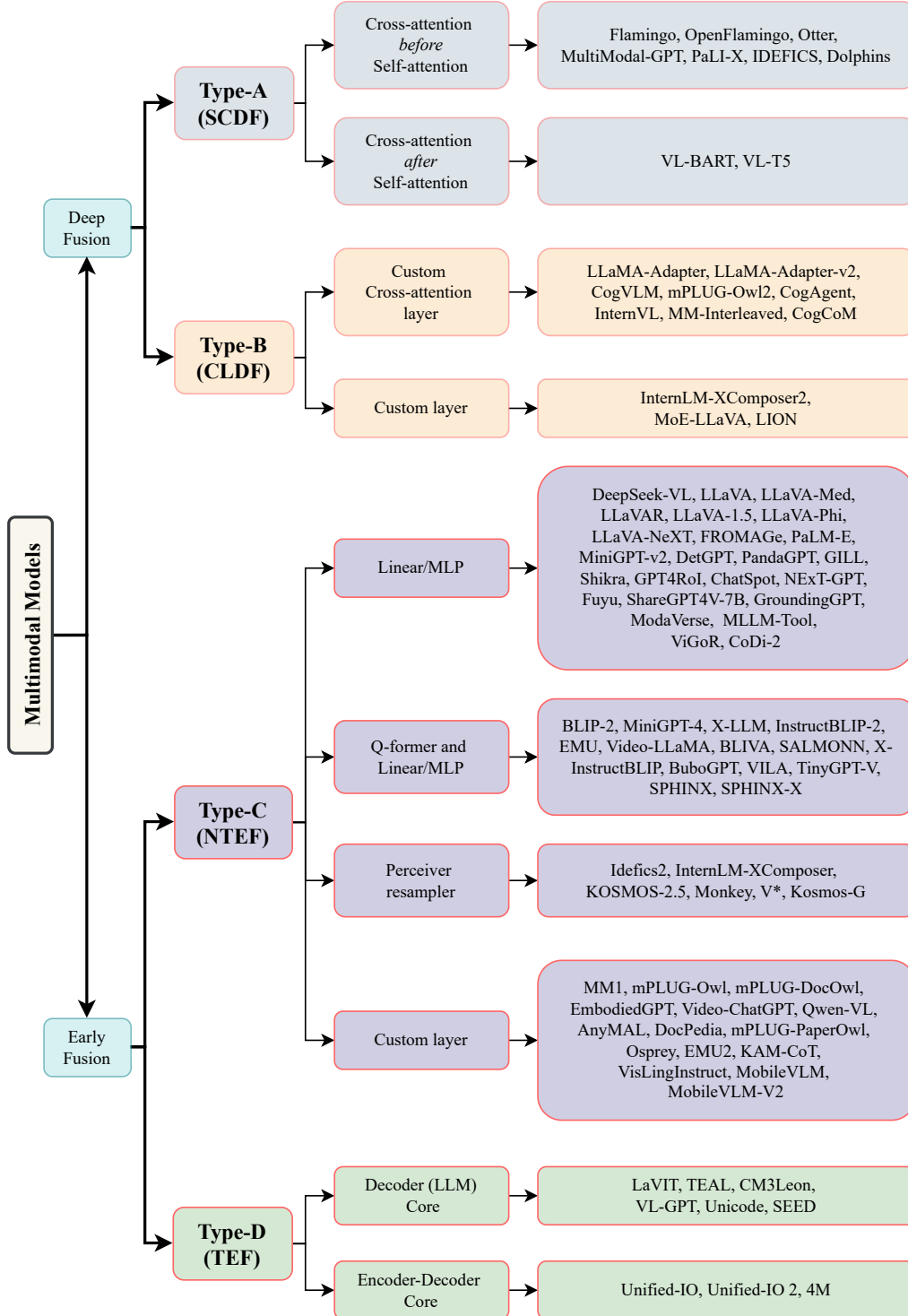


Figure 2: Taxonomy of multimodal model architectures. Four distinct types of multimodal architectures and their sub-types are outlined. Various models are systematically catalogued to the types and sub-types. Deep Fusion: Type-A and Type-B fuses multimodal inputs within the internal layers of the model. Early Fusion: Type-C and Type-D facilitate fusion at the input stage. Type-A uses standard cross-attention, whereas Type-B utilizes custom-designed cross-attention or specialized layers. Type-C is a non-tokenizing multimodal model architecture, while Type-D, employs input-tokenization (discrete tokens). SCDF: Standard Cross-attention based Deep Fusion. CLDF: Custom Layer based Deep Fusion. NTEF: Non-Tokenized Early Fusion. TEF: Tokenized Early Fusion.

Table 1: List of survey works related to recent multimodal developments.

Multimodal Survey Article	Year
MM-LLMs: Recent Advances in MultiModal LLMs Zhang et al. [2024a]	2024
A Survey on Multimodal Large Language Models Yin et al. [2024]	2024
The (R)Evolution of Multimodal LLMs: A Survey Caffagni et al. [2024]	2024
Large-scale Multi-modal Pre-trained Models: A Survey Wang et al. [2023b]	2023
Multimodal Large Language Models: A Survey Wu et al. [2023b]	2023
Multimodal Learning With Transformers: A Survey Xu et al. [2023]	2023
A Survey on Image-text Multimodal Models Guo et al. [2023]	2023

**Multimodal LLMs:** Zhang et al. [2024a], examines recent advancements in Multimodal Large Language Models (MLLMs), and explores NNs created by enhancing LLMs with mixed modality. It assesses the performance of mainstream MLLMs across 18 vision-language benchmarks. While it offers a comprehensive overview of the general architecture of MLLMs, it notably overlooks the critical inclusion of Type-D 3.4 multimodal model architecture. Type-D 3.4 is an emerging and popular multimodal model architecture type for developing any-to-any modality models. Yin et al. [2024], similar to Zhang et al. [2024a], offers insights into the intricate details of typical multimodal model architecture. However, it too lacks discussion on the Type-D multimodal architecture. It thoroughly presents details of pretraining, finetuning, and alignment methods, as well as the data used for Multimodal Large Language Models (MLLMs). Caffagni et al. [2024], shows a general multimodal model architecture, provides a comprehensive inventory of the components present in multimodal architectures, encompassing a diverse range of LLM variants, vision encoders, and vision-to-language connectors/adapters. It also compares these SOTA multimodal models on 14 multimodal benchmarks. Multimodal models tailored to specific domains, such as document understanding, medical vision learning, autonomous driving, and embodied AI, are discussed, but it too noticeably lacks information about Type-D multimodal architecture.

**Multimodal foundational models, training tasks, data and challenges:** Foundational multimodal models are explored in Wu et al. [2023b]. It extensively describes multimodal tasks and its related datasets for training the foundational multimodal models. Wu et al. [2023b] lacks details about the architectures of these models. The survey work Wang et al. [2023b], contains details of the models till 2022 and reviews multimodal pretraining datasets, pretraining tasks, pretrained model architectures and downstream multimodal tasks. Since the work only explores models till 2022, it lacks details of multimodal model architectures, which are currently prevalent. Baltrušaitis et al. [2018], highlights challenges in multimodal machine learning. It enumerates five multimodal challenges: representation, translation, alignment, fusion, and co-learning. Discussion about different types of model architectures is absent in this work. Xu et al. [2023], analyzes transformer model architecture from the perspective of multimodal learning. It lists various self-attention variants for input multimodal data/embeddings fusion. Based on the pretraining loss function, it categorizes the pretraining tasks used for multimodal learning with transformers. Similar to Baltrušaitis et al. [2018], it highlights challenges of multimodality fusion and alignment. Xu et al. [2023] too fails to discuss about different types of multimodal model architectures. Guo et al. [2023], specifically talks about models for image and text modalities. It investigates developments that have directly or indirectly influenced the evolution of multimodal large language models. It lists various image-text multimodal tasks and matches the widely used state-of-the-art models (till October 2023) with each of these tasks. The multimodal tasks include image captioning, visual reasoning, visual grounding and text-to-image generation.

### 3 Multimodal Model Architectures: A Taxonomy

The fusion of multimodal inputs into the deep neural network through various methods results in a range of architectural configurations. This study analyzes model architectures with mixed modalities and categorizes them into four distinct types based on the fusion of modalities. Two overarching categories are discernible: *Deep Fusion*, wherein the fusion of modalities occurs within the internal layers of the model, and *Early Fusion*, characterized by the fusion of modalities at the model’s input. Within each category, we observe two primary clusters. In the domain of Deep Fusion, the integration of modalities with internal layers manifests in: Type-A which employs standard cross-attention

layers, and Type-B which utilizes custom-designed layers. Conversely, in the domain of Early Fusion, multimodal inputs take two principal forms: non-tokenized<sup>2</sup> multimodal inputs as Type C, and discretely tokenized multimodal inputs as Type-D. These inputs are directly supplied to the input of the transformer model for early fusion, which can be either a decoder-only or an encoder-decoder style. Therefore, we define four distinct types of multimodal model architectures within the current landscape of multimodal models: Type-A 3.1, Type-B 3.2, Type-C 3.3 and Type-D 3.4.

Section 3 comprehensively outlines each architecture type, including information on their training data and computational requirements. Figure 2 maps together the multimodal model architecture types and the corresponding SOTA multimodal models. Figure 1 depicts the timeline of multimodal model development. The 4 types, Type-A, Type-B, Type-C and Type-D are described in Section 3.1, 3.2, 3.3, 3.4 respectively. Advantages and disadvantages of each multimodal model architecture type is listed in Section 4.

### 3.1 Type-A: Standard Cross-Attention based Deep Fusion (SCDF)

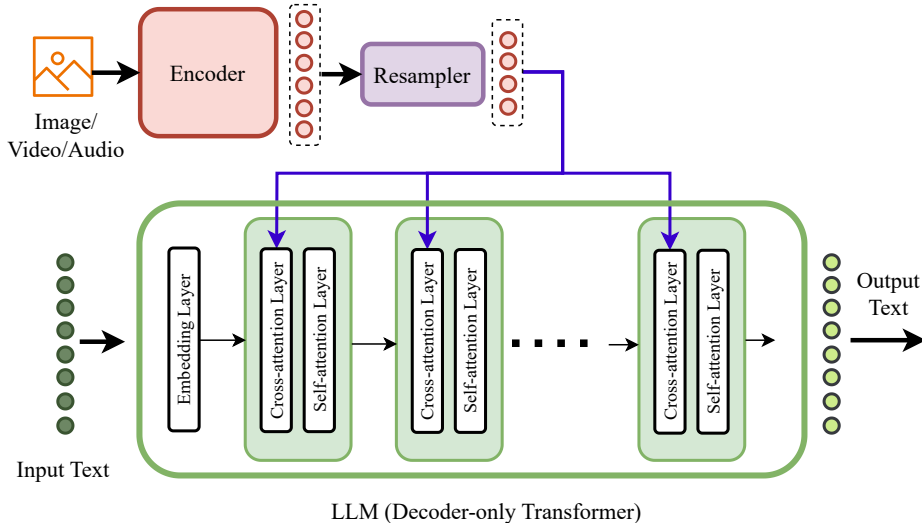


Figure 3: Type-A multimodal model architecture. The input modalities are deeply fused into the internal layers of the LLM using standard cross-attention layer. The cross-attention can be added either before (sub-type A.1) or after (sub-type A.2) the self-attention layer. Modality-specific encoders process the different input modalities. A resampler is used to output a fixed number of modality (visual/audio/video) tokens, given a variable number of input tokens at the input.

Type-A architecture mostly comprise of early multimodal models. Figure 3 shows a general Type-A model architecture. It typically involves a pre-trained LLM and integrating standard cross-attention layers into its internal architecture to achieve deep fusion of input modalities. The multimodal data (image/audio/video) is fed through modality specific encoders. A resampler is used to generate a fixed number of tokens that aligns with the requirements of the decoder layer. These resampler outputs are then directed to the internal layers of the LLM using cross-attention layers. The Type-A models exhibit a dichotomy in this aspect – the cross-attention layer can be added before or after the self-attention layer in the model’s architecture. This pre- and post-introduction of cross-attention w.r.t. self-attention in the LLM, results in the emergence of two distinct model architecture subtypes within Type-A. Section 3.1.1 and 3.1.2 details more about each sub-type. Models belonging to this architecture type include Flamingo Alayrac et al. [2022], OpenFlamingo (open-source replication of Flamingo) Awadalla et al. [2023], Otter (trained on MIMIC-IT dataset on top of OpenFlamingo) Li et al. [2023a], MultiModal-GPT (derived from OpenFlamingo) Gong et al. [2023], PaLI-X Chen et al. [2023b], IDEFICS (open-access reproduction of Flamingo) Laurençon et al. [2024a], Dolphins (based on OpenFlamingo architecture) Ma et al. [2023], VL-BART Cho et al. [2021] and VL-T5 Cho et al. [2021]. At present, models with this architecture commonly engage in processing image and

<sup>2</sup>No discrete tokenization

text modalities, subsequently producing textual outputs. Pretraining necessitates a substantial volume of data samples and computational resources, as observed in the resource-intensive implementations of Flamingo, OpenFlamingo, PaLI-X, and IDEFICS. Fine-tuning and/or instruction tuning these NNs can be achieved with minimal computational resources, a characteristic shared by the multimodal architectures discussed in this study. This trend is evident in the Otter, Multimodal-GPT, and Dolphins models, where fine-tuning or instruction tuning is exclusively performed using specific data, leveraging limited computational resources (typically less than or equal to 8 A100 GPUs). The comparative advantages and disadvantages of the Type-A multimodal model architecture, in relation to Types B, C, and D, are detailed in Section 4.1.

### 3.1.1 Subtype A.1

Figure 3 illustrates the model architecture sub-type, featuring the cross-attention layers *before* each self-attention layer within the decoder (LLM). Models belonging to this sub-group include Flamingo Alayrac et al. [2022], OpenFlamingo Awadalla et al. [2023], Otter Li et al. [2023a], MultiModal-GPT Gong et al. [2023], PaLI-X Chen et al. [2023b], IDEFICS Laurençon et al. [2024a] and Dolphins Ma et al. [2023]. Flamingo and its derivative multimodal models generally belong to this architectural sub-type.

**Training and data:** **Flamingo** model is trained with next-text-token prediction objective, where the input consists of interleaved image and text pairs and the model outputs text. For pretraining, it uses M3W Alayrac et al. [2022], ALIGN Jia et al. [2021], LTIP (Long Text & Image Pairs) Alayrac et al. [2022] and VTP (Video & Text Pairs) Alayrac et al. [2022]. For finetuning, VQAV2 Antol et al. [2015], COCO Chen et al. [2015], VATEX Wang et al. [2019], VizWiz Gurari et al. [2018], MSRVTQA Xu et al. [2017], VisDial Das et al. [2017], YouCook2 Zhou et al. [2018], and TextVQA Singh et al. [2019] datasets were used. **OpenFlamingo** models are also pretrained with the next-text-token prediction objective using 60M interleaved (MMC4 Zhu et al. [2024a]) examples and 120M LAION-2B Schuhmann et al. [2022] examples Awadalla et al. [2023]. It uses similar finetuning datasets as Flamingo. **Otter** created MIMIC-IT Li et al. [2023a] dataset, and trained OpenFlamingo model on it. This work improves OpenFlamingo’s instruction-following and in-context learning ability. **MultiModal-GPT** finetunes OpenFlamingo by adding LoRA Hu et al. [2021] weights to the LLM layers. This model too is trained using next-text-token prediction objective. Language training datasets include Dolly-15k and Alpaca-GPT4 Peng et al. [2023]. Vision-language datasets encompass A-OKVQA Schwenk et al. [2022], COCO Caption Karpathy and Fei-Fei [2015], OCR-VQA Mishra et al. [2019] and data from LLaVA Li et al. [2024a] & Mini-GPT4 Zhu et al. [2023a]. **PaLI-X** can process image, text and video inputs. Image and video inputs are encoded with ViT encoder. The text inputs and encoded image & video outputs are then processed by an encoder-decoder style transformer model. Most of the pretraining tasks (with the exception of the masked image token task) predict text-only output from the multimodal input Chen et al. [2023b]. Pretraining datasets include WebLI (image-text pairs) Chen et al. [2022], CC3M Sharma et al. [2018] and VTP (video data) datasets. Finetuning datasets include COCO (Karpathy split) Karpathy and Fei-Fei [2015], NoCaps Agrawal et al. [2019], VQAv2 Goyal et al. [2017], OKVQA Marino et al. [2019], TallyQA Acharya et al. [2019], VizWizCap Gurari et al. [2020], TextVQA Singh et al. [2019], STVQA Biten et al. [2019], OCRVQA Mishra et al. [2019], InfoVQA Mathew et al. [2022], DocVQA Mathew et al. [2021], ChartQA Masry et al. [2022], MSR-VTT Xu et al. [2016], Activity-Net Krishna et al. [2017a], VATEX Wang et al. [2019], SMIT Monfort et al. [2021], NExT-QA Xiao et al. [2021]. **IDEFICS** was trained on Wikipedia (Heafield [2011]; Laurençon et al. [2022]), Public Multimodal Dataset Singh et al. [2022], LAION Webster et al. [2023], and on a new 115B token dataset called OBELICS Laurençon et al. [2024a]. It follows Flamingo architecture style. LLaMA Touvron et al. [2023] is used as LLM and OpenClip as the vision encoder. **Dolphins** is based on OpenFlamingo. It is trained on driving data. This work utilizes BDD-X Kim et al. [2018] to establish instruction dataset, focusing on four key AV (autonomous vehicle) tasks like behavior comprehension, control signal forecasting, behavior analysis, and in-depth conversation Ma et al. [2023].

**Compute resources:** For **Flamingo**, all training and evaluations were performed on TPUv4 instances. The largest model containing 80 billion parameters was trained on 1536 chips for 15 days and sharded across 16 device Alayrac et al. [2022]. **OpenFlamingo** was trained using 64 A100 GPUs. While **Otter** only utilizes 1 A100 GPU or 4 RTX-3090 GPU, to instruction tune on MIMIC-IT Li et al. [2023a] dataset. The Otter models undergo solely instruction tuning without any pretraining process. **MultiModal-GPT** uses 8 A100 GPUs for finetuning training. The MultiModal-GPT model does

not undergo pretraining. For **PaLI-X**, no resource detail are provided. **IDEFICS**'s 9B-parameter models is trained on OBELICS-only Laurençon et al. [2024a] and LAION-only Schuhmann et al. [2022] datasets using 32 A100 (80GB) GPUs, and on OBELICS + LAION using 64 (80GB) A100s, for approximately 6 days. IDEFICS's large model is trained using 512 80GB A100 GPUs Laurençon et al. [2024a]. **Dolphins** is fully trained using 4 NVIDIA A100 GPUs.

### 3.1.2 Subtype A.2

This sub-type consists of a standard encoder-decoder transformer architecture, featuring a cross-attention layer placed *after* each self-attention layer within the decoder. Generally, language models follow a decoder-only structure. However, BART and T5 adopt an encoder-decoder style transformer architecture. Unlike sub-type A.1, sub-type A.2 does not utilize resampler in its architecture. Research work Cho et al. [2021], extended these language models to build mixed modality models named VL-BART and VL-T5.

**Training and data:** VL-T5 and VL-BART are designed to handle input comprising of images and text, and produces textual outputs. Visual embeddings from image are obtained using a CNN model. The image embeddings and text tokens are given as input to the encoder of the BART/T5. The model outputs text based on the task mentioned in the text input. A standard next-text-token prediction objective is used for training. Pretraining data includes, MS-COCO (Lin et al. [2014]; Chen et al. [2015]), Visual Genome Krishna et al. [2017b], VQA-v2.0 Goyal et al. [2017], GQA balanced version Hudson and Manning [2019], and Visual7W Zhu et al. [2016]. Finetuning data encompasses VQA Goyal et al. [2017], GQA, NLVR<sup>2</sup> Suhr et al. [2018], RefCOCOg Mao et al. [2016], COCO Caption Karpathy Karpathy and Fei-Fei [2015] and Multi30K Elliott et al. [2016].

**Compute resources:** VL-T5 and VL-BART are pretrained on 4 RTX 2080 Ti GPUs, for 4 days. Total 30 training epochs and batch size of 320 and 600 were used for VL-T5 and VL-BART respectively.

### 3.2 Type-B: Custom Layer based Deep Fusion (CLDF)

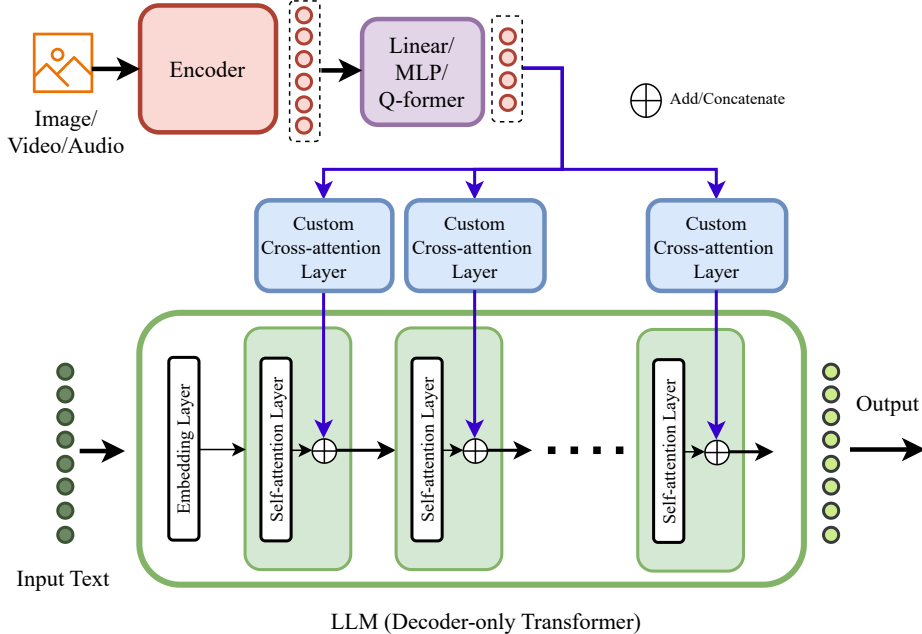


Figure 4: Type-B multimodal model architecture. The input modalities are deeply fused into the internal layers of the LLM using custom-designed layers. Custom cross-attention layers (sub-type A.1) or other custom layers (sub-type A.2) are used for modality fusion. A Linear Layer/MLP/Q-former is used to align different modalities with the decoder layer.

Type-B architecture is built using a pretrained LLM, learnable linear layer/MLP/Q-former, custom-cross-attention-layers or custom-layers and modality encoders. The difference between Type-A and

Type-B architecture types is that, *in Type-A, a standard cross-attention layer is utilized, while in Type-B, a custom-designed layer is or can be used.* Similar to Type-A, in Type-B the input modalities are deeply fused into the internal layers of the model. Example, LLaMA-Adapter-V2 model adds learnable embeddings to the output of the cross-attention layer before adding/concatenating the cross-attention output to the self-attention layer output. Additionally, a learnable gating factor is included to control the contribution of the cross-attention layer on the output of self-attention layer, which is not typically done in Type-A architecture. Models belonging to Type-B include LLaMA-Adapter Zhang et al. [2023a], LLaMA-Adapter-V2 Gao et al. [2023], CogVLM Wang et al. [2023a], mPLUG-Owl2 Ye et al. [2023a], CogAgent Hong et al. [2023], InternVL Chen et al. [2023a], MM-Interleaved Tian et al. [2024], CogCoM Qi et al. [2024], InternLM-XComposer2 Dong et al. [2024], MoE-LLaVA Lin et al. [2024], and LION Chen et al. [2023c]. Figure 4 shows general Type-B multimodal model architecture. Two architecture sub-types exists for Type-B. Sub-type B.1 3.2.1 adds custom-cross-attention layer to the internal layers of the LLM, while the sub-type B.2 3.2.2 uses custom learnable layer other than cross-attention layer. Sections 3.2.1 and 3.2.2 provide more details about sub-type B.1 and B.2 respectively. Models with multimodal input and text output dominate the Type-B architecture, similar to Type-A. If a Mixture-of-Experts layer (Jacobs et al. [1991]; Eigen et al. [2013]) is used in the architecture, an auxiliary loss is added to the standard auto-regressive loss (next-text-token prediction task) for load balancing. The comparative advantages and disadvantages of the Type-B multimodal model architecture, in relation to Types A, C, and D, are detailed in Section4.2.

### 3.2.1 Sub-type B.1: Custom Cross-Attention Layer

Here, the multimodal models are built through the process of using pretrained LLM and adding a *custom cross-attention layer* to the internal layers of the decoder. Models include LLaMA-Adapter Zhang et al. [2023a], LLaMA-Adapter-v2 Gao et al. [2023], CogVLM Wang et al. [2023a], mPLUG-Owl2 Ye et al. [2023a], CogAgent Hong et al. [2023], InternVL Chen et al. [2023a], MM-Interleaved Tian et al. [2024], and CogCoM Qi et al. [2024].

**CogVLM** learns separate query (Q), key (K), and value (V) embeddings for text and images in each decoder layer. These Q, K, V are initialized to same value as LLM at start of the training. A visual expert module processes the encoder outputs, and then inputs it to the custom cross-attention layers in the decoder (LLM). While **mPLUG-Owl2**, introduces a custom-cross-attention layer in which a common Q is learnt for separate K and V for each modality. This custom-cross-attention layer is called as ‘Modality adaptive module’ in this architecture. **MM-Interleaved**, introduces a custom-cross-attention layer called as MMFS (Multi-scale Multi-image Feature Synchronizer). This feature synchronizer is added after self-attention in each layer of the LLM. In **LLaMA-Adapter** and **LLaMA-Adapter-V2**, a custom-cross-attention layer called as ‘adapter’ is used. The Q in this cross-attention layer are learnable embeddings, and K, V are from the encoder outputs (after linear layer). Additional learnable embeddings are added to the output of the cross-attention before sending it to the decoder layer. This custom-cross-attention layer output is concatenated to the output of decoder layer.

**Training and data:** **LLaMA-Adapter** Zhang et al. [2023a] adds learnable prompts to layers of LLM for instruction following and multimodal learning. It uses standard next-text-token prediction objective for instruction tuning. The input consists of both text and image modalities, whereas the output is limited to text only. Dataset of size 52K samples is used to instruction tune LLaMA-Adapter. **LLaMA-Adapter-V2** Gao et al. [2023] further extends LLaMA-Adapter to create a visual instruction model. The input comprises text and images, while the output consists solely of text. This model too is trained on standard next-text-token prediction task. It uses 52K language instruction data Peng et al. [2023] & 567K image-text captioning data from COCO caption dataset Chen et al. [2015] and 80K conversation data collected by ShareGPT ShareGPT [2023] for finetuning/instruction tuning. In **CogVLM**, an additional attention layer and FFN (Linear Layer/s) are added in parallel to the LLM’s self-attention and FFN layer, for learning image features. Model takes image and text as input, and outputs text. Since output is text-only, this model too uses standard next-text-token prediction task for training. Pretraining data include LAION-2B, COYO-700M Byeon et al. [2022], LAION-115M Li et al. [2023b] and a subset of LAION-400M is used. Instruction tuning uses VQA<sub>v2</sub>, OKVQA, TextVQA, OCRVQA, ScienceQA Lu et al. [2022b], LLaVA-Instruct Liu et al. [2024a], LRV-Instruction Liu et al. [2023a], LLaVAR Zhang et al. [2024b], Flickr30K Entities Plummer et al. [2015], Ref-COCO Kazemzadeh et al. [2014], Visual7W, VisualGenome and Grounded CoT-VQA



Chen et al. [2023d] datasets. **mPLUG-Owl2** accepts both image and text inputs, producing text as output. Training objective is the standard next-text-token prediction. Pretraining datasets used are CC3M/CC12M Changpinyo et al. [2021], COCO Lin et al. [2014], Laion-en Schuhmann et al. [2022], COYO Byeon et al. [2022], DataComp Gadre et al. [2024]. Visual Instruction tuning datasets include TextCaps Sidorov et al. [2020], COCO, VQAv2, OKVQA, OCR-VQA, GQA, and A-OKVQA, Ref-COCO, VisualGenome, LLaVA-instruct-150K Liu et al. [2024a], ShareGPT-80K ShareGPT [2023], SlimOrca Lian et al. [2023]. **CogAgent** is trained using standard next-text-token prediction objective. It processes image and text as input, and outputs text. It is pretrained using datasets like LAION-2B, COYO, LAION-115M. To train model for GUI grounding, CogAgent created CCS400K (Common Crawl Screenshot 400K) dataset Hong et al. [2023]. Finetuning and alignment tuning uses Mind2Web Deng et al. [2024] and AITW Rawles et al. [2024]. **InternVL**, similar to other models in this architecture type, takes image and text as input and outputs text. Training progresses through three stages Chen et al. [2023a]. In stage 1, contrastive training is utilized to construct a 6 billion parameter vision model. In stage 2, a standard generative training is used with InternViT and frozen QLLaMA for image captioning task. In stage 3, supervised finetuning is done for visual QA and multimodal dialogue task. Stage 1 and 2 uses LAION-en, LAION-multi Schuhmann et al. [2022], COYO, Wukong Gu et al. [2022], LAION-COCO Schuhmann et al. [2022], LAION-en, CC12M, CC3M, SBU Ordonez et al. [2011] datasets. While stage 3 uses COCO Caption, TextCaps, VQAv2, OKVQA, A-OKVQA, IconQA Lu et al. [2021a], AI2D Kembhavi et al. [2016], GQA, OCR-VQA, ChartQA, DocVQA, ST-VQA, EST-VQA Wang et al. [2020a], InfoVQA, LLaVAR, Toloka Ustalov et al. [2023], LLaVA-150K, SVIT Zhao et al. [2023a], VisDial Das et al. [2017], LRV-Instruction, LLaVA-Mix-665K Liu et al. [2023b] datasets. **MM-Interleaved** model is trained end-to-end with next-text-token and next-image prediction task Tian et al. [2024]. The architecture is made up of visual encoder, resampler, LLM, feature-synchronizer and a diffusion model. Image and text are given as input, and model can output both image and text. Model is pretrained on a mixture of image-text pairs and interleaved image-text sequences, including MMC4, LAION-2B, LAION-COCO, CC-12M and Objects365 Shao et al. [2019]. Finetuning data include LLaVA-Mix-665K, COCO Caption, VQAv2, ChartQA, DocVQA, EST-VQA, InfoVQA, STVQA, TextCaps, LLaVAR, OCR-VQA, and DVQA, RefCOCO, RefCOCO+ Mao et al. [2016], and RefCOCOg Mao et al. [2016].

**Compute resources:** LLaMA-Adapter uses 8 A100 GPUs. Only 1 hour of training required on 52K language instruction following dataset using LLaMA-7b. LLaMA-Adapter-V2, since derived from LLaMA-Adapter, it too is trained using 8 A100 GPUs. Training time varies due to difference in training data sizes. CogVLM, mPLUG-Owl2, CogAgent and MM-Interleaved did not explicitly include details about training resources in their work. InternVL uses 640 A100 GPUs for stage 1 training, 160 A100 GPUs for stage 2 training and 8 to 32 A100 GPUs for stage 3 training.

### 3.2.2 Sub-type B.2: Custom Learnable Layer

Instead of adding custom cross-attention layers to internal LLM layers, models using *custom learnable layers* belong to sub-type B.2 of Type-B multimodal model architecture. Models include InternLM-XComposer2 Dong et al. [2024], MoE-LLaVA Lin et al. [2024] and LION Chen et al. [2023c]. **InternLM-XComposer2** adds LoRA weights to LLM layers for learning image modality. The LoRA weights are added in parallel to each decoder layer and only processes image tokens. No cross-attention layer is added in the decoder layers to build this model. **MoE-LLaVA** model is built on top of LLaVA multimodal model (LLaVA model belongs to Type-C). In addition to the changes that exists in LLaVA model, in MoE-LLaVA each decoder layer of LLaVA is modified to create MoE-LLaVA model. In each decoder layer, the FFN layer is modified to create Mixture-of-Expert (MoE) layer. MoE layer consists of a router and multiple FFN layers in parallel. **LION** modifies the FFN layer of each decoder layer. The learnable module 'Mixture of adapters with routers' is added in parallel to the FFN layers. This module consists of LoRA and MoE layer for learning image modality features.

**Training and data:** **InternLM-XComposer2** can process image and text as input, and outputs text. The model is trained using a standard next-text token prediction task. Its training process involves pretraining, finetuning and instruction tuning. Pretraining datasets include ShareGPT4V-PT Chen et al. [2023e], COCO, Nocaps, TextCaps, LAION-400M, SBU, CC-3M, Concept Data Zhang et al. [2023b], WanJuan He et al. [2023], Flicker, MMC-Instruction Liu et al. [2023c]. Finetuning data encompasses ShareGPT4V, COCO, Nocaps, VQAv2, GQA, OK-VQA, Science QA, AI2D, SQA, DVQA, ChartQA, MathQA Amini et al. [2019], Geometry3K Lu et al. [2021b], A-OKVQA,

KVQA Shah et al. [2019], LLaVA-150k, LVIS-Instruct4V Wang et al. [2023c]. Instruction tuning data include LLaVA-150k, LVIS-Instruct4V, ShareGPT-en&zh Chiang et al. [2023], InternLM-Chat Team [2023]. **MoE-LLaVA** too processes image and text as input, and outputs text. An auxiliary loss related to the mixture-of-experts load balancing is added to the standard auto-regressive loss. Training consists of three stages: one pretraining and two finetuning stages. Data used for pretraining is LLaVA 1.5-558k Liu et al. [2023b] dataset. First finetuning stage uses SViT-157k Zhao et al. [2023a], LVIS-220k, LRV-331k Liu et al. [2023a], and MIMIC-IT-256k datasets. Second finetuning stage uses LLaVA 1.5-mix-665k dataset. **LION** receives input in the form of images and text, and produces text as its output. It uses standard next-text-token prediction objective for training. LoRA weights and MoE layer are added to LLM layers for learning image modality. Training data includes LLaVA-Instruct-150K, OKVQA, A-OKVQA, VQAv2, OCR-VQA, COCO, TextCaps and Visual Genome datasets.

**Compute resources:** All MoE-LLaVA training steps use 8 A800 (80G) GPUs. InternLM-XComposer2 and LION does not explicitly provide resource details in their work.

### 3.3 Type-C: Non-Tokenized Early Fusion (NTEF)

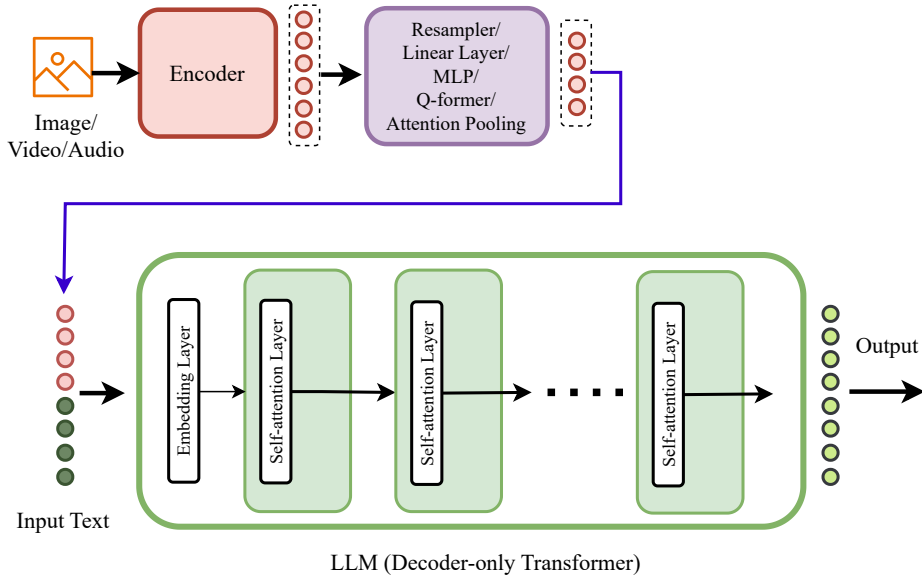


Figure 5: Type-C multimodal model architecture. The (non-tokenized) input modalities are directly fed to the model at its input, rather than to its internal layers, resulting in early fusion. Different types of modules are used to connect modality encoder outputs to the LLM (model) like a Linear-Layer/MLP (sub-type C.1), Q-former and a Linear-Layer/MLP (sub-type C.2), Perceiver resampler (sub-type C.3), Custom learnable layers (sub-type C.4).

Type-C stands out as the most widely adopted multimodal model architecture. Figure 5 depicts a general Type-C multimodal model architecture. The modular nature of this architecture type contributes to its simplicity in both construction and training. Differing from Type-A and Type-B, in Type-C and Type-D architectures, the modality encoder output is solely directed and fused at the input of the model, without involvement in the internal layers of the model. Hence, Type-C belongs to early fusion category (Figure 2). Pretrained LLM as decoder is used without any major architectural changes to its internal layers (some models belonging to this type may have LoRA weights added to their decoder layers). Pretrained image encoder or other modality encoder are used. The encoder/s and the decoder are combined together with learnable module like single Linear-Layer, MLP, Q-former, attention-pooling layer, convolutional layer, perceiver resampler or variants of Q-former. Q-former, first introduced in BLIP-2 Li et al. [2023b], is a lightweight transformer architecture consisting of two components. One utilizes learnable query embeddings and cross-attention for images, while the other employs self-attention for text processing. Both components share a self-attention layer, enhancing efficiency in multimodal tasks. The Perceiver resampler, similar to Q-former, utilizes

learnable queries and cross-attention mechanisms to process image data, generating a fixed number of visual tokens. However, unlike Q-former, it does not incorporate self-attention layers specifically tailored for text processing. Unlike Type-A and B, in Type-C, no major internal model architectural changes are added to either encoder or decoder. This allows the multimodal model belonging to Type-C, to incorporate off-the-shelf LLMs and encoders in their architecture. A substantial number of models fall under Type-C. Thus, for clarity, our study categorizes them according to the type of connectors utilized to link the modality encoder and LLM. Section 3.3.1, 3.3.2, 3.3.3 and 3.3.4 lists and describes the sub-types/categories. The comparative advantages and disadvantages of the Type-C multimodal model architecture, in relation to Types A, B, and D, are detailed in Section 4.3.

### 3.3.1 Sub-type C.1: Linear Layer/MLP

**Models using only Linear Layer/MLP for connecting Encoder to the LLM (decoder):** DeepSeek-VL Lu et al. [2024], LLaVA Liu et al. [2024a], LLaVA-Med Li et al. [2024a], LLaVAR Zhang et al. [2024b], LLaVA-1.5 Liu et al. [2023b], LLaVA-Phi Zhu et al. [2024b], LLaVA-NeXT Liu et al. [2024b], PaLM-E Driess et al. [2023], MiniGPT-v2 Chen et al. [2023f], DetGPT Pi et al. [2023], PandaGPT Su et al. [2023], GILL Koh et al. [2024], Shikra Chen et al. [2023d], GPT4RoI Zhang et al. [2023c], ChatSpot Zhao et al. [2023b], NExT-GPT Wu et al. [2023a], Fuyu Bavishi et al. [2023], FROMAGe Koh et al. [2023], ShareGPT4V-7B Chen et al. [2023e], GroundingGPT Li et al. [2024b], ModaVerse Wang et al. [2024a], MLLM-Tool Wang et al. [2024b], ViGoR Yan et al. [2024], CoDi Tang et al. [2024], CoDi-2 Tang et al. [2023a].

### 3.3.2 Sub-type C.2: Q-former and Linear Layer/MLP

**Models using Q-former and Linear Layer/MLP for connecting Encoder to the LLM (decoder):** BLIP-2 Li et al. [2023b], MiniGPT-4 Zhu et al. [2023a], X-LLM Chen et al. [2023g], InstructBLIP Dai et al. [2024], EMU Sun et al. [2023], Video-LLaMA Zhang et al. [2023d], BLIVA Hu et al. [2024], SALMONN Tang et al. [2023b], X-InstructBLIP Panagopoulou et al. [2023], BuboGPT Zhao et al. [2023c], VILA Lin et al. [2023a], TinyGPT-V Yuan et al. [2023a], SPHINX Lin et al. [2023b], SPHINX-X Gao et al. [2024].

### 3.3.3 Sub-type C.3: Perceiver Resampler

**Models using Perceiver resampler for connecting Encoder to the LLM (decoder):** Idefics2 Laurençon et al. [2024b], InternLM-XComposer Zhang et al. [2023b], KOSMOS-2.5 Lv et al. [2023], Monkey Li et al. [2023c], V\* Wu and Xie [2023], Kosmos-G Pan et al. [2024].

### 3.3.4 Sub-type C.4: Custom Learnable layer

**Models using custom-module/layer for connecting Encoder to the LLM (decoder):** MM1 McKinzie et al. [2024], mPLUG-Owl Ye et al. [2024], mPLUG-DocOwl Ye et al. [2023b], EmbodiedGPT Mu et al. [2024], Video-ChatGPT Maaz et al. [2023], Qwen-VL Bai et al. [2023], AnyMAL Moon et al. [2023], DocPedia Feng et al. [2023], mPLUG-PaperOwl Hu et al. [2023], Osprey Yuan et al. [2023b], EMU2 Sun et al. [2023], KAM-CoT Mondal et al. [2024], VisLingInstruct Zhu et al. [2024c], MobileVLM Chu et al. [2023], MobileVLM-V2 Chu et al. [2024].

### 3.3.5 Training Methods and Data

Many recent multimodal survey works like Zhang et al. [2024a], Yin et al. [2024], Guo et al. [2023], Caffagni et al. [2024] and Wu et al. [2023b] have explored Type-C multimodal model architecture in great detail. This section will highlight some important details provided in these work here, that are related to model training strategies, data and resources.

**Training and data:** Three stages of training used for Type-C multimodal model architecture are pre-training, instruction-tuning, and alignment tuning. General training procedure (each model may have some variations): Step 1 (Pretraining): Freeze LLM and Encoder. Only train the projection layer/s for Vision-Language alignment. Step 2 (Instruction and alignment tuning): Train projection layer and LLM for multimodal tasks. The encoder is trained optionally if required. But in general, only projection layer and LLM are trained. Pretraining mainly aims to align different modalities and learn multimodal world knowledge typically through caption data (text) for images, audio and videos Yin et al. [2024]. Since the output modality is text, standard next-text-token prediction objective is used

during pretraining. Table 2 lists pretraining data commonly used for Type-C architecture. It contains captioning data for images, audio and video modalities. LLMs and Multimodal-LLMs are being widely deployed in real-world applications, specially in chat-based applications (chat-bots). These applications require the model to understand user query (instruction) and with a greater attention on details provided by the user. In order to better align model for query (instruction) understanding, models are trained on instruction following datasets. The training process is called as instruction tuning. Instruction following datasets contain wide variety of tasks, hence making model more versatile across different tasks. This training process not only increases the instruction following capability, but also leads to improved few-shot and zero-shot performances. The survey works Zhang et al. [2024a] and Yin et al. [2024] lists different ways used to instruction tune a multimodal model of Type-C. Table 3 shows the datasets used for instruction tuning of Type-C models. To further align model for human interactions (chat applications), the model is trained using RLHF (Christiano et al. [2017]; Ziegler et al. [2019]; Stiennon et al. [2020]; Bai et al. [2022]; Ouyang et al. [2022]) or DPO Rafailov et al. [2024] training strategies using human preference data. Table 4 enumerates datasets used for model alignment.

**Compute resources:** Type-C multimodal model architecture is data and compute efficient. Hence, low training resources are required compared to all other types.

**LLaVA** use 8 A100s. Pretrained for 4 hours and finetuned within 10 hours. **LLaVA-Med** takes 7 and 8 hours for stage 1 and 2 training on 8 (40G) A100 GPUs. **LLaVAR** all experiments are run on NVIDIA A100 (80GB) GPUs. **LLaVA-1.5** full training completed within 1 day on a single 8-A100 node. **LLaVA-Phi** uses 8 A100 GPUs. Pretrained for 1.5 hours, 8 hours for visual instruction tuning. **LLaVA-NeXT** full trains within approximately 1 day with 32 A100s GPUs. **MiniGPT-v2** stage 1 training requires 8 A100 GPU for around 90 hours. Second stage is trained on 4 A100 GPU for roughly 20 hours. Last stage, training is executed on 4 A100 GPUs for around 7 hours. **PandaGPT** uses 8 A100 (40G) GPUs for around 7 hours for training. GILL training utilizes 2 A6000 GPUs for 2 days. **Shikra** model’s all training runs on 8 NVIDIA A100 GPUs. It takes around 100 hours for stage one training and 20 hours for stage two. All **FROMAGE** training steps are completed within 1 day (24 hours) using a single A6000 GPU. **DeepSeek-VL** 7B model utilized a cluster of 64 nodes, each comprising 8 Nvidia A100 GPUs for 5 days, while DeepSeek-VL-1B consumed 7 days with 16 nodes for all training steps Lu et al. [2024]. **ModaVerse** model’s full training completed in 20 hours on 4 A100 GPUs.

**BLIP-2** uses 16 A100 (40G) GPUs. In **MiniGPT-4**, stage 1 vision-language alignment training uses 4 A100 GPUs for 10 hours. And stage 2 finetuning takes 7 minutes on single A100 GPU. All **X-LLM** experiments use up to 8 A100 (40G) GPUs. **InstructBLIP** models are trained utilizing 16 Nvidia A100 (40G) GPUs for 1.5 days. **EMU** pretraining uses 128 NVIDIA A100 (80G) GPUs. While, instruction tuning stage utilizes 16 A100 (80G) GPUs. **X-InstructBLIP** requires 8 A100 (40GB) GPUs. **TinyGPT-V** can be trained on a 24GB memory GPU. Phi-2 and CLIP are used as LLM and vision encoder. **SPHINX** pre-trains for 125 hours on 32 A100 GPUs with a 7B language model and about takes twice amount of time for 13B language model. Fine-tuning takes about 38 hours with 16 A100 GPUs with a 13B language model.

**InternLM-XComposer** uses 128 Nvidia A100 GPUs. **Monkey** model’s whole training process takes 40 A800 days for one epoch. For **KOSMOS-G**, the training process took around 4 days with 256 NVIDIA V100 GPUs. No resource details are provided in KOSMOS-2.5, V\* and idefics2 works.

**Video-ChatGPT** 7B model training completed in 3 hours on 8 A100 (40GB) GPUs. **AnyMAL** was able to train a 70B model on a single A100 (80GB) VRAM GPU through quantization. Other models used a varying number of Nvidia A100 GPUs. **DocPedia** used 8 A100 GPUs for training. **mPLUG-PaperOwl** model is trained equivalent to costing 64 A100 days. **Osprey**’s training is conducted on 4 NVIDIA A100 GPUs with 80GB memory. In **MobileVLM**, first the language model pretraining is completed using 20 nodes equipped with 8 NVIDIA Tesla A100 GPUs each. Later, vision-language training is done in 5 hours with 8 NVIDIA Tesla A100 GPUs for MobileVLM 1.7B, and 8 hours for MobileVLM 3B. **MobileVLM-v2**, trained on top of MobileVLM, is pretrained on 8 NVIDIA A100 GPUs for about 5 hour. Finetuning is performed with 8 NVIDIA A100 GPUs for around 9 hours. Resource information are not clearly provided in MM1, mPLUG-Owl, mPLUG-DocOwl, EmbodiedGPT, Qwen-VL and EMU2.

Table 2: Pretraining data for Type-C. Yin et al. [2024], Wu et al. [2023b], Zhang et al. [2024c]

Dataset	Modality	Samples
ALLaVA	$I + T \rightarrow T$	709K
LVIS-Instruct4V	$I + T \rightarrow T$	111K
ShareGPT4V-PT	$I + T \rightarrow T$	1.2M
COYO-700M	$I + T \rightarrow T$	747M
LAION-COCO	$I + T \rightarrow T$	600M
LAION-2B	$I + T \rightarrow T$	2.3B
LAION-5B	$I + T \rightarrow T$	5.9B
CC-12M	$I + T \rightarrow T$	12.4M
CC-3M	$I + T \rightarrow T$	3.3M
SBU Captions	$I + T \rightarrow T$	1M
VTP	$V + T \rightarrow T$	27M
WebVid2M	$V + T \rightarrow T$	2.5M
YouCook2	$V + T \rightarrow T$	2.2K
MSR-VTT	$V + T \rightarrow T$	200K
AISHELL-1	$A + T \rightarrow T$	128K
AISHELL-2	$A + T \rightarrow T$	1M
WavCaps	$A + T \rightarrow T$	24K
Common Voice	$A + T \rightarrow T$	9.2K
LibriSpeech	$A + T \rightarrow T$	1K
MM5Product	$I + A + V + T$	6M
MSR-VTT	$I + A + V + T$	10K

Table 3: Instruction tuning data for Type-C. Zhang et al. [2024a], Yin et al. [2024]

Dataset	Modality	Samples
LLaVA-Instruct	$I + T \rightarrow T$	158K
LVIS-Instruct	$I + T \rightarrow T$	220K
ALLaVA	$I + T \rightarrow T$	1.4M
MIMIC-IT	$I/V + T \rightarrow T$	2.8M
Video-ChatGPT	$V + T \rightarrow T$	100K
VideoChat	$V + T \rightarrow T$	11K
Clotho-Detail	$A + T \rightarrow T$	3.9K
BuboGPT’s IT	$(I + A)/A + T \rightarrow T$	9K
T2M	$T \rightarrow I/V/A + T$	14.7K
MosIT	$I + V + A + T \rightarrow I + V + A + T$	5K

Table 4: Alignment tuning data for Type-C. Zhang et al. [2024a]

Dataset	Modality	Samples
VLGuard’s IT	$I + T \rightarrow T$	3K
RTVLM	$I + T \rightarrow T$	5K
MMViG	$I + T \rightarrow T$	16K
LLaVA-RLHF	$I + T \rightarrow T$	10K
RLHF-V’s IT	$I + T \rightarrow T$	1.4K

### 3.4 Type-D: Tokenized Early Fusion (TEF)

In Type-D, multimodal inputs are tokenized using a common tokenizer or modality specific tokenizers. The tokenized inputs are then given to a pretrained LLM 3.4.1 or an encoder-decoder transformer model 3.4.2, which generates multimodal outputs. Figure 6 represents a general Type-D multimodal

model architecture. Either pretrained modality specific tokenizers are used, or a tokenizer training stage is included in the training process. The fundamental advantage of tokenizing the inputs is that, the model now can be trained auto-regressively to generate image, audio and different modality tokens along with text tokens. Models belonging to Type-D include, LaVIT Jin et al. [2024], TEAL Yang et al. [2024], CM3Leon Yu et al. [2023], VL-GPT Zhu et al. [2023b], Unicode Zheng et al. [2024], SEED Ge et al. [2023], 4M Mizrahi et al. [2024], Unified-IO Lu et al. [2022a], Unified-IO-2 Lu et al. [2023]. The comparative advantages and disadvantages of the Type-D multimodal model architecture, in relation to Types A, B, and C, are detailed in Section 4.4.

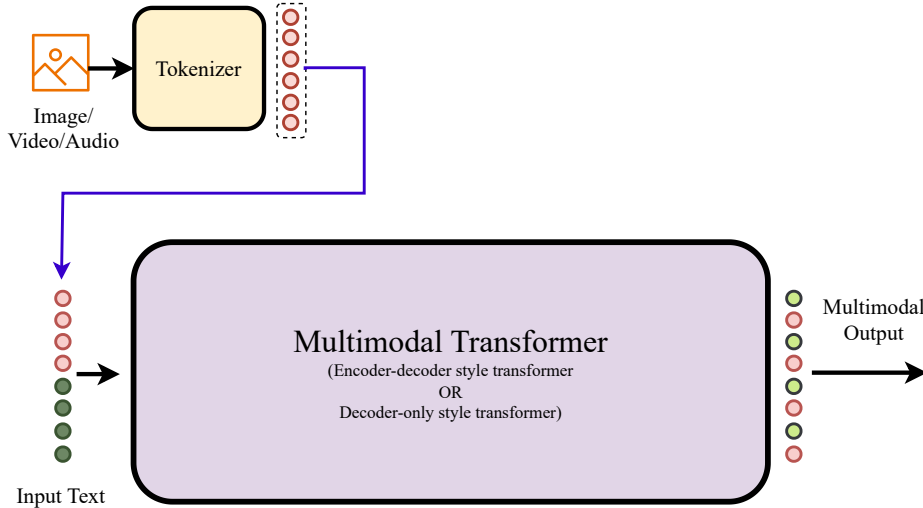


Figure 6: Type-D multimodal model architecture. The tokenized input modalities are directly fed into the model at its input. Either a decoder-only transformer (sub-type D.1) or an encoder-decoder style transformer (sub-type D.2) is used as the multimodal transformer in this architecture.

### 3.4.1 Subtype D.1: Models using LLM

Models that primarily use LLM are LaVIT, TEAL, CM3Leon, SEED, Unicode, VL-GPT. **LaVIT** aims at unified generative training for image and text modalities. It is achieved using a visual tokenizer. After image encoding using an image encoder, a visual tokenizer is used in LaVIT model architecture to tokenize the visual inputs. **TEAL**, tokenizes all modalities. It has a tokenizer and a detokenizer module in its architecture. A projection layer is used for connecting non-textual modalities to the LLM. **CM3Leon** uses image tokenizer to tokenize images and then directly provide it to the LLM. It uses OPT model as a LLM. In **VL-GPT**, a tokenizer is first trained to convert images to image tokens. Later, the trained tokenizer is used to feed the images to the LLM.

**Training and data:** **LaVIT** model is trained in two stages. In stage I, an image tokenizer is trained. In stage II, the main LaVIT model is trained. LaVIT utilizes general auto-regressive objective function where likelihood of each multi-modal sequence is directly maximized Jin et al. [2024]. Since both image and text are already represented as discrete tokens, the cross-entropy loss is used to supervise the token prediction at each location for both modalities (image and text) with a shared prediction head Jin et al. [2024]. **TEAL** employs pretrained tokenizers. During pretraining of TEAL, only the projection layers are trained. The LLM and the encoders are frozen for textual and non-textual embedding alignment during pretraining. It uses image-text and audio-text pairs for pretraining. Finetuning is done on downstream tasks. Datasets like COCO-Caption, Science-QA, and CoVoST 2 Wang et al. [2020b] are used. It uses general auto-regressive objective for training. **CM3Leon** too uses standard next-token prediction loss (auto-regressive). For pretraining, LAION and licensed images from shutterstock are used. Finetuning uses COCO Captioning (2015), Flickr30k, Image Paragraph Krause et al. [2017], Localized Narratives Pont-Tuset et al. [2020], VQA2, VizWiz, OKVQA, ScienceQA, and InstructPix2Pix Brooks et al. [2023] datasets. **VL-GPT**, also tokenizes the inputs, hence it too is able to use standard auto-regressive objective function for training. Pretraining involves utilizing both image-text pairs and interleaved image-text sequences. CC3M, LAION-Aesthetics, LAION-COCO, Multimodal-C4 (MMC4) and OBELICS datasets are used for pretraining.

Later, the model is instruction tuned using LLaVA data Liu et al. [2024a], SVIT, COCO Caption, InstructPix2Pix and Magicbrush Zhang et al. [2024d] datasets.

**Compute resources:** For LaVIT, 64 A100 GPUs and 12 hour training required to train tokenizer. 256 A100 GPUs and 36 hour training required for pretraining full LaVIT model. TEAL Yang et al. [2024] is fully trained with 8 A100 GPUs. CM3Leon Yu et al. [2023] was pretrained on 2 trillion tokens with 256 or 512 A100 GPUs and finetuned on 30 billion tokens with 64 or 128 A100 GPUs. In VL-GPT Zhu et al. [2023b], tokenizer training used 8 NVIDIA A100 (40G) GPUs for 10,000 iterations with batch size of 1024. Pretraining utilized 32 GPUs for 20,000 iterations with batch size of 4096 and instruction tuning was performed with 4 GPUs for 10,000 iterations using batch size of 512.

### 3.4.2 Subtype D.2: Models using Encoder-Decoder style Transformer

Models using encoder-decoder style transformer instead of LLM are Unified-IO, Unified-IO 2 and 4M. In **Unified-IO** Lu et al. [2022a] and **Unified-IO-2** Lu et al. [2023], an encoder-decoder style transformer is used. The input modalities are tokenized using VQ-GAN style tokenizers. To tokenize images and dense structures, VQ-GAN is employed. For audio, ViT-VQGAN is utilized. While these models may bear resemblance to Type-C, they diverge notably. The critical distinction lies in Type-D models’ utilization of discrete input and output modality-specific tokens. The **4M** model is a multimodal model capable of processing text, RGB images, depth, normals, semantic segmentation maps, and CLIP feature maps. In contrast to Unified-IO which uses VQ-GAN style tokenizers to tokenize all modalities, 4M utilizes modality specific tokenizers. Here, text is tokenized using WordPiece, and VQ-VAE is used to tokenize image and image-like modalities.

**Training and data:** **Unified-IO-2** is trained from scratch on a large multimodal pre-training data corpus from diverse sources with a multimodal Mixture of Denoisers (MoD) objective Lu et al. [2023]. UNIFIED-IO-2 contains 7 billion parameters and is pre-trained from scratch on an extensive variety of multimodal data – 1 billion image-text pairs, 1 trillion text tokens, 180 million video clips, 130 million interleaved image & text, 3 million 3D assets, and 1 million agent trajectories Lu et al. [2023]. The model is instruction-tuned with a massive multimodal data by combining more than 120 datasets covering 220 tasks across vision, language, audio, and action Lu et al. [2023]. **4M** uses MultiMAE Bachmann et al. [2022] pretraining strategy, where it takes small set of tokens from all modalities at its input, and performs cross-modal prediction coding. VQ-VAE is used for tokenizing image related modalities and WordPiece for text tokenization. Conceptual Captions 12M (CC12M) is used for pretraining. Finetuning datasets include ImageNet-21K, ImageNet-1K, COCO detection, ADE20K, and NYUv2.

**Compute resources:** 4M is pretrained on 500 billion tokens. Both pretraining and finetuning utilizes 64 or 128 A100 GPUs. For Unified-IO and Unified-IO-2, Google’s TPU are used. No other resource related details are provided in their work.

## 4 Advantages and Disadvantages

### 4.1 Type-A

Type-A (3.1) multimodal model architecture enables fine-grained control of how modality information flows in the model. It is end-to-end trainable and omits design of custom layers by using standard learnable layers of transformers. Example, standard cross-attention layer is used to fuse modalities in Type-A, while in Type-B (3.2), a specially designed layer is used to fuse modalities. Type-A requires large number of training data samples and computational resources as compared to Type-B and Type-C multimodal model architectures. Challenging to build model of this architecture type, due to the prerequisite understanding of the internal layers of the LLM. Architecture is difficult to scale as compared to Type-C (3.3), especially if pretraining step is involved, because of the large number of training parameters and computational requirements. Adding more modalities is challenging, because in Type-A, after adding image modality cross-attention layer to the LLM layer, adding other modalities to each LLM layer is difficult and has not been explored in current literature to best of our knowledge. Type-B addresses this challenge with a gating mechanism, which allows direct addition of input modalities to the output LLM layers. The gating mechanism involves a single learnable parameter, a multiplication and an addition operation. The input modality is multiplied with the learnable parameter determining the contribution of the modality, later, its output is directly added

to the LLM layer output. Number of trainable parameters can be large, due to addition of learnable cross-attention layer in each LLM layer. Unlike Type-D 3.4, which accommodate an autoregressive training objective across diverse modalities, Type-A encounter increased complexity when applying a standard autoregressive training objective to modalities beyond text.

## 4.2 Type-B

Similar to Type-A (3.1), Type-B (3.2) also benefits from fine-grained control of how modality information flows in the model. It is end-to-end trainable. In contrast to Type-A, custom-designed layers are used in Type-B. The custom design adds to more fine-grained control of modality fusion. Compared to Type-A, the efficient custom design of the layers and the architecture of the model mitigate the need for extensive training data samples and computational resources. Building these model require the knowledge of the internal layers of the LLM. In contrast to Type-A, Type-B architecture is more scalable, due to customizable nature and computational efficiency of the custom learnable connector layers. However, scaling may still present challenges in comparison to Type-C (3.3) architecture. Adding more modalities is simplified compared to Type-A. In Type-A, the addition of further modalities to each LLM layer becomes significantly more challenging after the inclusion of an image modality cross-attention layer, an area that remains underexplored in the literature. The Type-B, uniquely provides an alternative by introducing a gating mechanism which can be utilized to add other modalities. The gating mechanism enables direct addition of input modalities to the output LLM layers. Number of trainable parameters can be controlled by design of efficient custom connector layers, hence Type-B can be efficient in terms of number of trainable parameters. In contrast to Type-D 3.4, where an autoregressive training objective is readily applicable across various modalities, the implementation of a standard autoregressive training objective in Type-B for non-textual modalities presents greater complexity.

## 4.3 Type-C

Type-C 3.3 architecture is modular. Parts of the model architecture can be swapped and the resulting new model can trained efficiently for multimodal tasks. Unlike Type-A 3.1 and Type-B 3.2, Type-C does not have fine-grained control of how modality information flows in the model. Different modality inputs are fused only at the input of decoder (LLM). It is end-to-end trainable. Type-C requires less training data and computational resources as compared to Type-A, B and D 3.4 multimodal model architectures. It is easier to build compared to all other type of multimodal architectures, owing to its modular architecture. Unlike Type-A and Type-B architectures, Type-C models do not necessitate detailed knowledge of the internal layers of the LLM. Instead, only the interface details of the LLM or encoder being newly integrated are required. Type-C architecture is scalable due to its modular design, reduced training data requirements, and computational efficiency. Adding more modalities is easier in Type-C, compared to Type-A, B and D. A simple learnable Linear/MLP/Q-former/custom layer can be added between the modality encoder and the LLM and trained efficiently for augmenting different modalities. Number of trainable parameters is least in Type-C compared to Type-A, B and D. Hence, it is compute resource efficient from training perspective. Unlike Type-D, where an auto-regressive objective can be used to train different modalities, here in Type-C, it is challenging to utilize standard auto-regressive objective for modalities other than text (language). Type-C provides an alternate way to Type-D, for any-to-any multimodal model development due elimination of input modality tokenizers, its modular architecture, training efficiency and end-to-end trainable nature.

## 4.4 Type-D

Type-D 3.4 has a simplified model architecture due to tokenization of input and output modalities, when compared to Type-A 3.1, B 3.2 and C 3.3. It tokenizes all modalities. This characteristic can be perceived as both advantageous and disadvantageous. Tokenization offers the advantage of enabling all modality training through a standard auto-regressive objective function. However, the challenge lies in training a universal tokenizer or modality-specific tokenizers. In other architecture types, output embeddings from modality encoders are directly provided to the LLM without (discrete) tokenization. Unlike Type-A and Type-B, Type-D does not have fine-grained control of how modality information flows in the model. Different modality inputs are fused only at the input of the main transformer model. This main transformer can be a decoder (LLM) or an encoder-decoder style transformer. It is end-to-end trainable. It requires large training data and computational resources as compared to



Type-A, B and C multimodal model architectures. Type-D model architecture are comparatively easier to construct than Type-A and Type-B models but is more complex to build compared to Type-C architecture type. Type-D architecture is scalable, due to tokenization of modalities. Incorporating additional modalities poses challenges, as it necessitates training a new tokenizer for each new modality or adapting an existing multimodal tokenizer, which can be a non-trivial undertaking. All the modalities are learnt by the main transformer, and to add another modality or task, the model has to be trained, and efficient training strategies for this type of model architecture have not been extensively investigated in the literature, to the best of our knowledge. Adding additional modalities to the model architecture is simplest in the Type-C, when compared to Type-A, B and D. Type-D has large number of trainable parameters compared to Type-A, B and C. Since the LLM or transformer has to learn new modality tokens, the model has to be trained, and training in Type-D is computationally intensive. In Type-D, an auto-regressive objective can be used to train different (all) modalities. Like Type-C, Type-D provides a way for any-to-any multimodal model development, enabled by its tokenization of modalities and simplified training objective.

## 5 Next Generation Multimodal Architectures

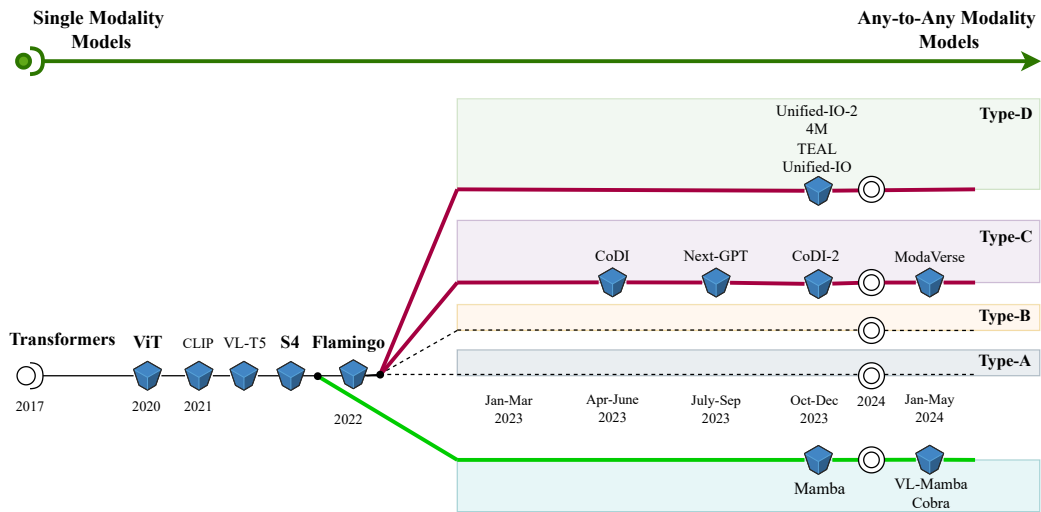


Figure 7: Any-to-any Multimodal Model development timeline. The evolution from single modality models (left) to any-to-any modality models (right) is depicted. Any-to-any multimodal models belonging to Type-C and Type-D are noted in the figure. An alternate development timeline (green line at the bottom) for non-transformer based models like SSM (State-space models) is shown. Mamba is a language model. VL-mamba and Cobra are vision-language models.

This section explores multimodal models with multimodal-input and multimodal-output. Plethora of models exist for any-input-modality to output-text-modality. In contrast, there are significantly fewer multimodal models capable of generating output modalities other than text. Multimodal output generation is one of the primary challenge in the multimodal domain. Type-C and Type-D multimodal architectures are at the forefront of development for any-to-any multimodal models. The representative models are highlighted in Figure 7. These dominant multimodal model architectures address some, though not all, challenging aspects of multimodal generation. Type-D simplifies the training process by utilizing input tokenization, enabling the use of a standard auto-regressive objective function for model training. However, it still faces limitations in addressing the challenge of accommodating large data sizes and computational demands necessary for building multimodal generative models. Type-C tackles the challenges associated with data and resources by leveraging pretrained components and integrating them with efficient connectors/adapters. Yet, the training process remains challenging because of the diverse objective functions associated with different components in the model architecture.

At present, there are three primary approaches for constructing any-to-any multimodal models: the first involves utilizing the end-to-end trainable Type-D model architecture, the second entails

leveraging the end-to-end trainable Type-C architecture, and the third method employs a combination of Type-C with agents, which is non-end-to-end trainable. **Type-D architecture:** Models generate multimodal outputs using tokenizers. Unified-IO, Unified-IO 2 and 4M are the models from Type-D which enable any-to-any multimodal model development. **Type-C architecture:** Models generate multimodal outputs without using tokenizers. NExt-GPT, CoDI and CoDI-2 models belong to Type-C assisting in any-to-any multimodal model development. **Type-C + agents:** In this method, a Type-C multimodal model is trained to generate specific text outputs with a general format to aid the frozen pretrained modality decoder models (like text-to-image models, text-to-video models) for multimodal generation. ModaVerse follows this process for creating an any-to-any multimodal model. The absence of an end-to-end training process results in performance that is not superior to the other two methods. Table 5 and 6 compares performance of any-to-any multimodal models.

Table 5: Comparing two next generation any-to-any models. Lu et al. [2023]

Model	Image generation		Audio generation & captioning				Video captioning
	FID↓	TIFA↑	FAD↓	IS↑	KL↓	CIDEr↑	CIDEr↑
UnifiedIO-2	13.39	81.3	2.64	5.89	1.80	48.9	48.8
CoDI	11.26	71.6	1.80	8.77	1.40	78.9	74.4

Table 6: Comparing next generation any-to-any models. Generation and captioning metrics for each modality are captured in the table. Wang et al. [2024a]

Model	Image		Audio		Video	
	FID↓	CIDEr↑	IS↑	CIDEr↑	CLIPSIM↑	METEOR↑
CoDI	11.26	149.9	8.77	0.789	0.2890	32.5
NExT-GPT	11.28	156.7	8.35	0.802	0.3085	38.5
ModaVerse	11.24	151.4	8.22	0.792	0.3014	35.2

State space models (SSMs) Gu et al. [2021], Smith et al. [2023], Gu and Dao [2023], are emerging as a viable alternative to transformers based model architectures. They tackle the inherent quadratic complexity of attention mechanisms in Transformers. Examples such as VL-Mamba Qiao et al. [2024] and Cobra Zhao et al. [2024] serve as compelling illustrations of how SSMs can be extended to incorporate multimodal learning capabilities. Their architecture closely resembles that of Type-C multimodal model architectures; therefore, it has been incorporated into the Type-C section in the Figure 1. Moreover, works such as MambaTalk Xu et al. [2024] and SpikeMba Li et al. [2024c] are enhancing SSMs by incorporating modalities such as audio and video, respectively. While any-to-any multimodal models based on SSMs have not yet been developed, the potential exists to construct such models. Therefore, in the future, SSMs may emerge as a robust alternative to Transformer-based Type-C and Type-D multimodal model architectures for any-to-any multimodal tasks.

## 6 Conclusion

This work uniquely identifies and categorizes existing multimodal model architectures in four types. Each architecture type is discussed in detail, including visualization of the general model architecture along with insights into their respective characteristics. Through a thorough examination of existing architectural patterns used in the development of any-to-any multimodal models, this research effort sheds light on the two prevalent approaches (Type-C and Type-D) that are currently driving advancements in this field. By comparing & contrasting architecture types to each other by describing their advantage & disadvantages, this work aids in model choices. This study map a broad spectrum of existing multimodal models to the four identified types. Though the model list is comprehensive, it is not exhaustive. By establishing a taxonomy of multimodal architectures, we can effectively track and capture the evolving trends and advancements within the multimodal domain.

## References

- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv preprint arXiv:2312.17172*, 2023.
- David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. *Advances in Neural Information Processing Systems*, 36, 2024.
- Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023a.
- Zhen Yang, Yingxue Zhang, Fandong Meng, and Jie Zhou. Teal: Tokenize and embed all for multi-modal large language models, 2024.
- Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context, interleaved, and interactive any-to-any generation. *arXiv preprint arXiv:2311.18775*, 2023a.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.
- Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning, 2023a.
- Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*, 2023.
- Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. *arXiv preprint arXiv:2312.00438*, 2023.
- Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023a.
- Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

- Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023a.
- Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023a.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023a.
- Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, et al. Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer. *arXiv preprint arXiv:2401.10208*, 2024.
- Bin Lin, Zhenyu Tang, Yang Ye, Jiayi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.
- Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2022a.
- Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models, 2024a.
- Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models, 2024.
- Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The (r) evolution of multimodal large language models: A survey. *arXiv preprint arXiv:2402.12451*, 2024.
- Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482, 2023b.
- Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and S Yu Philip. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023b.
- Ruifeng Guo, Jingxuan Wei, Linzhuang Sun, Bihui Yu, Guiyong Chang, Dawei Liu, Sibao Zhang, Zhengbing Yao, Mingjun Xu, and Liping Bu. A survey on image-text multimodal models. *arXiv preprint arXiv:2309.15857*, 2023.
- Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443, 2018.
- Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023b.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
- Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019.
- Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.
- Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 326–335, 2017.
- Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.
- Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.
- Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.
- Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

- Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023a.
- Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.
- Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cv conference on computer vision and pattern recognition*, pages 3195–3204, 2019.
- Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8076–8084, 2019.
- Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 417–434. Springer, 2020.
- Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluís Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019.
- Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.
- Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.
- Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017a.

- Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken moments: Learning joint audio-visual representations from video descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14871–14881, 2021.
- Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.
- Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197, 2011.
- Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826, 2022.
- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022.
- Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. On the de-duplication of laion-2b. *arXiv preprint arXiv:2303.12733*, 2023.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models (2023). *arXiv preprint arXiv:2302.13971*, 2023.
- Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017b.
- Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.
- Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.
- Desmond Elliott, Stella Frank, Khalil Sima’an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*, 2016.

- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.
- Ji Qi, Ming Ding, Weihang Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: Train large vision-language models diving into details through chain of manipulations. *arXiv preprint arXiv:2402.04236*, 2024.
- Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.
- Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. Lion: Empowering multi-modal large language model with dual-level visual knowledge. *arXiv preprint arXiv:2311.11860*, 2023c.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- David Eigen, Marc’ Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.
- Teams ShareGPT. Sharegpt: Share your wildest chatgpt conversations with one click, 2023. <https://sharegpt.com/>.
- Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. <https://github.com/kakaobrain/coyo-dataset>, 2022.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023b.
- Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022b.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024a.
- Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023a.
- Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. Llavav: Enhanced visual instruction tuning for text-rich image understanding, 2024b.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.
- Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023d.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.



- Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.
- Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification, 2023. URL <https://huggingface.co/Open-Orca/SlimOrca>.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024.
- Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36, 2024.
- Jiaxi Gu, Xiaojun Meng, Guansong Lu, Lu Hou, Niu Minzhe, Xiaodan Liang, Lewei Yao, Runhui Huang, Wei Zhang, Xin Jiang, et al. Wukong: A 100 million large-scale chinese cross-modal pre-training benchmark. *Advances in Neural Information Processing Systems*, 35:26418–26431, 2022.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.
- Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021a.
- Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.
- Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10126–10135, 2020a.
- Dmitry Ustalov, Nikita Pavlichenko, Sergey Koshelev, Daniil Likhobaba, and Alisa Smirnova. Toloka visual question answering benchmark. *arXiv preprint arXiv:2309.16511*, 2023.
- Bo Zhao, Boya Wu, and Tiejun Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023a.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023b.
- Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.
- Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023e.
- Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023b.

- Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, Jiaqi Wang, and Dahua Lin. Wanjuan: A comprehensive multimodal dataset for advancing english and chinese large models. *arXiv preprint arXiv:2308.10755*, 2023.
- Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv preprint arXiv:2311.10774*, 2023c.
- Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.
- Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021b.
- Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8876–8884, 2019.
- Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv preprint arXiv:2311.07574*, 2023c.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6, 2023.
- InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities, 2023.
- Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.
- Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multi-modal assistant with small language model. *arXiv preprint arXiv:2401.02330*, 2024b.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024b. URL <https://llava-vl.github.io/blog/2024-01-30-llava-next/>.
- Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.
- Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023f.
- Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*, 2023.
- Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.
- Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023c.

- Liang Zhao, En Yu, Zheng Ge, Jinrong Yang, Haoran Wei, Hongyu Zhou, Jianjian Sun, Yuang Peng, Runpei Dong, Chunrui Han, et al. Chatspot: Bootstrapping multimodal llms via precise referring instruction tuning. *arXiv preprint arXiv:2307.09474*, 2023b.
- Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşirlar. Introducing our multimodal models, 2023. URL <https://www.adept.ai/blog/fuyu-8b>.
- Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *International Conference on Machine Learning*, pages 17283–17300. PMLR, 2023.
- Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Van Tu Vu, Zhida Huang, and Tao Wang. Groundinggpt:language enhanced multi-modal grounding model, 2024b.
- Xinyu Wang, Bohan Zhuang, and Qi Wu. Modaverse: Efficiently transforming modalities with llms. *arXiv preprint arXiv:2401.06395*, 2024a.
- Chenyu Wang, Weixin Luo, Qianyu Chen, Haonan Mai, Jindi Guo, Sixun Dong, Xiaohua, Xuan, Zhengxin Li, Lin Ma, and Shenghua Gao. Mllm-tool: A multimodal large language model for tool agent learning, 2024b.
- Siming Yan, Min Bai, Weifeng Chen, Xiong Zhou, Qixing Huang, and Li Erran Li. Vigor: Improving visual grounding of large vision language models with fine-grained reward modeling. *arXiv preprint arXiv:2402.06118*, 2024.
- Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 36, 2024.
- Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*, 2023g.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.
- Quan Sun, Qiyang Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.
- Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023d.
- Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2256–2264, 2024.
- Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*, 2023b.
- Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv preprint arXiv:2311.18799*, 2023.
- Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023c.
- Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023a.

- Zhengqing Yuan, Zhaoxu Li, and Lichao Sun. Tinygpt-v: Efficient multimodal large language model via small backbones. *arXiv preprint arXiv:2312.16862*, 2023a.
- Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023b.
- Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv preprint arXiv:2402.05935*, 2024.
- Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024b.
- Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, et al. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2023.
- Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023c.
- Penghao Wu and Saining Xie. V\*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.
- Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhui Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models, 2024.
- Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruvi Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl: Modularization empowers large language models with multimodality, 2024.
- Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mplug-docowl: Modularized multimodal large language model for document understanding, 2023b.
- Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhui Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36, 2024.
- Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.
- Seungwhan Moon, Andrea Madotto, Zhaoyang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. Anymal: An efficient and scalable any-modality augmented language model. *arXiv preprint arXiv:2309.16058*, 2023.
- Hao Feng, Qi Liu, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *arXiv preprint arXiv:2311.11810*, 2023.
- Anwen Hu, Yaya Shi, Haiyang Xu, Jiabo Ye, Qinghao Ye, Ming Yan, Chenliang Li, Qi Qian, Ji Zhang, and Fei Huang. mplug-paperowl: Scientific diagram analysis with the multimodal large language model. *arXiv preprint arXiv:2311.18248*, 2023.

- Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. *arXiv preprint arXiv:2312.10032*, 2023b.
- Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. *arXiv preprint arXiv:2401.12863*, 2024.
- Dongsheng Zhu, Xunzhu Tang, Weidong Han, Jinghui Lu, Yukun Zhao, Guoliang Xing, Junfeng Wang, and Dawei Yin. Vislinginstruct: Elevating zero-shot learning in multi-modal language models with autonomous instruction optimization. *arXiv preprint arXiv:2402.07398*, 2024c.
- Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.
- Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.
- Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024c.
- Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, Di Zhang, Wenwu Ou, Kun Gai, and Yadong Mu. Unified language-vision pretraining in llm with dynamic discrete visual tokenization, 2024.
- Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023.
- Jinguo Zhu, Xiaohan Ding, Yixiao Ge, Yuying Ge, Sijie Zhao, Hengshuang Zhao, Xiaohua Wang, and Ying Shan. Vl-gpt: A generative pre-trained transformer for vision and language understanding and generation. *arXiv preprint arXiv:2312.09251*, 2023b.
- Sipeng Zheng, Bohan Zhou, Yicheng Feng, Ye Wang, and Zongqing Lu. Unicode: Learning a unified codebook for multimodal large language models. *arXiv preprint arXiv:2403.09072*, 2024.

- Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023.
- Changhan Wang, Anne Wu, and Juan Pino. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*, 2020b.
- Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–325, 2017.
- Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020.
- Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.
- Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024d.
- Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, pages 348–367. Springer, 2022.
- Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.
- Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling, 2023.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. Vl-mamba: Exploring state space models for multimodal learning. *arXiv preprint arXiv:2403.13600*, 2024.
- Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending mamba to multi-modal large language model for efficient inference. *arXiv preprint arXiv:2403.14520*, 2024.
- Zunnan Xu, Yukang Lin, Haonan Han, Sicheng Yang, Ronghui Li, Yachao Zhang, and Xiu Li. Mambataik: Efficient holistic gesture synthesis with selective state space models. *arXiv preprint arXiv:2403.09471*, 2024.
- Wenrui Li, Xiaopeng Hong, and Xiaopeng Fan. Spikemba: Multi-modal spiking saliency mamba for temporal video grounding. *arXiv preprint arXiv:2404.01174*, 2024c.