SCIENCE CHINA
Information Sciences

• REVIEW •

January 2026, Vol. 69, Iss. 1, 111301:1–111301:31
https://doi.org/10.1007/s11432-025-4498-2

An overview of domain-specific foundation model: key technologies, applications and challenges

Haolong CHEN[1,2], Hanzhi CHEN[1,10], Zijian ZHAO[1,6], Kaifeng HAN[4*], Guangxu ZHU[1,2*], Yichen ZHAO[5], Ying DU[3,4], Wei XU[7,8] & Qingjiang SHI[1,9]

[1]*Shenzhen Research Institute of Big Data, Shenzhen 518172, China*
[2]*School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China*
[3]*Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China*
[4]*The Mobile Communications Innovation Center, China Academy of Information and Communications Technology, Beijing 100191, China*
[5]*China Mobile Group Device Co., Ltd., Beijing 100033, China*
[6]*School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China*
[7]*National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China*
[8]*Purple Mountain Laboratories, Nanjing 211111, China*
[9]*School of Software Engineering, Tongji University, Shanghai 201804, China*
[10]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore*

Received 18 January 2025/Revised 3 June 2025/Accepted 13 June 2025/Published online 19 September 2025

Abstract The impressive performance of ChatGPT and other foundation-model-based products in human language understanding has prompted both academia and industry to explore how these models can be tailored for specific industries and application scenarios. This process, known as the customization of domain-specific foundation models (FMs), addresses the limitations of general-purpose models, which may not fully capture the unique patterns and requirements of domain-specific data. Despite its importance, there is a notable lack of comprehensive overview papers on building domain-specific FMs, while numerous resources exist for general-purpose models. To bridge this gap, this article provides a timely and thorough overview of the methodology for customizing domain-specific FMs. It introduces basic concepts, outlines the general architecture, and surveys key methods for constructing domain-specific models. Furthermore, the article discusses various domains that can benefit from these specialized models and highlights the challenges ahead. Through this overview, we aim to offer valuable guidance and reference for researchers and practitioners from diverse fields to develop their own customized FMs.

Keywords artificial intelligence, domain-specific foundation model, multi-modality foundation model, pre-training foundation model, fine-tuning

Citation Chen H L, Chen H Z, Zhao Z J, et al. An overview of domain-specific foundation model: key technologies, applications and challenges. Sci China Inf Sci, 2026, 69(1): 111301, https://doi.org/10.1007/s11432-025-4498-2

1 Introduction

ChatGPT has redefined people's understanding of artificial intelligence with its outstanding performance. As its core technology, the large language model (LLM) has become an essential tool for researchers and practitioners across various fields to improve their workflows. General-purpose foundation models (FMs) are usually trained on large public datasets, enabling them to learn and address a wide range of common problems. However, these datasets cannot fully encompass all the specialized knowledge and technical details of specific domains. As a result, although general-purpose FMs possess broad general knowledge, they lack the necessary depth to meet the complex needs of some specific fields [1]. Therefore, constructing domain-specific FMs tailored to the needs of particular industries has become particularly important. The term "domain" refers to a specific area of knowledge with logical completeness and knowledge complexity, such as healthcare, finance, or legal services. Often, only professionally trained one can master a certain domain's terminology, concepts, and techniques. Domain-specific FMs, or industry-specific FMs, are developed using data specific to a particular field. Compared to general-purpose FMs, they are trained with a large amount of domain-specific data, enabling them to more accurately understand the

* Corresponding author (email: hankaifeng@caict.ac.cn, gxzhu@sribd.cn)

Table 1 Examples of FMs with two different types.

Types of FMs	Examples
Single-modality FMs	VGG [2], ResNet [3], GPT-1 [4], GPT-2 [5], GPT-3 [6], GPT-3.5 turbo, BERT [7], GLM [8,9], LLaMA [10], LLaMA-2 [11], iGPT [12], LVM [13], SAM [14], BART [15], T5 [16], Time-LLM [17], UniTS [18], ST-LLM [19]
Multi-modality FMs	CoDi [20], CoDi-2 [21], Claude-3, GPT-4 [22], LLaVA [23], BriVL [24], ImageBind [25], NExT-GPT [26]

unique language, terminology, concepts, and nuances inherent to the specific domain and generate domain-specific professional content.

With the widespread use of ChatGPT-like products, the scope of the "foundation model" is gradually expanding. It is necessary to clearly define the foundation model discussed to support the subsequent discussion on customizing domain-specific FMs. The FMs mentioned in this article are neural network models that consist of at least one of the five modules (which will be detailed later) in a general multi-modality FM. These models also exhibit the following characteristics.

• **Big data.** The model utilizes a large volume of data covering various scenarios for training to provide ample knowledge for the model.

• **Large parameters.** The model possesses a huge number of parameters sufficient to embed the knowledge implied by the big data into the model's parameters.

• **Versatility.** The model's input data format and data processing workflow can adapt to different requirements across various task scenarios.

• **Fast adaptation capabilities.** The model possesses a vast knowledge repository, enabling it to deliver robust performance even in unknown data domains with minimal fine-tuning.

In the context of artificial intelligence, we use "modality" to categorize data with a consistent data structure and semantic structure. Such a definition brings the conceptual consistency of each modality, which is the theoretical foundation of multi-modality data analysis and processing. Based on the number of modalities an FM can process, they can be classified into single-modality FMs and multi-modality FMs, as shown in Table 1 [2–26]. With the certain modality an FM can process, we can use different names to categorize FMs, such as large language models, large vision models, and large vision-language models.

In constructing domain-specific FMs, a series of challenges will arise, especially in the data acquisition and preprocessing stages. For example, the domain-specific data required may not be open-source or readily accessible, as it often has high confidentiality. Additionally, the modalities of domain-specific data might differ from those used for training general-purpose FMs, making it difficult to adapt existing models to process this data. Furthermore, the environments where the domain-specific data is collected may differ significantly from those for the pre-training datasets, resulting in domain-specific knowledge that pre-trained models are unfamiliar with. In general, constructing a domain-specific FM is challenging and costly, with significant implications for technical security, but it is expected to yield high economic benefits. Therefore, it is essential to thoroughly review and explore the methodologies for constructing these models, providing guidelines for both researchers and practitioners. The organization structure of this article is summarized in Figure 1.

Notably, previous review articles have predominantly focused on developing general-purpose FMs. Recently, while several review articles have begun to explore the domain-specific adaptation of FMs, there is a significant gap in the literature for a comprehensive exploration of adaptation strategies that apply to FMs of all modalities, extending beyond language, vision, or any single modality, to various application domains. We summarize representative surveys or review articles on FMs in Table 2 [27–47]. This paper aims to provide researchers and practitioners interested in building domain-specific FM applications with methodological references by introducing the key methodologies. Also, practical examples and future directions will be discussed.

2 Preliminaries of multi-modality foundation models

This section will elaborate on the preliminary technological basis for customizing domain-specific FMs. We begin with the architecture of FMs, detailing all functional modules. Then, from two perspectives—model training and scaling gain—we will explain the foundational technologies that enable each module to contribute to the high performance of FMs. Finally, we compare the performance of various general-purpose large multi-modality models for an intuitive sense of the performance that FMs can achieve and guide model employers in choosing the most suitable FM.

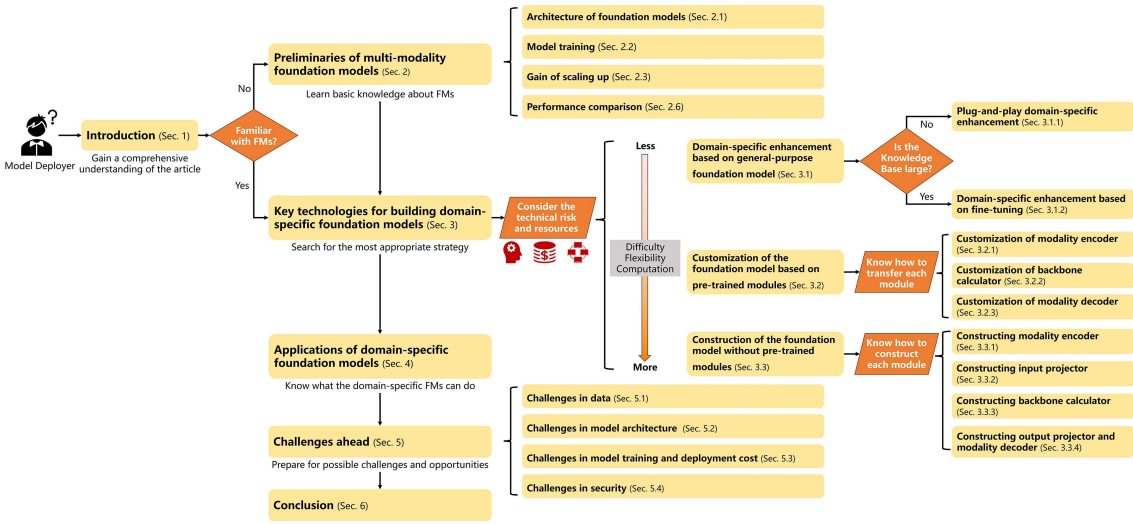

Figure 1 (Color online) Organization structure of this article.

Table 2 Summary of representative surveys or review articles on FMs.

Article	Modality	Domain adaptation	Domain	Year
Zhao et al. [27]	Language	No	General	2023
Liu et al. [28]	Graph	No	General	2023
Mao et al. [29]	Graph	No	General	2024
Du et al. [30]	Vision-language	No	General	2022
Yin et al. [31]	Multi-modality-language	No	General	2023
Yeh et al. [32]	Time-series data	No	General	2023
Jin et al. [33]	Time-series & spatio-temporal data	No	General	2023
Cao et al. [34]	Multi-modality	No	General	2023
Zhou et al. [35]	Multi-modality	No	General	2023
Zhang et al. [36]	Multi-modality-language	No	General	2024
Lai et al. [37]	Language	Yes	Law	2024
Ahn et al. [38]	Language	Yes	Mathematics	2024
Li et al. [39]	Language	Yes	Finance	2023
Thirunavukarasu et al. [40]	Language	Yes	Medicine	2023
Kazerouni et al. [41]	Vision	Yes	Medicine	2023
Shahriar [42]	Multi-modality	Yes	Arts	2022
Hou et al. [43]	Language	Yes	Software engineering	2023
Bariah et al. [44]	Language	Yes	Telecommunications	2024
Zhou et al. [45]	Language	Yes	Telecommunications	2024
Cui et al. [46]	Multi-modality	Yes	Autonomous driving	2024
Yang et al. [47]	Language	Yes	General	2024
Ours	Multi-modality	Yes	General	–

2.1 Architecture of foundation models

According to state-of-the-art research on FMs, it is widely considered that multi-modality FMs can encompass all functionalities and structures of single-modality FMs. Essentially, a single-modality FM implements only a subset of the functionalities of a multi-modality FM.

The five-module framework proposed for multi-modality FMs in [36] effectively encompasses the architecture of multi-modality FMs with language as the central modality. However, the recent emergence of non-language FMs, including vision FMs [13,14], graph FMs [28,29], time series FMs and spatio-temporal (ST) FMs [32,33], indicates that the backbone of FMs is expanding beyond language modalities. Therefore, we propose that the structure of multi-modality FMs can be divided into the following five modules: modality encoder, input projector, backbone calculator, output projector, and modality decoder. Figure 2 illustrates the framework of a multi-modality FM with language as the central modality.

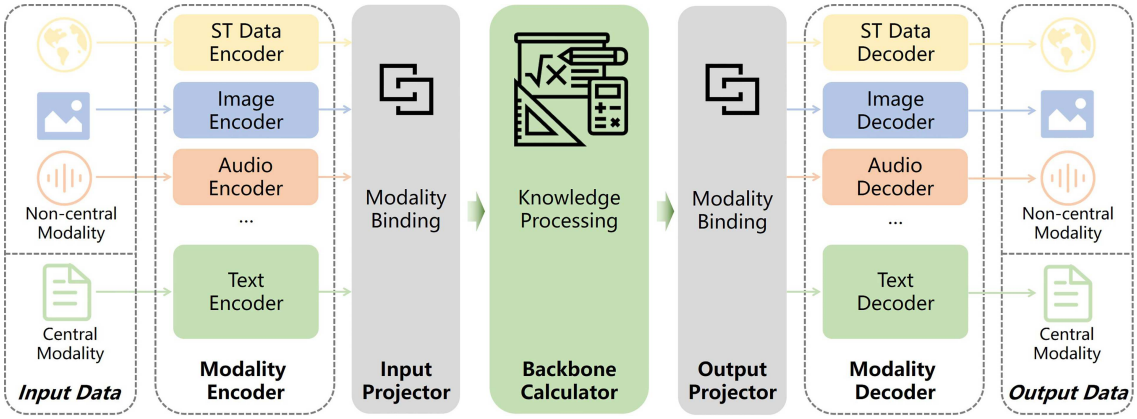

Figure 2 (Color online) Framework of multi-modality FMs with language as the central modality.

For multi-modality FMs, we define the set of all input modalities as M. Generally, a multi-modality FM has a central modality C. Using modality alignment techniques, the multi-modality FM projects all the modalities it can handle onto this central modality. Below, we define the five modules of the multi-modality FM and the input and output data for each module, laying down the foundation for describing the architecture of FMs in this paper.

Modality encoder (ME) encodes the data D_X of an input modality X into a feature vector F_X:

$$F_X = \mathrm{ME}_X(D_X), \quad X \in M. \tag{1}$$

Input projector (IP) projects the feature vector F_X of a modality X into the feature vector F_C of the central modality C:

$$F_C = \mathrm{IP}_{XC}(F_X), \quad X, C \in M. \tag{2}$$

Backbone calculator (BC) performs operations on the feature vector F_C of the central modality C, yielding results such as inference and generation $\hat{F}_C$:

$$\hat{F}_C = \mathrm{BC}_C(F_C), \quad C \in M. \tag{3}$$

Output projector (OP) projects the feature vector $\hat{F}_C$ of the central modality C into the feature vector $\hat{F}_X$ of a modality X:

$$\hat{F}_X = \mathrm{OP}_{CX}(\hat{F}_C), \quad X, C \in M. \tag{4}$$

Modality decoder (MD) decodes the feature vector $\hat{F}_X$ of the output modality X back into the original data format, resulting in the decoded data $\hat{D}_X$:

$$\hat{D}_X = \mathrm{MD}_X(\hat{F}_X), \quad X \in M. \tag{5}$$

According to the above definition, building a domain-specific FM entails choosing the necessary modules—some of which may be optional—and assembling a model tailored to the specific domain requirements, followed by training it with data pertinent to that domain.

Here, we provide a concrete example for better illustration. CoDi-2 [21] is a language-centric multi-modality FM capable of handling various modalities such as images, videos, text, and audio. It utilizes a large language model (LLM) to generate new content. CoDi-2 employs vision Transformers (ViTs) pre-trained by ImageBind [25] as the MEs for the image, video, and audio modalities. For the text modality, the ME is a text tokenizer. Since the output features of the ImageBind pre-trained ViT are specific to the image modality, CoDi-2 trains a multilayer perceptron (MLP) as the IP to map the features from the image, video, and audio MEs to the central modality, text. Subsequently, these text features are fed into its BC, LLaMA2—an LLM—that generates content matching the user's input prompt in the text modality. Then, the generated image and audio content are mapped back to their respective feature spaces through an OP, another MLP. Finally, pre-trained diffusion models are used as image, video, and audio MDs to generate the content.

2.2 Model training

After constructing an FM module by module, we need to train it. Model training involves setting up various training tasks to enable FM to specifically extract the features required for solving each task. Feature extraction concerns

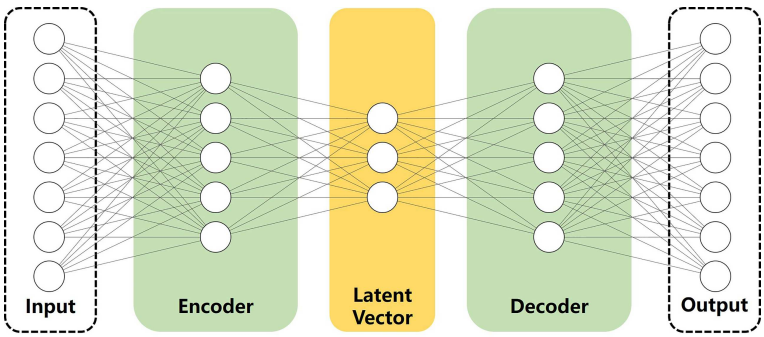

Figure 3 (Color online) Structure of autoencoder.

extracting representative features from raw data. As raw data often contains a significant amount of redundant and noisy information, feature extraction, which maps the data into a more information-dense feature space, enables the models to understand data structure and patterns more effectively. Feature space is the set of coordinates representing data samples across multiple feature dimensions, which mathematically refer to a high-dimensional space. Note that "raw data" and "feature" are used in this paper to differentiate whether a dataset has undergone feature extraction, which is essentially a set of mathematical calculations that increase the information density of the data. In deep learning, neural networks can perform end-to-end feature extraction from raw data. However, this often requires large amounts of data and computational resources to ensure good performance and generalization. In each layer of neural networks, it transfers the output of the previous layer to a new vector space. This structure allows for a flexible definition of the output dimensions of each layer without explicitly specifying the transformation.

Numerous training methods are available to achieve better feature extraction. They construct different loss functions to teach the model how to focus on different aspects of the data, thereby extracting various feature representations. If labeled information is available as the target for a specific task during training, supervised learning can be used to train the model's feature extraction capabilities. For example, suppose we have labeled data for a classification task. In that case, we can construct a loss function such as cross-entropy loss to measure the difference between the model's predicted probability distribution and the true label's probability distribution. This provides the model with a specific direction for optimization to achieve the best feature extraction for the classification task. Sometimes, we may not have labeled data for a specific task but still wish to mine features from the data to perform better on downstream tasks in the future. In such cases, unsupervised learning methods can be employed. Unsupervised training involves learning from data without the need for labeled examples. Common methods include autoencoders, which learn to compress and reconstruct input data, as shown in Figure 3, capturing important features. Another approach is using generative models like generative adversarial networks (GANs) or variational autoencoders (VAEs) to learn the distribution of the data. Self-supervised training is also considered a subset of unsupervised training [48], such as in training LLMs where autoregressive prediction tasks are used. By generating subsequent content based on the previous content, the model inherently captures the distribution of text sequences. Combining unsupervised and self-supervised training can form semi-supervised training, which is well-suited for situations where there is a limited amount of labeled data and a large amount of unlabeled data.

Humans can also play a significant guiding role in training the model's feature extraction capabilities. Reinforcement learning from human feedback (RLHF) is a method that combines reinforcement learning with human feedback, adjusting the model based on human evaluations of its behavior to learn feature representations that align more closely with human expectations. This technique is often mentioned in the context of training LLMs. First, the model generates a series of outputs and requests a human evaluation of these outputs, such as scoring, ranking, or providing other forms of feedback. These feedback data are then used to train a reward model that assesses the quality of different outputs based on human preferences. Then, using the reward model, the initial model is fine-tuned through reinforcement learning algorithms. During this fine-tuning process, the model continuously adjusts its behavior strategy based on the feedback from the reward model to maximize cumulative rewards. Through this process, the model can better understand user preferences and needs, generating outputs more aligned with human expectations.

For a single-modality FM, cross-modality data is unavailable, so the architecture excludes IPs and OPs. In contrast, multi-modality FMs must accommodate the processing of various data modalities, including the primary and ancillary modalities. To achieve data conversion between modalities through IPs and OPs, the key is to apply modality alignment. The goal of modality alignment is to process the feature vectors of different modalities into a common feature space with the same dimension by using a loss function to characterize the correlation between

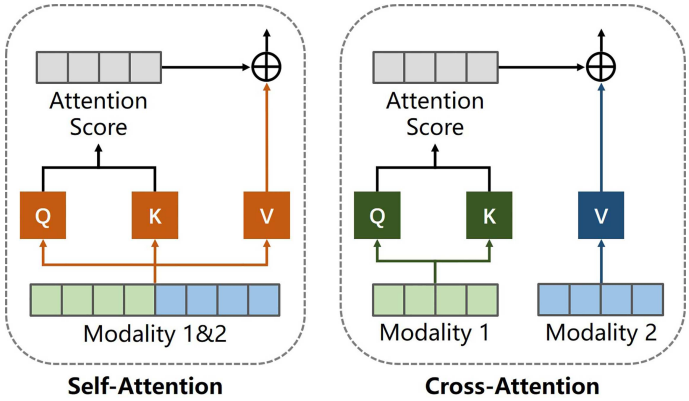

Figure 4 (Color online) Structure of fusion encoder.

feature vectors. Ideally, modality alignment should ensure that raw data of different modalities carrying the same semantic information are represented as the same point in the target feature space, facilitating cross-modality information transfer.

There are two main implementation methods for modality alignment: the fusion encoder architecture and the dual encoder architecture [30].

(1) **Fusion encoder.** Fusion encoder encodes the multi-modality data by the self-attention or cross-attention mechanism from transformer [49]. The attention mechanism can be represented as follows:

$$\text{Attenton}(Q,K,V)=\text{softmax}\left(\frac{Q^TK}{\sqrt{d_k}}\right)V, \tag{6}$$

where query Q, key K, and value V are all intermediate features based on the input feature, and d_k refers to the dimension of Q and K.

Self-attention-based mechanisms require concatenating the feature vectors of the main and auxiliary modalities as input to the transformer to generate Q, K, and V, allowing the model to automatically focus on the features of different modalities and achieve cross-modality information fusion. For example, VL-BERT [50] concatenates the feature vectors of text and images, and then uses the self-attention mechanism of the transformer to aggregate and align language-visual features. On the other hand, methods based on the cross-attention mechanism calculate Q, K, and V separately for the feature vectors of the two modalities, thereby achieving cross-modality information fusion. For example, DiT [51] captures the relationship between text and images by a self-attention mechanism and then realizes the image generation controlled by text. We illustrate the two fusion encoder architectures in Figure 4.

(2) **Dual encoder.** As a multi-modality learning strategy, dual encoder trains a dedicated encoder for each modality independently. The core idea of this architecture is to guide the learning process of the two encoders in synchronization through semantic similarity metrics by leveraging contrastive learning methods. Then, the output feature vectors of different encoders can be projected into the same vector space. Specifically, this model is based on the hypothesis that if the feature vectors output by the two encoders belong to the same feature space, the feature vectors with paired labels should be closer in the vector space and vice versa. Through this alignment method, we can expect that the output results of encoders for different modalities describing similar objects or scenes will be close enough. And even in an ideal state, they will converge to the same point in the feature space. The key to achieving this goal is to construct a reasonable model architecture and thoroughly train it on a large-scale dataset. Figure 5 shows the processing logic of the dual encoder.

However, the cost of one-to-one aligning each pair of modalities would be very high. Besides, obtaining a dataset with each pair of modalities aligned is also challenging. To this end, some researchers have proposed the bridging alignment (also known as binding alignment) strategy. This method matches all other modalities with a central modality, thereby achieving the alignment of all modalities in the semantic space. For example, ImageBind [25] takes the image as the central modality, while CoDi [20] takes text as the central modality. In this way, they effectively simplify the training process of multi-modality alignment and improve the practicality and efficiency of the model. In contrastive learning, the class imbalance in long-tailed datasets can lead to reduced effectiveness, as most contrastive pairs consist of major classes instead of tail classes, making the model ineffective for the minor classes. Therefore, Ref. [52] proposed a method to solve this problem.

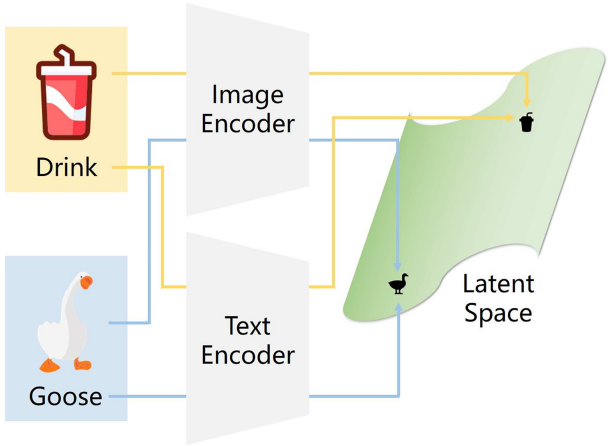

Figure 5 (Color online) Structure of dual encoder.

2.3 Gain of scaling up

As the training scale of artificial intelligence (AI) models continues to grow, their capabilities in various tasks demonstrated significant leaps in performance. When the model size reaches the FM level, researchers have introduced the concepts of scaling laws and emergent phenomena to illustrate the quantitative and qualitative relationships between training scale and model performance.

Scaling law refers to the mathematical pattern of how the performance of a system changes as the scale of the system increases. In AI, especially in the research and application of FMs, the scaling law describes rules and phenomena about how model performance changes as the model scale expands, including the number of parameters, dataset size, and computational resources. It uses quantitative analysis methods to reveal the intrinsic mechanism of the performance improvement of FMs.

Ref. [53] discussed how the inductive biases of different models affect the relationship between model scale expansion and performance. They found that model architecture is indeed one of the key factors affecting the benefits of model expansion. They also pointed out that although the standard transformer architecture may not always achieve the best performance, it does exhibit the best scalability. In computer vision [54] and natural language process [55], models based on the transformer architecture have shown an exponential relationship between model scale and model performance.

Besides, Ref. [56] examined the impact of the number of downstream tasks and model scale on the performance of instruction fine-tuning. They fine-tuned the models on various tasks by a multi-task joint training method. As a result, the language model could learn a broader language representation and knowledge, enhancing its generalization ability on unseen tasks. During the joint training process, knowledge transfer between different tasks is promoted through parameter sharing, significantly improving the model's generalization ability and performance. In addition, joint training also reduces the time and computational resources required to train each task individually, improving training efficiency. This phenomenon of model performance improving as task diversity increases is a manifestation of the scaling law. Ref. [57] verified the phenomenon of model performance increasing with the number of tasks on the large-scale benchmark OPT-IML Bench. Additionally, some studies have separately provided the model performance of natural language models [55] and autoregressive generative models of various modalities [58] at different scales.

Although there is no unified form for the quantitative representation of the scaling law, it can be generally represented as an exponential relationship between the model loss function and the model's parameters, dataset size, and computational resources. We use the loss function $L(\cdot)$ to characterize the model's performance, where a smaller loss function value represents better model performance. Eq. (7) describes the performance of a model with a given number of parameters trained to convergence on a sufficiently large dataset, where $L(N)$ is the loss function, N is the number of trainable parameters of the model, N_c is a constant, and α_N is the power law exponent. Eq. (8) provides the performance of a suitably sized model trained on a sufficiently large dataset under a given computational resource constraint, where $L(C)$ is the loss function, C is the given computational resource, C_c is a constant, and α_C is the power law exponent. Eq. (9) describes the performance of an FM trained with an early stopping strategy on a given dataset size, where $L(D)$ is the loss function, D is the dataset size (in tokens), D_c is

a constant, and α_D is the power law exponent.

$$L(N) \propto \left(\frac{N_c}{N}\right)^{\alpha_N}, \tag{7}$$

$$L(C) \propto \left(\frac{C_c}{C}\right)^{\alpha_C}, \tag{8}$$

$$L(D) \propto \left(\frac{D_c}{D}\right)^{\alpha_D}. \tag{9}$$

According to the above equations, when not constrained by other conditions, the model's loss function decreases exponentially as the number of parameters, computational resources, and training data volume increase. This means that by increasing the model's parameters, investing more in computational resources, and expanding the training data volume, the model's performance can also be exponentially improved.

The scaling law reveals that the scaling up of model size can lead to incremental improvements in model performance. On the contrary, the emergent phenomenon refers to the new properties exhibited by the model after reaching a critical point of scale expansion, one of which is a significant improvement in model performance [59].

The emergent phenomenon essentially reveals the source of the superior performance of FMs. In deep learning, especially in the domain of LLM, the emergent phenomenon has been widely observed. For example, models like LLaMA have demonstrated exceptional comprehension, generation capabilities, and even a certain degree of logical reasoning ability, which small language models cannot achieve. As the model scale increases, it can have more parameters and a more complex structure, allowing it to capture the complex features and patterns in the data. FMs often exhibit strong generalization capabilities because their abundant parameters can store rich knowledge, enabling them to make accurate inferences and predictions on unseen data. It can provide them adaptability and universality to different tasks and even allows the model to truly learn the underlying principles and reasoning methods in the data. Ref. [59] suggested that the task and the prompting method used can influence the emergence point of the emergent phenomenon in LLMs. Specifically, using a chain-of-thought prompting approach can significantly enhance the LLMs' ability to handle complex reasoning tasks [60], thereby bringing forward the emergence point of the emergent phenomenon.

When building domain-specific FMs, developers must select an appropriate model scale, considering both the requirements of potential downstream tasks and user prompting patterns. An increase in the number of model parameters raises the demand for computational resources and the risk of overfitting. Consequently, there is a limit to the extent that parameters can be increased. The trade-offs highlighted by this phenomenon are crucial for model deployment, offering significant insights for both model design and application.

2.4 Performance comparison

In this subsection, we present the evaluation results of current multi-modality FMs to assist readers in selecting the appropriate model for their needs. We summarize some common vision-language tasks in Table 3 [10, 11, 23, 26, 31, 56, 61–93] according to [26, 83, 87, 94]. We select three mainstream vision-language tasks as follows.

• **Image captioning.** This task involves generating descriptive captions for images, demonstrating models' ability to understand visual content.

• **Image question answering.** In this task, models are required to answer questions about the content of images, demonstrating their ability to comprehend and reason about visual information.

• **Benchmark toolkit.** This refers to a collection of standardized metrics and datasets used to evaluate the performance of multi-modal models across various tasks like conversation, complex reasoning, and detailed description, ensuring a consistent assessment framework.

In summary, InternVL-Chat [87] has the best results in image captioning and image question answering tasks. While considering language tasks such as benchmark toolkits, the best results are provided by NExT-GPT [26].

3 Key technologies for building domain-specific foundation models

This section delves into the technical pathways toward customizing domain-specific FMs. We will explain how to flexibly select and combine the appropriate modules from five key components—MEs, IPs, BCs, OPs, and MDs based on the specific requirements of different domains. Additionally, we will analyze concrete cases to help readers better understand and apply the methodologies discussed in this section.

Table 3 Performance of various large multi-modality models on vision-language tasks. * represents InternVL-Chat with IViT-6B as the visual encoder, QLLaMA as the glue layer and 13B trained parameters. Bold fonts represent the best results. Underlined fonts represent the second-best results.

Model	LLM	Image captioning			Image question answering						Benchmark toolkit						
		NoCaps [61]	Flickr30K [62]	COCO [63]	VQA^{v2} [64]	GQA [65]	VizWiz [66]	SQA^{I} [67]	VQA^{T} [68]	OKVQA [69]	POPE [70]	MME [31]	MMB [71]	LLAVA^{W} [23]	SEED [72]	MM-Vet [73]	QBench [74]
mPLUG-Owl [75]	LLaMA-7B [10]	117.0	80.3	119.3	–	–	39.0	–	–	–	–	–	46.6	–	34.0	–	–
Emu [76]	LLaMA-7B [10]	–	–	117.7	40.0	–	35.4	–	–	34.7	–	–	–	–	–	–	–
I-80B [77]	LLaMA-65B [10]	65.0	53.7	91.8	60.0	45.2	36.0	–	30.9	–	–	–	54.5	–	–	–	–
LLAVA [23]	LLaMA2-7B [11]	120.7	82.7	–	–	–	–	–	–	–	–	–	36.2	–	–	–	–
MobileVLM [78]	MobileLLaMa-2.7B [79]	–	–	–	–	59.0	–	61.0	47.5	–	84.9	1288.9	59.6	–	–	–	–
DREAMLLM [80]	Vicuna-7B [81]	–	–	115.4	56.6	–	45.8	–	–	44.3	–	–	49.9	–	–	–	–
Video-LLAVA [82]	Vicuna-7B [81]	–	–	–	74.7	–	48.1	–	–	–	–	–	60.9	–	–	–	–
NExT-GPT [26]	Vicuna-7B [81]	123.7	84.5	124.9	66.7	–	48.4	–	–	**52.1**	–	–	58.0	–	57.5	–	–
ShareGPT4V-7B [83]	Vicuna-7B [81]	–	–	–	80.6	–	57.2	68.4	–	–	–	1567.4	**68.8**	**72.6**	**69.7**	**37.6**	**63.4**
LLAVA-1.5 [84]	Vicuna-7B [81]	–	–	–	78.5	62.0	50.0	–	58.2	–	85.9	1510.7	–	–	–	–	–
LLAVA-1.5 [84]	Vicuna-13B [81]	–	–	–	80.0	63.3	53.6	**71.6**	61.3	–	85.9	1531.3	67.7	70.7	68.2	35.4	62.1
InstructBLIP [85]	Vicuna-7B [81]	123.1	82.4	102.2	–	–	34.5	60.5	50.1	33.9	–	–	36.0	60.9	53.4	26.2	56.7
InstructBLIP [85]	Vicuna-13B [81]	121.9	82.8	–	–	49.5	33.4	–	50.7	–	78.9	1212.8	–	–	–	–	–
Shikra [86]	Vicuna-13B [81]	–	73.9	117.5	77.4	–	–	–	–	–	–	–	58.8	–	–	–	54.7
InternVL-Chat* [87]	Vicuna-13B [81]	**126.2**	**92.2**	**146.2**	**81.2**	**66.6**	**58.5**	–	61.5	–	**87.6**	**1586.4**	–	–	–	–	–
Qwen-VL [88]	Qwen-7B [88]	121.4	85.8	–	78.8	59.3	35.2	67.1	**63.8**	–	–	–	38.2	–	56.3	–	59.4
Qwen-VL-Chat [89]	Qwen-7B [88]	120.2	81.0	–	78.2	57.5	38.9	68.2	61.5	–	–	1487.5	60.6	–	58.2	–	–
TinyGPT-V [90]	Phi2-2.7B [91]	–	–	–	–	33.6	33.4	–	–	–	–	–	–	–	–	–	–
LLAVA-Phi [92]	Phi2-2.7B [91]	–	–	–	71.4	–	35.9	68.4	48.6	–	85.0	1335.1	59.8	–	–	28.9	–
BLIP-2 [93]	FLAN-T5 [56]	103.9	71.6	–	41.0	41.0	19.6	61.0	42.5	–	85.3	1293.8	–	38.1	46.4	22.4	–

Table 4 Customization methods of domain-specific FMs.

Customization method	Customization degree	Difficulty	Flexibility	Computation
Based on general-purpose FMs	Low, only the input method of domain knowledge is customized	Low	Low	Low
Based on pre-trained modules	Medium, some modules can be customized	Medium	Medium	Medium
Without pre-trained modules	High, each module can be customized	High	High	High

Table 5 Specific-domain enhancement with the entire general-purpose FMs.

Customization method			Parameter modification	Addition of new modules	Domain knowledge provider	Specific techniques
Plug-and-play	Utilizing existing knowledge	Hard prompting	No	No	Deployer	PET [95]
		Soft prompting	No	Yes	Deployer	Prefix tuning [96], P-tuning [97]
	Adding new knowledge	Prompts	No	No	User	LongRoPE [98], Transformer-XL [99]
		External knowledge base	No	Yes	Deployer	RAG [100], GraphRAG [101], RankRAG [102]
Fine-tuning-based	Adapter-based		Yes	Yes	Deployer	Adapter [103], AdapterFusion [104], IA3 [105], model reprogramming [106]
	Low-rank matrix decomposition		Yes	Yes	Deployer	LoRA [107], LoHa [108], LoKr [109]
	Full parameter fine-tuning		Yes	No	Deployer	PEFT [110]

We can categorize the customization of domain-specific FMs into three levels, ranging from low to high level of customization (i.e., from high to low reliance on general-purpose FMs or pre-trained modules).

(1) Domain-specific enhancement based on general-purpose FMs.

(2) Customization of the FM based on pre-trained modules.

(3) Construction of the FM without pre-trained modules.

Table 4 summarizes the characteristics of these three domain-specific FM customization methods.

3.1 Domain-specific enhancement based on general-purpose foundation model

General-purpose FMs offer comprehensive capabilities, making them suitable for various task scenarios. When such a general-purpose FM can fully handle the required data modalities, any modifications to its underlying architecture for model developers are rendered unnecessary. Instead, they can focus on implementing domain-specific enhancements.

Depending on whether the domain-specific enhancement requires altering the parameters of the FM, we can further divide this into plug-and-play domain-specific enhancement and fine-tuning-based domain-specific enhancement. Table 5 [95–110] categorizes and summarizes the methods of domain-specific enhancement based on a full-architecture general-purpose FM.

3.1.1 *Plug-and-play domain-specific enhancement*

The generality, generalization capabilities, and reasoning abilities of general-purpose FMs enable them to serve as the foundation for domain-specific models. To achieve plug-and-play domain enhancement without modifying the parameters of the FM, two approaches can be utilized: leveraging existing knowledge or embedding new knowledge. The first approach aims to leverage domain knowledge already stored within the general-purpose FM, as illustrated in Figure 6(a). The second approach, embedding new knowledge, involves equipping the FM with the ability to handle domain tasks by introducing domain-specific knowledge. This can be divided into two methods: embedding knowledge through prompts and embedding knowledge via an external knowledge base. These methods are depicted in Figures 6(b) and (c), respectively. The following sections will provide a detailed explanation of these techniques.

(1) **Invoking existing knowledge for domain enhancement.** A general-purpose FM may have already enclosed domain knowledge during the training process. Prompt tuning improves prompts to better invoke the model's inherent domain knowledge, where "tuning" refers to the optimization of prompts. Specifically, it involves inserting a carefully crafted prompt into the input data as context to improve the generated output. These carefully

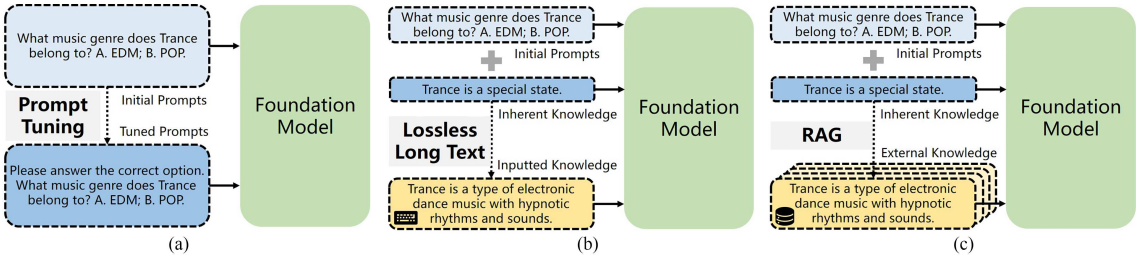

Figure 6 (Color online) Plug-and-play domain-specific enhancement. (a) Invoking existing knowledge; (b) knowledge embedding by prompts; (c) knowledge embedding by external knowledge base.

crafted prompts can be natural language descriptions, examples, rules, or other text or embedding vectors that guide the model to understand the task requirements. The model will consider these carefully crafted prompts to produce task-relevant results when generating outputs. Prompt tuning can be categorized into hard prompts and soft prompts.

(a) **Hard prompts.** Hard prompt methods are common techniques in natural language processing (NLP). They use interpretable and reusable handcrafted words and tokens to guide the language model's output. Hard prompts are typically manually designed and tailored for specific tasks, making them difficult to modify. PET (pattern exploiting training) [95] is a classic hard prompt learning method that models questions as cloze tasks and optimizes the final output words. This method trains the model on a small amount of supervised data and performs ensembling predictions on unsupervised data to guide the model.

(b) **Soft prompts.** Designing hard prompts requires experimental exploration and expertise, and manually designed prompts may not align well with the model's data processing methods. To simplify this process and enhance the flexibility of prompt tuning, researchers proposed soft prompt-based tuning methods. Prefix tuning [96] is a form of soft prompt tuning that adapts to specific downstream tasks by adding learnable prefix vectors (soft prompts) to the beginning of the input sequence. As part of the input, these prefix vectors guide the model's output to meet task requirements. The advantage of prefix tuning is that it only updates these prefix vectors rather than the model's parameters, significantly reducing computational and storage resource demands while retaining the rich knowledge learned by the pre-trained model. Building on prefix tuning, researchers introduced the P-tuning method [97]. P-tuning replaces fixed or manually designed words and tokens with learnable soft prompts. Its core idea is to treat prompts as part of the model that can be learned, allowing the model to learn not only how to respond to given tasks but also how to generate the best prompts. These soft prompts are typically a series of embedding vectors processed alongside the actual text input. Through end-to-end training, the model automatically learns to adjust these embedding vectors for better task performance. P-tuning combines the parameter efficiency of prefix tuning with the flexibility of traditional hard prompt tuning. Soft prompts give the model greater freedom to generate answers, potentially producing more diverse outputs and increasing the risk of generating inaccurate or irrelevant responses.

(2) **Knowledge embedding for domain enhancement.** When the existing knowledge of a general-purpose FM is insufficient to solve domain tasks, introducing new knowledge by embedding additional background information can achieve higher-quality output. This method is known as knowledge embedding for domain enhancement.

(a) **Knowledge embedding by prompts.** Prompts, serving as a direct interface between the user and the large language model, can be used to incorporate domain knowledge. However, the method of knowledge embedding through prompts has a significant limitation: the amount of embedded domain knowledge is restricted by the maximum prompt length of the model. The limitation on the model's ability to process longer text inputs stems from three core issues of the Transformer architecture.

• **Limitations of positional encoding.** Transformer models typically generate fixed-length positional encodings using sine and cosine functions, where each position in the sequence is uniquely encoded. However, when the sequence length exceeds the maximum length used during training, the model cannot effectively handle the additional text because it cannot generate valid encodings for the new positions.

• **Resource consumption of the attention mechanism.** The attention mechanism is the core of Transformer models, allowing the model to compute attention weights for each element in the sequence. However, as the sequence length increases, this mechanism's computational complexity and memory requirements grow quadratically, leading to significant resource consumption.

• **Long-distance dependency issue.** When handling long sequences, the Transformer needs to span a large number of input tokens, which often results in problems such as gradient vanishing or exploding. This makes it

difficult for the model to capture dependencies between elements far apart in the sequence.

To address the above issues, lossless long text technology has emerged. It aims to enhance the model's ability to process long texts that exceed its input length limitations, allowing users to input a large amount of domain knowledge directly through prompts into the large language model as contextual information for domain enhancement. Lossless long text technology expands the long-text input capability of large language models in two directions: extrapolation and interpolation.

(i) **Extrapolation.** Extrapolation involves extending the model's context window to handle new texts that exceed the length of the training data. This typically involves improving the positional encoding mechanism so the model can understand and process longer sequences. Longformer [111] extends the ability to handle long texts effectively by combining local and global attention mechanisms; BigBird [112] uses sparse attention mechanisms and reversible layers to extrapolate the model's long-sequence processing capabilities; LongRoPE [98] improves positional encoding by introducing rotational transformations in self-attention, allowing the model to handle long-distance dependencies and support inputs up to two million tokens without impacting computational efficiency.

(ii) **Interpolation.** Interpolation refers to enhancing the model's ability to process long texts within its existing sequence length capacity by adjusting and optimizing the attention mechanism. This typically involves improvements to the attention mechanism so the model can more effectively handle long-distance information. The BERT model [7] enhances text understanding through pre-training with a bidirectional Transformer. XLNet [113] improves long text processing by using permutation language modeling and generalized autoregressive pre-training to enhance the model's internal representations. Transformer-XL [99] is an improved Transformer model that introduces a recurrence mechanism to address the issue of gradient vanishing in long text processing. This allows the model to retain information from previous sequences while processing the current sequence, thereby better understanding and generating long text content.

(b) **Knowledge embedding by external knowledge base.** In practical applications, users may be unable to provide sufficient domain knowledge to enhance a general-purpose FM. To address this issue, model deployers can augment the general-purpose FM with a dedicated domain knowledge base. This method allows the general-purpose FM to reference this external knowledge base when generating answers or performing tasks, thereby obtaining the necessary domain information and context to provide more accurate and targeted responses or solutions. Retrieval-augmented generation (RAG) [1, 100, 114] technology was developed to achieve this purpose. RAG technology aims to enhance the language model's generation capabilities by leveraging an external document base without retraining the model. It is particularly suitable for tasks requiring a customizable dynamic knowledge base, such as question-answering, text summarization, and fact-checking. The core of RAG technology is the integration of a retrieval component that can quickly find information relevant to the current task in a large document database during the generation process. Once the relevant documents are retrieved, this information is used as additional contextual information to aid the generation process. The advantage of RAG technology is that it combines the generation capabilities of large language models with the knowledge provided by external retrieval systems without requiring domain knowledge themselves. Furthermore, because the external knowledge base can be replaced as needed, RAG technology offers high flexibility and adaptability. To have an intuitive understanding of RAG, we present an example in Figure 7 [115].

In RAG, the two core challenges are retrieving useful information from the database and organizing that information to augment the generation of FMs. For the retrieval task, the earliest and most naive approach uses sparse retrieval that directly matches data based on raw data like BM25 [116]. Inspired by the field of information retrieval, dense retrieval like DPR [117] has been proposed, which projects raw data into a high-dimensional space, potentially helping to capture semantic similarity better. However, a significant challenge is that even if the retrieved data is highly related to the original input in the semantic space, we cannot guarantee that it will help the model generation due to issues like polysemy. To address this problem, some researchers have introduced other technologies like knowledge graphs to aid the retrieval process [101]. Ref. [118] showed that adding noise can help enhance the model performance compared to traditional dense retrieval methods. Additionally, the organization of the retrieved information can directly influence the output quality. For example, providing a ranking of the retrieved data [102] can improve model performance.

While the aforementioned technologies were initially proposed to enhance large language models in specific domains, their applications are not limited to language models. With the development of FM domains, these technologies are expected to be extended to FMs of other modalities.

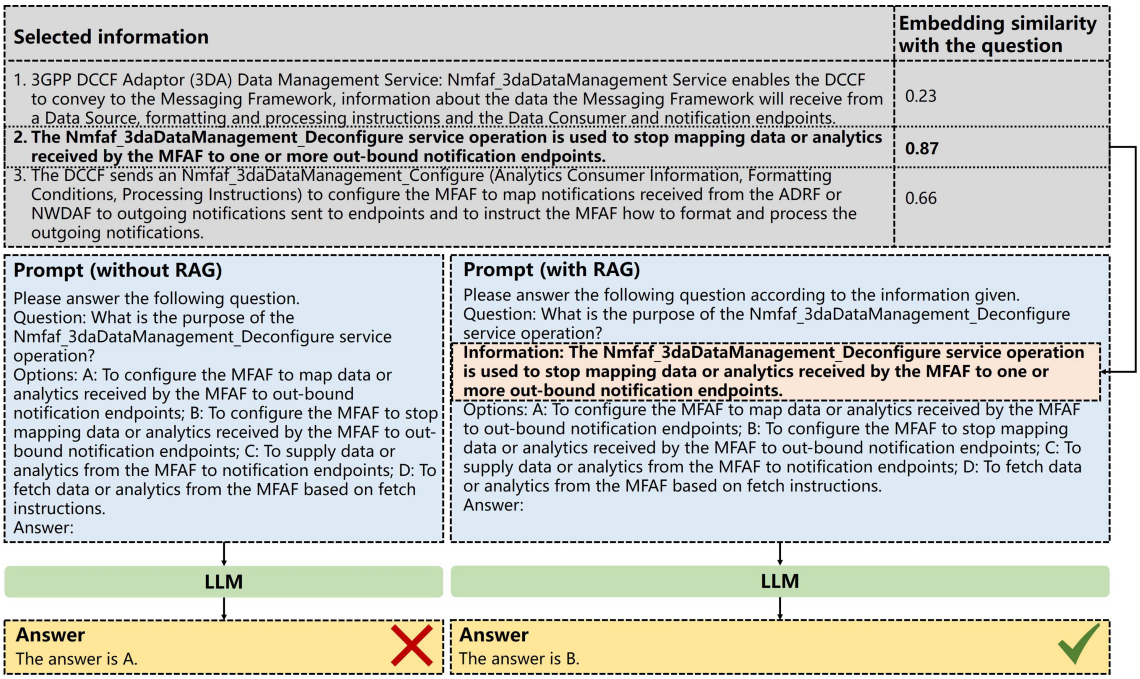

Figure 7 (Color online) Example of using LLM to answer multiple-choice communication questions. RAG improves the accuracy of answers by retrieving additional knowledge for the LLM from a knowledge base based on the relevance to the question text. The data used in the example is from [115].

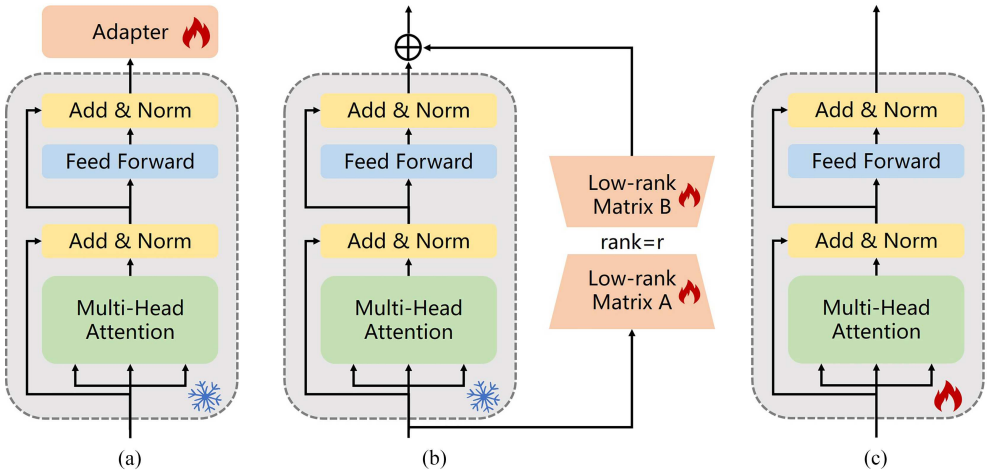

Figure 8 (Color online) Fine-tuning-based domain-specific enhancement. (a) Adapter-based tuning; (b) low-rank-decomposition-based tuning; (c) full fine-tuning.

3.1.2 *Domain-specific enhancement based on fine-tuning*

When plug-and-play domain enhancement techniques are difficult to implement or require embedding too much domain knowledge into the general-purpose FM, or when deep modifications to the general-purpose FM are necessary, we can turn to the strategy of domain enhancement based on fine-tuning. This strategy aims to customize the required domain-specific FM while preserving the pre-trained knowledge of the general-purpose FM as much as possible by specific domain enhancement [119].

Fine-tuning techniques can be divided into three main types: adapter-based fine-tuning, low-rank matrix decomposition-based fine-tuning, and full-parameter fine-tuning. Figures 8(a)–(c) illustrate these three technical pathways, respectively. In the following sections, this paper will elaborate on these techniques, sorted from low to high resource requirements and the complexity needed for fine-tuning.

(1) **Adapter-based fine-tuning.** Adapter-based fine-tuning [103] is a method that inserts small trainable adapter modules into pre-trained models, aiming to efficiently adapt the model to specific downstream tasks.

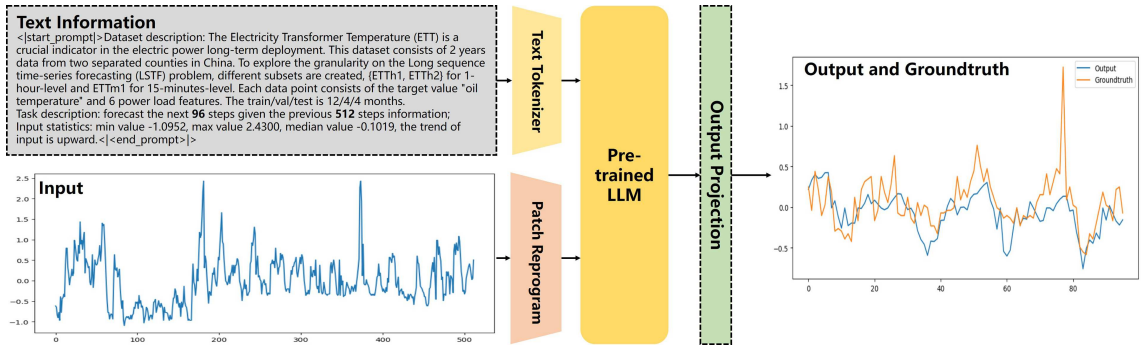

Figure 9 (Color online) Example of model reprogramming. The input time-series data is embedded by a patch reprogram module, which patches the data and then projects each patch embedding into the text space. Besides the time-series part, text information of the dataset is also inputted for LLM to better understand the prediction task. The data used in the example is from [17].

During fine-tuning, only the parameters of adapter modules are updated, while the original parameters of the pre-trained model remain unchanged, reducing computational resources and storage requirements while retaining the rich knowledge learned by the model during pre-training. AdapterFusion [104] is an extension of adapter-based fine-tuning that allows the model to learn multiple tasks or adapt to various data distributions simultaneously by fusing multiple adapter modules, with each adapter module focusing on capturing task-specific features. Infused adapter by inhibiting and amplifying inner activations (IA3) [105] scales the activation layers by injecting learned vectors into the attention and feed-forward modules of the Transformer architecture. Since these learned vectors are only trainable parameters during fine-tuning, IA3 significantly reduces the number of trainable parameters compared to traditional adapter-based fine-tuning, thereby reducing training costs and improving training efficiency. Additionally, IA3 does not introduce inference latency because its adapter weights can be merged with the FM while maintaining the flexibility and adaptability of the model, allowing customized fine-tuning for different tasks and datasets. Ref. [106] proposed a method called model reprogramming that converts both the input and output domains of an FM by adding two adapter layers, enabling more flexible adaptation. Unlike inserting adapters into each layer in the FM, model reprogramming serves as a more direct way in which only the input and output adapters are inserted upstream and downstream of the FM. During reprogramming, the pre-trained FM is frozen, which means the core of the model still works in the original domain. Such a process reduces the cost of training resources and enables the simultaneous adaptation of a single pre-trained FM to multiple domains. However, because of less trainable adapters, such a method depends more on the similarity between the source domain of the pre-trained FM and the target domain. For example, time-series data can be considered similar to text data since they all persist in a series form [17]. To better illustrate the ability of model reprogramming, we present an example in Figure 9.

(2) **Low-rank matrix decomposition based fine-tuning.** Fine-tuning based on low-rank matrix decomposition reduces the number of parameters that need to be updated by decomposing the weight matrices in the pre-trained model into the product of low-rank matrices. Low-rank matrix decomposition can capture the most important information in the weight matrices while keeping the original pre-trained parameters unchanged during fine-tuning, updating only low-rank decomposition matrices, thus reducing the computational and storage requirements during fine-tuning. This method improves fine-tuning efficiency while maintaining or approaching the performance of full-parameter fine-tuning. In low-rank matrix decomposition-based fine-tuning methods, two low-rank matrices, A and B, share the same rank r, are introduced to construct an update parameter matrix. The low-rank property of both A and B enables this process to be computationally efficient while still enabling the model to adapt effectively to new tasks.

(a) **Low-rank adaptation (LoRA).** LoRA [107] achieves efficient fine-tuning performance by introducing low-rank adaptation matrices to the pre-trained model, rather than updating the original weight matrix directly. LoRA modifies the model by adding a low-rank update to the original weights, as described by the equation:

$$W_{\text{new}} = W_{\text{old}} + \Delta W, \tag{10}$$

where W_{old} is the original weight matrix, ΔW is the low-rank update matrix, and $+$ is the point-wise addition of two matrices with the same shape. The key idea is that ΔW is constructed as the product of two low-rank matrices A and B, such that

$$\Delta W = AB. \tag{11}$$

Here, A and B are low-rank matrices with ranks typically much smaller than that of W_{old}. During fine-tuning, only the matrices A and B are updated, while W_{old} remains fixed. These low-rank matrices are learned to adapt the model efficiently while keeping the overall number of trainable parameters minimal.

One limitation of LoRA is that it typically applies the same low-rank structure to all layers, ignoring the varying importance of different layers and parameters for downstream tasks. Adaptive low-rank adaptation (AdaLoRA) [120] is an improved method based on LoRA, which adaptively determines which layer parameters need to be updated. By employing adaptive learning rates and task-specific parameter adjustment strategies, AdaLoRA enables the model to automatically adjust the intensity and scope of fine-tuning according to the specific requirements of the task.

Researchers have also found that LoRA's continuous pre-training performance is unsatisfactory on some large-scale datasets. Thus, the layerwise importance sampled AdamW (LISA) [121] strategy is proposed, where the weight norm distribution of different layers exhibits uncommon skewness. LISA adopts an importance sampling strategy by randomly activating different layers in the FM for optimization. Specifically, LISA consistently updates the bottom embedding layers and the top linear head while randomly updating a small number of intermediate self-attention layers. With memory consumption comparable to LoRA, this method outperforms LoRA and even full-parameter fine-tuning in various downstream fine-tuning tasks.

(b) **Low-rank Hadamard product (LoHa).** LoHa [108] updates the model's weights by introducing the Hadamard product of low-rank matrices. The principle of LoHa can be represented by the following equation:

$$W_{\text{new}} = W_{\text{old}} + \Delta W, \tag{12}$$

where W_{old} is the original weight matrix and ΔW is the update matrix approximated by a low-rank matrix. The update matrix ΔW can be further decomposed into the Hadamard product of two low-rank matrices:

$$\Delta W = A \odot B. \tag{13}$$

Here, A and B are two low-rank matrices that adjust elements of the original weight matrix by learning key information extracted from the input data, and $\odot$ is the Hadamard product. By updating only parameters in A and B, LoHa achieves efficient model fine-tuning while maintaining adaptability to new tasks.

(c) **Low-rank Kronecker product (LoKr).** LoKr [109] is another parameter-efficient fine-tuning method that emerged after LoHa. LoKr utilizes the properties of the Kronecker product to expand the dimensions of the weight matrix while keeping the increase in the number of parameters within a manageable range. The Kronecker product allows the model to learn complex interactions across different dimensions, particularly useful for capturing high-order relationships in the input data. The updating process of LoKr can be represented as

$$W_{\text{new}} = W_{\text{old}} + \Delta W, \tag{14}$$

where ΔW is the Kronecker product of two low-rank matrices A and B:

$$\Delta W = A \otimes B. \tag{15}$$

In this formula, two low-rank matrices, A and B, are used by the algorithm to compute a large matrix through the Kronecker product (denoted as $\otimes$), which is then updated during the fine-tuning process. LoKr is particularly suitable for tasks that require increasing the model's dimensions to capture more complex relationships while maintaining similar parameter efficiency to LoHa. However, LoKr may require more complex mathematical operations to handle the Kronecker product, and in some cases, its computational cost may be higher than that of LoHa.

(3) **Full fine-tuning.** Full fine-tuning is not constrained by pre-training tasks or data distributions, making it flexible to adapt to various downstream tasks. It allows the model to be directly optimized end-to-end on the data of the final task without the need for additional adapter modules. However, because it requires updating all parameters in the model, full fine-tuning demands significant computational resources and longer training times. Moreover, FMs have a vast number of parameters, and insufficient fine-tuning data may lead to overfitting. Additionally, the intermediate variables generated during full fine-tuning consume a large amount of GPU memory. Therefore, researchers have proposed many parameter-efficient fine-tuning methods mentioned earlier, which can reduce resource consumption and training time while maintaining performance [110].

3.2 Customization of the foundation model based on pre-trained modules

FMs may contain millions or even billions of parameters. Through domain-specific customization, it is possible to reduce parts of the model that need training, thereby significantly lowering training costs. This approach is known

as the customization of the FM based on pre-trained modules. Such customization means utilizing the knowledge embedded in model parameters during the pre-training process when constructing a new model. As mentioned earlier, in the architecture of FMs, there are typically five main modules: MEs, IPs, BCs, OPs and MDs. Among them, the ME, BC, and MD carry a large amount of knowledge as they are directly involved in data encoding, processing, and decoding. In contrast, the IP and OP themselves carry less model knowledge. In some cases, it is unnecessary to explicitly train these modules. Therefore, when customizing FMs, the IP and OP are often excluded.

Next, this paper will detail how to customize FMs based on the pre-trained ME, BC, and MD. Through this approach, we can effectively leverage pre-trained models' knowledge while reducing the demand for computational resources, making the model more suitable for specific tasks and environments.

3.2.1 *Customization of modality encoder*

General-purpose FMs often adapt to the distribution characteristics of a vast dataset during training, internalizing ample domain knowledge in their model parameters. They are thus well-suited as feature extraction modules for customized FMs. By utilizing the pre-trained model's pre-existing feature extraction module as the ME for domain data, downstream task modules can be seamlessly integrated after this module to fulfill task requirements. The ME encapsulates knowledge about crucial features of the data. There are two main approaches to customizing MEs.

• **Customization within the same modality across different data domains.** This approach involves transferring the knowledge of the ME trained on the source data domain to the target data domain. It typically entails aligning the feature distribution of the source and target domain data. Fine-tuning the pre-trained ME from the source domain on the target domain enables it to adapt to the characteristics of the new data. Specifically, this can be achieved by adjusting or adding layers to the encoder. Additionally, domain adaptation techniques such as domain adversarial training or domain-invariant feature extraction techniques can reduce the distribution discrepancy between the source and target domains.

• **Customization across modalities.** When it comes to multi-modality FMs, there are often cases that lack corresponding pre-trained encoders for certain modalities. In such situations, a cross-modality customization strategy can be utilized, adapting encoders to process new data modalities similar to the original modality. Although the modality is different, the source and target modality with the same data structure, such as natural images and thermal images, can be simultaneously fed into one ME. Such customization involves fine-tuning a pre-trained ME using task-specific datasets. For instance, ImageBind treats depth and thermal imaging data as single-channel images, using image encoders to extract features for thermal data. Initializing the model with weights pre-trained on image datasets can lead to faster convergence than random initialization and enhance generalization. This strategy is particularly useful in applications like autonomous driving, where vehicles need to process diverse data modalities, including visual, depth, thermal, and LiDAR (light detection and ranging) data. To more directly understand this process, we present another example in Figure 10 [122]. In this example, natural images captured by multi-view cameras are fed into the pre-trained vision encoder with LiDAR data.

By adapting image encoders to handle thermal imaging, the model can better understand and process thermal data, improving the overall performance of the autonomous driving system. This cross-modal customization increases the flexibility of multi-modal systems and enhances their adaptability and robustness, providing a powerful tool for solving complex problems.

3.2.2 *Customization of backbone calculator*

For multi-modality FMs, the BC is the core computational component for processing encoded feature vectors and performing tasks such as classification and generation. The BC of customized FMs can be customized from pre-trained models to leverage the complex feature processing and task execution capabilities learned from large-scale datasets. This approach avoids training the BC from scratch but requires constructing appropriate previous modules to encode the data into feature vectors that the BC can process. For example, NExT-GPT [26] converts raw data from various modalities into language modality feature vectors before feeding them into a pre-trained LLM, allowing the model to process inputted tokens according to task requirements. There are two main approaches to customize pre-trained BCs:

• **Customization from a single pre-trained BC.** Utilizing a general-purpose FM (e.g., LLaMA) as the BC to process data from the central modality. Customizing a pre-trained BC to a specific domain application typically requires fine-tuning to adapt to domain-specific data characteristics and task requirements. This step can be performed on a limited domain dataset, fine-tuning the model parameters to optimize its processing capabilities for the domain-specific data.

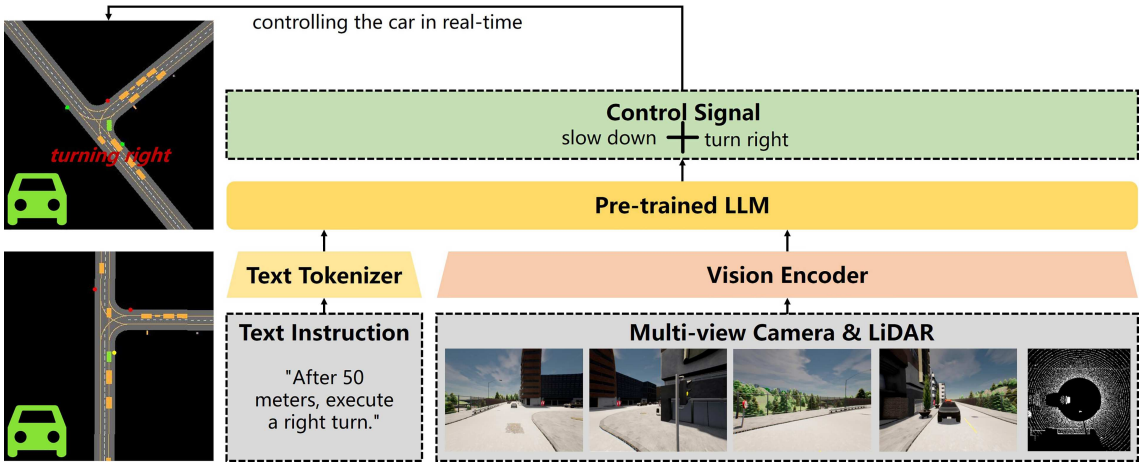

Figure 10 (Color online) Example of using LLM as the pre-trained BC to adopt to the autonomous driving area. In this example, multi-view camera data and LiDAR data are sent into a vision encoder aligned with LLM's feature space. By incorporating the user's instructions and environmental information, the LLM can generate real-time control signals for the car. The data used in the example is from [122].

• **Modular combination of multiple pre-trained BCs.** Modular combination is a flexible design method in deep learning architectures that allows integrating multiple specialized pre-trained models into a unified framework based on task requirements. The mixture of experts (MoE) model [123] can serve as an effective mechanism to further optimize this modular combination. The MoE model introduces multiple expert networks and uses a gating mechanism and mixing strategy to dynamically select and combine the outputs of these experts, enabling specialized processing for different tasks or data subsets. Recent advancements in large-scale foundation models have adopted sparse MoE architectures to achieve a favorable balance between model capacity and computational efficiency. For instance, DeepSeek-V3 employs a large-scale MoE setup with a total of 671 billion parameters, yet activates only 37 billion parameters per token using a top-2 expert routing strategy [124]. It introduces an efficient expert-balancing mechanism without auxiliary loss, and leverages a multi-token prediction objective for improved generalization. Similarly, Mixtral-8x7B [125] activates only two out of eight expert subnetworks (each a 7B model) per forward pass, enabling performance comparable to much larger dense models while significantly reducing inference costs. These examples demonstrate how sparse MoE designs have become core architectural elements in state-of-the-art foundation models, offering scalable specialization with manageable computational overhead.

The primary function of the gating mechanism is to determine how input data should be distributed among experts. It generates a weight or score for each expert based on the characteristics of the input data, reflecting each expert's capability or suitability for processing the current input. The output of the gating mechanism is typically used to guide the mixing strategy, indicating the importance of each expert for the current input. The mixing strategy then combines the outputs of multiple experts according to specific rules to generate the final model output. This strategy can be simple, such as averaging or weighted averaging, or more complex, involving probability distributions of model outputs or other advanced methods.

For example, for complex tasks requiring both image recognition and language understanding, one expert network might excel at identifying edges in images. In contrast, another expert network might be adept at understanding semantic relationships in natural language. The MoE model's gating mechanism can automatically adjust each expert network's participation level based on the input data's characteristics and the task requirement, allowing the model to flexibly invoke the most appropriate expert network when dealing with mixed visual and language inputs, achieving optimal performance. Moreover, MoE models are highly scalable. They can adapt to new task requirements or data types by adding new expert networks and updating the gating mechanism, making them suitable for constructing flexible customized FMs.

3.2.3 *Customization of modality decoder*

MDs play a crucial role in multi-modality foundation pre-trained models for converting processed feature vectors back into the form of the original data. In generative tasks, such as converting text to images or audio to text, MDs need to accurately decode the feature vectors to reconstruct understandable original data and exhibit a certain level of creativity. Some pre-trained MDs can also understand and process multi-modality feature inputs. For instance, CoDi-2 can utilize both text and audio as conditions to control image generation. By customizing such pre-trained decoders, there is no need to train complex decoder structures from scratch, allowing them to be directly applied

to image generation tasks.

Here are methods to effectively utilize pre-trained MDs for customization.

(1) **Fine-tuning pre-trained MDs.** Similar to MEs, MDs can be fine-tuned on a specific task's dataset to adapt to new task requirements. This process usually involves adjusting the last few layers of the decoder or adding new layers to better capture the data characteristics of the specific domain.

(2) **Customizing cross-modality generative MDs.** In cross-modality generation tasks, pre-trained MDs can be directly used to generate data in the target modality. The conditional information is first encoded into feature vectors through a conditional encoder and then combined with the feature vectors of the original data to achieve conditional generation. The prerequisite for this functionality is ensuring that the decoder can correctly understand input feature vectors, which may involve adjustments to the BC and OP.

3.3 Construction of the foundation model without pre-trained modules

When it is not possible to construct an FM through transferring from pre-trained models, it becomes necessary to design and train the corresponding modules. We will first provide a general analysis of the architectures of single-modality and multi-modality FMs as the foundation for constructing each component.

Single-modality FMs consist of three core modules: the ME, the BC, and the MD. For example, in LLaMA-2 [11], the ME and decoder are specifically designed for the language modality, utilizing the byte pair encoding (BPE) algorithm for encoding and decoding functions. And the BC is a massive autoregressive Transformer model. In this way, LLaMA-2 achieves a complete processing workflow of "input raw text – input text feature vectors – output text feature vectors – output raw text". Additionally, Bai et al. [13] introduced the concept of visual sentences and proposed a large visual model (LVM) that can autoregressively generate the required image based on visual sentences. It realizes in-context learning within the pure image modality, which enables the model to infer tasks directly from image modality prompts and generate corresponding results. This not only explores the potential of pure visual input but also provides a new perspective for constructing domain-specific FMs—the central modality does not have to be limited to language but any modality widely used in a specific domain.

Multi-modality FMs require the additional IPs and OPs to achieve modality alignment. For instance, CoDi-2 [21] first utilizes multiple MEs proposed in ImageBind [25] to process input data, aligning all corresponding modalities to the image modality. Then, the feature vectors of the image modality are transformed into the feature space of the language modality through an MLP. In detail, it uses the pre-trained autoregressive Transformer of the LLM LLaMA-2-7b-chat-hf as the BC's foundation. The image and audio features processed by the BC are converted back to the image domain via two MLPs. They are then used as control vector input of a diffusion-based generative model to obtain the final image and text results. The training losses include text generation, modality conversion, and data generation loss. So that the multi-modality feature processing capability of the BC and the modality conversion capability of the two MLPs can be trained simultaneously in an end-to-end way. The model's modality alignment is reflected in two terms. On the one hand, the model aligns the feature vectors of multiple modalities to the imaging modality through the pre-trained MEs of ImageBind. On the other hand, it also converts between image feature vectors and text feature vectors via the MLP.

In summary, constructing an FM begins with determining the data modalities and selecting the central modality. Next, the ME and IP are implemented to convert raw data from different modalities into central modality feature vectors, which the BC will then process. Following this, OP and MD are designed to convert the feature vectors from the BC back into the original data forms of each modality. Once the model structure is constructed, the training process can begin. In the following, we will provide a detailed introduction to each module's implementation principles and construction methods.

3.3.1 *Constructing modality encoder*

Constructing an ME means designing a neural network capable of extracting feature vectors from data. The general steps for building an ME are as follows.

(1) **Preprocessing into suitable data structures.** Choose an appropriate data structure based on the characteristics of the data modality for subsequent model usage. For example, in audio processing, a common approach is to convert time-domain signals into spectrograms and then use a neural network designed for images to extract features. Another method is to view audio as sequential vectors and use neural networks for sequences, like time series or neural language, to process it. When selecting the target data structure, researchers need to balance task requirements and processing difficulty, ensuring that the data structure represents domain knowledge sufficiently and is suitable for downstream model processing. Additionally, since domain-specific FMs need to be

Table 6 Comparison between encoder-only structure, decoder-only structure, and encoder-decoder structure.

Model architecture	Generative capability	Understanding capability	Computation	Examples
Encoder-only	Low	High	Low	BERT [7]
Decoder-only	High	Low	Low	GPT series [4–6, 22], LLaMA series [10, 11]
Encoder-decoder	High	High	High	BART [15], T5 [16]

functionally versatile, the compatibility of various task inputs should be considered when choosing the target data structure.

(2) **Designing the network architecture.** Design the network architecture according to the characteristics of the input data structure. For example, Transformer architectures can be used for text data to capture long-range dependencies, while CNN-based or ViT-based models can be employed for image data to extract features.

(3) **Training the ME.** Pre-train the ME using a dataset with sufficient quantity and variety of samples to learn the general features and distributions of the modality data. Pre-training is the process of infusing the model with knowledge. If the dataset size or diversity is inadequate, the model might not learn a complete representation of the modality data. One method to train an ME is to combine the ME and MD into an autoencoder and perform unsupervised training by minimizing reconstruction error. Another training method is designing and training a model for a specific task by supervised training, then transferring the model's upstream part as the ME. However, this method cannot get a compatible MD, which might affect the design and functionality of subsequent modules. Therefore, when designing the ME, it is essential to consider the consistency of the entire FM architecture.

3.3.2 *Constructing input projector*

The role of IPs is to project data from different modalities into a common feature space. As discussed in Subsection 2.2, modality alignment can be achieved through either fusion encoders or dual encoders. When constructing IPs, the key decision is whether to use a bridging strategy to integrate input vectors from different modalities or to use a fine-tuning approach to bring the projectors of different modalities closer together. These two strategies correspond to the concepts of fusion encoders and dual encoders, respectively. During training, using loss functions from multi-modality understanding tasks, such as multi-modality classification or generation task losses, can train the model's cross-modality projection capability. Additionally, an end-to-end training approach can optimize the overall performance of the FM and train cross-modality projections at the same time. As previously mentioned, the CoDi-2 model [21] utilizes ImageBind [25] encoders aligned by CLIP as the image and audio MEs and part of the IP. It then incorporates an MLP as another part of the IP. During the end-to-end training process of the FM, the MLP is optimized to align the image and audio to text.

3.3.3 *Constructing backbone calculator*

The BC is capable of understanding and generating feature vectors of the central modality. Constructing a BC for a specific domain begins with identifying the most common and information-rich data modality in that domain as the processed modality by the BC. The model architecture is then designed based on this modality. Currently, mainstream model architectures are based on Transformers. A complete Transformer model consists of an encoder and a decoder, where the encoder analyzes the input data to extract compact feature representations, and the decoder uses these feature representations to generate the output content. Since the structures of the encoder and decoder are different, generally, the encoder has stronger comprehension capabilities, while the decoder has more powerful generative capabilities. Transformer-based FM BCs have three main architectural forms: encoder-only architectures, decoder-only architectures, and encoder-decoder architectures. The characteristics of these three architectures are summarized in Table 6 [4–7, 10, 11, 15, 16, 22].

(1) **BC based on encoder-only architecture.** The encoder-only architecture consists solely of the encoder part of the Transformer. It is typically used for tasks that require understanding input text rather than generating new text sequences, such as text classification and sentiment analysis. Due to containing only the encoder part, the encoder model structure is relatively simple but can only produce fixed-length outputs, resulting in limited generation capability. Regarding generation tasks, BCs based on an encoder-only model can only handle tasks such as sequence completion. BERT [7] is a well-known example of an encoder-only model.

(2) **BC based on decoder-only architecture.** In the decoder-only architecture, the decoder directly processes the input sequence and generates the output sequence without a separate encoder to compactly represent the input sequence. This reduces parameter count and computational overhead but also makes it more challenging for the

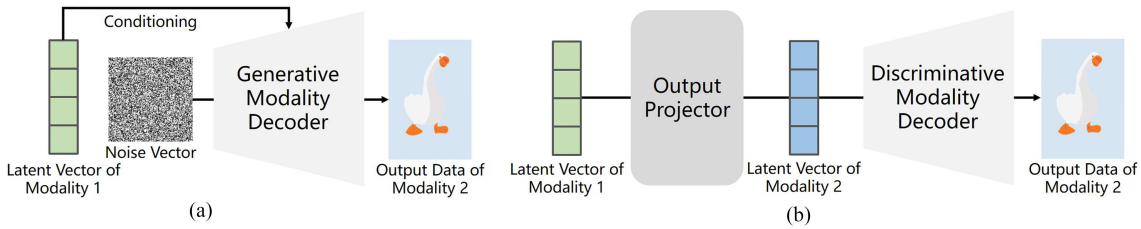

Figure 11 (Color online) Workflow of OP and MD. (a) Generative MD; (b) discriminative MD.

model to understand input sequences, thus limiting its ability to handle long sequences. This architecture does not require explicit context representation when generating output sequences but captures information automatically within the sequence through self-attention mechanisms. BCs based on decoder-only architecture typically employ autoregressive generation, meaning they generate words or characters one by one based on previously generated content to complete sequence text generation tasks. Models like the GPT series [4–6, 22] and the LLaMA series [10, 11] of foundation language models belong to the decoder-only architecture.

(3) **BC based on encoder-decoder architecture.** The encoder-decoder architecture can simultaneously possess the understanding capability of an encoder and the generation capability of a decoder, but also results in higher parameter count and computational costs. This architecture is adopted by models like Meta's BART [15] and Google's T5 [16].

3.3.4 *Constructing output projector and modality decoder*

MDs come in two types: generative and discriminative. Generative MDs can produce high-quality data samples given conditional information. Discriminative MDs, on the other hand, excel in the precise reconstruction of data samples based on input vectors. The type of MDs also affects the design of OPs. Thus it requires a joint consideration of these two modules. Figures 11(a) and (b) respectively illustrate the operational processes of generative and discriminative MDs.

(1) **Generative MD.** A generative MD utilizes a generative model as its neural network, which can control the generation process using conditional information. It takes other modality features as conditional information and generates data as decoding output. Such a generative model can be termed as a generative MD without the need for explicit construction of OPs. For instance, DiT [51], a diffusion-based image generation model, can progressively generate images based on previous results and conditional vectors, where cross-attention mechanisms serve the function of the OP. Similarly, the VAR model [126] also uses the self-attention mechanism as the OP function. By incorporating modality labels in the starting position, VAR knows which content needs to be controlled for generation and generates images in an autoregressive way.

Generative MDs typically employ end-to-end training strategies. During training, the generation part of the model and the modality interaction part are optimized simultaneously. The training objective usually involves minimizing the difference between generated data and real data while ensuring that generated data meets given conditions. For example, if the model's objective is to generate images based on textual descriptions, a large number of text-image pairs are used during training. The model is optimized by comparing the similarity between generated images and real images. This similarity can be measured using pixel-level loss functions (such as mean squared error) or more advanced perceptual losses (such as VGG loss). Additionally, adversarial training can be employed to enhance generation quality.

(2) **Discriminative MD.** A discriminative MD is only for directly reconstructing feature vectors into their original data form. Thus, it requires an explicit OP to convert feature vectors from other modalities to the target modality for use. For example, when using the decoder of VQGAN [127] as the discriminative MD in a multi-modality FM, it is necessary to first explicitly construct an OP to transform feature vectors from other modality domains to the image modality, and then use the decoder portion to decode the feature vectors into their original data form. This explicit OP is typically trained using a supervised way, where the model takes feature vectors from other modalities as input and learns to project these features into feature vectors of the target modality by minimizing the error between the two feature vectors. Besides, MDs are trained together with MEs as autoencoders, aiming to improve reconstruction performance by minimizing reconstruction errors.

Table 7 Performance comparison of telecom multiple-choice question (MCQ) answering on TeleQnA [115] dataset of mainstream LLMs and different telecom-specific LLMs obtained during different stages of TelecomGPT's training pipeline. We bold the best result within the same model family on each task.

LLMs	Lexicon	Research overview	Research publications	Standards overview	Standards specifications	Avg.
Llama3-8B	72	51.92	65.11	56.45	36.17	56.20
Llama3-8B-Instruct	80	67.31	69.77	59.68	50	64.80
Llama3-8B-TI	**96**	69.23	**74.88**	**74.19**	56.38	**71.20**
Llama3-8B-TI-TA	92	**73.08**	71.63	72.58	**58.51**	70.60
Mistral-7B	72	49.04	51.16	50	34.04	48.40
Mistral-7B-Instruct	**84**	64	65	56	51	62
Mistral-7B-TI	**84**	67.3	70.69	56.45	**51.06**	**65.2**
Mistral-7B-TI-TA	**84**	**70.19**	**73.95**	**61.29**	48.94	64
LlaMA-2-7B	62.5	52.24	49.18	48.28	40	48.94
LlaMA-2-7B-TI	**84**	57.69	63.26	56.45	**50**	59.80
LlaMA-2-7B-TP-TI	81.82	**63.92**	**67**	**70**	47.48	**63.79**

4 Applications of domain-specific foundation models

FMs have emerged as powerful tools with extensive application prospects across various domains. These FMs possess the capability not only to handle massive data and complex tasks but also to bring about new breakthroughs and innovations.

• **Telecommunications.** FMs are expected to be widely applied in sensing, transmission, and network performance optimization. For example, an LLM fine-tuned for the telecommunications domain can be used to process network log data, model, and solve specific network problems [44, 128]. Additionally, FMs in telecommunications, through the utilization of spatio-temporal correlations and knowledge reasoning [45], are expected to identify and prevent experience issues caused by degraded service quality and delayed fault response times. Recent studies also successfully incorporate telecommunication standards into LLMs' reasoning [115, 129]. As a pioneering work on how to introduce knowledge of the wireless communication field to LLMs, Ref. [130] presented a comprehensive framework that elaborates on the principles and methods for constructing a "WirelessLLM" and provides examples to demonstrate its functionality. Specifically, TelecomGPT [131] acquires domain knowledge in telecom-specific datasets through continual pre-training, instruction tuning, and alignment tuning, after which it is able to answer related questions and generate codes in the telecom domain. This lays the foundation for fine-grained network optimization in real time. We illustrate the advantages of domain-specific enhancement by presenting the results from [131], as summarized in Table 7. This table highlights the performance of LLMs that have undergone domain-specific enhancements, denoted by TP (continual pretraining), TI (instruct tuning), and TA (alignment tuning) in their names. It is evident that these enhancements lead to significant improvements in performance across various metrics. The efficiency of foundation models is crucial in use [132]. In response to the significant challenges in data and communication faced when training FMs, Ref. [133] proposed multiple new paradigms to explore how federated learning can be utilized in a wireless context to address these challenges. In industrial scenarios, the business understanding capability of FMs is also expected to help optimize signal transmission and scheduling strategies, improving network efficiency. In research on network FMs, the NetGPT architecture proposed in [134] is expected to become an effective approach to achieve endogenous intelligence in telecommunications networks, while Ref. [135] discussed the challenges and issues that may be encountered in constructing FMs for telecommunications. The NetLLM proposed in [136] adapts LLM to serve several specific downstream communication tasks. Ref. [137] introduced a new memory mechanism to facilitate the embedding of FMs into the semantic communication process. By leveraging the FMs' capability to understand and generate multimodal data, it becomes feasible to propose new and more complex transmission approaches that enhance transmission performance. In telecommunications, training domain-specific FMs on large datasets from telecommunication systems involves significant costs for data storage, processing power, and specialized hardware. Deployment costs are often high due to the need for real-time processing capabilities, especially in edge environments [138]. Additionally, ongoing operational costs arise from continuous monitoring, model updating, and adaptation to evolving network conditions.

• **Autonomous driving.** FMs play a core role in various key aspects such as vehicle perception, decision-making, and motion control [46]. Specifically, perception tasks in autonomous driving involve real-time monitoring of the vehicle's surroundings, including other vehicles, pedestrians, traffic signs, and road conditions [139]. FMs, by analyzing data collected from sensors such as cameras, radar, and LiDAR, can identify various objects and construct detailed maps of the vehicle's surroundings. This advanced perception capability is fundamental to achieving safe

Table 8 Experimental results on mathematical evaluation datasets including math problems from elementary, high school, and college levels. Math-SFT? means whether the model has been instruction-tuned on any math reasoning datasets. We bold the best result on each dataset.

Model	Base	Math-SFT?	GSM8K	MATH	AQuA	NumGLUE	Avg.
Llama-1	–	No	10.7	2.9	22.6	24.7	15.5
Llama-2	–	No	14.6	2.5	30.3	29.9	19.3
Galactica-6.7B	GAL	GAL-Instruct	10.2	2.2	25.6	25.8	15.9
Code-Llama (PoT)	–	No	25.2	13.0	24.0	26.8	22.2
AQuA-SFT	Llama-2	AQuA	11.2	3.6	35.6	12.2	15.6
Llama-1 RFT	Llama-1	GSM8K	46.5	5.2	18.8	21.1	22.9
WizardMath	Llama-2	GSM8K+MATH	54.9	10.7	26.3	36.1	32.0
MAmmoTH	Llama-2	MathInstruct	53.6	31.5	44.5	61.2	47.7
MAmmoTH-Coder	Code-Llama	MathInstruct	**59.4**	**33.4**	**47.2**	**66.4**	**51.6**

autonomous driving. At the decision-making level, FMs need to make rapid and accurate judgments based on perceived information, such as avoiding obstacles, selecting appropriate driving paths, and devising optimal driving strategies in complex traffic situations [140, 141]. Multi-modality FMs like DriveGPT [142] can not only process visual data but also understand and respond to language-mode instructions, such as planning routes based on voice input destinations. Additionally, pFedLVM [143] can utilize the powerful performance of pre-trained visual FMs for image feature extraction, serving as the basis for downstream tasks. Considering the scarcity of high-quality public datasets for training LLMs in autonomous driving scenarios, Ref. [144] employed federated instruction tuning and expanded the dataset by generating new instruction tracking data to mitigate the data scarcity. DriveMLM [139] is an LLM-based framework that can perform close-loop autonomous driving. It uses driving rules, user commands, and sensor inputs as its multi-modality LLM core inputs to model behavior planning. It standardizes decision states to connect language decisions with vehicle control commands. Training domain-specific FMs for autonomous driving requires extensive labeled data collection from real-world scenarios, which is costly and time-consuming. Additionally, the training demands substantial computational resources, particularly with large sensor data like LiDAR, radar, and camera feeds. Deploying these models involves significant expenses for hardware integration, safety certifications, and real-time vehicle processing.

• **Mathematics.** Since many general-purpose LLMs are typically pre-trained on datasets that include open-source mathematical corpora, they inherently possess some capacity to address mathematical problems, and through domain-specific enhancements, this capacity can be significantly improved [38, 145–148]. The MAmmoTH proposed in [146] combined chain-of-thought and program-of-thought, fully leveraging the understanding capability of large language models and the computational power of programming languages, achieving good performance in mathematical reasoning. To provide a clear demonstration of the benefits of domain-specific enhancements, we collect the results from [146] in Table 8. MAmmoTH-Coder shows a clear performance advantage with general-purpose open-source LLMs and is comparable with closed-source LLMs. Ref. [147] indicated through experiments that the seamless integration of LLMs' programming and reasoning abilities enables them to model and solve complex problems progressively. Due to the inherent complexity of mathematical problems and the need for vast, specialized datasets, training domain-specific foundation models in mathematics for tasks such as theorem proving or algorithm optimization can be challenging.

• **Medicine.** The applications of FMs in medicine encompass various aspects, including disease diagnosis, patient treatment, analysis and prediction of genetic and protein structure data, and medical education [40,41,149–152]. The HuatuoGPT proposed in [151] can simulate the diagnosis and treatment process of doctors, providing preliminary medical consultation and advice for patients. This model not only reduces the workload of doctors but also enables patients to receive timely medical services in remote or resource-limited environments. To illustrate the advantages of domain-specific enhancements effectively, we present part of the experimental results from [151] in Table 9 [153]. When contrasted with LLMs without medical domain-specific enhancements, HuatuoGPT demonstrates distinct comparative advantages. Ref. [152] proposed BiomedGPT, which is capable of performing multimodal medical tasks and has achieved excellent performance in understanding and generating medical-related content. In the medical area, the training datasets are often expensive to acquire due to the stringent requirements for privacy and regulatory compliance. Moreover, it is expensive to pay medical experts to annotate data.

• **Law.** FMs can conduct in-depth analysis of legal documents and identify key information and legal concepts in the text, thereby assisting lawyers and legal advisors in more precise case analysis and legal consultation [37,154,155]. For example, FMs can identify clauses in contracts, extract important legal elements such as obligations, rights, and conditions, help lawyers quickly understand document contents, and identify potential legal risks. Moreover, FMs can be used for case logical reasoning, predicting possible outcomes of cases by analyzing historical cases and

Table 9 Experimental results on Chinese medical QA dataset [153]. We bold the best result on each dataset.

Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	GLEU	ROUGE-1	ROUGE-2	ROUGE-L	Distinct-1	Distinct-2
cMedQA2	T5 (fine-tuned)	20.88	11.87	7.69	5.09	7.62	27.16	9.30	20.11	0.41	0.52
	DoctorGLM	13.51	7.10	3.72	2.00	5.11	22.78	5.68	12.22	**0.85**	0.96
	ChatGPT	19.21	7.43	3.14	1.24	5.06	20.13	3.10	12.57	0.69	**0.99**
	ChatGLM-6B	24.90	12.74	6.99	3.87	8.49	28.52	7.19	**18.21**	0.68	**0.99**
	Ziya-LLaMA-13B	27.03	13.87	7.48	4.09	7.77	28.24	7.10	14.81	0.78	0.93
	HuatuoGPT	**27.39**	**14.38**	**8.06**	**4.55**	**8.52**	**29.26**	**8.02**	15.46	0.74	0.93
webMedQA	T5 (fine-tuned)	21.42	13.79	10.06	7.38	8.94	31.00	13.85	25.78	0.37	0.46
	DoctorGLM	9.91	5.20	2.78	1.54	4.67	23.01	5.68	11.96	**0.84**	**0.95**
	ChatGPT	18.06	6.74	2.73	1.09	4.71	20.01	2.81	12.58	0.65	0.87
	ChatGLM-6B	23.42	12.10	6.73	3.83	**8.04**	**28.30**	6.87	**18.49**	0.63	0.87
	Ziya-LLaMA-13B	22.16	11.70	6.53	3.74	6.91	27.41	6.80	13.52	0.76	0.93
	HuatuoGPT	**24.85**	**13.42**	**7.72**	**4.51**	7.50	**28.30**	**7.72**	14.50	0.73	0.93
Huatuo-26M	T5 (fine-tuned)	26.63	16.74	11.77	8.46	11.38	33.21	13.26	24.85	0.51	0.68
	DoctorGLM	11.50	6.00	3.14	1.69	4.65	22.39	5.47	12.14	**0.85**	**0.96**
	ChatGPT	18.44	6.95	2.87	1.13	4.87	19.60	2.82	12.46	0.69	0.89
	ChatGLM-6B	24.46	12.75	7.20	4.13	**8.50**	28.44	7.31	**18.58**	0.67	0.89
	Ziya-LLaMA-13B	25.58	13.39	7.46	4.24	7.30	28.14	7.18	14.78	0.77	0.93
	HuatuoGPT	**27.42**	**14.84**	**8.54**	**4.96**	8.01	**29.16**	**8.29**	15.84	0.74	0.93

relevant legal provisions and providing data support for lawyers to formulate defense strategies. The ChatLaw [156] can provide real-time legal consultation and answers, helping non-professionals understand complex legal issues and even generate drafts of legal documents, reducing the workload of lawyers. Additionally, FMs can assist in legal research, quickly retrieve relevant legal literature and precedents, and provide solid evidence for legal arguments. However, LLMs can sometimes generate incorrect or fabricated information, known as "AI hallucinations", which is a crucial challenge for domain-specific FMs in law. Another key challenge is the controversy about legal liability when using AI models for legal consultation. These challenges should be carefully addressed by practitioners in the process of developing legal models.

• **Arts.** The application of FMs is exploring and changing the ways of creative expression and the process of artistic production. By learning from a large number of artworks and creative concepts, FMs can generate novel artworks, music, literary works, etc., providing creative inspiration and support for artists [42]. For example, generative models can be used to produce artworks [157, 158], and fine-tuning techniques can be applied to adjust the style and content of the generated works [159,160]. Currently, in the field of video generation, Sora, developed by OpenAI, allows users to control generated content using text, producing lifelike video works. Furthermore, FMs also show promising performance in art understanding [161,162], which provides great potential in the field of art teaching and study. Acquiring training datasets in the art domain is highly contingent upon legal restrictions, particularly concerning copyright and intellectual property. Moreover, there is significant debate within artistic ethics about the appropriateness of employing AI models. These challenges are the key factors for FMs to contribute to art.

• **Finance.** FMs can cover various tasks such as risk modeling, investment strategy management, and market forecasting, providing powerful tools for financial institutions and investors to optimize decision-making [39, 163]. For example, FMs can be used to build credit scoring systems to assess borrowers' credit risks. These models analyze factors such as borrowers' historical credit records, financial conditions, and debt levels to predict their ability to repay loans and, based on this, decide whether to approve loan applications and the loan interest rates. Alternatively, FMs can be used to consider the historical performance, correlations, risks, and expected returns of various asset classes, as well as historical market data, macroeconomic indicators, and political events, to make optimal investor decisions. BloombergGPT, as detailed in [164], was pre-trained on a large-scale financial dataset, resulting in superior performance across many benchmarks. Ref. [165] introduced FinGPT, an open-source financial LLM, and positions it not only as a model but also as an open-source framework for Financial LLMs (FinLLMs), which has the potential to fuel innovation among researchers. Using time series analysis models to predict future stock trends and other financial information is common in the financial area. However, the accuracy of these models can significantly impact investors' decisions. Additionally, for LLMs to serve as effective investment advisors, they must possess more robust numerical analysis capabilities, which presents a crucial challenge in leveraging their full potential.

5 Challenges ahead

The development of FM technology has achieved remarkable progress, but with the continuous advancement of technology, new challenges and issues arise, pointing out future research directions.

5.1 Challenges in data

Acquiring domain-specific data and modeling data structures are the top priority and most important challenges. FMs typically require vast amounts of high-quality data for training [166], which may be costly and time-consuming to obtain in specific domains. Moreover, strengthening privacy regulations imposes more restrictions on data collection and usage. To address this challenge, future research can focus on developing new data collection and annotation techniques and utilizing synthetic data and weakly supervised learning methods to reduce reliance on extensively labeled data [167, 168]. This reduces costs and effectively utilizes data resources while ensuring privacy protection. On the other hand, effective modeling of data structures in specific domains requires researchers and practitioners to deeply understand domain-specific business processes, extract key business data, and establish comprehensive data preprocessing pipelines.

Understanding multi-modality data poses another critical challenge. Although existing FMs excel in processing textual data, their understanding of other modalities, such as images and sounds, still needs improvement [36], not to mention various new data modalities that may emerge in specific domains. Constructing unified models capable of comprehensively processing various modalities of data is essential for enhancing the model's performance and generalization on multi-modality tasks. This requires researchers and practitioners to not only deeply understand the characteristics of different modalities of data but also explore effective multi-modality fusion and interaction understanding mechanisms.

The availability of aligned multi-modality data presents a potential risk for the long-term development of multi-modal models. While techniques like bridging alignment can reduce the dependency on aligned data [20, 25], the performance and generalization capabilities of multi-modal FMs still heavily rely on annotated data. This data is costly in terms of both time and financial resources. Therefore, new alignment methods are necessary to address these challenges [52].

5.2 Challenges in model architecture

In the architectural design of FMs in specific domains, a core challenge is how to build models that can effectively capture and express deep semantic information specific to different domain [20, 25, 26]. This requires models to have a broad knowledge base and understand and adapt to domain-specific knowledge and input modalities. FMs tailored for specific domains need to achieve high modularity and customizability in architecture, enabling adjustment and optimization according to specific application scenarios to adapt to the data characteristics and task requirements of different domains [4, 7, 16]. Furthermore, interpretability of models is particularly important in specific domains. When designing architectures, researchers need to consider how to construct models so that their decision-making processes and output results can be understood and trusted by domain experts and end-users. This may involve developing new model mechanisms, introducing interpretable model intermediate representations, or designing visualization tools to demonstrate the internal workings of the model.

5.3 Challenges in model training and deployment cost

The demands on computational, memory, and energy resources put significant limitations on the development of FM technologies. Therefore, their cost-intensive nature should be primarily considered when designing domain-specific FMs. Research on improving the efficiency of model training and inference [169] and reducing resource consumption has become an urgent issue. Potential research directions include developing more efficient memory-saving and acceleration techniques, such as quantization, efficient attention, and distributed learning. These technologies can help model deployers reduce costs when maintaining model performance while building and deploying a domain-specific FM.

5.3.1 *Challenges in cost-efficient training*

The impressive capabilities of FMs come at a significant cost of computing, memory, and energy during training. For example, in some Internet of Things (IoT) scenarios, such costs could be the primary concern for model deployers. On one hand, they may need to use edge devices to collect real-time data for training large models rather than centrally pre-training a model and then deploying it. On the other hand, the resource demands of training models

pose a hard constraint on edge devices. Since building domain-specific FMs is costly, it is beneficial to help model deployers better manage their costs.

Memory efficiency. FMs' huge parameter scale put a serious challenge for memory resources. Mixed precision training [170] is a common technique to convert parts of high-precision variables into low-precision dynamically implemented by automatic mixed precision [171] in each training step. This can accelerate model training and reduce GPU memory usage while minimally impacting model accuracy. A huge memory is occupied while focusing on masked training in long-sequence training. Therefore, positional encoding compression methods like ALiBi [172] and reset attention mask [173] have been proposed.

Computation efficiency. Most existing FMs have transformer-based architecture in which the attention mechanism has a powerful capability but is also computationally intensive. Researchers are now working to create efficient transformers. FlashAttention [174] significantly improves the efficiency of attention computation by IO-awareness block attention algorithm, and researchers also proposed other attention algorithms like LSH attention [175] and local attention [111]. Parallel training is another computational accelerating approach, including traditional data parallel and emerging methods like ZeRO method [176] distribute data and model parameters to achieve fast training under limited hardware constraints. Additionally, parameter-efficient fine-tuning methods [103,107] are also the key approach to improve the computation efficiency.

Communication efficiency. Federated learning [177] is applied when the training data cannot be acquired and centralized in advance due to privacy concerns. Reducing communication overhead is essential since FL involves frequent model parameter exchanges. Parameter-efficient fine-tuning methods have been combined with federated learning for efficient communication [178–181].

Energy efficiency. The energy consumption of FM training is an emerging area as it generates significant carbon footprints [182], leading to environmental impacts at the same time as increasing economic costs.

5.3.2 *Challenges in cost-efficient deployment*

The deployment of domain-specific FMs requires enormous resources, not only the training process but also the deployment of domain-specific FMs. In domain-specific business scenarios, the user's requirements for model response speed and performance may be an important consideration. To meet these demands under the resource constraints on the hardware, cost-efficient deployment strategies are highly desired, as introduced below.

Inference efficiency. The inference time of domain-specific FMs drastically increases with a huge amount of data. To solve such a problem, inference acceleration is crucial for deploying domain-specific FMs. KV cache [183] is a popular method to speed up FM inference by only generating a single row of the Q matrix. Furthermore, multi-query attention [184] could reduce the size of the KV cache, and paged attention [185] utilizes the block table to make full use of the large KV cache. In addition to the improvement on transformer architecture, speculative decoding [186] generates the next token with the help of a small "draft" model, which provides a set of token candidates for the main FM, followed by various methods such as staged speculative decoding [187], guided generation [188], lookahead decoding [189] and prompt lookup decoding [190]. Additionally, edge AI draws researchers' attention as it reduces the latency of providing AI services to devices [191–193].

Resource efficiency. The resource requirements of FMs often make it difficult to deploy them in mobile devices and edge computational scenarios [194]. To enable FMs to run in resource-constrained scenarios, lightweight model architectures and deployment strategies need to be developed. This may involve simplifying, distilling, and optimizing models to reduce their resource requirements while maintaining or improving their performance. Alternatively, employing cloud-edge collaboration strategies can distribute the training and inference processes of FMs across various server tiers, enabling a cooperative deployment. In cloud-edge learning, it is essential to consider the joint optimization of sensing, computation, and communication [195–197]. Such joint optimization for the performance of neural networks and resource consumption holds vast potential for innovation in research and can generate significant value in practical application scenarios. Furthermore, energy consumption in the inference phase is becoming the new trend within the community [198]. There are trade-offs between performance and power usage to achieve energy-efficient FM inference [199–202]. Recently, edge AI inference [203,204] also shows potential to enable various devices without adequate computing resources to access the FMs.

5.4 Challenges in security

Security issues are also challenges that FM technology must face. FMs may be used to generate false information, infringe on privacy, or be maliciously exploited, and the models themselves may also be subject to adversarial attacks [205]. To ensure the security and reliability of models, relevant work includes but is not limited to the following key points. Firstly, strengthening the robustness of models to resist potential adversarial attacks [206]. This may

involve developing advanced adversarial training techniques and implementing more stringent data cleaning and preprocessing steps. In addition to traditional adversarial training, a representative line of early work proposed certified robustness methods-such as randomized smoothing [207], which provide guarantees against perturbations. More recent studies have explored robust fine-tuning strategies, including adversarial contrastive learning and consistency regularization, to enhance generalization under distribution shifts. In parallel, robust representation learning has emerged as a promising direction, aiming to learn perturbation-invariant and semantically consistent embeddings, as exemplified by adversarial contrastive objectives in robust pre-training [208]. These complementary techniques contribute to a layered defense system that strengthens the reliability of foundation models in open environments. Secondly, developing and deploying efficient malicious input detection mechanisms, using anomaly detection algorithms and real-time monitoring systems to identify and prevent malicious behavior. Furthermore, ensuring privacy protection is achieved by diminishing the model's reliance on sensitive data, thereby safeguarding the security and privacy of user information [209]. In addition to technical efforts, enhancing security in workflow and policy aspects is also necessary. For example, implementing security audit and certification processes to comprehensively assess the security of models and ensure that model development and deployment comply with ethical and legal standards, continuously updating security policies to address emerging security threats and challenges.

6 Conclusion

This paper provides a comprehensive overview of recent advancements in domain-specific foundation models (FMs). We begin by discussing the foundational concepts of FMs, including their architectures, training methodologies, the benefits of scaling, and a comparison of their performance across different contexts. Next, we delve into the key technologies for developing domain-specific FMs, which encompass three primary approaches: enhancing general-purpose FMs with domain-specific knowledge, customizing FMs using pre-trained modules, and building FMs from scratch without relying on pre-trained components. We then explore the diverse applications of domain-specific FMs across various fields, including telecommunications, autonomous driving, mathematics, medicine, law, the arts, and finance. Following this, we address the significant challenges faced by practitioners in constructing and deploying domain-specific FMs. These challenges span multiple dimensions, such as data availability and quality, model architecture design, training processes, deployment complexities, and security concerns. By synthesizing these insights, we aim to provide researchers and practitioners with a valuable resource for navigating the development and application of domain-specific FMs. We hope this paper serves as a foundation to guide future innovation and encourage the creation of tailored FMs that address specific domain requirements effectively.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant No. 62371313), National Science and Technology Major Project (Grant No. 2024ZD1300500), Guangdong Major Project of Basic and Applied Basic Research (Grant No. 2023B0303000001), Guangdong Young Talent Research Project (Grant No. 2023TQ07A708), Shenzhen-Hong Kong-Macao Technology Research Programme (Type C) (Grant No. SGDX20230821091559018), Shenzhen Science and Technology Program (Grant No. JCYJ20241202124934046), and Young Elite Scientists Sponsorship Program by China Association for Science and Technology (Grant No. 2022QNRC001).

References

1 Ding Y, Fan W, Ning L, et al. A survey on rag meets LLMs: towards retrieval-augmented large language models. 2024. ArXiv:2405.06211
2 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations, 2015
3 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 770–778
4 Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training. 2018. https://gwern.net/doc/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf
5 Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. OpenAI Blog, 2019, 1: 9
6 Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 1877–1901
7 Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019. 4171–4186
8 Du Z, Qian Y, Liu X, et al. GLM: general language model pretraining with autoregressive blank infilling. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022. 320–335
9 Zeng A, Liu X, Du Z, et al. GLM-130B: an open bilingual pre-trained model. In: Proceedings of the 11th International Conference on Learning Representations, 2023
10 Touvron H, Lavril T, Izacard G, et al. LLaMA: open and efficient foundation language models. 2023. ArXiv:2302.13971
11 Touvron H, Martin L, Stone K, et al. LLaMA 2: open foundation and fine-tuned chat models. 2023. ArXiv:2307.09288
12 Chen M, Radford A, Child R, et al. Generative pretraining from pixels. In: Proceedings of International Conference on Machine Learning, 2020. 1691–1703
13 Bai Y, Geng X, Mangalam K, et al. Sequential modeling enables scalable learning for large vision models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 22861–22872

14 Kirillov A, Mintun E, Ravi N, et al. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 4015–4026
15 Lewis M, Liu Y, Goyal N, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. 7871–7880
16 Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res, 2020, 21: 1–67
17 Jin M, Wang S, Ma L, et al. Time-LLM: time series forecasting by reprogramming large language models. In: Proceedings of the 12th International Conference on Learning Representations, 2024
18 Gao S, Koker T, Queen O, et al. Units: building a unified time series model. 2024. ArXiv:2403.00131
19 Liu C, Yang S, Xu Q, et al. Spatial-temporal large language model for traffic prediction. In: Proceedings of the 25th IEEE International Conference on Mobile Data Management, 2024. 31–40
20 Tang Z, Yang Z, Zhu C, et al. Any-to-any generation via composable diffusion. In: Proceedings of Advances in Neural Information Processing Systems, 2024
21 Tang Z, Yang Z, Khademi M, et al. CoDi-2: in-context interleaved and interactive any-to-any generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 27425–27434
22 Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report. 2023. ArXiv:2303.08774
23 Liu H, Li C, Wu Q, et al. Visual instruction tuning. In: Proceedings of Advances in Neural Information Processing Systems, 2024
24 Fei N, Lu Z, Gao Y, et al. Towards artificial general intelligence via a multimodal foundation model. Nat Commun, 2022, 13: 3094
25 Girdhar R, El-Nouby A, Liu Z, et al. IMAGEBIND: one embedding space to bind them all. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 15180–15190
26 Wu S, Fei H, Qu L, et al. Next-GPT: any-to-any multimodal LLM. In: Proceedings of the 41st International Conference on Machine Learning, 2024
27 Zhao W X, Zhou K, Li J, et al. A survey of large language models. 2023. ArXiv:2303.18223
28 Liu J, Yang C, Lu Z, et al. Towards graph foundation models: a survey and beyond. 2023. ArXiv:2310.11829
29 Mao H, Chen Z, Tang W, et al. Graph foundation models. 2024. ArXiv:2402.02216
30 Du Y, Liu Z, Li J, et al. A survey of vision-language pre-trained models. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 5436–5443
31 Yin S, Fu C, Zhao S, et al. A survey on multimodal large language models. 2023. ArXiv:2306.13549
32 Yeh C M, Dai X, Chen H, et al. Toward a foundation model for time series data. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023. 4400–4404
33 Jin M, Wen Q, Liang Y, et al. Large models for time series and spatio-temporal data: a survey and outlook. 2023. ArXiv:2310.10196
34 Cao Y, Li S, Liu Y, et al. A comprehensive survey of AI-generated content (AIGC): a history of generative AI from GAN to ChatGPT. 2023. ArXiv:2303.04226
35 Zhou C, Li Q, Li C, et al. A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT. 2023. ArXiv:2302.09419
36 Zhang D, Yu Y, Dong J, et al. MM-LLMs: recent advances in multimodal large language models. In: Proceedings of Findings of the Association for Computational Linguistics, 2024. 12401–12430
37 Lai J, Gan W, Wu J, et al. Large language models in law: a survey. AI Open, 2024, 5: 181–196
38 Ahn J, Verma R, Lou R, et al. Large language models for mathematical reasoning: progresses and challenges. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, 2024. 225–237
39 Li Y, Wang S, Ding H, et al. Large language models in finance: a survey. In: Proceedings of the 4th ACM International Conference on AI in Finance, 2023. 374–382
40 Thirunavukarasu A J, Ting D S J, Elangovan K, et al. Large language models in medicine. Nat Med, 2023, 29: 1930–1940
41 Kazerouni A, Aghdam E K, Heidari M, et al. Diffusion models in medical imaging: a comprehensive survey. Med Image Anal, 2023, 88: 102846
42 Shahriar S. GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network. Displays, 2022, 73: 102237
43 Hou X, Zhao Y, Liu Y, et al. Large language models for software engineering: a systematic literature review. 2023. ArXiv:2308.10620
44 Bariah L, Zhao Q, Zou H, et al. Large generative AI models for telecom: the next big thing? IEEE Commun Mag, 2024, 62: 84–90
45 Zhou H, Hu C, Yuan Y, et al. Large language model (LLM) for telecommunications: a comprehensive survey on principles, key techniques, and opportunities. 2024. ArXiv:2405.10825
46 Cui C, Ma Y, Cao X, et al. A survey on multimodal large language models for autonomous driving. In: Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, 2024. 958–979
47 Yang J, Jin H, Tang R, et al. Harnessing the power of LLMs in practice: a survey on ChatGPT and beyond. ACM Trans Knowl Discov Data, 2024, 18: 1–32
48 Gui J, Chen T, Zhang J, et al. A survey on self-supervised learning: algorithms, applications, and future trends. IEEE Trans Pattern Anal Mach Intell, 2024, 46: 9052–9071
49 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017
50 Su W, Zhu X, Cao Y, et al. VL-BERT: pre-training of generic visual-linguistic representations. In: Proceedings of the 8th International Conference on Learning Representations, 2020
51 Peebles W, Xie S. Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 4195–4205
52 Du C, Wang Y, Song S, et al. Probabilistic contrastive learning for long-tailed visual recognition. IEEE Trans Pattern Anal Mach Intell, 2024, 46: 5890–5904
53 Tay Y, Dehghani M, Abnar S, et al. Scaling laws vs model architectures: how does inductive bias influence scaling? In: Proceedings of Findings of the Association for Computational Linguistics, 2023. 12342–12364
54 Zhai X, Kolesnikov A, Houlsby N, et al. Scaling vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 12104–12113
55 Kaplan J, McCandlish S, Henighan T, et al. Scaling laws for neural language models. 2020. ArXiv:2001.08361

56 Chung H W, Hou L, Longpre S, et al. Scaling instruction-finetuned language models. J Mach Learn Res, 2024, 25: 1–53
57 Iyer S, Lin X V, Pasunuru R, et al. OPT-IML: scaling language model instruction meta learning through the lens of generalization. 2022. ArXiv:2212.12017
58 Henighan T, Kaplan J, Katz M, et al. Scaling laws for autoregressive generative modeling. 2020. ArXiv:2010.14701
59 Wei J, Tay Y, Bommasani R, et al. Emergent abilities of large language models. 2022. ArXiv:2206.07682
60 Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 24824–24837
61 Agrawal H, Desai K, Wang Y, et al. Nocaps: novel object captioning at scale. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 8948–8957
62 Young P, Lai A, Hodosh M, et al. From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Transa Assoc Comput Linguist, 2014, 2: 67–78
63 Karpathy A, Li F F. Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3128–3137
64 Goyal Y, Khot T, Summers-Stay D, et al. 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, 2017. 6904–6913
65 Hudson D A, Manning C D. 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, 2019. 6700–6709
66 Gurari D, Li Q, Stangl A J, et al. Vizwiz grand challenge: answering visual questions from blind people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3608–3617
67 Lu P, Mishra S, Xia T, et al. Learn to explain: multimodal reasoning via thought chains for science question answering. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 2507–2521
68 Singh A, Natarajan V, Shah M, et al. Towards VQA models that can read. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 8317–8326
69 Marino K, Rastegari M, Farhadi A, et al. OK-VQA: a visual question answering benchmark requiring external knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 3195–3204
70 Li Y, Du Y, Zhou K, et al. Evaluating object hallucination in large vision-language models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2023. 292–305
71 Liu Y, Duan H, Zhang Y, et al. MMBench: is your multi-modal model an all-around player? In: Proceedings of European Conference on Computer Vision, 2025. 216–233
72 Li B, Ge Y, Ge Y, et al. Seed-bench: benchmarking multimodal large language models. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 13299–13308
73 Yu W, Yang Z, Li L, et al. MM-VET: evaluating large multimodal models for integrated capabilities. In: Proceedings of the 41st International Conference on Machine Learning, 2024. 57730–57754
74 Wu H, Zhang Z, Zhang E, et al. Q-Bench: a benchmark for general-purpose foundation models on low-level vision. In: Proceedings of the 12th International Conference on Learning Representations, 2024
75 Ye Q, Xu H, Xu G, et al. MPLUG-OWL: modularization empowers large language models with multimodality. 2023. ArXiv:2304.14178
76 Sun Q, Yu Q, Cui Y, et al. EMU: generative pretraining in multimodality. In: Proceedings of the 12th International Conference on Learning Representations, 2024
77 Laurençon H, Saulnier L, Tronchon L, et al. Obelics: an open web-scale filtered dataset of interleaved image-text documents. In: Proceedings of Advances in Neural Information Processing Systems, 2024
78 Chu X, Qiao L, Lin X, et al. MobileVLM: a fast, reproducible and strong vision language assistant for mobile devices. 2023. ArXiv:2312.16886
79 Kan K B, Mun H, Cao G, et al. Mobile-LLaMA: instruction fine-tuning open-source LLM for network analysis in 5G networks. IEEE Netw, 2024, 38: 76–83
80 Dong R, Han C, Peng Y, et al. DreamLLM: synergistic multimodal comprehension and creation. In: Proceedings of the 12th International Conference on Learning Representations, 2024
81 Chiang W L, Li Z, Lin Z, et al. Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. 2023. https://vicuna.lmsys.org
82 Lin B, Ye Y, Zhu B, et al. Video-LLaVA: learning united visual representation by alignment before projection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024. 5971–5984
83 Chen L, Li J, Dong X, et al. ShareGPT4V: improving large multi-modal models with better captions. In: Proceedings of European Conference on Computer Vision, 2025. 370–387
84 Liu H, Li C, Li Y, et al. Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 26296–26306
85 Dai W, Li J, Li D, et al. InstructBLIP: towards general-purpose vision-language models with instruction tuning. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 49250–49267
86 Chen K, Zhang Z, Zeng W, et al. Shikra: unleashing multimodal LLM's referential dialogue magic. 2023. ArXiv:2306.15195
87 Chen Z, Wu J, Wang W, et al. InternVL: scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 24185–24198
88 Bai J, Bai S, Chu Y, et al. Qwen technical report. 2023. ArXiv:2309.16609
89 Bai J, Bai S, Yang S, et al. Qwen-VL: a frontier large vision-language model with versatile abilities. 2023. ArXiv:2308.12966
90 Yuan Z, Li Z, Huang W, et al. TinyGPT-V: efficient multimodal large language model via small backbones. In: Proceedings of the 2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ ICML 2024), 2024
91 Javaheripi M, Bubeck S, Abdin M, et al. Phi-2: the surprising power of small language models. Microsoft Research Blog, 2023, 1: 3
92 Zhu Y, Zhu M, Liu N, et al. LLaVA-Phi: efficient multi-modal assistant with small language model. In: Proceedings of the 1st International Workshop on Efficient Multimedia Computing under Limited, 2024. 18–22
93 Li J, Li D, Xiong C, et al. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of International Conference on Machine Learning, 2022. 12888–12900
94 Lin B, Tang Z, Ye Y, et al. MoE-LLaVA: mixture of experts for large vision-language models. 2024. ArXiv:2401.15947

95 Schick T, Schütze H. Exploiting cloze questions for few shot text classification and natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, 2021. 255–269
96 Li X L, Liang P. Prefix-tuning: optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021. 4582–4597
97 Liu X, Zheng Y, Du Z, et al. GPT understands, too. AI Open, 2024, 5: 208–215
98 Ding Y, Zhang L L, Zhang C, et al. LongRoPE: extending LLM context window beyond 2 million tokens. In: Proceedings of the 41st International Conference on Machine Learning, 2024
99 Dai Z, Yang Z, Yang Y, et al. Transformer-XL: attentive language models beyond a fixed-length context. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019. 2978–2988
100 Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 9459–9474
101 Edge D, Trinh H, Cheng N, et al. From local to global: a graph RAG approach to query-focused summarization. 2024. ArXiv:2404.16130
102 Yu Y, Ping W, Liu Z, et al. RankRAG: unifying context ranking with retrieval-augmented generation in LLMs. In: Proceedings of the 38th Annual Conference on Neural Information Processing Systems, 2024
103 Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP. In: Proceedings of International Conference on Machine Learning, 2019. 2790–2799
104 Pfeiffer J, Kamath A, Rücklé A, et al. Adapterfusion: non-destructive task composition for transfer learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, 2021. 487–503
105 Liu H, Tam D, Muqeeth M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 1950–1965
106 Chen P Y. Model reprogramming: resource-efficient cross-domain machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 22584–22591
107 Hu E J, Shen Y, Wallis P, et al. LoRA: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations, 2022
108 Kim J, On K W, Lim W, et al. Hadamard product for low-rank bilinear pooling. In: Proceedings of the 5th International Conference on Learning Representations, 2017
109 Hackbusch W, Khoromskij B N. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions. Computing, 2006, 76: 177–202
110 Ding N, Qin Y, Yang G, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat Mach Intell, 2023, 5: 220–235
111 Beltagy I, Peters M E, Cohan A. Longformer: the long-document transformer. 2020. ArXiv:2004.05150
112 Zaheer M, Guruganesh G, Dubey K A, et al. Big bird: transformers for longer sequences. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 17283–17297
113 Yang Z, Dai Z, Yang Y, et al. XLNet: generalized autoregressive pretraining for language understanding. In: Proceedings of Advances in Neural Information Processing Systems, 2019
114 Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: a survey. 2023. ArXiv:2312.10997
115 Maatouk A, Ayed F, Piovesan N, et al. TeleQnA: a benchmark dataset to assess large language models telecommunications knowledge. 2023. ArXiv:2310.15051
116 Chen D, Fisch A, Weston J, et al. Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017. 1870–1879
117 Karpukhin V, Oguz B, Min S, et al. Dense passage retrieval for open-domain question answering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. 6769–6781
118 Cuconasu F, Trappolini G, Siciliano F, et al. The power of noise: redefining retrieval for RAG systems. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024. 719–729
119 Han Z, Gao C, Liu J, et al. Parameter-efficient fine-tuning for large models: a comprehensive survey. 2024. ArXiv:2403.14608
120 Zhang Q, Chen M, Bukharin A, et al. Adaptive budget allocation for parameter-efficient fine-tuning. In: Proceedings of the 11th International Conference on Learning Representations, 2022
121 Pan R, Liu X, Diao S, et al. LISA: layerwise importance sampling for memory-efficient large language model fine-tuning. 2024. ArXiv:2403.17919
122 Shao H, Hu Y, Wang L, et al. LMDrive: closed-loop end-to-end driving with large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 15120–15130
123 Ma J, Zhao Z, Yi X, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 1930–1939
124 Liu A, Feng B, Xue B, et al. DeepSeek-V3 technical report. 2024. ArXiv:2412.19437
125 Jiang A Q, Sablayrolles A, Roux A, et al. Mixtral of experts. 2024. ArXiv:2401.04088
126 Tian K, Jiang Y, Yuan Z, et al. Visual autoregressive modeling: scalable image generation via next-scale prediction. 2024. ArXiv:2404.02905
127 Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 12873–12883
128 Cui Q M, You X H, Wei N, et al. Overview of AI and communication for 6G network: fundamentals, challenges, and future research opportunities. Sci China Inf Sci, 2025, 68: 171301
129 Chen T, Chen J, Zhao Z, et al. First token probability guided RAG for telecom question answering. 2025. ArXiv:2501.06468
130 Shao J, Tong J, Wu Q, et al. WirelessLLM: empowering large language models towards wireless intelligence. J Commun Inf Netw, 2024, 9: 99–112
131 Zou H, Zhao Q, Tian Y, et al. TelecomGPT: a framework to build telecom-specfic large language models. 2024. ArXiv:2407.09424
132 Bommasani R, Liang P, Lee T. Holistic evaluation of language models. Ann New York Acad Sci, 2023, 1525: 140–146
133 Chen Z, Yang H H, Tay Y C, et al. The role of federated learning in a wireless world with foundation models. IEEE Wireless Commun, 2024, 31: 42–49
134 Chen Y, Li R, Zhao Z, et al. NetGPT: an AI-native network architecture for provisioning beyond personalized generative services. IEEE Netw, 2024, 38: 404–413

135 Tong W, Peng C, Yang T, et al. Ten issues of NetGPT. 2023. ArXiv:2311.13106
136 Wu D, Wang X, Qiao Y, et al. Large language model adaptation for networking. 2024. ArXiv:2402.02338
137 Xie H, Qin Z, Tao X, et al. Toward intelligent communications: large model empowered semantic communications. IEEE Commun Mag, 2025, 63: 69–75
138 You X H, Huang Y M, Zhang C, et al. When AI meets sustainable 6G. Sci China Inf Sci, 2025, 68: 110301
139 Wang W, Xie J, Hu C, et al. DriveMLM: aligning multi-modal large language models with behavioral planning states for autonomous driving. 2023. ArXiv:2312.09245
140 Cui Y, Huang S, Zhong J, et al. DriveLLM: charting the path toward full autonomous driving with large language models. IEEE Trans Intell Veh, 2024, 9: 1450–1464
141 Fu D, Li X, Wen L, et al. Drive like a human: rethinking autonomous driving with large language models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024. 910–919
142 Xu Z, Zhang Y, Xie E, et al. DriveGPT4: interpretable end-to-end autonomous driving via large language model. IEEE Robot Autom Lett, 2024, 9: 8186–8193
143 Kou W B, Lin Q, Tang M, et al. PFEDLVM: a large vision model (LVM)-driven and latent feature-based personalized federated learning framework in autonomous driving. 2024. ArXiv:2405.04146
144 Chen J, He J, Chen F, et al. Empowering IoT-based autonomous driving via federated instruction tuning with feature diversity. IEEE Int Things J, 2025, 12: 6095–6108
145 Yang Z, Ding M, Lv Q, et al. GPT can solve mathematical problems without a calculator. 2023. ArXiv:2309.03241
146 Yue X, Qu X, Zhang G, et al. MAmmoTH: building math generalist models through hybrid instruction tuning. In: Proceedings of the 12th International Conference on Learning Representations, 2024
147 Wang K, Ren H, Zhou A, et al. MathCoder: seamless code integration in LLMs for enhanced mathematical reasoning. In: Proceedings of the 12th International Conference on Learning Representations, 2024
148 Wu Y, Jia F, Zhang S, et al. MathChat: converse to tackle challenging math problems with LLM agents. In: Proceedings of ICLR 2024 Workshop on Large Language Model (LLM) Agents, 2024
149 Garg R K, Urs V L, Agrawal A A, et al. Exploring the role of ChatGPT in patient care (diagnosis and treatment) and medical research: a systematic review. Health Promot Perspect, 2023, 13: 183–191
150 Xi Z, Chen W, Guo X, et al. The rise and potential of large language model based agents: a survey. 2023. ArXiv:2309.07864
151 Zhang H, Chen J, Jiang F, et al. HuatuoGPT, towards taming language model to be a doctor. In: Proceedings of Findings of the Association for Computational Linguistics, 2023. 10859–10885
152 Zhang K, Yu J, Yan Z, et al. BiomedGPT: a unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. 2023. ArXiv:2305.17100
153 Li J, Wang X, Wu X, et al. Huatuo-26m, a large-scale Chinese medical QA dataset. 2024. ArXiv:2305.01526
154 Zhou Z, Shi J X, Song P X, et al. LawGPT: a Chinese legal knowledge-enhanced large language model. 2024. ArXiv:2406.04614
155 Fei Z, Zhang S, Shen X, et al. InternLM-Law: an open source Chinese legal large language model. 2024. ArXiv:2406.14887
156 Cui J, Li Z, Yan Y, et al. ChatLaw: open-source legal large language model with integrated external knowledge bases. 2023. ArXiv:2306.16092
157 Schumacher D, LaBounty Jr F, Schellekens M. Enhancing Suno's bark text-to-speech model: addressing limitations through Meta's encodec and pre-trained hubert. SSRN 4443815, 2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4443815
158 Liang X, Zhao Z, Zeng W, et al. PianoBART: symbolic piano music generation and understanding with large-scale pre-training. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2024. 1–6
159 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 6840–6851
160 Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 3836–3847
161 Zhao Z. Adversarial-MidiBERT: symbolic music understanding model based on unbias pre-training and mask fine-tuning. 2024. ArXiv:2407.08306
162 Garcia N, Vogiatzis G. How to read paintings: semantic art understanding with multi-modal retrieval. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018
163 Lopez-Lira A, Tang Y. Can ChatGPT forecast stock price movements? Return predictability and large language models. 2023. ArXiv:2304.07619
164 Wu S, Irsoy O, Lu S, et al. BloombergGPT: a large language model for finance. 2023. ArXiv:2303.17564
165 Yang H, Liu X Y, Wang C D. FinGPT: open-source financial large language models. 2023. ArXiv:2306.06031
166 Du F, Ma X J, Yang J R, et al. A survey of LLM datasets: from autoregressive model to AI Chatbot. J Comput Sci Technol, 2024, 39: 542–566
167 Zhou Z H. A brief introduction to weakly supervised learning. Natl Sci Rev, 2018, 5: 44–53
168 van Engelen J E, Hoos H H. A survey on semi-supervised learning. Mach Learn, 2020, 109: 373–440
169 Wang Z, Zhou Y, Shi Y, et al. Federated fine-tuning for pre-trained foundation models over wireless networks. 2024. ArXiv:2407.02924
170 Micikevicius P, Narang S, Alben J, et al. Mixed precision training. In: Proceedings of International Conference on Learning Representations, 2018
171 Lam M O, Hollingsworth J K, de Supinski B R, et al. Automatically adapting programs for mixed-precision floating-point computation. In: Proceedings of the 27th International ACM Conference on International Conference on Supercomputing, 2013. 369–378
172 Press O, Smith N, Lewis M. Train short, test long: attention with linear biases enables input length extrapolation. In: Proceedings of International Conference on Learning Representations, 2022
173 Dubey A, Jauhri A, Pandey A, et al. The LLaMA 3 herd of models. 2024. ArXiv:2407.21783
174 Dao T, Fu D, Ermon S, et al. Flashattention: fast and memory-efficient exact attention with IO-awareness. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 16344–16359
175 Kitaev N, Kaiser L, Levskaya A. Reformer: the efficient transformer. In: Proceedings of International Conference on Learning Representations, 2020
176 Rajbhandari S, Rasley J, Ruwase O, et al. Zero: memory optimizations toward training trillion parameter models. In: Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, 2020. 1–16

177 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Artificial Intelligence and Statistics, 2017. 1273–1282
178 Sun Y, Li Z, Li Y, et al. Improving LoRA in privacy-preserving federated learning. In: Proceedings of the 12th International Conference on Learning Representations, 2024
179 Cho Y J, Liu L, Xu Z, et al. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024. 12903–12913
180 Zhang Z, Yang Y, Dai Y, et al. FedPETuning: when federated learning meets the parameter-efficient tuning methods of pre-trained language models. In: Proceedings of Annual Meeting of the Association of Computational Linguistics, 2023. 9963–9977
181 Kim Y, Kim J, Mok W L, et al. Client-customized adaptation for parameter-efficient federated learning. In: Proceedings of Findings of the Association for Computational Linguistics, 2023. 1159–1172
182 Gupta U, Elgamal M, Hills G, et al. ACT: designing sustainable computer systems with an architectural carbon modeling tool. In: Proceedings of the 49th Annual International Symposium on Computer Architecture, 2022. 784–799
183 Luohe S, Zhang H, Yao Y, et al. Keep the cost down: a review on methods to optimize LLM's KV-cache consumption. In: Proceedings of the 1st Conference on Language Modeling, 2024
184 Xu Y, Li X, Yuan H, et al. Multi-task learning with multi-query transformer for dense prediction. IEEE Trans Circ Syst Video Technol, 2024, 34: 1228–1240
185 Kwon W, Li Z, Zhuang S, et al. Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th Symposium on Operating Systems Principles, 2023. 611–626
186 Leviathan Y, Kalman M, Matias Y. Fast inference from transformers via speculative decoding. In: Proceedings of International Conference on Machine Learning, 2023. 19274–19286
187 Spector B F, Re C. Accelerating LLM inference with staged speculative decoding. In: Proceedings of Workshop on Efficient Systems for Foundation Models, 2023
188 Willard B T, Louf R. Efficient guided generation for large language models. 2023. ArXiv:2307.09702
189 Fu Y, Bailis P, Stoica I, et al. Break the sequential dependency of LLM inference using lookahead decoding. In: Proceedings of the 41st International Conference on Machine Learning, 2024
190 Saxena A. Prompt lookup decoding. 2023. https://github.com/apoorvumang/prompt-lookup-decoding
191 Xing H, Zhu G, Liu D, et al. Task-oriented integrated sensing, computation and communication for wireless edge AI. IEEE Netw, 2023, 37: 135–144
192 Yang B, Cao X, Yuen C, et al. Offloading optimization in edge computing for deep-learning-enabled target tracking by Internet of UAVs. IEEE Int Things J, 2020, 8: 9878–9893
193 Yang B, Cao X, Xiong K, et al. Edge intelligence for autonomous driving in 6G wireless system: design challenges and solutions. IEEE Wireless Commun, 2021, 28: 40–47
194 Yin W, Xu M, Li Y, et al. LLM as a system service on mobile devices. 2024. ArXiv:2403.11805
195 Xu W, Yang Z, Ng D W K, et al. Edge learning for B5G networks with distributed signal processing: semantic communication, edge computing, and wireless sensing. IEEE J Sel Top Signal Process, 2023, 17: 9–39
196 Zhu G X, Lyu Z H, Jiao X, et al. Pushing AI to wireless network edge: an overview on integrated sensing, communication, and computation towards 6G. Sci China Inf Sci, 2023, 66: 130301
197 Wen D, Liu P, Zhu G, et al. Task-oriented sensing, computation, and communication integration for multi-device edge AI. IEEE Trans Wireless Commun, 2024, 23: 2486–2502
198 Ding Y, Shi T. Sustainable LLM serving: environmental implications, challenges, and opportunities. In: Proceedings of the 15th International Green and Sustainable Computing Conference (IGSC), 2024. 37–38
199 Hisaharo S, Nishimura Y, Takahashi A. Optimizing LLM inference clusters for enhanced performance and energy efficiency. Authorea Preprints, 2024. https://www.techrxiv.org/doi/full/10.36227/techrxiv.172348951.12175366
200 Argerich M F, Patiño-Martínez M. Measuring and improving the energy efficiency of large language models inference. IEEE Access, 2024, 12: 80194–80207
201 Liu S, Wen D, Li D, et al. Energy-efficient optimal mode selection for edge AI inference via integrated sensing-communication-computation. IEEE Trans Mobile Comput, 2024, 23: 14248–14262
202 Stojkovic J, Choukse E, Zhang C, et al. Towards Greener LLMs: bringing energy-efficiency to the forefront of LLM inference. 2024. ArXiv:2403.20306
203 Zhuang Z, Wen D, Shi Y, et al. Integrated sensing-communication-computation for over-the-air edge AI inference. IEEE Trans Wireless Commun, 2024, 23: 3205–3220
204 Zhang P, Wen D, Zhu G, et al. Collaborative edge AI inference over cloud-RAN. IEEE Trans Commun, 2024, 72: 5641–5656
205 Yao Y, Duan J, Xu K, et al. A survey on large language model (LLM) security and privacy: the good, the bad, and the ugly. High-Confidence Comput, 2024, 4: 100211
206 Zou A, Wang Z, Kolter J Z, et al. Universal and transferable adversarial attacks on aligned language models. 2023. ArXiv:2307.15043
207 Cohen J, Rosenfeld E, Kolter Z. Certified adversarial robustness via randomized smoothing. In: Proceedings of International Conference on Machine Learning, 2019. 1310–1320
208 Jiang Z, Chen T, Chen T, et al. Robust pre-training by adversarial contrastive learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 16199–16210
209 Feng Q, Kasa S R, Yun H, et al. Exposing privacy GAPs: membership inference attack on preference data for LLM alignment. 2024. ArXiv:2407.06443