IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2025, accepted 19 March 2025, date of publication 26 March 2025, date of current version 4 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3554586

SURVEY

Distilling Wisdom: A Review on Optimizing Learning From Massive Language Models

DINGZONG ZHANG[1], DEVI LISTIYANI[1], PRIYANKA SINGH[1], AND MANORANJAN MOHANTY[2]

[1]School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane, QLD 4072, Australia
[2]School of Electrical Engineering and Computer Science, Carnegie Mellon University in Qatar, Ar-Rayyan, Qatar

Corresponding author: Devi Listiyani (d.listiyani@student.uq.edu.au)

This work was funded by the Indonesia Endowment Fund for Education (LPDP) Scholarship.

ABSTRACT In the era of Large Language Models (LLMs), Knowledge Distillation (KD) enables the transfer of capabilities from proprietary LLMs to open-source models. This survey provides a detailed discussion of the basic principles, algorithms, and implementation methods of knowledge distillation. It explores KD's impact on LLMs, emphasizing its utility in model compression, performance enhancement, and self-improvement. Through the analysis of practical examples such as DistilBERT, TinyBERT, and MobileBERT, the paper demonstrates how knowledge distillation can markedly enhance the efficiency and applicability of large language models in real-world scenarios. The discussion encompasses the varied applications of KD across multiple domains, including industrial systems, embedded systems, Natural Language Processing (NLP), multi-modal processing, and vertical domains, such as medicine, law, science, finance, and materials science. This survey outlines current KD methodologies and future research directions, highlighting its role in advancing AI technologies and fostering innovation across different sectors.

INDEX TERMS Artificial intelligence (AI), large language model (LLM), knowledge distillation (KD), optimization.

I. INTRODUCTION

Proprietary LLMs like GPT-3.5 [1], GPT-4 [2], Gemini [3], and Claude2 have become groundbreaking technologies in the rapidly changing field of artificial intelligence (AI), profoundly altering our understanding of natural language processing(NLP). These models, which stand out for their enormous size and complexity, have opened up new possibilities. They can now produce writing that resembles that of a person and have advanced problem-solving abilities. Their primary importance is found in their emergent skills [4], where they exhibit capacities above and beyond their stated training goals, allowing them to do a wide range of activities with impressive efficacy. Their deep understanding of context, nuance, and the intricacies of human language enable them to excel in a wide array of applications, from creative content generation to complex problem-solving [2].

Despite these advantages of proprietary models, users may still prefer open-source alternatives, even though they exhibit certain limitations, such as suboptimal performance [5]. To mitigate these performance issues, knowledge distillation can be utilized to transfer sophisticated capabilities from proprietary LLMs to open-source models [6], [7]. Knowledge distillation is a model compression technique in which a smaller model (student model) learns from a larger model (teacher model) to enhance its performance [8], as illustrated in Figure 1. Open-source models can also use knowledge distillation for self-improvement by employing themselves as teacher models to improve their performance continuously [7]. Figure 2 illustrates the role of knowledge distillation in LLMs.

A. EVOLUTION

While private LLMs like GPT-4 [2] and Gemini [3] have amazing capabilities, they are not without drawbacks. Their greater cost, restricted accessibility, and adaptability are major disadvantages [2]. These proprietary models are more

The associate editor coordinating the review of this manuscript and approving it for publication was Olarik Surinta.

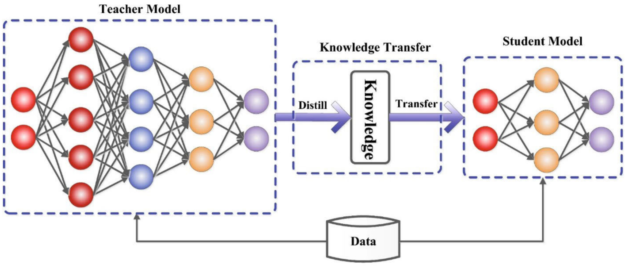


FIGURE 1. **Teacher–student framework for KD [6].**

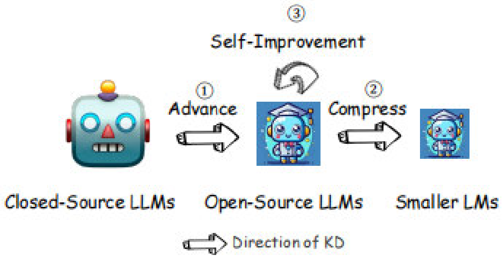


FIGURE 2. **KD plays three key roles in LLMs [7].**

expensive for individuals and smaller organisations to use and frequently have restricted access. Concerns concerning data privacy and security are raised by the fact that using these proprietary LLMs frequently requires transferring sensitive data to external servers [9]. This particular component is of particular importance for users who handle sensitive data. The general-purpose architecture of proprietary LLMs does not always match the particular requirements of specialised applications.

In contrast, open-source models such as LLaMA [10] and Mistral [11] offer a number of noteworthy advantages over proprietary LLMs. The accessibility and adaptability of open-source models is one of its main advantages. These models are more easily accessible to a wider range of users, from lone researchers to smaller organisations, since they are not restricted by licencing costs or usage guidelines. This transparency encourages innovation and a wide range of applications by creating a more inclusive and collaborative research environment for AI. Furthermore, because open-source LLMs are customisable, more specialised solutions can be created to fulfil certain requirements that may not be addressed by large-scale, generic models.

However, open-source LLMs also have a set of disadvantages of their own, primarily due to their relatively smaller size and resources in comparison to their proprietary equivalents. The smaller model scale is one of the biggest drawbacks, as it frequently leads to poorer performance on real-world jobs requiring a lot of instructions [5]. With fewer parameters, these models may struggle to represent the breadth and depth of knowledge found in larger models such as GPT-4. Moreover, there is usually less pre-training expenditure required for these open-source models. This lower investment may result in a smaller set of pre-training data, which could restrict the models' capacity to comprehend and manage a variety of specialised or diversified subjects [12]. Additionally, open-source models frequently go through fewer fine-tuning stages due to resource limitations. A model's performance must be optimised for certain jobs or sectors, and insufficient fine-tuning can reduce the model's usefulness in niche applications. When these models are contrasted with the highly optimised proprietary LLMs, which are frequently designed to perform well in a broad range of difficult circumstances, this shortcoming becomes very clear [2].

As a result of the differences in performance between proprietary and open-source LLMs, KD approaches have become increasingly popular [6]. KD uses advanced proprietary models like GPT-4 or Gemini to enhance open-source LLMs, similar to a student learning from a skilled instructor.

B. PURPOSE AND STRUCTURE OF THIS SURVEY

This paper aims to systematically summarize the current research status and application progress of knowledge distillation in large language models. The specific contributions are as follows:

- Review of Basic Concepts and Technical Methods: This paper provides a detailed discussion on the basic principles, algorithms, and implementation methods of knowledge distillation, including core techniques such as soft targets, feature matching, and attention transfer. By introducing these techniques, the paper offers a comprehensive theoretical foundation for readers. These techniques will be explained in detail in the next section.
- Exploration of Practical Applications Across Various Layers: The paper analyzes the practical applications of knowledge distillation in different layers, including industrial systems, embedded systems, natural language processing, multi-modal processing, and vertical domains. These examples demonstrate the broad application prospects of knowledge distillation in real-world operations.
- Analysis of the Role in Improving Model Performance and Efficiency: Through actual case evaluations, the paper explores how knowledge distillation enhances model performance and efficiency. It focuses not only on the accuracy of the models but also discusses their performance in reducing latency, improving throughput, and decreasing model size, highlighting the multifaceted benefits of knowledge distillation in practical applications.
- Prospects for Future Development: The paper discusses the current challenges facing knowledge distillation and proposes future research directions and potential applications. It emphasizes the importance of multi-modal knowledge distillation, adaptive and online knowledge distillation methods, and the development of more

efficient and scalable distillation techniques to guide and inspire future research.
- Empirical Case Studies: By analyzing real-world cases such as DistilBERT, TinyBERT, and MobileBERT, the paper illustrates how knowledge distillation can significantly improve the efficiency and feasibility of large language models in practical applications. These cases validate the effectiveness of the theoretical concepts and provide valuable references for practical operations.

This survey aims to systematically summarize the current research status and application progress of knowledge distillation in large language models. The specific objectives include reviewing the basic concepts and technical methods of knowledge distillation with a detailed discussion on its principles, algorithms, and implementation methods; exploring practical applications across various layers such as industrial systems [13], embedded systems [14], natural language processing [15], [16], multi-modal processing [17], and vertical domains [7], [18]; analyzing the role of knowledge distillation in improving model performance and efficiency through actual case evaluations; and looking ahead to future development directions by discussing current challenges and proposing future research directions and potential applications.

The structure of the survey is outlined as follows: The second section introduces the basic concepts and development history of LLMs. The third section provides an overview of KD and LLMs' basic principles and technical methods. The fourth section explores the specific applications of KD in different application layers. The fifth section analyzes the role of KD in optimizing model performance. The sixth section addresses current challenges and future development directions in the field. The seventh section concludes the survey and proposes directions for future research initiatives.

II. RELATED WORK

Knowledge distillation (KD) has gained significant attention in the machine learning community as a crucial technique for model compression and efficiency enhancement. This section reviews notable works and surveys in the field of KD, emphasizing advancements and applications across various domains.

A. OVERVIEW OF LARGE LANGUAGE MODELS (LLMS)

The development of large language models have undergone several key stages. From early statistical language models to neural network-based deep learning models and the current Transformer-based models, these models have continuously improved their ability to handle natural language tasks [19], [20].

1) KEY STAGES OF DEVELOPMENT

Early language models were predominantly founded on statistical methods, which leveraged techniques such as n-grams and Markov models to predict subsequent words in a sequence. An n-gram model, for instance, would estimate the probability of a word based on the occurrence of the preceding n-1 words. Markov models, on the other hand, utilized the Markov assumption, which states that the probability of transitioning to the next state (or word) depends only on the current state and not on the sequence of events that preceded it [21]. Despite their simplicity and computational efficiency, these models were significantly constrained by their limited ability to capture long-range dependencies within text. This limitation arose because statistical models typically considered only a fixed window of previous words, which resulted in a loss of context and coherence over longer passages [22]. The emergence of neural network-based models heralded a transformative era in language modeling. Recurrent Neural Networks(RNNs) introduced the capability to process sequences of arbitrary length by maintaining a hidden state that could, in theory, encode information from all previous time steps in a sequence [23]. However, in practice, RNNs encountered difficulties with learning long-range dependencies due to issues like vanishing and exploding gradients [24]. To address these challenges, Long Short-Term Memory (LSTM) networks were developed. LSTMs are a type of RNN specifically designed to capture long-term dependencies by incorporating memory cells that can retain information across many time steps [25]. This innovation allowed LSTMs to significantly outperform traditional RNNs in tasks requiring the modeling of longer contexts. However, despite these advancements, both RNNs and LSTMs remained computationally intensive and still struggled with very long-range dependencies, paving the way for the development of more sophisticated architectures [20].

2) INTRODUCTION OF TRANSFORMER ARCHITECTURE

The introduction of the Transformer architecture marked a significant turning point in the development of large language models. Transformers leverage a self-attention mechanism that allows them to process all tokens in a sequence simultaneously, rather than sequentially, enabling parallel processing during training and inference. This innovation greatly enhances the models' efficiency and effectiveness, allowing them to capture complex dependencies and contextual information more effectively than previous models [26].

- Self-Attention Mechanism: The self-attention mechanism is central to the Transformer's ability to capture long-range dependencies and contextual information. By computing the similarity between each element in the input sequence and all other elements, the self-attention mechanism enables the model to focus on the most relevant parts of the input when generating each part of the output [20]. This approach allows the model to weigh the importance of different words differently, depending on their relevance to the current context, which significantly improves the handling of long-distance relationships in text.

- Positional Encoding: Unlike RNNs and LSTMs, Transformers do not process tokens in a sequential manner and thus lack a built-in mechanism to understand the order of tokens. To address this, Transformers employ positional encoding, which provides information about the position of each token in the sequence [27]. Positional encoding involves adding a unique positional vector to each token embedding, allowing the model to learn the positional relationships between tokens. This enables the Transformer to maintain the sequence order and capture the positional context, which is crucial for tasks like language modeling where word order impacts meaning.

The combination of these features allows Transformers to handle long-range dependencies more effectively and efficiently, leading to significant improvements in performance on a variety of natural language processing tasks. The Transformer architecture has become the foundation for many state-of-the-art language models, including BERT, GPT-3, and T5, which have set new benchmarks in various NLP tasks [27].

3) PROMINENT LLMs

Developed by OpenAI, the GPT (Generative Pre-trained Transformer) series has become representative of generative pre-trained models. GPT-3, with its 175 billion parameters, has demonstrated an unprecedented ability to generate human-like text and perform a variety of NLP tasks with few-shot learning [28], [29]. This model utilizes a transformer architecture that allows it to process text in a parallel manner, enhancing its efficiency and capability to handle complex language tasks. GPT-3's versatility in generating coherent text, answering questions, translating languages, and summarizing content has set new benchmarks in the field of NLP. The latest version, GPT-4, builds upon the foundation laid by its predecessors, further enhancing the model's generative and comprehension capabilities. GPT-4 showcases significant improvements in understanding context and generating coherent and contextually accurate responses [2]. Its enhanced architecture and increased parameter count enable it to perform more complex tasks and provide more nuanced and accurate outputs, solidifying its position as a leading model in the generative pre-trained category.

Introduced by Google, BERT (Bidirectional Encoder Representations from Transformers) represents a significant advancement in language modeling by employing a bidirectional encoder representation. Unlike unidirectional models that process text from either left-to-right or right-to-left, BERT considers context from both directions during training [27]. This bidirectional approach allows BERT to understand the context of a word more effectively, leading to substantial improvements in various NLP tasks such as question answering, named entity recognition, and sentiment analysis. BERT's ability to capture the intricate relationships between words in a sentence has made it a powerful tool for many NLP applications. Its architecture, which includes multiple layers of transformers, enables it to learn deep contextual representations, making it highly effective in understanding the subtleties of human language.

Also from Google, T5 (Text-to-Text Transfer Transformer) represents a unification of handling multiple NLP tasks by converting all tasks into a text-to-text format. This innovative approach simplifies the model architecture and training process by treating every NLP task as a text generation problem [30]. Whether the task is translation, summarization, or question answering, T5 processes the input text and generates the desired output text, leveraging the same underlying model. This text-to-text framework greatly enhances T5's versatility and performance across a wide range of NLP tasks. By training on diverse datasets and tasks, T5 learns to generalize well across different types of text processing challenges, making it a robust and adaptable model for many applications in natural language understanding and generation.

4) CORE PRINCIPLES OF LLMs

The core principles of LLMs are fundamentally rooted in the Transformer architecture [26]. This architecture has revolutionized the field of natural language processing by introducing several key features that enhance model performance and efficiency. LLMs leverage a two-phase training approach that includes pre-training on large-scale text data followed by fine-tuning on specific tasks. This strategy allows the models to learn rich linguistic knowledge and contextual information, making them highly versatile and powerful for a wide range of applications.

- Self-Attention Mechanism: One of the most significant innovations in the Transformer architecture is the self-attention mechanism. This mechanism captures long-range dependencies and contextual information by computing the similarity between each element in the input sequence and all other elements. Unlike traditional sequential processing methods, which process tokens one at a time, the self-attention mechanism allows the model to focus on different parts of the input simultaneously. This parallel processing capability improves both training and inference efficiency, enabling the model to handle larger datasets and more complex tasks [26]. The self-attention mechanism ensures that the model can dynamically weigh the importance of different words in a sequence based on their relevance to the current context, leading to more accurate and coherent text generation.
- Simultaneous Processing: Another crucial advantage of the Transformer architecture is its ability to process all elements of the input sequence simultaneously. This is a departure from the sequential nature of earlier models like RNNs and LSTMs, which process tokens in a step-by-step manner. The simultaneous processing enabled by Transformers not only accelerates the training

process but also allows the model to maintain a global view of the entire sequence, enhancing its ability to capture complex dependencies and interactions between words.

- Pre-Training Phase: During the pre-training phase, the model is exposed to a vast corpus of text data, learning to predict the next word in a sentence. This unsupervised learning process enables the model to acquire a deep understanding of syntax, semantics, and world knowledge. By learning from diverse and extensive text data, the model builds a comprehensive language representation that can be applied to various downstream tasks [28]. The pre-training phase is crucial as it equips the model with a broad base of knowledge that can be fine-tuned for specific applications.
- Fine-Tuning Phase: Following pre-training, the model undergoes fine-tuning on specific tasks to enhance its performance in those areas. Fine-tuning involves adjusting the pre-trained model using task-specific labeled data, allowing the model to specialize in particular tasks such as question answering, text classification, or sentiment analysis. This phase ensures that the model can adapt its general language understanding to meet the requirements of specific applications, achieving higher accuracy and relevance in its outputs [27], [28].

In summary, the evolution of LLMs from statistical models to sophisticated Transformer-based architectures has dramatically enhanced their ability to understand and generate human language. The continuous improvements in model architecture, training techniques, and the leveraging of large-scale datasets have propelled the capabilities of these models, making them indispensable tools in the field of natural language processing. Understanding the history of large language models is crucial (Figure 3).

Additionally, Minghao et al. extensively reviewed recent advancements in large language models (LLMs) and their various architectures, including encoder-only, decoder-only, and encoder-decoder models [32]. Encoder-only models, such as BERT and RoBERTa, transform input data into lower-dimensional vectors, capturing contextual information. In contrast, decoder-only models like GPT-1 through GPT-4o and Mistral utilize uni-directional attention mechanisms to generate tokens based solely on previous ones. Encoder-decoder models, represented by T5 and the Switch Transformer, combine both encoder and decoder components of the transformer architecture. A notable aspect of the Switch Transformer is its use of Mixture of Experts (MoE) technology, which selectively activates specific parts of the model in response to input, thereby improving computational efficiency. Furthermore, they also argued that the GLM model employs a unique approach by omitting sequences of words from the input and concentrating on their reconstruction instead of merely masking tokens. This strategy enables GLM

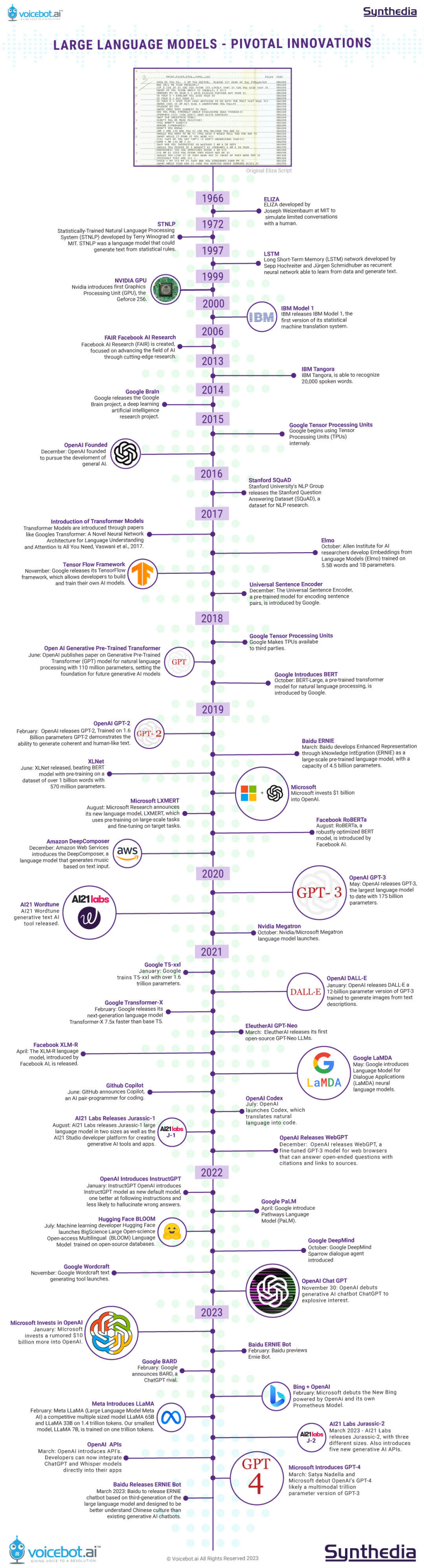

voicebot.ai
Synthedia
LARGE LANGUAGE MODELS - PIVOTAL INNOVATIONS
Original Eliza Script
1966
ELIZA
ELIZA developed by Joseph Weizenbaum at MIT to simulate limited conversations with a human.
1972
STNLP
Statistically-Trained Natural Language Processing System (STNLP) developed by Terry Winograd at MIT. STNLP was a language model that could generate text from statistical rules.
1997
LSTM
Long Short-Term Memory (LSTM) network developed by Sepp Hochreiter and Jürgen Schmidhuber as recurrent neural network able to learn from data and generate text.
1999
NVIDIA GPU
Nvidia introduces first Graphics Processing Unit (GPU), the Geforce 256.
2000
IBM Model 1
IBM releases IBM Model 1, the first version of its statistical machine translation system.
2006
FAIR Facebook AI Research
Facebook AI Research (FAIR) is created, focused on advancing the field of AI through cutting-edge research.
2013
IBM Tangora
IBM Tangora, is able to recognize 20,000 spoken words.
2014
Google Brain
Google releases the Google Brain project, a deep learning artificial intelligence research project.
2015
Google Tensor Processing Units
Google begins using Tensor Processing Units (TPUs) internaly.
OpenAI Founded
December: OpenAI founded to pursue the develoment of general AI.
2016
Stanford SQuAD
Stanford University's NLP Group releases the Stanford Question Answering Dataset (SQuAD), a dataset for NLP research.
2017
Introduction of Transformer Models
Transformer Models are introduced through papers like Googles Transformer: A Novel Neural Network Architecture for Language Understanding and Attention Is All You Need, Vaswani et al., 2017.
Elmo
October: Allen Institute for AI researchers develop Embeddings from Language Models (Elmo) trained on 5.5B words and 1B parameters.
Tensor Flow Framework
November: Google releases its TensorFlow framework, which allows developers to build and train their own AI models.
Universal Sentence Encoder
December: The Universal Sentence Encoder, a pre-trained model for encoding sentence pairs, is introduced by Google.
2018
Google Tensor Processing Units
Google Makes TPUs availabe to third parties.
Open AI Generative Pre-Trained Transformer
June: OpenAI publishes paper on Generative Pre-Trained Transformer (GPT) model for natural language processing with 110 million parameters, setting the foundation for future generative AI models
GPT
Google Introduces BERT
October: BERT-Large, a pre-trained transformer model for natural language processing, is introduced by Google.
2019
OpenAI GPT-2
February: OpenAI releases GPT-2, Trained on 1.6 Billion parameters GPT-2 demonstrates the ability to generate coherent and human-like text.
GPT- 2
Baidu ERNIE
March: Baidu develops Enhanced Representation through kNowledge IntEgration (ERNIE) as a large-scale pre-trained language model, with a capacity of 4.5 billion parameters.
XLNet
June: XLNet released, beating BERT model with pre-training on a dataset of over 1 billion words with 570 million parameters.
Microsoft
Microsoft invests $1 billion into OpenAI.
Microsoft LXMERT
August: Microsoft Research announces its new language model, LXMERT, which uses pre-training on large-scale tasks and fine-tuning on target tasks.
Facebook RoBERTa
August: RoBERTa, a robustly optimized BERT model, is introduced by Facebook AI.
Amazon DeepComposer
December: Amazon Web Services introduces the DeepComposer, a language model that generates music based on text input.
aws
2020
OpenAI GPT-3
May: OpenAI releases GPT-3, the largest language model to date with 175 billion parameters.
GPT- 3
AI21 Wordtune
AI21 Wordtune generative text AI tool released.
AI21labs
Nvidia Megatron
October: Nvidia/Microsoft Megatron language model launches.
2021
Google T5-xxl
January: Google trains T5-xxl with over 1.6 trillion parameters.
OpenAI DALL-E
January: OpenAI releases DALL-E a 12-billion parameter version of GPT-3 trained to generate images from text descriptions.
DALL·E
Google Transformer-X
February: Google releases its next-generation language model Transformer-X 7.5x faster than base T5.
EleutherAI GPT-Neo
March: EleutherAI releases its first open-source GPT-Neo LLMs.
Facebook XLM-R
April: The XLM-R language model, introduced by Facebook AI, is released.
Google LaMDA
May: Google introduces Language Model for Dialogue Applications (LaMDA) neural language models.
LaMDA
Github Copilot
June: GitHub announces Copilot, an AI pair-programmer for coding.
OpenAI Codex
July: OpenAI launches Codex, which translates natural language into code.
AI21 Labs Releases Jurassic-1
August: AI21 Labs releases Jurassic-1 large language model in two sizes as well as the AI21 Studio developer platform for creating generative AI tools and apps.
AI21labs J-1
OpenAI Releases WebGPT
December: OpenAI releases WebGPT, a fine-tuned GPT-3 model for web browsers that can answer open-ended questions with citations and links to sources.
2022
OpenAI Introduces InstructGPT
January: InstructGPT OpenAI introduces InstructGPT model as new default model, one better at following instructions and less likely to hallucinate wrong answers.
Google PaLM
April: Google introduce Pathways Language Model (PaLM).
Hugging Face BLOOM
July: Machine learning developer Hugging Face launches BigScience Large Open-science Open-access Multilingual (BLOOM) Language Model trained on open-source databases.
Google DeepMind
October: Google DeepMind Sparrow dialogue agent introduced
Google Wordcraft
November: Google Wordcraft text generating tool launches.
OpenAI Chat GPT
November 30: OpenAI debuts generative AI chatbot ChatGPT to explosive interest.
2023
Microsoft Invests in OpenAI
January: MIcrosoft invests a rumored $10 billion more into OpenAI.
Baidu ERNIE Bot
February: Baidu previews Ernie Bot.
Google BARD
February: Google announces BARD, a ChatGPT rival.
Bing + OpenAI
February: Microsoft debuts the New Bing powered by OpenAi and its own Prometheus Model.
Meta Introduces LLaMA
February: Meta LLaMA (Large Language Model Meta AI) a competitive multiple sized model LLaMA 65B and LLaMA 33B on 1.4 trillion tokens. Our smallest model, LLaMA 7B, is trained on one trillion tokens.
AI21 Labs Jurassic-2
March 2023 - AI21 Labs releases Jurassic-2, with three different sizes. Also introduces five new generative AI APIs.
AI21labs J-2
OpenAI APIs
March: OpenAI introduces API's. Developers can now integrate ChatGPT and Whisper models directly into their apps
GPT 4
Microsoft Introduces GPT-4
March: Satya Nadella and Microsoft debut OpenAI's GPT-4 likely a multimodal trillion parameter version of GPT-3
Baidu Releases ERNIE Bot
March 2023: Baidu to release ERNIE chatbot based on third-generation of the large language model and designed to be better understand Chinese culture than existing generative AI chatbots.
voicebot.ai GIVING VOICE TO A REVOLUTION
© Voicebot.ai All Rights Reserved 2023
Synthedia

FIGURE 3. History of LLMs. Souce from:voicebot.ai [31].

to outperform BERT while using fewer parameters. Figure 4 illustrates the evolutionary tree that highlights the progression of LLMs across different architectural.

B. OVERVIEW OF KNOWLEDGE DISTILLATION

Knowledge distillation is a technique used to compress and transfer knowledge from a large, complex model(the teacher) to a smaller, simpler model(the student). This process involves the student model learning to mimic the behavior of the teacher model, thereby achieving similar performance with fewer parameters and reduced computational requirements [33], [34]. The primary objective of knowledge distillation is to retain the accuracy and capabilities of the large model while significantly reducing its size and inference time.

The core techniques of knowledge distillation revolve around three main approaches: soft targets, feature matching, and attention transfer. Instead of using hard labels for training, the student model is trained on the soft targets provided by the teacher model, which include the probability distribution of the output classes, helping the student model capture the teacher model's learned knowledge more effectively [33], [34]. The term "soft target" denotes the probability distribution from a teacher model that provides confidence scores for various answers, helping the student model learn more efficiently than just using the correct answer [35], [36]. Feature matching involves using the intermediate features (activations) of the teacher model to guide the learning process of the student model, allowing the student to learn richer and more nuanced features [34], [37]. Attention transfer involves transferring the attention maps of the teacher model to the student model, highlighting important areas in the input data that the teacher model focuses on, thus helping the student learn where to pay attention [37]. By guiding a smaller model to concentrate on the significant aspects of the data identified by a larger model, this technique contributes to improved accuracy in predictions [38], [39].

The current research of KD focuses on several key areas, including cross-modal knowledge distillation, self-distillation, online knowledge distillation, and distillation for robustness and generalization. Cross-modal KD extends distillation techniques to scenarios where the teacher and student models operate on different modalities, such as distilling knowledge from a vision model to a language model [40], [41]. Self-distillation involves a model distilling knowledge into itself, improving its own performance without the need for a separate teacher model, which can be particularly useful in iterative training processes [42]. Online KD trains multiple student models simultaneously, allowing them to learn from each other in an online setting without a predefined teacher model, enhancing the performance of all participating models [17]. Research also explores how KD can improve the robustness and generalization capabilities of models, making them more resilient to adversarial attacks and better at handling diverse data distributions [43], [44]. An overview of this survey on KD of large language models is presented in Figure 5.

1) ADDITIONAL TECHNIQUES AND APPLICATIONS

Based on the current situation, there are some optimizations for the student model. This allows the student model to adapt to a variety of fields.

Parameter Pruning: Parameter pruning is a technique aimed at reducing the size of a model by removing redundant or less significant parameters. By identifying and eliminating weights that contribute minimally to the model's performance, pruning helps in streamlining the model, thereby reducing its complexity and memory footprint. This process not only lowers the computational cost but also enhances inference speed, making the model more efficient for deployment in resource-constrained environments [45]. The pruning process involves techniques such as weight magnitude pruning, where parameters with the smallest absolute values are removed, or more sophisticated methods that consider the impact of parameters on the overall network performance.

Quantization: Quantization is another effective technique used to improve model efficiency. This method involves reducing the precision of the weights and activations from higher precision (such as 32-bit floating-point) to lower precision (such as 8-bit integer). By converting the model parameters to lower bit-widths, quantization significantly reduces the model size and the amount of computation required for inference [46]. Despite the reduction in precision, quantized models often maintain comparable performance levels to their full-precision counterparts due to careful calibration and optimization processes. Quantization can be particularly beneficial in scenarios where computational resources are limited, such as on edge devices or mobile platforms. Together, parameter pruning and quantization contribute to making LLMs more practical for real-world applications by enhancing their efficiency without substantially compromising their performance.

Adaptive KD: Adaptive methods involve dynamically adjusting the distillation process based on the difficulty of the samples [47]. This can help to focus the training on harder samples where the student model needs more guidance from the teacher model. In adaptive KD, the training process is tailored to focus more on challenging samples where the student model requires additional guidance. This adaptive approach ensures that the student model receives more focused and effective learning signals, which can lead to better generalization and robustness. For instance, if the student model performs well on easier samples but struggles with harder ones, the distillation process will emphasize these harder samples, providing more detailed and nuanced knowledge transfer from the teacher model. This targeted training helps in improving the student model's performance more efficiently compared to uniform distillation methods.

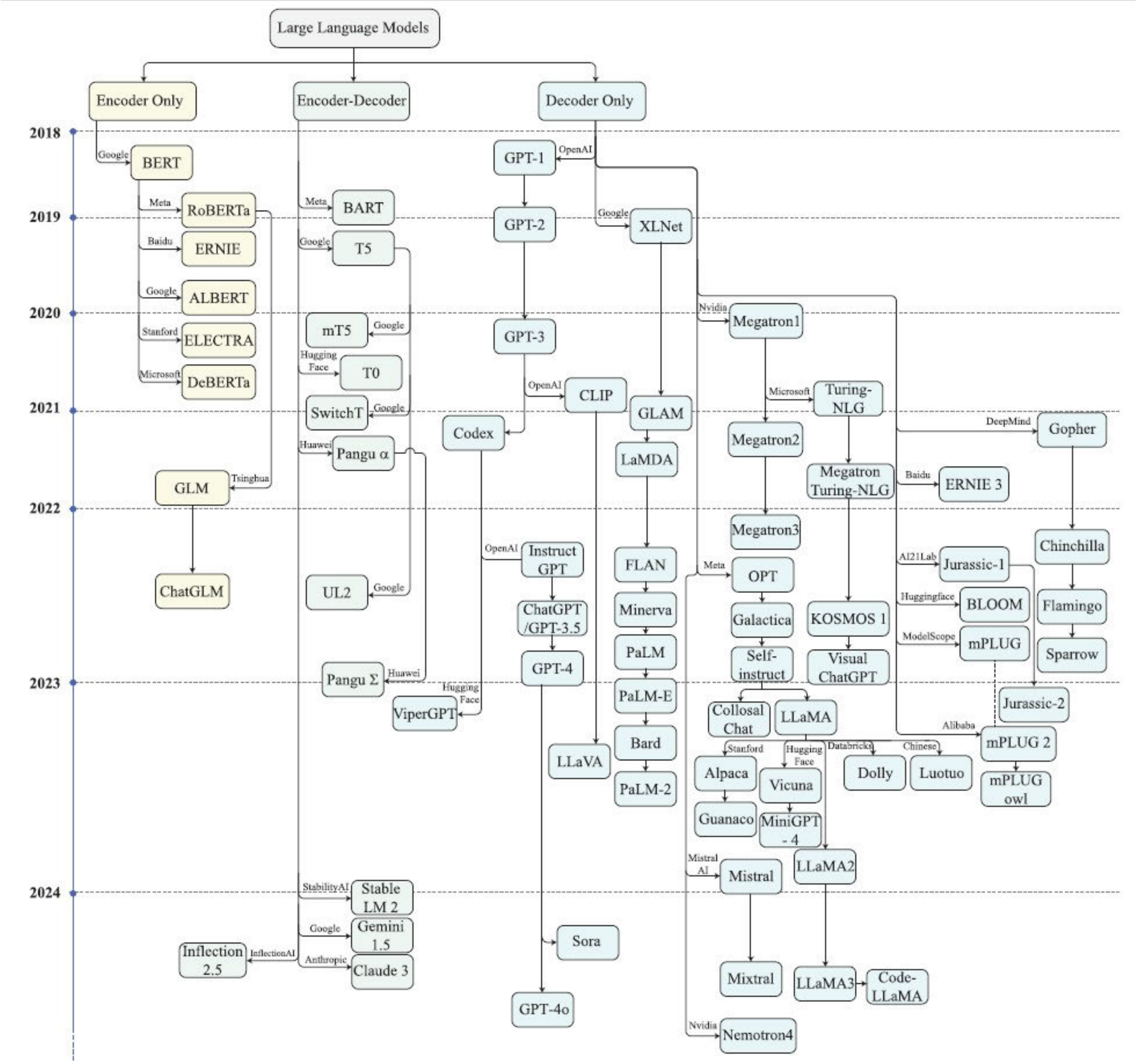


FIGURE 4. The evolutionary tree of LLMs [32].

Adaptive KD can be implemented using various techniques, such as adjusting the weight of each sample's loss based on its difficulty or using an adaptive temperature in the softmax function to control the smoothness of the teacher model's predictions. These methods ensure that the distillation process is more responsive to the learning needs of the student model, leading to more efficient and effective training outcomes.

Model compression techniques like KD, pruning, quantization, and adaptive KD each have unique pros and cons. KD transfers knowledge from a larger teacher model to a smaller student model, significantly reducing size while maintaining accuracy [6], especially in natural language processing [16]. However, it relies on large labeled datasets and struggles with very small models [6]. Pruning removes redundant weights or neurons for better compression and lower computational demands, but often sacrifices accuracy and requires fine-tuning [48], [49]. Quantization reduces model precision, saving memory and speeding up inference, though it can degrade accuracy in complex tasks without specialized hardware [50]. Adaptive KD customizes the distillation process based on input difficulty or model alignment, improving performance but increasing training complexity [47], [51]. While KD and its adaptive forms balance performance and size, pruning and quantization excel in extreme compression scenarios, highlighting the complementary nature of these methods.

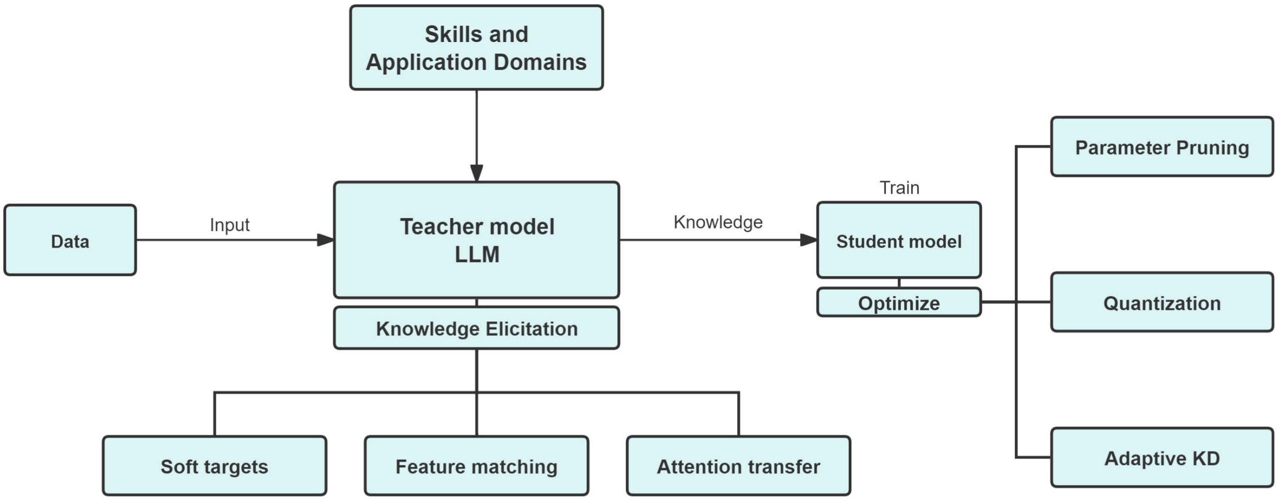


FIGURE 5. Overview of KD of LLM.

2) EMERGING TRENDS

With the development of KD of LLMs, there are many prospects for the future direction of this technique. Here are some of the main possible directions.

Multi-Modal KD: Integrating information from multiple modalities, such as text, images, and audio, into a single model is an emerging trend in KD. This approach leverages the complementary strengths of different data types to create more robust and versatile models. By distilling knowledge from models trained on diverse modalities into a unified model, researchers can develop systems capable of handling a wider range of tasks and providing richer, context-aware responses. For instance, a multi-modal model can understand and generate descriptions for images, respond to audio queries, and interpret text, making it highly useful in applications like virtual assistants and automated customer service [40]. Figure 6 illustrates the Multimodal Hierarchical Knowledge Distillation for Medical Visual Question Answering (MHKD-MVQA), which comprises five distinct modules: Multimodal Pretrain, Multimodal Feature Extraction, Multimodal Hierarchical Knowledge Distillation, Medical VQA, and Answer Prediction [52]. The integration of multiple modalities enhances the model's ability to capture nuanced information and improve its generalization capabilities across various tasks.

Distillation for Model Robustness: Recent studies have focused on using KD to improve the robustness of models against adversarial attacks. Adversarial attacks involve manipulating input data in subtle ways to deceive a model into making incorrect predictions. By training a teacher model with adversarial examples and then transferring these robust features to a student model, KD can help the student model become more resilient to such attacks. This process involves exposing the student model to clean and adversarially perturbed examples during training, allowing it to learn robust representations less susceptible to manipulation. This technique improves the security and reliability of AI systems, particularly in sensitive applications such as cybersecurity and autonomous driving [43], [44].

Continual Learning and Online Distillation: In dynamic environments where data continuously evolve, traditional static models often struggle to remain relevant. Online distillation methods address this challenge by enabling models to be updated in real time as new data becomes available. This continuous learning approach ensures that the model stays up-to-date with the latest information and can quickly adapt to new patterns. By continuously transferring knowledge from an updated teacher model to a student model, online distillation helps maintain high performance without the need for complete retraining from scratch. This is particularly beneficial in scenarios such as real-time analytics, personalized recommendations, and evolving user preferences [17]. In various industrial applications, online distillation enhances machine health prognosis on edge devices [53], while also contributing to improvements in robustness and accuracy across diverse domains, including video surveillance and autonomous driving [54]. Continual learning through online distillation helps to maintain the effectiveness and relevance of the model over time.

Self-Supervised KD: In many practical situations, labeled data are scarce and expensive to obtain. Self-supervised learning combined with KD offers a solution to this problem. In this approach, the student model learns useful representations using unlabeled data [55]. This process enables the student model to acquire meaningful patterns and features from the vast amounts of available unlabeled data, enhancing its performance on downstream tasks. Self-supervised KD is particularly useful in domains such as natural language

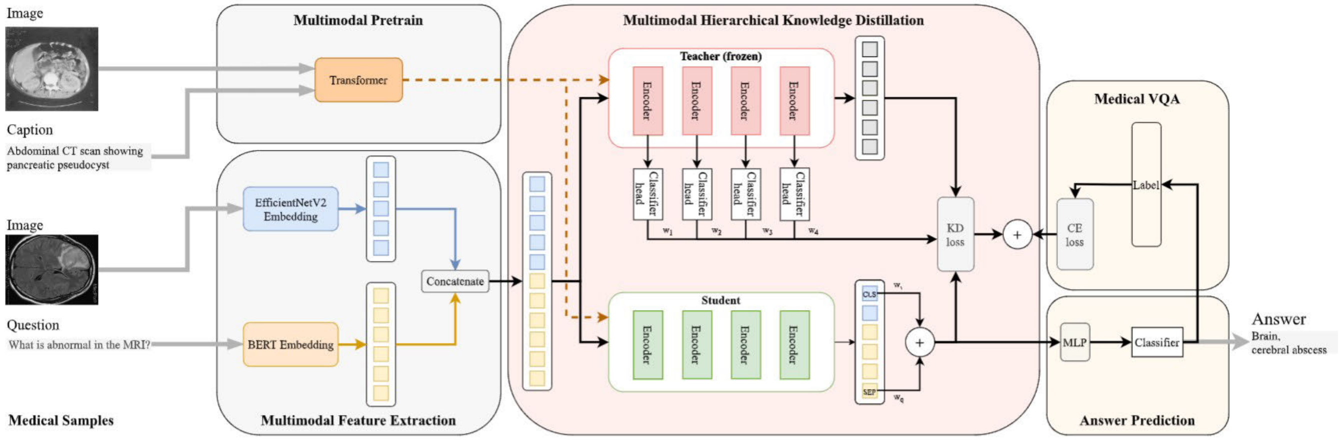


FIGURE 6. Model architecture of multimodal hierarchical knowledge distillation for MHKD-MVQA [52].

processing and computer vision, where the abundance of unlabeled data can be leveraged to improve model accuracy and efficiency [56].

3) CASE STUDIES

KD is a versatile and powerful technique that significantly enhances the deployment of LLMs by creating smaller, more efficient versions without a substantial loss of performance. The ongoing research and emerging trends in KD continue to expand its applicability, making advanced AI models more accessible and practical for various real-world applications. Here are some case studies.

DistilBERT: Developed by Hugging Face, DistilBERT is a distilled version of BERT (Bidirectional Encoder Representations from Transformers) that retains 97 percent of BERT's performance while being 60 percent faster and 40 percent smaller [15]. DistilBERT achieves this remarkable efficiency by applying KD techniques to compress the original BERT model. During the distillation process, a smaller student model learns to mimic the outputs of the larger BERT teacher model. The result is a model that maintains a high level of accuracy in natural language understanding tasks but is much more efficient in terms of computational resources. DistilBERT's reduced size and increased speed make it particularly useful for applications where computational efficiency is critical, such as real-time language processing on web servers.

TinyBERT: Similar to DistilBERT, TinyBERT is another compressed version of the BERT model designed to achieve significant reductions in model size and inference time while maintaining minimal performance loss [16]. TinyBERT utilizes a two-stage learning framework that includes general distillation and task-specific distillation. In the general distillation stage, the student model learns from the teacher model's intermediate representations and predictions. In the task-specific distillation stage, the student model is further fine-tuned on downstream tasks to optimize its performance. TinyBERT's efficiency and performance make it suitable for deployment in scenarios with limited computational resources, such as mobile applications and edge devices.

MobileBERT: MobileBERT is an optimized version of BERT specifically designed for mobile and edge devices. It combines KD with architectural optimizations to deliver high performance in resource-constrained environments [57]. MobileBERT achieves its efficiency through several techniques, including bottleneck structures and a carefully designed teacher-student learning framework. This model is significantly smaller and faster than the original BERT, making it ideal for on-device AI applications where memory and processing power are limited. MobileBERT's ability to perform complex language tasks on mobile devices without relying on cloud-based resources enables a wide range of applications, from personal assistants to real-time translation services.

Sun et al. conducted a comparative analysis of the various models and demonstrated that MobileBERT significantly outperforms all other models of smaller or comparable sizes [57], as illustrated in Tables 1 and 2. The analysis in Table 1 utilized the SQuAD development datasets based on Exact Match (EM) and F1 scores, alongside model parameter size (#Params) for each model. SQuAD is a reading comprehension dataset, where version 1.1 includes guaranteed answers, while version 2.0 introduces unanswerable questions to enhance complexity. Conversely, Table 2 employed the GLUE benchmark for its comparative assessment, which includes a variety of language understanding tasks like CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. The evaluation metrics include model parameters (#Params), FLOPS (computational cost), latency, and task-specific accuracy or correlation scores. The GLUE score is an aggregated metric representing overall performance across these tasks.

Table 1 demonstrates the trade-offs between model size, efficiency, and performance across different use cases. BERT $_{\text{BASE}}$ serves as a baseline with strong performance but a high parameter count, while smaller models like DistilBERT and

TABLE 1. Comparison of models on SQuAD dev datasets [57].

Model	#Params	SQuAD v1.1		SQuAD v2.0	
		EM	F1	EM	F1
BERT$_{BASE}$	109M	80.8	88.5	74.2†	77.1†
DistilBERT$_{BASE\text{-}6L}$	66.6M	79.1	86.9	-	-
DistilBERT$_{BASE\text{-}6L}$†	66.6M	78.1	86.2	66.0	69.5
DistilBERT$_{BASE\text{-}4L}$†	52.2M	71.8	81.2	60.6	64.1
TinyBERT	14.5M	72.7	82.1	65.3	68.8
MobileBERT$_{TINY}$	15.1M	81.4	88.6	74.4	77.1
MobileBERT	25.3M	82.9	90.0	76.2	79.2
MobileBERT w/o OPT	25.3M	**83.4**	**90.3**	**77.6**	**80.2**

TinyBERT reduce parameters at the cost of some accuracy. On the other hand, larger models, such as BERT and MobileBERT, without optimizations, offer improved accuracy but come with higher computational costs. MobileBERT stands out as a balanced option, combining efficiency with strong performance, making it suitable for tasks that require both speed and accuracy.

Table 2 further explores these trade-offs in natural language processing, showing that while BERT$_{BASE}$ is a high-performing benchmark, its latency and resource demands limit its use in real-time applications. Models like DistilBERT and those using progressive knowledge distillation (PKD) effectively minimize size and inference costs while maintaining competitive accuracy. MobileBERT stands out with a GLUE score of 78.5 and reduced latency, making it suitable for high-efficiency applications. Overall, optimized models like MobileBERT can achieve state-of-the-art performance while remaining computationally efficient, making them suitable for edge devices and resource-constrained environments.

There are some more KD techniques and models that each have unique skills and application areas. XLNet [58] employs an auto-regressive pre-training method, leveraging large-scale text data through permutation language modeling to enhance NLP task performance and contextual understanding. ERNIE [59] integrates knowledge, combining large-scale text data and knowledge graphs to achieve enhanced contextual representation. RoBERTa [60] focuses on robust optimization, using dynamic masking to improve NLP task performance and robustness. Electra [61] enhances pre-training efficiency and speed through replaced token detection. DeBERTa [62] employs a disentangled attention mechanism, improving language understanding through enhanced contextual encoding. SqueezeBERT [63] achieves efficient inference and edge deployment through model compression. ERNIE 2.0 [64] integrates incremental knowledge, combining large-scale text data and domain-specific knowledge to achieve domain adaptability and versatility. SpanBERT [65] employs span-based pre-training, using span boundary objectives to enhance NLP task performance and entity recognition. CamemBERT [66] focuses on language-specific adaptation for French, using language-specific pre-training to improve French NLP task performance. These models demonstrate the broad application of KD across various tasks and domains, highlighting the unique advantages of different models in compression, efficiency, performance, and versatility.

III. APPLICATIONS OF KNOWLEDGE DISTILLATION IN VARIOUS DOMAINS

KD has found extensive applications across multiple domains, leveraging its ability to transfer the knowledge from large models to smaller ones, thereby improving efficiency and performance. Based on the provided diagram, here are the specific applications of KD in various domains.

A. AGENT

1) INDUSTRIAL SYSTEM

KD plays a crucial role in enhancing all aspects of industrial systems, such as for remaining useful life (RUL) prediction of machine [13]. In predictive maintenance, KD allows smaller models to more accurately predict equipment failures, enabling timely maintenance operations, reducing downtime and improving overall operational efficiency [67]. Table 4 demonstrates the improvement in performance achieved through KD, which leads to increased accuracy in monitoring industrial processes. In addition, in automated safety systems, distillation models can quickly and accurately detect anomalies, which is critical to early identification and resolution of potential safety issues, thereby maintaining safer industrial operations [68]. KD optimizes the utilization of software library calls and mechanical equipment. Smaller models learn from larger models to make efficient use of resources and improve the efficiency of software development and mechanical operation, especially in environments with limited computing resources [69]. By learning from the experience of larger models, smaller models can achieve efficient use of resources, resulting in time and cost savings during development and operation of industrial systems.

In the operation process, it supports automated task assignment, schedule prediction, and risk assessment. Distilling complex scheduling algorithms into smaller, more efficient models enables organizations to manage tasks more accurately and efficiently [13]. This approach allows accurate prediction and adjustment of progress, ensuring optimal resource allocation and timely completion of tasks. In addition, the distillation model enhances risk assessment capabilities to better identify and mitigate potential risks, thereby improving overall operational resilience [70].

In industrial product defect detection, models based on attention mechanism and KD can effectively improve the detection accuracy, especially when dealing with complex backgrounds and diverse product types [71]. This is of great significance for ensuring product quality and reducing the scrap rate in the production process. In real-time industrial applications, a wide range of KD models can provide effective end-to-end anomaly detection to ensure system stability and reliability [72]. Multi-stage attention mechanism

TABLE 2. Comparison of models on GLUE benchmark [57].

Model	#Params	#FLOPS	Latency	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m/mm	QNLI	RTE	GLUE
BERT$_{\text{BASE}}$	109M	22.5B	342 ms	**52.1**	**93.5**	**88.9**	**85.8**	71.2	**84.6/83.4**	90.5	66.4	78.3
DistilBERT$_{\text{BASE-6L}}$†	66.2M	11.3B	-	-	92.0	85.0		70.7	81.5/81.0	89.0	65.5	-
DistilBERT$_{\text{BASE-4L}}$†	52.2M	7.6B	-	32.8	91.4	82.4	76.1	68.5	78.9/78.0	85.2	54.1	-
TinyBERT*	14.5M	1.2B	-	43.3	92.6	86.4	79.9	71.3	82.5/81.8	87.7	62.9	75.4
MobileBERT$_{\text{TINY}}$	15.1M	3.1B	40 ms	46.7	91.7	87.9	80.1	68.9	81.5/81.6	89.5	65.1	75.8
MobileBERT	25.3M	5.7B	62 ms	50.5	92.8	88.8	84.4	70.2	83.3/82.6	90.6	66.2	77.7
MobileBERT w/o OPT	25.3M	5.7B	192 ms	51.1	92.6	88.8	84.8	70.5	84.3/**83.4**	**91.6**	**70.4**	**78.5**

TABLE 3. Overview of mentioned knowledge distillation techniques and models.

Techniques and Models	Skills	Seed Knowledge	Knowledge Distillation	Objectives
DistilBERT [15]	Compression	BERT Dataset	Teacher-Student	Compression, Speed
TinyBERT [16]	Compression	BERT Dataset	Teacher-Student	Compression, Speed
MobileBERT [57]	Compression, Edge Deployment	BERT Dataset	Teacher-Student	Edge Deployment
Multi-Modal Knowledge Distillation [41]	Multi-Modal Processing	Text, Image Datasets	Multi-Modal	Versatility, Robustness
Distillation for Model Robustness [44]	Adversarial Robustness	Adversarially Perturbed Data	Robust Features	Adversarial Robustness
Self-Supervised Knowledge Distillation [55]	Unlabeled Data Utilization	Unlabeled Data	Self-Supervised	Representation Learning
GPT-3 Series [1]	Generative Pre-training	Large-Scale Text Data	Pre-training	Few-Shot Learning
GPT-4 [2]	Generative Pre-training	Large-Scale Text Data	Pre-training	Contextual Understanding
Gemini [3]	Multi-Modal Pre-training	Large-Scale Text, Image, and Audio Data	Multi-Modal Integration	Versatility, Robustness
BERT [27]	Contextual Representation	Large-Scale Text Data	Bidirectional Training	NLP Tasks Performance
T5 [30]	Text-to-Text Transfer Learning	Large-Scale Text Data	Text-to-Text Framework	Versatility, Performance
XLNet [58]	Auto-regressive Pre-training	Large-Scale Text Data	Permutation Language Modeling	NLP Tasks Performance, Contextual Understanding
ERNIE [59]	Knowledge Integration	Large-Scale Text Data, Knowledge Graphs	Knowledge Integration	Enhanced Contextual Representation
RoBERTa [60]	Robust Optimization	Large-Scale Text Data	Dynamic Masking	NLP Tasks Performance, Robustness
Electra [61]	Sample Efficiency	Large-Scale Text Data	Replaced Token Detection	Efficient Pre-training, Speed
DeBERTa [62]	Disentangled Attention Mechanism	Large-Scale Text Data	Enhanced Contextual Encoding	Improved Language Understanding
SqueezeBERT [63]	Efficient Inference	Large-Scale Text Data	Model Compression	Speed, Edge Deployment
ERNIE 2.0 [64]	Incremental Knowledge Integration	Large-Scale Text Data, Domain-Specific Knowledge	Continual Learning	Domain Adaptability, Versatility
SpanBERT [65]	Span-based Pre-training	Large-Scale Text Data	Span Boundary Objectives	NLP Tasks Performance, Enhanced Entity Recognition
CamemBERT [66]	Language-Specific Adaptation (French)	Large-Scale Text Data	Language-Specific Pre-training	NLP Tasks Performance for French

sample correlation KD technology significantly improves the performance and robustness of the model by extracting and utilizing the correlation information between samples [73]. This is especially important for maintaining an efficient model performance in a diverse and dynamically changing industrial environment [74].

2) TOOL USE

Recent developments in LLMs have demonstrated significant advancements in handling various tasks. However, these models often encounter challenges when dealing with large numerical values or performing complex mathematical calculations [75], [76]. Consequently, there has been

TABLE 4. Accuracy comparison of different methods [67].

	DLFN	DLFN-KD	W/O Mem & Hard	W/O Hard	KD-SCL
Acc	80,67%	**82,47%**	85,43%	87,23%	**88,82%**

a growing focus on equipping LLM agents with tool-use capabilities. Traditional methods have predominantly relied on human-curated data for training [77] or prompt engineering [78].

More recently, distillation-based approaches have emerged as promising alternatives [78], [79], [80]. Toolformer utilizes a self-supervised approach, minimizing the need for extensive human annotations by identifying essential APIs and distilling this knowledge into the model. This method has shown superior performance, with the GPT-J-based Toolformer outperforming models like OPT(66B) and GPT-3(175B) [77]. Graph-ToolFormer focuses on enabling LLMs to reason over complex graph data by integrating external graph reasoning APIs. This model uses ChatGPT to create a large graph reasoning dataset for training [81]. Gorilla addresses inaccuracies in generating API inputs, reducing hallucination by collecting numerous models from platforms like HuggingFace and Torch Hub, and utilizing GPT-4 for generating synthetic instruction data [82]. GPT4Tools enhances open-source LLMs such as LLaMA and OPT with multimodal tool-use capabilities previously exclusive to advanced proprietary models like ChatGPT and GPT-4. This involves creating an instruction-following dataset using multimodal contexts and the Low-Rank Adaptation optimization [83].

ToolAlpaca [84] proposes a framework for augmenting compact language models with tool-use skills for embodied intelligence. It compiles a dataset with nearly 4000 instances from over 400 real-world tool APIs across 50 categories, with documentation generated by ChatGPT. ToolLLM offers a comprehensive framework for enhancing tool-use proficiency in LLMs, focusing on data creation, model training, and evaluation by distilling knowledge from ChatGPT. Their ToolLLaMA model excels in executing complex instructions and managing new APIs [85]. CRAFT introduces a general framework for tool creation and retrieval, leveraging GPT-4 to generate code snippets that smaller LLMs can use during inference [86]. Confucius employs a tiered training strategy for mastering tool use through a curriculum and iterative self-instruction from introspective feedback [87]. MLLM-Tool integrates multimodal encoders with open-source LLMs, enabling interpretation of visual or audio content embedded instructions. This method also uses GPT-4 to generate initial instruction-answer pairs [88]. Shen et al. propose a multi-LLM framework to enhance tool-use capabilities by decomposing the ability into planner, caller, and summarizer roles, supported by a two-stage training strategy using ChatGPT and GPT-4 for collecting execution trajectories [89]. Yuan et al. address the issue of lengthy tool documentation hindering tool utilization by proposing EASYTOOL, which distills essential information from extensive documentation using ChatGPT for ground truth summarization [86].

3) PLANNING

In the context of high-level task decomposition, LLMs have demonstrated their ability to generate plausible goal-driven action plans without prior training, as shown by Huang et al. [90]. Their research introduces non-invasive tools to enhance model executability and evaluates these methods through human assessment to balance executability and semantic accuracy. Most existing methods utilize prompting strategies for task planning or rely on human-curated data for training.

Recent advancements have also seen the emergence of distillation methods. FireAct [95] refines LLMs by fine-tuning smaller models using agent trajectories derived from various tasks and prompting techniques, demonstrating performance enhancement with GPT-4-generated trajectories. AgentTuning [78] enhances LLM performance in executing agent tasks by utilizing a dataset called AgentInstruct and applying a hybrid instruction-tuning approach. Lumos [96] introduces a framework for training agents using a unified data format and modular architecture, facilitating the decomposition of tasks into subgoals and actionable steps. TPTU-v2 [97] improves task planning and tool usage in real-world scenarios with a framework that includes an API Retriever, an LLM Finetuner, and a Demo Selector. AUTOACT [80] proposes a self-instruct method to generate planning trajectories with limited initial data, employing a division-of-labor strategy to create specialized sub-agents for different task aspects.

Distillation also supports the training of embodied multi-modal agents. For example, Sumers [91] enhance AI agents' ability to follow instructions by using pretrained vision-language models for supervision, leveraging model distillation and hindsight experience replay in a simulated 3D environment. Emma [86] addresses the inefficiency of training embodied agents in noisy visual worlds by using imitation learning in a simulated environment, guided by an expert Language Model.

In terms of planning, KD provides essential support for automated task allocation [92], schedule forecasting [93], and risk assessment [94]. By distilling complex scheduling algorithms into smaller, more efficient models, organizations achieve more accurate and efficient task management [92]. This process enables precise forecasting and adjustment of schedules, ensuring optimal resource allocation and timely task completion. Additionally, the enhanced risk assessment capabilities of distillation models help organizations identify and mitigate potential risks more effectively, improving overall operational resilience [94].

In conclusion, the application of KD in industrial systems demonstrates its potential to transform various operational processes. By enhancing predictive maintenance, optimizing tool use, and improving programs, KD helps improve the

TABLE 5. Overview of KD in agent.

Aspect	Description	References
Industrial System - Predictive Maintenance	Predict failures, timely maintenance, reduce downtime, improve efficiency.	[13], [67]
Industrial System - Automated Safety Systems	Detect anomalies, early identification, resolve safety issues, maintain safer operations.	[68]
Industrial System - Resource Utilization	Optimize resource utilization, improve efficiency, limited computing resources.	[69]
Industrial System - Task Assignment	Automated task assignment, schedule prediction, risk assessment, manage tasks efficiently.	[13], [70]
Industrial System - Product Defect Detection	Improve detection accuracy, complex backgrounds, diverse product types, ensure quality.	[71]
Industrial System - Anomaly Detection	End-to-end anomaly detection, system stability, reliability, real-time applications.	[72]
Industrial System - Sample Correlation	Sample correlation, enhance performance, robustness, diverse environments.	[73], [74]
Tool Use - Toolformer	Self-supervised, identify APIs, distill knowledge, superior performance.	[77]
Tool Use - Graph-ToolFormer	Graph reasoning, external APIs, ChatGPT dataset.	[81]
Tool Use - Gorilla	Reduce inaccuracies, collect models, synthetic instruction data.	[82]
Tool Use - GPT4Tools	Enhance multimodal tool-use, instruction-following dataset.	[83]
Tool Use - ToolAlpaca	Tool-use skills, embodied intelligence, real-world tool APIs.	[84]
Tool Use - ToolLLM	Tool-use proficiency, data creation, model training, evaluation.	[85]
Tool Use - CRAFT	Tool creation, retrieval, code snippets, inference.	[86]
Tool Use - Confucius	Tiered training, curriculum, self-instruction, introspective feedback.	[87]
Tool Use - MLLM-Tool	Multimodal encoders, interpret visual/audio content, GPT-4 instructions.	[88]
Tool Use - Multi-LLM	Decompose ability, planner, caller, summarizer roles, training strategy.	[89]
Tool Use - EASYTOOL	Distill essential information, extensive documentation, ChatGPT summarization.	[86]
Planning - High-level Task Decomposition	Generate action plans, executability, semantic accuracy.	[90]
Planning - Embodied Multi-Modal Agents	Follow instructions, vision-language models, model distillation, simulated environment.	[86], [91]
Planning - Task Allocation	Automated task allocation, accurate task management.	[92]
Planning - Schedule Forecasting	Schedule forecasting, adjustment, optimal resource allocation.	[93]
Planning - Risk Assessment	Risk assessment, identify, mitigate risks, improve resilience.	[94]

efficiency, safety, and reliability of industrial operations. These improvements highlight KD's value in industrial environments where maximizing performance and minimizing downtime are key goals. Integrating KD into these systems not only takes advantage of large models but also ensures that the benefits of advanced AI technology are fully realized in resource-constrained environments.

B. ASSIST

1) EMBEDDED

KD significantly enhances the deployment and performance of compact models in a variety of applications in embedded systems, such as smart home devices, mobile devices, vehicle systems, drones, and robotics. Despite the typical limitations in compute and storage resources within these embedded systems, KD enables smaller models to maintain performance levels comparable to larger models [14]. For instance, in smart home devices, KD improves speech recognition and natural language understanding, thereby making interactions with virtual assistants more efficient and accurate. This improvement ensures that users can enjoy seamless and intuitive smart home controls, enhancing the overall user experience. As Jaiswal and Gajjar [98] discuss, deep neural network compression via KD is particularly beneficial for embedded applications, enabling sophisticated functionalities without the need for extensive computational resources.

In mobile devices, KD plays a crucial role in saving battery life while still delivering advanced features. By learning from larger models, smaller models can efficiently perform tasks such as image recognition and personal assistance without consuming excessive power. This balance of performance and efficiency allows users to enjoy features like real-time image analysis and responsive virtual assistants while maintaining long battery life, making their devices more practical and satisfying. Xie et al. [99] highlight how distillation embedded absorbable pruning can be employed to achieve fast object re-identification, which is essential for many mobile applications.

Furthermore, KD is pivotal in vehicle systems, drones, and robotics, where efficient and accurate perception is critical. Shaw et al. [100] demonstrated the application of teacher-student KD for radar perception on embedded accelerators, which underscores the potential of KD in enhancing the capabilities of resource-constrained embedded systems.

In the realm of fault diagnosis and health monitoring, Gong et al. developed a lightweight method based on KD for embedded systems, emphasizing the method's ability to maintain high diagnostic accuracy with reduced computational overhead [101]. Similarly, Chen et al. [102] presented a lightweight deep learning network for efficient crack segmentation on embedded devices, highlighting the practical applications of KD in structural health monitoring. In addition, Cho and Lee [103] focused on building compact convolutional neural networks for embedded intelligent sensor systems using group sparsity and KD, demonstrating significant improvements in both model compactness and performance. Li et al. [104] introduced embedded mutual learning, a novel online distillation method that integrates diverse knowledge sources, further showcasing the versatility and efficacy of KD in various embedded applications.

Wang et al. [105] explored collaborative KD for heterogeneous information network embedding, which can be crucial for improving the interoperability and performance of embedded systems in complex environments. Xiao et al. [106] proposed a distillation sparsity training algorithm to accelerate convolutional neural networks in embedded systems, emphasizing the importance of efficient training techniques in constrained environments. Lastly, Xiong et al. [107] investigated ability-aware KD for resource-constrained embedded devices, highlighting how KD can be tailored to the specific capabilities of different devices to optimize their performance.

In summary, KD significantly advances the deployment of compact models in embedded systems by enabling high performance with limited resources. This makes KD an invaluable tool for enhancing the functionality and user experience of various embedded applications, from smart home devices to mobile phones and beyond.

2) SOFTWARE PLUG-IN

KD significantly enhances the capabilities of software plugins, including browser plugins, text editor plugins, and content management systems (CMS). The distillation model enables these plugins to provide advanced functionality without consuming excessive system resources. For example, browser plugins can provide language translation and contextual recommendations in real-time, increasing user productivity and convenience. This approach allows for the efficient use of resources while delivering high performance, as demonstrated by Chen [108] in their work on distilling crowd knowledge from software-specific QA discussions to assist developers' knowledge search. Similarly, text editor plugins can provide enhanced syntax checking, style suggestions, and predictive text capabilities to make writing and editing tasks more efficient. These functionalities, supported by KD, ensure that even lightweight models can offer robust assistance, as shown by Guo et al. [109], who developed a lightweight CNN for object detection with sparse models and KD.

In CMS applications, KD allows for advanced content management features such as automatic tagging, categorization, and personalized content push, while ensuring the system remains responsive and efficient. Shi et al. [110] demonstrated the effectiveness of compressing pre-trained models into compact sizes without losing significant performance, which is essential for maintaining responsive CMS applications.

Additionally, KD has proven useful in various other applications. Sun et al. [111] explored a lightweight dual Siamese network for onboard hyperspectral object tracking via joint spatial-spectral knowledge distillation, emphasizing the importance of KD in resource-constrained environments. Li et al. [112] highlighted deep generative knowledge distillation by likelihood finetuning, showcasing how KD can enhance generative models in different contexts.

Moreover, Fluri [113] discussed change distilling to enrich software evolution analysis with fine-grained source code change histories, illustrating the broad applicability of KD in software development and maintenance. Yuan et al. [114] investigated the influence mechanism of the knowledge network allocation on the distillation process in high-tech enterprises, further broadening the understanding of KD's potential.

Finally, Yao et al. [115] introduced GKT, a novel guidance-based knowledge transfer framework for efficient cloud-edge collaboration LLM deployment, underlining KD's role in optimizing complex deployments across distributed systems. KD significantly advances the functionality and efficiency of various software plugins by enabling high performance with minimal resource consumption. This makes KD an invaluable tool for enhancing user experience and system responsiveness in diverse applications.

Overall, KD's use in embedded systems and software plugins demonstrates its profound impact on improving performance and efficiency. By leveraging the benefits of larger models, KD ensures that compact models can deliver advanced functionality even in resource-constrained environments. This approach not only maximizes the utility and effectiveness of embedded systems and software plugins but also ensures the accessibility and utility of advanced AI capabilities in everyday applications.

C. NATURAL LANGUAGE PROCESSING (NLP)

In the field of NLP, KD has significantly enhanced the capabilities of smaller models, enabling them to perform complex tasks that previously required larger, more resource-intensive models. Significant obstacles that NLP projects frequently confront include noisy data, interpretability problems, data shortages, and privacy issues. Our survey's "Knowledge" section outlines several techniques for extracting information from large language models, which successfully gets student models ready to take on a variety of NLP tasks. By information augmentation, this condensed knowledge functions as supervision for student model training. Student models are better prepared to handle a variety of NLP problems by using knowledge from LLMs, which improves task performance and lessens the constraints brought on by inadequate data.

1) NATURAL LANGUAGE UNDERSTANDING (NLU)

In terms of NLU, KD improves the performance of dialogue systems, sentiment analysis, logical analysis, and machine translation. For example, smaller models trained by KD can approach the performance of larger models on tasks such as sentiment analysis and machine translation, making them well suited for real-time applications [16]. This ability to achieve approximate performance in a compact form factor is critical to deploying AI technology in scenarios where computing resources and response time are critical.

NLU is a crucial NLP task involving the comprehension and interpretation of human language. Distilled knowledge from LLMs is often integrated into encoder-based language models like BERT [27] and RoBERTa [60] to enhance their performance. This knowledge transfer, through methods such as data labeling and augmentation, is particularly beneficial for classification tasks. For instance, AugGPT, developed by Dai et al. [116], addresses text classification in both general and clinical domains. It tackles the challenges posed by small-scale clinical datasets, which often lack expert

TABLE 6. Overview of KD in assist.

Aspect	Description	References
Embedded - Smart Home Devices	Improves speech recognition and natural language understanding, enhances user experience.	[98]
Embedded - Mobile Devices	Saves battery life, performs tasks like image recognition efficiently.	[99]
Embedded - Vehicle Systems, Drones, and Robotics	Efficient and accurate perception, enhances capabilities of resource-constrained systems.	[100]
Embedded - Fault Diagnosis and Health Monitoring	Maintains high diagnostic accuracy with reduced computational overhead.	[101]
Embedded - Structural Health Monitoring	Efficient crack segmentation, practical for structural health monitoring.	[102]
Embedded - Intelligent Sensor Systems	Compact CNNs using group sparsity and KD, improves model compactness and performance.	[103]
Embedded - Embedded Mutual Learning	Online distillation method, integrates diverse knowledge sources.	[104]
Embedded - Heterogeneous Information Network	Collaborative KD for improving interoperability and performance.	[105]
Embedded - Accelerated CNNs	Distillation sparsity training algorithm, accelerates CNNs.	[106]
Embedded - Resource-Constrained Devices	Ability-aware KD tailored to specific device capabilities.	[107]
Software Plugin - Browser Plugins	Provides language translation, contextual recommendations, improves productivity.	[108]
Software Plugin - Text Editor Plugins	Enhanced syntax checking, style suggestions, predictive text capabilities.	[109]
Software Plugin - CMS Applications	Automatic tagging, categorization, personalized content push.	[110]
Software Plugin - Hyperspectral Object Tracking	Lightweight dual Siamese network for object tracking.	[111]
Software Plugin - Generative Models	Enhances generative models via likelihood finetuning.	[112]
Software Plugin - Software Evolution Analysis	Change distilling for software evolution analysis.	[113]
Software Plugin - High-Tech Enterprises	Influence mechanism of knowledge network allocation on KD process.	[114]
Software Plugin - Cloud-Edge Collaboration	Guidance-based knowledge transfer for cloud-edge collaboration.	[115]

annotation and are restricted by privacy regulations, by using LLMs to rephrase training sentences. This technique generates multiple semantically distinct yet conceptually similar samples, enriching the dataset's diversity and robustness.

Another method was shown by Gilardi et al. [117], who classified inputs using ChatGPT as an annotator. It has been discovered that their approach outperforms crowd-workers in a number of tasks, including frame identification, stance, relevance, and themes. Targeted Data Generation (TDG) is a novel technique that Gao et al. [118] presented. It generates fresh data specifically for problematic subgroups within a dataset using human-in-the-loop and LLMs. As a result, the dataset is enhanced and the model performs better on tasks involving sentiment analysis and natural language inference.

By employing LLMs to extract a variety of clinical samples, including instances and distinct seeds of clinical entities, Tang et al. [119] also made a substantial contribution to the process of improving the extraction of clinical information. Several NLU tasks have been the subject of additional research. Gao et al. [118] annotated inputs with labels and explanations using GPT-3.5 for a variety of NLU tasks, such as BoolQ, WiC, and user input and keyword relevance assessment. In order to increase the amount of high-quality training data using GPT-3 and improve the overall quality of the dataset, Wang et al. [120] used few-shot prompts. Ding et al. investigated the use of labelling, expansion, and curation techniques in conjunction with GPT-3 to extract information for NLP tasks at the token and sequence levels.

2) NATURAL LANGUAGE GENERATION (NLG)

In the field of NLG, KD enhances various applications, such as dialogue generation, content creation, report, and summary generation, and data narration. By learning from larger generative models, smaller models can produce text that is both coherent and contextually accurate. This capability allows distillation models to efficiently handle tasks requiring high-quality text generation, such as automated report writing or digital media content creation. For instance, Dai et al. [121] demonstrated the effectiveness of KD in enabling multimodal generation on CLIP through vision-language KD, showcasing how smaller models can benefit from the rich multimodal information in larger models to generate more accurate and contextually relevant content. Similarly, Fantazzini et al. [122] explored efficient KD techniques for creating green NLP models, highlighting the potential of KD to bridge the gap with large language models while maintaining environmental sustainability [122].

In the domain of autoregressive text generation, Lin et al. [123] utilized imitation learning for autoregressive KD, which helps smaller models mimic the behavior of larger models, thereby enhancing their text generation capabilities. Quteineh et al. [124] focused on enhancing task-specific distillation in small data regimes through language generation, emphasizing the importance of KD in improving performance even when data is limited.

Liu and Lin [125] provided a comprehensive review of unsupervised pre-training for natural language generation, underlining how KD can be integrated into pre-training processes to improve the efficiency and effectiveness of NLG models. Furthermore, Yu et al. [126] conducted a survey on knowledge-enhanced text generation, demonstrating various ways in which knowledge distillation can be leveraged to augment text generation models with external knowledge. Grünwald et al. [127] discussed simple, efficient, and high-quality evaluation metrics for NLG, which can be used to assess the performance of distillation models and ensure they meet the required standards for various applications. Jiang et al. [128] explored knowledge-augmented methods for NLG, providing insights into how KD can be combined with other techniques to enhance the overall quality and utility of generated text.

Wang et al. [129] investigated the use of conditional variational autoencoders with KD for generating long financial reports, highlighting the ability of KD to support complex and specialized text generation tasks. These advancements in KD have also led to significant improvements in conversation generation, making conversational AI systems more responsive and natural, thus providing a more satisfying user experience.

3) INFORMATION RETRIEVAL

Information retrieval systems benefit significantly from KD, which enhances search engines, recommendation systems, literature retrieval, and information analysis. By utilizing KD, these systems can deliver faster and more accurate search results and recommendations, thereby increasing their efficiency and user satisfaction. For example, search engines employing KD can return relevant results more quickly. Shakeri et al. [130] illustrated the impact of KD on document retrieval, demonstrating how KD improves the relevance and speed of search results by distilling the knowledge from larger, more complex models into smaller, efficient ones. Dong et al. [131] explored the idea of distillation as a form of early stopping in overparameterized neural networks, emphasizing the utility of harvested dark knowledge in refining search algorithms.

Recommendation systems also see substantial benefits from KD, enabling them to better personalize content and enhance user experience. Huang et al. [132] discussed how optimal transport distillation can improve cross-lingual information retrieval for low-resource languages, showcasing the ability of KD to bridge language gaps and provide accurate recommendations in multilingual contexts. Vakili Tahami et al. [133] highlighted the application of KD in fast retrieval-based chatbots, showing that distilled models can maintain high performance while responding rapidly to user queries.

Moreover, Xiao et al. [134] introduced Distill-VQ, a method for learning retrieval-oriented vector quantization by distilling knowledge from dense embeddings, which enhances the efficiency of retrieval systems. Gao et al. [135] examined BERT rankers under distillation, demonstrating how KD helps in understanding and improving the ranking capabilities of these models.

KD also plays a crucial role in zero-shot sketch-based image retrieval, as shown by Tian et al. [136], who used relationship-preserving KD to maintain the integrity of relationships within the data, thus improving retrieval accuracy. Izacard and Grave [137] demonstrated the use of KD from reader to retriever models for question answering, highlighting the cross-domain benefits of KD in information retrieval tasks.

Additionally, Passalis et al. [138] discussed heterogeneous KD using information flow modeling, which improves the performance of retrieval systems by leveraging diverse knowledge sources. Lu et al. [139] introduced TwinBERT, a method that distills knowledge to twin-structured compressed BERT models for large-scale retrieval, emphasizing the scalability of KD-enhanced systems. Finally, Hofstätter et al. [140] focused on improving neural ranking models with cross-architecture KD, demonstrating the efficacy of KD in refining and optimizing different architectures within information retrieval systems.

4) CODE

In code-related tasks, KD optimizes code generation, refactoring, and automated testing. By leveraging the knowledge from larger models, smaller models can generate efficient and high-quality code, automate testing procedures, and significantly reduce the time and effort required for software development. This efficiency is especially valuable in agile development environments that demand rapid iteration and deployment. By automating repetitive tasks throughout numerous development cycles, KD allows developers to concentrate more on innovation. For instance, Song et al. [141] discussed the concept of spot-adaptive knowledge distillation, which focuses on optimizing specific parts of the model to improve overall performance. This approach can be applied to code generation, where targeted optimization ensures that the generated code is both efficient and maintainable. Kim and Rush [142] introduced sequence-level knowledge distillation, which can enhance the process of code generation by ensuring that the generated sequences (i.e., lines of code) maintain logical consistency and high quality.

Ruffy and Chahal [143] highlighted the state of KD for classification tasks, emphasizing how KD can streamline complex classification problems into more manageable and efficient processes. This principle can be translated to automated testing, where distilled models can classify and prioritize test cases, making the code-testing process faster and more accurate. Matsubara [144] presented torchdistill, a modular framework for KD, which can be adapted to various aspects of software development, including code refactoring and optimization, to ensure that the software remains efficient and scalable.

Khan et al. [145] explored the development of multilingual and code-mixed visual question answering systems using KD, showcasing how distillation techniques can be applied to handle diverse and complex datasets. This methodology is particularly useful in automated testing scenarios, where test cases may need to handle multiple languages and formats. Li et al. [146] introduced knowledge condensation distillation, which focuses on condensing essential knowledge from large models into smaller, more efficient ones. This technique is crucial for code refactoring, ensuring that the refactored code retains its functionality while becoming more efficient.

Zhao et al. [147] discussed decoupled knowledge distillation, which separates the distillation process into distinct stages to improve the overall efficiency and effectiveness. This approach can be particularly beneficial for code generation and automated testing, where distinct stages of development and testing can be optimized individually. Xu et al. [148] combined knowledge distillation with self-supervision, demonstrating how self-supervised learning can enhance the distillation process. This technique can be applied to software development to ensure continuous improvement and learning from the generated code and test results.

Kanellopoulos et al. [149] presented an improved methodology for information distillation by the mining program source code, emphasizing the importance of extracting valuable information from existing codebases. This approach can be integrated with KD to optimize code refactoring and generation, ensuring that the new code is both efficient and easy to maintain. KD significantly enhances the efficiency and quality of code-related tasks, including generation, refactoring, and automated testing. By automating repetitive tasks and enabling rapid iteration, KD empowers developers to focus on innovation, making it an indispensable tool in modern software development.

In summary, the application of KD to various NLP tasks demonstrates its transformative potential. By enabling smaller models to approach larger ones in performance, KD not only makes advanced AI capabilities more ubiquitous but also enables these capabilities to be deployed in resource-constrained environments. This expands the scope of NLP technology, enabling it to be integrated into a wider range of products and services, from mobile applications to large enterprise systems. The integration of KD in NLP tasks represents an important advance in making advanced AI more accessible, efficient, and practical.

D. MULTI-MODALITY

Multimodal Large Language Models (MLLMs) understand and process information across multiple modalities, surpassing classic language-only LLMs. This skill allows for a greater variety of practical uses and more closely resembles human perception. Creating MLLMs that can obey multimodal instructions and increase task interaction is becoming more and more popular. Numerous studies have focused on multimodal KD from LLMs in an effort to address the limited availability of multimodal instruction-following data and take advantage of the common sense and world knowledge embedded in teacher LLMs. KD has been crucial in enhancing the efficacy and efficiency of small models in computer vision and audio processing, allowing them to execute intricate tasks that were previously the domain of huge, resource-hungry models.

1) VISION

In the field of vision, KD significantly improves the effectiveness of image classification, object detection, image generation, and optical character recognition. By distilling knowledge from larger models, small models can achieve high accuracy in image recognition and classification while reducing computational load. For example, KD enables these compact models to efficiently perform tasks such as object recognition in images and generating new images that maintain high fidelity. Moreover, in OCR applications, distillation models are capable of accurately identifying and converting various texts in images into machine-coded text, facilitating a wide range of applications from document digitization to real-time translation of foreign language text captured by cameras [150]. For instance, Habib et al. [151] provided a critical review of knowledge distillation in vision transformers, highlighting how KD improves the performance of smaller models, enabling them to perform complex tasks with high accuracy. Chen et al. [152] introduced Data-Efficient Early Knowledge Distillation (DearKD) for vision transformers, demonstrating that early-stage KD can significantly boost the performance of vision models with limited data.

In object detection, Gu et al. [153] explored open-vocabulary object detection using vision and language knowledge distillation, showing that KD enables models to recognize objects beyond their training vocabulary by transferring knowledge from models trained on vast datasets. This approach allows smaller models to efficiently perform tasks such as object recognition in images while maintaining high fidelity.

Moreover, Liu et al. [154] discussed semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition, which ensures that the distilled models retain crucial semantic information, thereby improving action recognition performance. Ni et al. [155] further examined cross-modal knowledge distillation for vision-to-sensor action recognition, underscoring the effectiveness of KD in enhancing multimodal recognition tasks. In the domain of OCR, KD enables compact models to accurately identify and convert various texts in images into machine-coded text. This facilitates a wide range of applications from document digitization to real-time translation of foreign language text captured by cameras. Liu et al. [156] highlighted the benefits of KD in vision-and-language pretraining with object KD, which enhances the OCR capabilities of vision models by incorporating object-specific knowledge.

Additionally, Wu et al. [157] introduced TinyViT, a method for fast pretraining distillation for small vision transformers, demonstrating how KD can expedite the training process while maintaining high model performance. Xu et al. [177] explored cross-modality medical image segmentation with online mutual KD, showcasing how KD can improve segmentation tasks by transferring knowledge between modalities.

Finally, Zhang et al. [158] investigated knowledge distillation from multi-modality to single-modality for person verification, illustrating how KD can effectively transfer knowledge from complex, multimodal systems to simpler, unimodal models while preserving performance. KD significantly improves the effectiveness and efficiency of various vision-related tasks, including image classification, object detection, image generation, and OCR. By enabling smaller models to perform at levels comparable to larger models, KD enhances the applicability and performance of vision systems in real-world scenarios.

2) AUDIO

In the field of audio, KD significantly enhances a variety of speech processing tasks, including speech recognition, speech synthesis, speech sentiment analysis, and speech translation. Small models trained using KD are able to accurately transcribe spoken words into text, providing highly reliable speech recognition capabilities for virtual assistants, transcription services, and voice-activated applications. For instance, Gao et al. [160] demonstrated the effectiveness of

TABLE 7. Overview of KD in NLP.

Aspect	Description	References
NLU - Dialogue Systems	Improves performance, real-time applications.	[16], [60]
NLU - Sentiment Analysis	Achieves near large model performance, critical for resource-limited scenarios.	[16]
NLU - Logical Analysis	Enhances comprehension and logical analysis.	[16]
NLU - Machine Translation	Approaches large model performance, suitable for real-time translation.	[16]
NLU - Text Classification	Improves performance in classification tasks with data augmentation.	[27], [60]
NLU - Data Annotation	Uses LLMs for annotation, surpasses crowd-workers.	[116], [117]
NLU - Clinical Information Extraction	Enhances extraction with diverse clinical samples.	[119]
NLG - Report Generation	Enables automated and accurate report writing.	[121], [123]
NLG - Data Narration	Handles complex data narration tasks.	[121], [123]
NLG - Multimodal Generation	Enhances generation with vision-language distillation.	[121]
NLG - Autoregressive Text Generation	Imitation learning for better text generation.	[123]
NLG - Green NLP Models	Creates efficient, environmentally sustainable models.	[122]
NLG - Unsupervised Pre-training	Integrates KD into pre-training processes.	[125]
NLG - Knowledge-Augmented Generation	Augments generation models with external knowledge.	[126], [128]
NLG - Conversational AI	Improves responsiveness and natural interaction.	[127], [129]
Information Retrieval - Search Engines	Enhances relevance and speed of search results.	[130], [131]
Information Retrieval - Recommendation Systems	Personalizes content, enhances user experience.	[132], [133]
Information Retrieval - Literature Retrieval	Improves accuracy and efficiency of retrieval.	[134]
Information Retrieval - Information Analysis	Enhances analysis and search capabilities.	[135]
Code - Code Generation	Generates high-quality, efficient code.	[141], [142]
Code - Code Refactoring	Optimizes and condenses code for efficiency.	[141], [144]
Code - Automated Testing	Automates testing, prioritizes test cases.	[141], [145]

TABLE 8. Overview of KD in multi-modality.

Aspect	Description	References
Vision - Image Classification	Improves accuracy, reduces computational load.	[150]–[152]
Vision - Object Detection	Recognizes objects beyond training vocabulary.	[153]
Vision - Image Generation	Generates high-fidelity images efficiently.	[150], [152]
Vision - Optical Character Recognition (OCR)	Accurately converts text in images to machine-coded text.	[150], [156]
Vision - Action Recognition	Enhances recognition performance with semantic information.	[154], [155]
Vision - Medical Image Segmentation	Transfers knowledge between modalities for better segmentation.	[159]
Vision - Person Verification	Transfers knowledge from multimodal to unimodal models.	[158]
Audio - Speech Recognition	Accurately transcribes spoken words to text.	[160], [161]
Audio - Speech Synthesis	Generates natural, fluid speech from text input.	[162]
Audio - Speech Sentiment Analysis	Accurately analyzes and interprets emotional tone.	[163]
Audio - Speech Translation	Translates spoken language into different languages with high accuracy.	[164]
Audio - Audio Classification	Improves accuracy and efficiency of audio models.	[160], [161]
Audio - Audio-Visual Content Generation	Generates high-quality audio-visual content.	[162]
Audio - Acoustic Scene Classification	Improves detection and classification of audio scenes.	[163]
Audio - Audio Tagging	Streamlines complex audio tagging processes.	[165]
Audio - Audio Question Answering	Enhances spoken question-answering systems.	[166]
Audio - Emotion Classification	Enhances emotion detection capabilities in audio data.	[167]

multi-representation KD in audio classification, showing how KD can improve the accuracy and efficiency of audio models. Gong et al. [161] proposed a CNN/transformer-based cross-model KD technique for audio classification, which further illustrates the power of KD in enhancing audio processing models.

In addition to improving speech recognition, KD also enhances speech synthesis. Smaller models can generate natural, fluid speech from text input, which is critical for creating more realistic dialogue agents and automatic answering systems. Chen et al. [162] explored distilling audio-visual knowledge through compositional contrastive learning, highlighting how KD can be used to generate high-quality audio-visual content.

Moreover, KD significantly benefits speech sentiment analysis. By training small models with knowledge distilled from larger, more comprehensive models, these models can accurately analyze and interpret the emotional tone of speech. Jung et al. [163] demonstrated the use of KD in acoustic scene classification, showing its potential to improve the detection and classification of various audio scenes.

In the realm of speech translation, KD allows small models to effectively translate spoken language into different languages while maintaining high accuracy. Fukuda et al. [164] discussed efficient KD from an ensemble of teachers, emphasizing the potential of KD to enhance the performance of translation models.

Furthermore, KD improves large-scale audio tagging and audio-based question answering systems. Schmid et al. [165] showcased efficient large-scale audio tagging via transformer-to-CNN KD, demonstrating KD's ability to streamline complex audio tagging processes. You et al. [166] introduced MRD-Net, a multi-modal residual KD network for spoken question answering, which highlights the utility of KD in developing sophisticated audio question-answering systems. Additionally, KD has been applied to fine-grained emotion classification in audio. Kim and Kang [167] utilized cross-modal distillation with audio–text fusion for emotion classification using BERT and Wav2vec 2.0, illustrating the integration of KD in enhancing emotion detection capabilities in audio data.

KD significantly enhances various speech-processing tasks by enabling small models to achieve high accuracy and efficiency in speech recognition, synthesis, sentiment analysis, and translation. These advancements are crucial for developing more responsive and natural-sounding virtual assistants, transcription services, and other voice-activated applications, thereby improving user experiences and operational efficiency.

Furthermore, KD enables small models to perform speech emotion analysis, accurately detecting and interpreting emotional intonation in spoken language. This ability is particularly valuable in customer service applications, where understanding a customer's emotional state can lead to a more empathetic and effective response. In terms of speech translation, KD allows for real-time translation of spoken words, eliminating language barriers, and facilitating seamless communication between different languages. The distillation model performs these tasks efficiently, making it suitable for deployment in mobile applications and other resource-constrained environments that require fast and accurate audio processing.

In summary, KD's applications in visual and audio processing demonstrate its transformative impact in enhancing the capabilities of small models. By leveraging the benefits of larger models, KD ensures that compact models can deliver high-performance results in image- and voice-related tasks while maintaining efficiency and reducing computational requirements. This makes advanced AI technology more accessible and practical in a wide range of applications from consumer electronics to enterprise systems. KD's integration in visual and audio processing demonstrates its importance in driving the next generation of intelligent, efficient, and versatile AI solutions.

E. VERTICAL DOMAINS

KD plays a transformative role in multiple professional fields such as medicine, law, science, finance, and materials science, enhancing the efficiency of these fields by enabling smaller models to perform complex tasks efficiently and accurately.

1) MEDICINE

In the field of medicine, KD has significantly enhanced the effectiveness of surgical assistance, drug development, automatic monitoring, and assisted diagnosis [18]. By learning from larger models, smaller models can provide surgeons with real-time insights and recommendations during surgery, leading to improved surgical accuracy and safety. Meng et al. provided a comprehensive survey on the application of KD in medical data mining, highlighting how distilled models can deliver high performance while maintaining efficiency [168]. In surgical assistance, KD allows compact models to process and analyze surgical data in real-time, offering surgeons critical insights that enhance precision and safety. This real-time assistance is crucial during complex surgical procedures, where timely and accurate information can significantly impact outcomes.

In drug development, KD enables smaller models to analyze large datasets efficiently, identifying potential drug candidates more rapidly. This accelerates the drug discovery process and enhances the ability to find effective treatments. Qin et al. [169] demonstrated the effectiveness of KD in medical image segmentation, a technique essential in drug research for analyzing medical images and identifying biological markers. For patient monitoring, KD facilitates continuous and automated health monitoring systems. These systems can alert medical providers to abnormal or critical conditions in a timely manner, improving patient care and response times. Xing et al. [170] discussed the use of KD in medical image classification, which plays a crucial role in monitoring and diagnosing health conditions through imaging technologies.

In the realm of assisted diagnosis, KD models can accurately interpret medical images and patient data, providing reliable diagnostic support that helps doctors make informed decisions. Li and Shen [171] developed a hybrid framework based on KD for explainable disease diagnosis, illustrating how KD can enhance diagnostic accuracy while making the decision-making process more transparent. Wang et al. [172] introduced prototype KD for medical segmentation with missing modality, which improves the robustness and accuracy of segmentation tasks even when some data modalities are absent. This is particularly useful in scenarios where complete data is not always available.

Moreover, KD has been applied to multimodal hierarchical knowledge distillation for medical visual question answering (MHKD-MVQA), as explored by Wang et al. [52]. This approach integrates various data types to answer complex medical queries accurately, showcasing the versatility of KD in handling diverse medical data. Jaiswal et al. [173] presented ROS-KD, a robust stochastic KD approach for noisy medical imaging, demonstrating how KD can improve the robustness of models in dealing with noisy and imperfect data, which is common in medical imaging. KD significantly enhances various aspects of medical practice, from surgical assistance and drug development to patient monitoring and assisted diagnosis. By enabling smaller models to perform complex tasks with high accuracy and efficiency, KD improves healthcare delivery and patient outcomes.

2) LAW

In the field of law, KD supports legal search and classification of legal documents. By learning from large models, smaller models are able to handle a wide range of legal research tasks more efficiently, including quick retrieval of relevant case law, regulations, and legal precedents. This capability simplifies the research process for legal professionals, saves time, and improves accuracy [7]. Additionally, KD helps analyze complex regulations, ensuring that legal professionals can comply with changing legal standards. In the classification of legal documents, the distillation model can classify and organize documents accurately, facilitating the access and management of legal information. For example, Yuan et al. [174] highlighted the potential of deep learning-based legal judgment prediction, demonstrating how KD can enhance the efficiency and accuracy of legal search and judgment tasks. These distilled models can process vast amounts of legal texts to identify pertinent information quickly.

In auxiliary judgments, KD helps smaller models analyze complex legal cases and provide recommendations or insights that assist judges and legal professionals in making informed decisions. Ma [175] discussed artificial intelligence-assisted decision-making methods for legal judgments, emphasizing the role of KD in supporting these advanced analytical tasks.

Regarding regulatory analysis, KD allows smaller models to keep up with the evolving legal landscape, ensuring that legal professionals can comply with changing legal standards. Yang et al. [176] explored self-knowledge distillation techniques, which can be applied to continuously update and refine the understanding of complex regulations, thereby improving compliance and regulatory analysis.

In the classification of legal documents, KD enables the accurate organization and management of extensive legal information. Xu et al. [177] demonstrated how KD could help distinguish confusing law articles for legal judgment prediction, which is crucial for accurately categorizing and organizing legal documents.

By utilizing KD, legal professionals can enhance their workflow, ensure compliance with up-to-date regulations, and access organized and relevant legal information more efficiently. This results in more streamlined legal processes and improved legal outcomes.

3) SCIENCE

In scientific research, KD significantly enhances the capabilities of mathematical formula analysis, research assistance, and chemical and physical analysis. By leveraging knowledge from larger models, smaller models can efficiently process and analyze complex mathematical data, providing precise calculations and insights that are essential for advanced scientific research. Ma et al. [178] discussed the use of large language models for enhanced KD in scientific question answering, illustrating how distilled models can handle intricate scientific queries with high accuracy.

In the realm of chemical and physical analysis, KD enables smaller models to accurately analyze chemical reactions and physical processes. This is crucial for discovering new compounds and materials. Julka and Granitzer [179] applied KD with the Segment Anything Model (SAM) for planetary geological mapping, illustrating the effectiveness of KD in enhancing the precision and efficiency of geological analysis. Phuong and Lampert [75] provided insights into the mechanisms of KD, emphasizing how KD improves the performance of smaller models by transferring the expertise of larger models. This foundational understanding supports the application of KD in various scientific domains.

In mathematical analysis, Zhang et al. [180] discussed the use of teacher-student networks with multiple decoders for solving math word problems. KD helps these smaller models to break down complex problems into manageable parts, leading to more accurate solutions.

KD also plays a significant role in learning gravitational dynamics with unknown disturbances. Lin et al. [181] conducted an initial feasibility study on physical knowledge distillation, demonstrating how KD can enhance the learning of deep nets for understanding gravitational dynamics, which is vital for advancements in robotics and automation. Furthermore, KD has been applied to accelerate molecular graph neural networks, as shown by Kelvinius et al. [182]. This application of KD in chemical research helps in the efficient processing and analysis of molecular structures, aiding in the discovery of new drugs and materials.

Van Keulen et al. [183] discussed teaching and learning distillation in chemistry laboratory courses, highlighting the educational benefits of KD in simplifying complex scientific concepts for students. This approach not only improves learning outcomes but also equips future scientists with the tools to handle sophisticated scientific analyses. KD significantly enhances scientific research by enabling smaller models to perform complex mathematical, chemical, and physical analyses efficiently. This advancement facilitates automated data interpretation, supports innovative research, and contributes to the discovery of new scientific insights.

4) FINANCE

In the financial sector, KD has significantly enhanced the effectiveness of financial forecasting [184], risk management, financial planning [185], and automated stock trading volume prediction [186]. By leveraging the knowledge from larger models, smaller financial models can deliver accurate and timely financial forecasts, enabling investors to make well-informed decisions [184]. For instance, Fang and Lin [187] explored prior KD based on financial time series, demonstrating how KD can improve the accuracy and efficiency of financial forecasting models. These models can process vast amounts of financial data and provide precise predictions, helping stakeholders navigate the complex financial landscape. Floratos et al. [184] discussed online KD for financial timeseries forecasting, highlighting the real-time application of KD in continuously updating and refining financial predictions.

In terms of risk management, KD enhances the ability to identify and mitigate potential financial risks, ensuring the stability and safety of financial operations. Tang and Liu [188] introduced a distributed knowledge distillation framework for financial fraud detection based on transformers, showing how KD can be applied to detect and prevent fraudulent activities efficiently.

KD also plays a pivotal role in financial planning. By distilling knowledge from larger, more comprehensive models, smaller models can offer personalized advice and strategies to help individuals and businesses achieve their financial goals. Yi et al. [189] proposed a long-short dual-mode knowledge distillation framework for empirical asset pricing models in digital financial networks, illustrating how KD can be utilized to provide robust financial planning tools that adapt to dynamic market conditions.

In the realm of automated trading, KD has enabled the development of efficient trading algorithms that can execute trades quickly and accurately based on real-time market data. Shen and Kurshan [190] discussed temporal knowledge distillation for time-sensitive financial services applications, emphasizing the importance of KD in creating responsive and effective automated trading systems. These distilled models can analyze market trends, predict price movements, and execute trades with precision, optimizing trading strategies and outcomes.

Moreover, Wang et al. [61] explored generating long financial reports using conditional variational autoencoders with KD, demonstrating how KD can streamline the process of financial reporting and analysis, making it more efficient and comprehensive. KD significantly improves various financial sector applications by enabling smaller models to perform complex tasks with high accuracy and efficiency. This advancement supports financial forecasting, risk management, financial planning, and automated trading, thereby enhancing the overall effectiveness and reliability of financial services.

5) MATERIAL

In the field of materials science, KD significantly enhances material discovery and design, material property prediction and analysis, and synthesis path optimization. By distilling knowledge from larger models, smaller models can accelerate the discovery of new materials, analyze datasets of chemical compounds, and predict their properties with high precision. KD also improves the efficiency of material design, providing insights into optimal structure and composition. In synthesis path optimization, distillation models can identify the most efficient and cost-effective methods for synthesizing new materials, speeding up the development process and reducing costs. For instance, VECCHIO's work on StableMaterials demonstrates how KD, combined with semi-supervised learning, can enhance the diversity and efficiency of material generation. This approach allows smaller models to leverage the extensive knowledge embedded in larger models to generate innovative material compositions [191].

Das et al. [192] introduced Crysgnn, a framework that distills pre-trained knowledge to improve property prediction for crystalline materials. This technique enables smaller models to accurately predict material properties, which is crucial for identifying suitable materials for various applications. Similarly, Chiang et al. [193] developed LLaMP, a large language model tailored for high-fidelity materials knowledge retrieval and distillation, illustrating the potential of KD in extracting and applying detailed materials knowledge efficiently.

In the realm of material design, Zhang and Saniie [194] showcased the use of knowledge distillation-based transformer neural networks for characterizing steel material microstructures. This method enhances data-efficient ultrasonic nondestructive evaluation (NDE) systems, enabling precise material analysis and quality control. Additionally, Smith et al. [195] explored a defect detection model for industrial products that combines attention mechanisms with KD, highlighting its effectiveness in identifying defects in materials and improving quality assurance processes.

KD also plays a vital role in synthesis path optimization. By learning from larger, comprehensive models, smaller models can determine the most efficient and cost-effective synthesis routes for new materials. This capability is essential for accelerating the development of new materials and reducing associated costs, thereby fostering innovation in material science. KD significantly contributes to materials science by enhancing material discovery and design, property prediction, and synthesis path optimization. These advancements enable smaller models to perform complex analyses with high precision and efficiency, supporting the development of new materials and improving existing processes.

Overall, KD's application in these different areas demonstrates its great potential to enhance the capabilities of smaller models. By enabling high-performance tasks to be executed on models with lower computational requirements, KD makes advanced AI technologies more accessible and practical across a wide range of specialized fields. This expands the range of applications of AI technology and promotes innovation and efficiency in medicine, law, science, finance and materials science. Figure 7 illustrates the classification of application fields for KD LLM.

IV. PERFORMANCE OPTIMIZATION

KD plays a key role in optimizing the performance of LLMs by employing various techniques to compress and accelerate their performance. KD is widely used to compress LLMs, allowing them to become more efficient without significantly reducing performance. This process involves transferring knowledge from a larger, more complex model (the teacher model) to a smaller, more efficient model (the student model), thereby reducing the computational resources required for deployment.

TABLE 9. Overview of KD in vertical domains.

Aspect	Description	References
Medicine - Surgical Assistance	Real-time insights, improved accuracy and safety.	[18], [52], [168], [173]
Medicine - Drug Development	Efficient analysis of large datasets, rapid drug discovery.	[169]
Medicine - Automatic Monitoring	Continuous health monitoring, timely alerts.	[170]
Medicine - Assisted Diagnosis	Accurate interpretation of medical images, reliable support.	[171]
Law - Legal Search	Efficient retrieval of case law, regulations, precedents.	[7], [174]
Law - Auxiliary Judgments	Analyze cases, provide recommendations.	[175]
Law - Regulatory Analysis	Compliance with evolving standards.	[176]
Law - Classification of Legal Documents	Organize and manage legal information.	[177]
Science - Mathematical Formula Analysis	Efficient processing of complex data, precise calculations.	[178], [180]
Science - Research Assistance	Handle scientific queries with high accuracy.	[178]
Science - Chemical and Physical Analysis	Analyze chemical reactions, physical processes.	[179], [181]–[183]
Finance - Financial Forecasting	Accurate and timely financial forecasts.	[184], [187]
Finance - Risk Management	Identify and mitigate financial risks.	[188]
Finance - Financial Planning	Personalized advice and strategies.	[189]
Finance - Automated Trading	Efficient trading algorithms, real-time market data.	[61], [190]
Materials Science - Material Discovery and Design	Accelerate material discovery, analyze chemical compounds.	[191]–[193]
Materials Science - Property Prediction and Analysis	Accurately predict material properties.	[192], [194]
Materials Science - Synthesis Path Optimization	Identify efficient synthesis methods.	[195]

Parameter pruning and quantification are the key techniques in this process. Parametric pruning reduces the size and computational complexity of a model by removing less important parameters. Pruning can be done at different levels, such as the neuron level, the hierarchy level, or the entire network level, depending on the trade-off between model size and precision [45]. The pruned model retains the basic features while being lighter and faster to execute. On the other hand, quantization reduces the accuracy of the model parameters, thus significantly reducing the size and computational load of the model. This process involves converting the floating-point weights of the model to a low-level representation, such as an 8-bit integer, while maintaining an acceptable level of precision [46]. The quantified model is particularly suitable for deployment on edge devices with limited computing power. In addition, model pruning and structural optimization techniques are designed to simplify the architecture of LLMs to improve their performance. Structured pruning, such as removing entire neurons, filters, or layers that contribute least to the model output, results in a more efficient network structure [196]. Combining KD with structural modifications can create student models that are not only smaller but also better suited to specific tasks, resulting in more efficient and specialized models [197].

Soft targets are used to smooth the probability distribution output by the teacher model, making it easier for the student model to learn. These soft targets are controlled by the temperature parameter T, which can amplify or diminish the logits of the teacher model [33]. When the temperature T is increased, the softmax function's output probability distribution becomes smoother, making it easier for the student model to learn the knowledge.

$$P_{\text{teacher}}(i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

where:

- $P_{\text{teacher}}(i)$ is the soft target probability of the teacher model for class i,
- z_i is the logit (i.e., the unnormalized output) for class i,
- T is the temperature parameter, usually greater than 1.

The loss function for KD combines the standard cross-entropy loss with the KD loss from the teacher model's soft targets. This combination ensures that the student model learns both the actual labels and mimics the teacher model's output distribution [33].

$$L = \alpha L_{\text{hard}}(y, y_{\text{student}}) + (1-\alpha) L_{\text{soft}}(P_{\text{teacher}}, P_{\text{student}}) \tag{2}$$

where:

- L is the total loss,
- $L_{\text{hard}}(y, y_{\text{student}})$ is the cross-entropy loss of the student model based on the true labels y and the student model's output y_{student},
- $L_{\text{soft}}(P_{\text{teacher}}, P_{\text{student}})$ is the cross-entropy loss between the soft targets from the teacher model and the student model's output,
- α is a balancing factor to weight the two loss components.

The cross-entropy loss functions are defined as:

$$L_{\text{hard}}(y, y_{\text{student}}) = -\sum_i y_i \log(y_{\text{student},i}) \tag{3}$$

$$L_{\text{soft}}(P_{\text{teacher}}, P_{\text{student}}) = -\sum_i P_{\text{teacher},i} \log(P_{\text{student},i}) \tag{4}$$

The quantization process converts floating-point weights to a lower-bit representation (e.g., 8-bit integers) to reduce model size and computational load [46]. This formula first normalizes the floating-point weights to the [0, 1] range, then maps them to the integer range of the quantization bits, and finally converts them back to the original range in a lower-bit

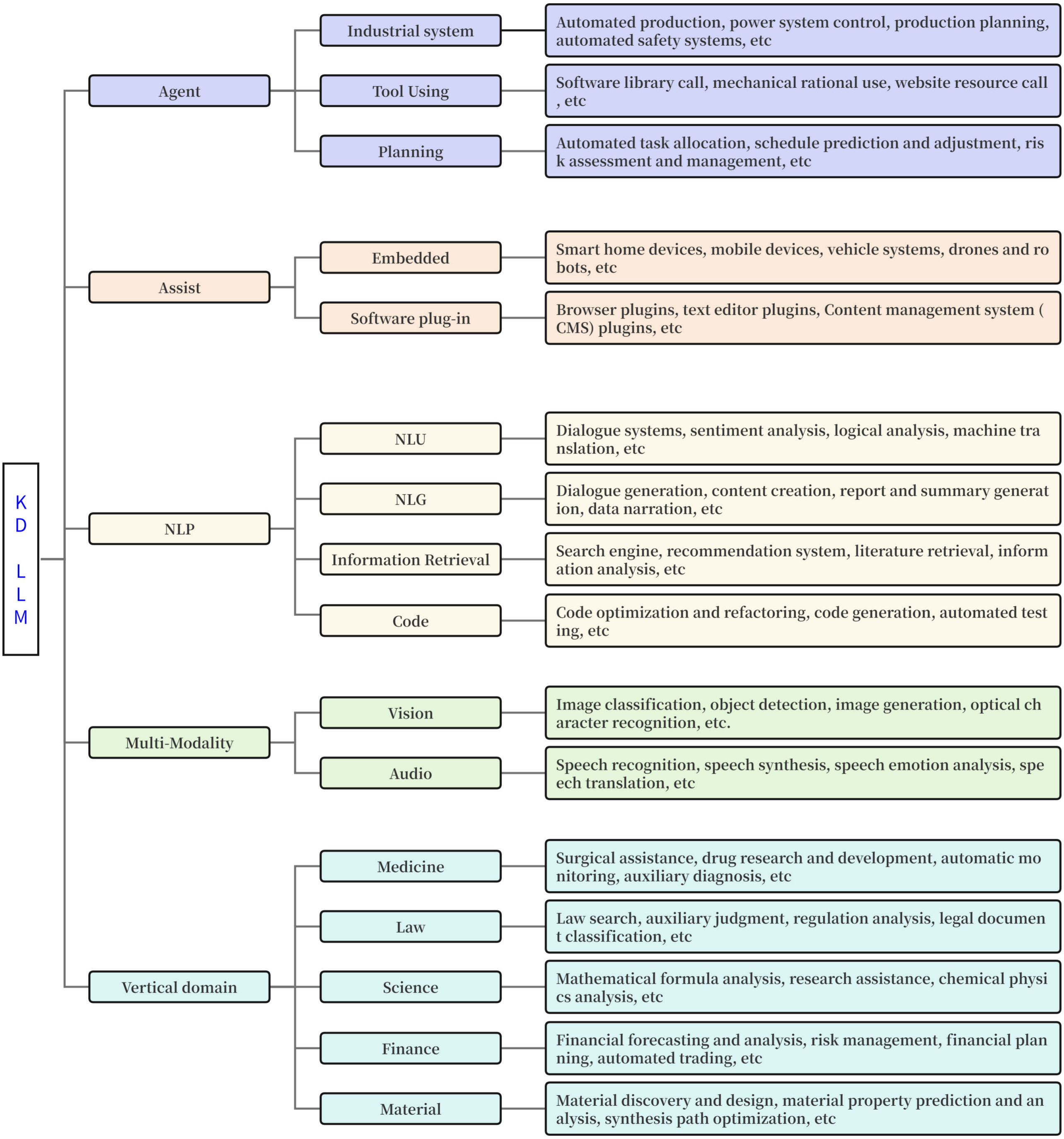

KD LLM
Agent
Industrial system
Automated production, power system control, production planning, automated safety systems, etc
Tool Using
Software library call, mechanical rational use, website resource call , etc
Planning
Automated task allocation, schedule prediction and adjustment, risk assessment and management, etc
Assist
Embedded
Smart home devices, mobile devices, vehicle systems, drones and robots, etc
Software plug-in
Browser plugins, text editor plugins, Content management system (CMS) plugins, etc
NLP
NLU
Dialogue systems, sentiment analysis, logical analysis, machine translation, etc
NLG
Dialogue generation, content creation, report and summary generation, data narration, etc
Information Retrieval
Search engine, recommendation system, literature retrieval, information analysis, etc
Code
Code optimization and refactoring, code generation, automated testing, etc
Multi-Modality
Vision
Image classification, object detection, image generation, optical character recognition, etc.
Audio
Speech recognition, speech synthesis, speech emotion analysis, speech translation, etc
Vertical domain
Medicine
Surgical assistance, drug research and development, automatic monitoring, auxiliary diagnosis, etc
Law
Law search, auxiliary judgment, regulation analysis, legal document classification, etc
Science
Mathematical formula analysis, research assistance, chemical physics analysis, etc
Finance
Financial forecasting and analysis, risk management, financial planning, automated trading, etc
Material
Material discovery and design, material property prediction and analysis, synthesis path optimization, etc

FIGURE 7. Application field classification.

representation.

$$Q(w) = \text{round}\left(\frac{w - w_{\min}}{w_{\max} - w_{\min}} \cdot (2^b - 1)\right) \cdot \frac{w_{\max} - w_{\min}}{2^b - 1} + w_{\min} \tag{5}$$

where:

- $Q(w)$ is the quantized weight,
- w is the original floating-point weight,
- $w_{\min}$ and $w_{\max}$ are the minimum and maximum values of the original weights,
- b is the number of bits for quantization (e.g., 8 for 8-bit integers).

These equations and techniques are fundamental to the process of knowledge distillation, enabling the creation of smaller, faster, and more efficient models while maintaining high performance.

In order to evaluate the performance of a distillation model, appropriate indicators need to be selected and benchmarks set to reflect the efficiency and accuracy of the model. Common performance metrics include accuracy, latency, throughput, and model size. Accuracy measures how correct a model is when performing a task, such as classification or generation, while delaying evaluation of the time it takes for the model to generate output is critical for real-time applications. Throughput evaluates the number of tasks handled by the model per unit time, and model size refers to the storage requirements of the model, which is particularly important for deployment on resource-constrained devices. It is very important to select and set the appropriate evaluation index to accurately evaluate the performance of distillation model. These metrics should be consistent with specific use cases and deployment environments. For example, in real-time applications such as conversational AI, latency and throughput are key metrics, while for on-device AI applications, model size and accuracy are a top priority [33], [45].

Many case studies highlight the effectiveness of KD in improving LLMs performance. DistilBERT, for example, is a small, fast version of BERT implemented via KD that retains 97 percent of BERT language understanding while increasing speed by 60percent and reducing volume by 40 percent [15]. Similarly, TinyBERT uses KD to compress BERT, resulting in a model that is 7.5 times smaller and 9.4 times faster than Bert-Base, with minimal performance loss [16]. MobileBERT optimizes performance on mobile and edge devices by combining KD with structural modifications to maintain competitive accuracy [57]. These case studies show how KD can significantly improve the efficiency and feasibility of LLMs in real-world applications, making it more accessible and usable in a variety of environments without sacrificing performance.

V. CHALLENGES AND FUTURE DIRECTIONS OF KNOWLEDGE DISTILLATION

KD faces several significant challenges that need to be addressed to maximize its potential. One of the primary challenges is the heavy reliance on large datasets and the associated high cost of data labeling [6]. Effective KD requires extensive and diverse training data to ensure that the student model can accurately capture the knowledge of the teacher model. However, acquiring and labeling such large datasets is both time-consuming and expensive, posing a barrier to the widespread adoption of KD [33]. Another major challenge is ensuring the effectiveness and robustness of knowledge transfer. The student model must not only replicate the performance of the teacher model but also generalize well to new, unseen data. Achieving this requires sophisticated techniques to prevent overfitting and to ensure that the distilled knowledge remains relevant across different contexts and applications [6].

Another challenge related to the use of KD in sensitive fields like healthcare and legal systems is the rise of significant ethical concerns. In healthcare, KD's dependence on large datasets for model training raises issues surrounding patient privacy, data security, and the potential misuse of sensitive medical information [198]. Similarly, the legal sector faces the critical need for client confidentiality, privacy, and accountability in sensitive legal tasks, which is essential for ethical practice [7], [199]. As a result, both areas require the development of robust ethical frameworks to mitigate the potential risks associated with the application of KD.

One of the main challenges in KD is addressing the transfer gap between teacher and student models. As Niu et al. [200] highlight, the effectiveness of KD is often hindered by the differences in architecture and capacity between the large teacher model and the smaller student model. This transfer gap can lead to inefficient knowledge transfer, where the student model struggles to fully utilize the distilled knowledge. Additionally, Zhang et al. [201] note that in the context of vision transformers, there are unique challenges related to maintaining the fine-grained spatial information and ensuring the robustness of distilled models. Another significant challenge is the application of KD in federated learning environments, where data privacy and distributed training add layers of complexity. According to Qin et al. [202], federated learning introduces issues such as inconsistent data distributions and communication overhead, complicating the KD process further.

A new research area in KD is focused on exploring the performance gap between teacher and student models. Huang et al. introduced an innovative KD technique named DiffKD, which aims to bridge this gap by explicitly denoising and aligning features through the use of diffusion models [203]. Diffusion models represent a cutting-edge category of deep generative models that have demonstrated exceptional success across various applications, such as image synthesis, video generation, and molecular design [204]. These models are a prime example of self-supervised learning, as they operate without requiring labeled data [205]. For instance, some studies have indicated that a diffusion model can be trained using just a single image of the "Marina Bay Sands," enabling it to generate similar images that incorporate additional towers resembling the "Sands Skypark" [206]. Prior research has highlighted the challenges associated with sample size in KD [6], particularly in domains like medicine, where obtaining data samples can be both challenging and expensive [206]. Therefore, applying the diffusion model offers a promising approach to address these sampling challenges.

Future research in KD should focus on developing more sophisticated methods to bridge the transfer gap between teacher and student models. One promising direction is the exploration of adaptive KD techniques that dynamically adjust the distillation process based on the specific characteristics of the models involved, as suggested by the work of Niu et al. [200]. Additionally, as Habib et al. [201] propose, there is a need for advanced strategies in vision transformers to preserve essential spatial information and enhance model robustness. In the realm of federated learning, Qin et al. [202] recommend the development of new algorithms that can handle heterogeneous data distributions and reduce communication costs, making KD more feasible in decentralized environments. Finally, integrating KD with other machine learning paradigms, such as semi-supervised and unsupervised learning, could further expand its applicability and effectiveness across various domains, as discussed by Alkhulaifi et al. [14].

Looking ahead, several trends are likely to shape the future of knowledge distillation. One such trend is the integration of multi-modal KD, which involves transferring knowledge across different modalities such as text, images, and audio. This approach can create more versatile and comprehensive models capable of handling a wider range of tasks and data types [40]. Additionally, there is growing interest in adaptive and online knowledge distillation methods. These approaches involve continuously updating and refining the student model as new data becomes available, enabling the model to adapt to changing conditions and requirements in real-time. This can significantly enhance the model's performance and applicability in dynamic environments [17]. The development of more efficient and scalable distillation techniques will also play a crucial role in the future of KD. Techniques that reduce the computational overhead of distillation and improve the efficiency of the training process are essential for making KD more accessible and practical for a broader range of applications. Furthermore, advances in understanding the theoretical foundations of knowledge distillation could lead to the development of more effective and robust distillation strategies, ultimately enhancing the performance and reliability of distilled models.

VI. CONCLUSION

KD serves as a powerful technique for optimizing the performance of LLMs by compressing them into more efficient, smaller models without significant loss of accuracy. Throughout this survey, we have explored various aspects of KD, including its basic concepts, core techniques, and diverse applications across multiple domains such as industrial systems, embedded systems, natural language processing, multimodality, and specialized vertical domains. The application of KD in these fields has demonstrated substantial improvements in efficiency and performance, making advanced AI capabilities more accessible and practical for real-world deployment.

Despite its successes, KD faces several significant challenges that need to be addressed to maximize its potential. These challenges include the reliance on large, annotated datasets, the effectiveness and robustness of knowledge transfer, and the computational overhead associated with the distillation process. Multi-modal KD is confronted with additional complexities, such as the need to align diverse data representations, maintain inter-modal relationships, and navigate the intricacies of various architectures, which are often compounded by a scarcity of annotated datasets. Likewise, online KD faces difficulties in adapting to evolving data streams, reducing computational demands, and ensuring synchronized and scalable knowledge transfer in real-time settings. Future research directions in KD emphasize the necessity of integrating multi-modal knowledge distillation, developing adaptive and online distillation strategies, and enhancing the efficiency and scalability of distillation techniques. Progress in these domains will significantly improve the applicability and effectiveness of KD, solidifying its role as an essential component in the advancement of AI technologies.

Overall, KD represents a vital strategy in the field of AI, enabling the creation of efficient and high-performing models that are well-suited for a wide range of applications. Continued research and innovation in KD will undoubtedly contribute to the advancement of AI, providing more powerful, versatile, and accessible solutions to meet the growing demands of various industries and research domains.

REFERENCES

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. E. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 27730–27744.

[2] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 1877–1901.

[3] G. Team, "Gemini: A large language model," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 831–842.

[4] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," 2022, *arXiv:2206.07682*.

[5] S. Badshah and H. Sajjad, "Quantifying the capabilities of LLMs across scale and precision," 2024, *arXiv:2405.03146*.

[6] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021.

[7] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," 2024, *arXiv:2402.13116*.

[8] D. Walawalkar, Z. Shen, and M. Savvides, "Online ensemble model compression using knowledge distillation," in *Proc. 16th Eur. Conf. Comput. Vision (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 18–35.

[9] B. Chandra Das, M. Hadi Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," 2024, *arXiv:2402.00888*.

[10] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[11] S. Barandoni, F. Chiarello, L. Cascone, E. Marrale, and S. Puccio, "Automating customer needs analysis: A comparative study of large language models in the travel industry," 2024, *arXiv:2404.17975*.

[12] Z. Allen-Zhu and Y. Li, "Physics of language models: Part 3.1, knowledge storage and extraction," 2023, *arXiv:2309.14316*.

[13] L. Ren, T. Wang, Z. Jia, F. Li, and H. Han, "A lightweight and adaptive knowledge distillation framework for remaining useful life prediction," *IEEE Trans. Ind. Informat.*, vol. 19, no. 8, pp. 9060–9070, Aug. 2022.

[14] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *PeerJ Comput. Sci.*, vol. 7, p. e474, Apr. 2021.

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[16] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.

[17] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.

[18] C.-H. Wang, T. Lin, G. Chen, M.-R. Lee, J. Tay, C.-Y. Wu, M.-C. Wu, H. R. Roth, D. Yang, C. Zhao, W. Wang, and C.-H. Huang, "Deep learning-based diagnosis and localization of pneumothorax on portable supine chest X-ray in intensive and emergency medicine: A retrospective study," *J. Med. Syst.*, vol. 48, no. 1, pp. 1–10, Dec. 2023.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 3111–3119.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*.

[21] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, NJ, USA: Pearson Education, Inc., 2008.

[22] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[23] J. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Jun. 1990.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[27] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Naacl-HLT*, Jan. 2018, p. 2.

[28] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2021, pp. 8748–8763.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[31] (2024). *Large Language Models: A Complete History*. Accessed: Jun. 24, 2024. [Online]. Available: https://voicebot.ai/large-language-models-history-timeline/

[32] M. Shao, A. Basit, R. Karri, and M. Shafique, "Survey of different large language model architectures: Trends, benchmarks, and challenges," *IEEE Access*, vol. 12, pp. 188664–188706, 2024.

[33] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[34] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[35] A. Jafari, M. Rezagholizadeh, P. Sharma, and A. Ghodsi, "Annealing knowledge distillation," 2021, *arXiv:2104.07163*.

[36] D. Hwang, K. Chai Sim, Y. Zhang, and T. Strohman, "Comparison of soft and hard target RNN-T distillation for large-scale ASR," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[37] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2016, *arXiv:1612.03928*.

[38] Y. Qu, W. Deng, and J. Hu, "H-AT: Hybrid attention transfer for knowledge distillation," in *Proc. 3rd Chin. Conf. Pattern Recognit. Comput. Vis. (PRCV)*, Nanjing, China. Cham, Switzerland: Springer, Jan. 2020, pp. 249–260.

[39] J. Gou, L. Sun, B. Yu, S. Wan, and D. Tao, "Hierarchical multi-attention transfer for knowledge distillation," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 20, no. 2, pp. 1–20, Feb. 2024.

[40] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2827–2836.

[41] K. Gupta, D. Gautam, and R. Mamidi, "CViL: Cross-lingual training of vision-language models using knowledge distillation," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 1734–1741.

[42] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2018, pp. 1607–1616.

[43] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[44] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," 2018, *arXiv:1802.05668*.

[45] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–11.

[46] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[47] S. Ganguly, R. Nayak, R. Rao, U. Deb, and A. P. Prathosh, "AdaKD: Dynamic knowledge distillation of ASR models using adaptive loss weighting," 2024, *arXiv:2405.08019*.

[48] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," 2018, *arXiv:1810.05270*.

[49] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag, "What is the state of neural network pruning," *Proc. Mach. Learn. Res.*, vol. 2, pp. 129–146, Mar. 2020.

[50] Y. Choi, M. El-Khamy, and J. Lee, "Learning low-precision neural networks without straight-through estimator," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 8, pp. 96963–96974, May 2020, doi: 10.1109/ACCESS.2020.2996936. [Online]. Available: https://ieeexplore.ieee.org/document/9098870

[51] X. Yang, J. Ye, and X. Wang, "Factorizing knowledge in neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2022, pp. 73–91.

[52] J. Wang, S. Huang, H. Du, Y. Qin, H. Wang, and W. Zhang, "MHKD-MVQA: Multimodal hierarchical knowledge distillation for medical visual question answering," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2022, pp. 567–574.

[53] Y. Cao, Q. Ni, M. Jia, X. Zhao, and X. Yan, "Online knowledge distillation for machine health prognosis considering edge deployment," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 27828–27839, Aug. 2024.

[54] J. Houyon, A. Cioppa, Y. Ghunaim, M. Alfarra, A. Halin, M. Henry, B. Ghanem, and M. Van Droogenbroeck, "Online distillation with continual learning for cyclic domain shifts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 2437–2446.

[55] X. He, I. Nassar, J. Kiros, G. Haffari, and M. Norouzi, "Generate, annotate, and learn: NLP with synthetic text," 2021, *arXiv:2106.06168*.

[56] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2Vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 1298–1312.

[57] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," 2020, *arXiv:2004.02984*.

[58] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," 2019, *arXiv:1906.08237*.

[59] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*.

[60] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[61] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," 2020, *arXiv:2003.10555*.

[62] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," 2020, *arXiv:2006.03654*.

[63] F. Iandola, A. Shaw, R. Krishna, and K. Keutzer, "SqueezeBERT: What can computer vision teach NLP about efficient neural networks?" in *Proc. SustaiNLP, Workshop Simple Efficient Natural Lang. Process.*, 2020, pp. 124–135.

[64] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "Ernie 2.0: A continual pre-training framework for language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 5, pp. 8968–8975. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6428

[65] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, Dec. 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.5

[66] L. Martin, B. Müller, P. J. O. Surez, Y. Dupont, L. Romary, É. V. de La Clergerie, D. Seddah, and B. Sagot, "Camembert: A tasty French language model," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7203–7219. [Online]. Available: https://aclanthology.org/2020.acl-main.645

[67] M. Ai, Y. Xie, S. X. Ding, Z. Tang, and W. Gui, "Domain knowledge distillation and supervised contrastive learning for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9452–9462, Sep. 2023.

[68] Z. Wang, Z. Li, D. He, and S. Chan, "A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117671.

[69] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Industrial cyber-physical systems-based cloud IoT edge for federated heterogeneous distillation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5511–5521, Aug. 2021.

[70] Z. Shi, Y. Li, and C. Liu, "Knowledge distillation-based information sharing for online process monitoring in decentralized manufacturing system," *J. Intell. Manuf.*, vol. 36, no. 3, pp. 2177–2192, Mar. 2025.

[71] Z.-K. Zhang, M.-L. Zhou, R. Shao, M. Li, and G. Li, "A defect detection model for industrial products based on attention and knowledge distillation," *Comput. Intell. Neurosci.*, vol. 2022, Oct. 2022, Art. no. 6174255.

[72] A. A. U. Rakhmonov, B. Subramanian, B. Olimov, and J. Kim, "Extensive knowledge distillation model: An end-to-end effective anomaly detection model for real-time industrial applications," *IEEE Access*, vol. 11, pp. 69750–69761, 2023.

[73] J. Gou, L. Sun, B. Yu, S. Wan, W. Ou, and Z. Yi, "Multilevel attention-based sample correlations for knowledge distillation," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 7099–7109, May 2023.

[74] L. Guan, F. Qiao, X. Zhai, and D. Wang, "Model evolution mechanism for incremental fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[75] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5142–5151. [Online]. Available: http://proceedings.mlr.press/v97/phuong19a.html

[76] Q. Lin, B. Xu, Z. Huang, and R. Cai, "From large to tiny: Distilling and refining mathematical expertise for math word problems with weakly supervision," in *Proc. Int. Conf. Intell. Comput.* Cham, Switzerland: Springer, Jan. 2024, pp. 251–262.

[77] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. Victoria Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.

[78] A. Zeng, M. Liu, R. Lu, B. Wang, X. Liu, Y. Dong, and J. Tang, "AgentTuning: Enabling generalized agent abilities for LLMs," 2023, *arXiv:2310.12823*.

[79] Y. Da, F. Brahman, A. Ravichander, K. R. Chandu, K.-W. Chang, Y. Choi, and B. Y. Lin, "Lumos: Learning agents with unified data, modular design, and open-source LLMs," in *Proc. ICLR Workshop Large Lang. Model (LLM) Agents*, Jan. 2023, pp. 1–11.

[80] S. Qiao, N. Zhang, R. Fang, Y. Luo, W. Zhou, Y. E. Jiang, C. Lv, and H. Chen, "AutoAct: Automatic agent learning from scratch for QA via self-planning," 2024, *arXiv:2401.05268*.

[81] J. Zhang, "Graph-ToolFormer: To empower LLMs with graph reasoning ability via prompt augmented by ChatGPT," 2023, *arXiv:2304.11116*.

[82] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez, "Gorilla: Large language model connected with massive Apis," 2023, *arXiv:2305.15334*.

[83] Y. Yang, E. Chern, X. Qiu, G. Neubig, and P. Liu, "Alignment for honesty," 2023, *arXiv:2312.07000*.

[84] Q. Tang, Z. Deng, H. Lin, X. Han, Q. Liang, B. Cao, and L. Sun, "ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases," 2023, *arXiv:2306.05301*.

[85] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun, "ToolLLM: Facilitating large language models to master 16000+ real-world Apis," 2023, *arXiv:2307.16789*.

[86] L. Yuan, Y. Chen, X. Wang, Y. R. Fung, H. Peng, and H. Ji, "CRAFT: Customizing LLMs by creating and retrieving from specialized toolsets," 2023, *arXiv:2309.17428*.

[87] S. Gao, Z. Shi, M. Zhu, B. Fang, X. Xin, P. Ren, Z. Chen, J. Ma, and Z. Ren, "Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 16, pp. 18030–18038.

[88] C. Wang, W. Luo, Q. Chen, H. Mai, J. Guo, S. Dong, X. Xiaohua, Z. Li, L. Ma, and S. Gao, "MLLM-tool: A multimodal large language model for tool agent learning," 2024, *arXiv:2401.10727*.

[89] W. Shen, C. Li, H. Chen, M. Yan, X. Quan, H. Chen, J. Zhang, and F. Huang, "Small LLMs are weak tool learners: A multi-LLM agent," 2024, *arXiv:2401.07324*.

[90] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 9118–9147.

[91] T. Sumers, K. Marino, A. Ahuja, R. Fergus, and I. Dasgupta, "Distilling internet-scale vision-language models into embodied agents," 2023, *arXiv:2301.12507*.

[92] C. Yang, J. Pan, X. Gao, T. Jiang, D. Liu, and G. Chen, "Cross-task knowledge distillation in multi-task recommendation," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 4, pp. 4318–4326.

[93] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2599–2608.

[94] C. Sun, T. Jiang, S. Zonouz, and D. Pompili, "Fed2KD: Heterogeneous federated learning for pandemic risk assessment via two-way knowledge distillation," in *Proc. 17th Wireless Demand Netw. Syst. Services Conf. (WONS)*, Mar. 2022, pp. 1–8.

[95] Y. Li et al., "Personal LLM agents: Insights and survey about the capability, efficiency and security," 2024, *arXiv:2401.05459*.

[96] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[97] Y. Kong, J. Ruan, Y. Chen, B. Zhang, T. Bao, S. Shi, G. Du, X. Hu, H. Mao, Z. Li, X. Zeng, and R. Zhao, "TPTU-v2: Boosting task planning and tool usage of large language model-based agents in real-world systems," 2023, *arXiv:2311.11315*.

[98] B. Jaiswal and N. Gajjar, "Deep neural network compression via knowledge distillation for embedded applications," in *Proc. Nirma Univ. Int. Conf. Eng. (NUiCONE)*, Nov. 2017, pp. 1–4.

[99] Y. Xie, H. Wu, J. Zhu, and H. Zeng, "Distillation embedded absorbable pruning for fast object re-identification," *Pattern Recognit.*, vol. 152, Aug. 2024, Art. no. 110437.

[100] S. Shaw, K. Tyagi, and S. Zhang, "Teacher-student knowledge distillation for radar perception on embedded accelerators," in *Proc. 57th Asilomar Conf. Signals, Syst., Comput.*, Oct. 2023, pp. 1035–1038.

[101] R. Gong, C. Wang, J. Li, and Y. Xu, "Lightweight fault diagnosis method in embedded system based on knowledge distillation," *J. Mech. Sci. Technol.*, vol. 37, no. 11, pp. 5649–5660, Nov. 2023.

[102] J. Chen, Y. Liu, and J.-A. Hou, "A lightweight deep learning network based on knowledge distillation for applications of efficient crack segmentation on embedded devices," *Struct. Health Monitor.*, vol. 22, no. 5, pp. 3027–3046, Sep. 2023.

[103] J. Cho and M. Lee, "Building a compact convolutional neural network for embedded intelligent sensor systems using group sparsity and knowledge distillation," *Sensors*, vol. 19, no. 19, p. 4307, Oct. 2019.

[104] C. Li, G. Li, H. Zhang, and D. Ji, "Embedded mutual learning: A novel online distillation method integrating diverse knowledge sources," *Int. J. Speech Technol.*, vol. 53, no. 10, pp. 11524–11537, May 2023.

[105] C. Wang, S. Zhou, K. Yu, D. Chen, B. Li, Y. Feng, and C. Chen, "Collaborative knowledge distillation for heterogeneous information network embedding," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 1631–1639.

[106] P. Xiao, T. Xu, X. Xiao, W. Li, and H. Wang, "Distillation sparsity training algorithm for accelerating convolutional neural networks in embedded systems," *Remote Sens.*, vol. 15, no. 10, p. 2609, May 2023.

[107] Y. Xiong, W. Zhai, X. Xu, J. Wang, Z. Zhu, C. Ji, and J. Cao, "Ability-aware knowledge distillation for resource-constrained embedded devices," *J. Syst. Archit.*, vol. 141, Aug. 2023, Art. no. 102912.

[108] C. Chen, "Distilling crowd knowledge from software-specific Q&A discussions for assisting developers knowledge search," Ph.D. dissertation, School Comput. Sci. Eng., Nanyang Technological Univ., Singapore, 2018.

[109] J.-M. Guo, J.-S. Yang, S. Seshathiri, and H.-W. Wu, "A light-weight CNN for object detection with sparse model and knowledge distillation," *Electronics*, vol. 11, no. 4, p. 575, Feb. 2022.

[110] J. Shi, Z. Yang, B. Xu, H. J. Kang, and D. Lo, "Compressing pre-trained models of code into 3 MB," in *Proc. 37th IEEE/ACM Int. Conf. Automated Softw. Eng.*, Oct. 2022, pp. 1–12.

[111] C. Sun, X. Wang, Z. Liu, Y. Wan, L. Zhang, and Y. Zhong, "SiamOHOT: A lightweight dual Siamese network for onboard hyperspectral object tracking via joint spatial–spectral knowledge distillation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5521112.

[112] J. Li, X. Chen, P. Zheng, Q. Wang, and Z. Yu, "Deep generative knowledge distillation by likelihood finetuning," *IEEE Access*, vol. 11, pp. 46441–46453, 2023.

[113] B. Fluri, "Change distilling. enriching software evolution analysis with fine-grained source code change histories," Ph.D. dissertation, Dept. Inform., Univ. Zurich, Zürich, Switzerland, 2008.

[114] J. Yuan, Q. Jiang, and Y. Pan, "The influence mechanism of knowledge network allocation mechanism on knowledge distillation of high-tech enterprises," *Comput. Intell. Neurosci.*, vol. 2022, Apr. 2022, Art. no. 8246234.

[115] Y. Yao, Z. Li, and H. Zhao, "GKT: A novel guidance-based knowledge transfer framework for efficient cloud-edge collaboration LLM deployment," 2024, *arXiv:2405.19635*.

[116] H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, N. Liu, S. Li, D. Zhu, H. Cai, L. Sun, Q. Li, D. Shen, T. Liu, and X. Li, "AugGPT: Leveraging ChatGPT for text data augmentation," 2023, *arXiv:2302.13007*.

[117] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT outperforms crowd workers for text-annotation tasks," *Proc. Nat. Acad. Sci. USA*, vol. 120, no. 30, Jul. 2023, Art. no. 2305016120.

[118] J. Gao, R. Pi, Y. Lin, H. Xu, J. Ye, Z. Wu, X. Liang, Z. Li, and L. Kong, "Self-guided noise-free data generation for efficient zero-shot learning," in *Proc. 11th Int. Conf. Learn. Represent.*, Jan. 2023, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=h5OpjGd_lo6

[119] R. Tang, X. Han, X. Jiang, and X. Hu, "Does synthetic data generation of LLMs help clinical text mining?" 2023, *arXiv:2303.04360*.

[120] Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," 2023, *arXiv:2302.01560*.

[121] W. Dai, L. Hou, L. Shang, X. Jiang, Q. Liu, and P. Fung, "Enabling multimodal generation on CLIP via vision-language knowledge distillation," 2022, *arXiv:2203.06386*.

[122] S. Fantazzini, P. Torroni, A. E. Ziri, D. F. Alise, and F. Ruggeri, "Efficient knowledge distillation for green NLP models: Bridging the gap with large language models," Master's thesis, Dept. Comput. Sci. Eng., Univ. Bologna, Bologna, Italy, 2024. [Online]. Available: https://amslaurea.unibo.it/id/eprint/31796/

[123] A. Lin, J. Wohlwend, H. Chen, and T. Lei, "Autoregressive knowledge distillation through imitation learning," 2020, *arXiv:2009.07253*.

[124] H. Quteineh, S. Samothrakis, and R. Sutcliffe, "Enhancing task-specific distillation in small data regimes through language generation," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, vol. 29, no. 1, pp. 5955–5965.

[125] Y. Liu and Z. Lin, "Unsupervised pre-training for natural language generation: A literature review," 2019, *arXiv:1911.06171*.

[126] W. Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H. Ji, and M. Jiang, "A survey of knowledge-enhanced text generation," *ACM Comput. Surveys*, vol. 54, no. 11s, pp. 1–38, Jan. 2022.

[127] D. Larionov, J. Grünwald, C. Leiter, and S. Eger, "EffEval: A comprehensive evaluation of efficiency for MT evaluation metrics," 2022, *arXiv:2209.09593*.

[128] M. Jiang, B. Lin, S. Wang, Y. Xu, W. Yu, and C. Zhu, "Knowledge-augmented methods for natural language generation," in *Proc. Knowl.-Augmented Methods Natural Lang. Process.* Cham, Switzerland: Springer, Jan. 2024, pp. 41–63.

[129] Z. Wang, Y. Ren, X. Zhang, and Y. Wang, "Generating long financial report using conditional variational autoencoders with knowledge distillation," *IEEE Trans. Artif. Intell.*, vol. 5, no. 4, pp. 1669–1680, Apr. 2024.

[130] S. Shakeri, A. Sethy, and C. Cheng, "Knowledge distillation in document retrieval," 2019, *arXiv:1911.11065*.

[131] B. Dong, J. Hou, Y. Lu, and Z. Zhang, "Distillation $\approx$ early stopping? Harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network," 2019, *arXiv:1910.01255*.

[132] Z. Huang, P. Yu, and J. Allan, "Improving cross-lingual information retrieval on low-resource languages via optimal transport distillation," in *Proc. 16th ACM Int. Conf. Web Data Mining*, Feb. 2023, pp. 1048–1056.

[133] A. Vakili Tahami, K. Ghajar, and A. Shakery, "Distilling knowledge for fast retrieval-based chat-bots," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 2081–2084.

[134] S. Xiao, Z. Liu, W. Han, J. Zhang, D. Lian, Y. Gong, Q. Chen, F. Yang, H. Sun, Y. Shao, and X. Xie, "Distill-VQ: Learning retrieval oriented vector quantization by distilling knowledge from dense embeddings," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1513–1523.

[135] L. Gao, Z. Dai, and J. Callan, "Understanding BERT rankers under distillation," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, Sep. 2020, pp. 149–152.

[136] J. Tian, X. Xu, Z. Wang, F. Shen, and X. Liu, "Relationship-preserving knowledge distillation for zero-shot sketch based image retrieval," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 5473–5481.

[137] G. Izacard and E. Grave, "Distilling knowledge from reader to retriever for question answering," 2020, *arXiv:2012.04584*.

[138] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2336–2345.

[139] W. Lu, J. Jiao, and R. Zhang, "TwinBERT: Distilling knowledge to twin-structured compressed BERT models for large-scale retrieval," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2645–2652.

[140] S. Hofstätter, S. Althammer, M. Schröder, M. Sertkan, and A. Hanbury, "Improving efficient neural ranking models with cross-architecture knowledge distillation," 2020, *arXiv:2010.02666*.

[141] J. Song, Y. Chen, J. Ye, and M. Song, "Spot-adaptive knowledge distillation," *IEEE Trans. Image Process.*, vol. 31, pp. 3359–3370, 2022.

[142] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," 2016, *arXiv:1606.07947*.

[143] F. Ruffy and K. Chahal, "The state of knowledge distillation for classification," 2019, *arXiv:1912.10850*.

[144] Y. Matsubara, "Torchdistill: A modular, configuration-driven framework for knowledge distillation," in *Proc. Int. Workshop Reproducible Res. Pattern Recognit.* Cham, Switzerland: Springer, Jan. 2021, pp. 24–44.

[145] H. Raj Khan, D. Gupta, and A. Ekbal, "Towards developing a multilingual and code-mixed visual question answering system by knowledge distillation," 2021, *arXiv:2109.04653*.

[146] C. Li, M. Lin, Z. Ding, N. Lin, Y. Zhuang, Y. Huang, X. Ding, and L. Cao, "Knowledge condensation distillation," in *Proc. Eur. Conf. Comput. Vis. (ECCV).* Cham, Switzerland: Springer, 2022, pp. 19–35.

[147] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11943–11952.

[148] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision," in *Proc. Eur. Conf. Comput. Vis.(ECCV).* Cham, Switzerland: Springer, Jan. 2020, pp. 588–604.

[149] Y. Kanellopoulos, C. Makris, and C. Tjortjis, "An improved methodology on information distillation by mining program source code," *Data Knowl. Eng.*, vol. 61, no. 2, pp. 359–383, May 2007.

[150] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 6105–6114.

[151] G. Habib, T. Jan Saleem, and B. Lall, "Knowledge distillation in vision transformers: A critical review," 2023, *arXiv:2302.02108*.

[152] X. Chen, Q. Cao, Y. Zhong, J. Zhang, S. Gao, and D. Tao, "DearKD: Data-efficient early knowledge distillation for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12042–12052.

[153] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," 2021, *arXiv:2104.13921*.

[154] Y. Liu, K. Wang, G. Li, and L. Lin, "Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 5573–5588, 2021.

[155] J. Ni, R. Sarbajna, Y. Liu, A. H. H. Ngu, and Y. Yan, "Cross-modal knowledge distillation for vision-to-sensor action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 4448–4452.

[156] Y. Liu, C. Wu, S.-y. Tseng, V. Lal, X. He, and N. Duan, "KD-VLP: Improving end-to-end vision-and-language pretraining with object knowledge distillation," 2021, *arXiv:2109.10504*.

[157] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan, "TinyViT: Fast pretraining distillation for small vision transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2022, pp. 68–85.

[158] L. Zhang, Z. Chen, and Y. Qian, "Knowledge distillation from multi-modality to single-modality for person verification," in *Proc. Interspeech*, Aug. 2021, pp. 1897–1901.

[159] K. Li, L. Yu, S. Wang, and P. Heng, "Towards cross-modality medical image segmentation with online mutual knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, Jan. 2020, pp. 775–783.

[160] L. Gao, K. Xu, H. Wang, and Y. Peng, "Multi-representation knowledge distillation for audio classification," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5089–5112, Feb. 2022.

[161] Y. Gong, S. Khurana, A. Rouditchenko, and J. Glass, "CMKD: CNN/transformer-based cross-model knowledge distillation for audio classification," 2022, *arXiv:2203.06760*.

[162] Y. Chen, Y. Xian, A. S. Koepke, Y. Shan, and Z. Akata, "Distilling audio-visual knowledge by compositional contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7012–7021.

[163] J.-W. Jung, H.-S. Heo, H.-J. Shim, and H.-J. Yu, "Knowledge distillation in acoustic scene classification," *IEEE Access*, vol. 8, pp. 166870–166879, 2020.

[164] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Proc. Interspeech*, Aug. 2017, pp. 3697–3701. [Online]. Available: https://www.isca-speech.org/archive/Interspeech_2017/pdfs/0862.PDF

[165] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-CNN knowledge distillation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10096110/

[166] C. You, N. Chen, and Y. Zou, "MRD-net: Multi-modal residual knowledge distillation for spoken question answering," in *Proc. Thirtieth Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 3985–3991. [Online]. Available: https://www.ijcai.org/proceedings/2021/550

[167] D. Kim and P. Kang, "Cross-modal distillation with audio-text fusion for fine-grained emotion classification using BERT and wav2vec 2.0," *Neurocomputing*, vol. 506, pp. 168–183, Aug. 2022.

[168] H. Meng, Z. Lin, F. Yang, Y. Xu, and L. Cui, "Knowledge distillation in medical data mining: A survey," in *Proc. 5th Int. Conf. Crowd Sci. Eng.*, Oct. 2021, pp. 175–182.

[169] D. Qin, J.-J. Bu, Z. Liu, X. Shen, S. Zhou, J.-J. Gu, Z.-H. Wang, L. Wu, and H.-F. Dai, "Efficient medical image segmentation based on knowledge distillation," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3820–3831, Dec. 2021.

[170] X. Xing, Y. Hou, H. Li, Y. Yuan, H. Li, and M. Q.-H. Meng, "Categorical relation-preserving contrastive knowledge distillation for medical image classification," in *Medical Image Computing and Computer Assisted Intervention—MICCAI*, vol. 12901. Cham, Switzerland: Springer, 2021, pp. 163–173.

[171] X. Li and Q. Shen, "A hybrid framework based on knowledge distillation for explainable disease diagnosis," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121844.

[172] S. Wang, Z. Yan, D. Zhang, H. Wei, Z. Li, and R. Li, "Prototype knowledge distillation for medical segmentation with missing modality," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[173] A. Jaiswal, K. Ashutosh, J. F. Rousseau, Y. Peng, Z. Wang, and Y. Ding, "RoS-KD: A robust stochastic knowledge distillation approach for noisy medical imaging," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2022, pp. 981–986.

[174] L. Yuan, J. Wang, S. Fan, Y. Bian, B. Yang, Y. Wang, and X. Wang, "Automatic legal judgment prediction via large amounts of criminal cases," in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2019, pp. 2087–2091.

[175] W. Ma, "Artificial intelligence-assisted decision-making method for legal judgment based on deep neural network," *Mobile Inf. Syst.*, vol. 2022, pp. 1–9, Oct. 2022, doi: 10.1155/2022/4636485.

[176] Z. Yang, A. Zeng, Z. Li, T. Zhang, C. Yuan, and Y. Li, "From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17185–17194, doi: 10.1109/ICCV51070.2023.01576.

[177] N. Xu, P. Wang, L. Chen, L. Pan, X. Wang, and J. Zhao, "Distinguish confusing law articles for legal judgment prediction," 2020, *arXiv:2004.02557*.

[178] Y. Ma, C. Fan, and H. Jiang, "Sci-CoT: Leveraging large language models for enhanced knowledge distillation in small models for scientific QA," in *Proc. 9th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2023, pp. 2394–2398, doi: 10.1109/ICCC59590.2023.10507622.

[179] S. Julka and M. Granitzer, "Knowledge distillation with segment anything (SAM) model for planetary geological mapping," in *Proc. Int. Conf. Mach. Learn., Optim., Data Sci. (LOD)*. Cham, Switzerland: Springer, Jan. 2023, pp. 68–77, doi: 10.48550/arxiv.2305.07586.

[180] J. Zhang, L. Wang, Y. Zhang, B. Wang, F. Zhuang, H. Xiong, and Q. He, "Teacher-student networks with multiple decoders for solving math word problems," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jun. 2020, pp. 4011–4017, doi: 10.24963/ijcai.2020/554.

[181] H. Lin, Y. Sun, Y. Yuan, Y. Wang, and X. Cao, "Learning deep nets for gravitational dynamics with unknown disturbance through physical knowledge distillation: Initial feasibility study," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2658–2665, Feb. 2021, doi: 10.1109/LRA.2021.3060435.

[182] F. E. Kelvinius, D. Georgiev, A. P. Toshev, and J. Gasteiger, "Accelerating molecular graph neural networks via knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2024, pp. 1–11.

[183] H. V. Keulen, T. H. Mulder, M. J. Goedhart, and A. H. Verdonk, "Teaching and learning distillation in chemistry laboratory courses," *J. Res. Sci. Teaching*, vol. 32, no. 7, pp. 715–734, 1995, doi: 10.1002/tea.3660320707.

[184] P. Floratos, A. Tsantekidis, N. Passalis, and A. Tefas, "Online knowledge distillation for financial timeseries forecasting," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2022, pp. 1–6.

[185] M. Biehler, M. Guermazi, and C. Starck, "Using knowledge distillation to improve interpretable models in a retail banking context," 2022, *arXiv:2209.15496*.

[186] L. Li, Z. Zhang, R. Bao, K. Harimoto, and X. Sun, "Distributional correlation-aware knowledge distillation for stock trading volume prediction," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, Jan. 2022, pp. 105–120.

[187] J. Fang and J. Lin, "Prior knowledge distillation based on financial time series," in *Proc. IEEE 18th Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2020, pp. 429–434.

[188] Y. Tang and Z. Liu, "A distributed knowledge distillation framework for financial fraud detection based on transformer," *IEEE Access*, vol. 12, pp. 62899–62911, 2024.

[189] Y. Yi, K. Cui, M. Xu, L. Yi, K. Yi, X. Zhou, S. Liu, and G. Zhou, "A long-short dual-mode knowledge distillation framework for empirical asset pricing models in digital financial networks," *Connection Sci.*, vol. 36, no. 1, Dec. 2024, Art. no. 2306970.

[190] H. Shen and E. Kurshan, "Temporal knowledge distillation for time-sensitive financial services applications," 2023, *arXiv:2312.16799*.

[191] G. Vecchio, "StableMaterials: Enhancing diversity in material generation via semi-supervised learning," 2024, *arXiv:2406.09293*.

[192] K. Das, B. Samanta, P. Goyal, S. Lee, S. Bhattacharjee, and N. Ganguly, "CrysGNN: Distilling pre-trained knowledge to enhance property prediction for crystalline materials," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 6, pp. 7323–7331.

[193] Y. Chiang, E. Hsieh, C.-H. Chou, and J. Riebesell, "LLaMP: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation," 2024, *arXiv:2401.17244*.

[194] X. Zhang and J. Saniie, "Steel material microstructure characterization using knowledge distillation based transformer neural networks for data-efficient ultrasonic NDE system," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2022, pp. 1–4.

[195] J. Smith, L. Jones, and A. Brown, "Advanced techniques in neural network training: Knowledge distillation and beyond," *Neural Comput.*, vol. 34, no. 5, pp. 1234–1245, 2023.

[196] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*.

[197] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," 2017, *arXiv:1702.03044*.

[198] A. Mahajan and A. Bhat, "A survey on application of knowledge distillation in healthcare domain," in *Proc. 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2023, pp. 762–768.

[199] R. Zhang, H. Li, Y. Wu, Q. Ai, Y. Liu, M. Zhang, and S. Ma, "Evaluation ethics of LLMs in legal domain," 2024, *arXiv:2403.11152*.

[200] Y. Niu, L. Chen, C. Zhou, and H. Zhang, "Respecting transfer gap in knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 21933–21947.

[201] T. Zhang, X. Wang, G. Xu, and L. Zhang, "Efficient distillation of vision transformers via attention map and feature map alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Oct. 2022, pp. 1234–1243.

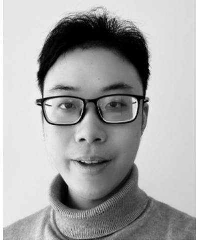
[202] L. Qin, T. Zhu, W. Zhou, and P. S. Yu, "Knowledge distillation in federated learning: A survey on long lasting challenges and new solutions," 2024, *arXiv:2406.10861*.

[203] T. Huang, Y. Zhang, M. Zheng, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge diffusion for distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023, pp. 65299–65316. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/cdddf13f06182063c4dbde8cbd5a5c21-Paper-Conference.pdf

[204] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–39, Nov. 2023, doi: 10.1145/3626235.

[205] M. Fuest, P. Ma, M. Gui, J. Schusterbauer, V. T. Hu, and B. Ommer, "Diffusion models and representation learning: A survey," 2024, *arXiv:2407.00783*.

[206] M. Abdollahzadeh, T. Malekzadeh, C. T. H. Teo, K. Chandrasegaran, G. Liu, and N.-M. Cheung, "A survey on generative modeling with limited data, few shots, and zero shot," 2023, *arXiv:2307.14397*.



DINGZONG ZHANG received the Master of Cyber Security degree from The University of Queensland, Brisbane, Australia, in 2024. His research interests include cyber security, large language models, and AI.

DEVI LISTIYANI is currently pursuing the Master of Cyber Security degree with The University of Queensland, Brisbane, Australia. She is an IT Professional in Indonesia. She is an awardee of Indonesia Endowment Fund for Education (LPDP) Scholarship. Her research interests include cyber security, data privacy, and AI.

PRIYANKA SINGH received the Ph.D. degree in image forensics. She is currently a Lecturer of cyber security with The University of Queensland, Brisbane, Australia. She earned the prestigious Postdoctoral Fellowships from Dartmouth College and University at Albany, USA. Her research interests include cyber security, encrypted domain processing, multimedia security, and digital forensics.

MANORANJAN MOHANTY received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2014. He was a Lecturer with the Center for Forensic Science, School of Mathematical and Physical Science, The University of Technology Sydney (UTS). He is currently an Assistant Teaching Professor of information systems with Carnegie Mellon University in Qatar. Before joining UTS, he was a Lecturer in digital security with The University of Auckland, New Zealand. His research interests include digital forensics and cyber security, with a current focus mainly on source camera attribution, child-explicit content detection, fake food detection, privacy-aware forensics, cloud and IoT forensics, and the application of deep learning and blockchain for forensics.