KCM: KAN-BASED COLLABORATION MODELS ENHANCE PRETRAINED LARGE MODELS

Guangyu Dai, Siliang Tang, Yueting Zhuang

Zhejiang University

ABSTRACT

In recent years, Pretrained Large Models(PLMs) have demonstrated exceptional performance across diverse tasks in multiple domains. Nevertheless, these large-scale models face challenges including substantial computational overhead, API dependency, and inaccuracies in domain-specific knowledge. Consequently, researchers proposed large-small model collaboration frameworks, leveraged easily trainable small models to assist large models, aiming to (1) reduce computational resource consumption while maintaining comparable accuracy, and (2) enhance large model performance in specialized domain tasks. However, this collaborative paradigm suffers from issues such as significant accuracy degradation, exacerbated catastrophic forgetting, and amplified hallucination problems induced by small model knowledge. To address these challenges, we propose a KAN-based Collaborative Model (KCM) as an improved approach to large-small model collaboration. The KAN utilized in KCM represents an alternative neural network architecture distinct from conventional MLPs. Compared to MLPs, KANs exhibit superior visualizability and interpretability while mitigating catastrophic forgetting. We deployed KCM in large-small model collaborative systems across three scenarios: language, vision, and vision-language cross-modal tasks. The experimental results demonstrate that, compared with pure large model approaches, the large-small model collaboration framework utilizing KCM as the collaborative model significantly reduces the number of large model inference calls while maintaining near-identical task accuracy, thereby substantially lowering computational resource consumption. Concurrently, the KAN-based small collaborative model markedly mitigates catastrophic forgetting, leading to significant accuracy improvements for long-tail data. Furthermore, in ablation studies, we compared the large-small model collaboration methods employing MLP-based small collaborative models (MCM) against those using KAN-based collaborative models (KCM). The results reveal that KCM demonstrates superior performance across all metrics compared to MCM.

***Index Terms*—** KAN, collaborative model, data distillation

1. INTRODUCTION

Recently, Pre-trained Large Models (PLMs) have revolutionized artificial intelligence research. These models, trained on vast datasets, adapt to numerous downstream tasks—such as GPT in language modality and BLIP-2 in vision-language modality. With their widespread adoption, inherent limitations have emerged, including high computational costs and insufficient domain-specific knowledge. Consequently, researchers have proposed collaborative frameworks where small models assist large models to mitigate these issues. Examples include leveraging small models for Retrieval-Augmented Generation (RAG) tasks[1, 2, 3], facilitating In-context Learning [4], and reducing overall computational expenditure through MLP-based collaborative small models[5, 6]. Studies reveal that while small models underperform large models on identical tasks, this performance gap primarily manifests in challenging samples (e.g. tail region samples in long-tail datasets). Thus, classifying data and substituting small models for large models on specific subsets significantly reduces computational costs.

This study introduces KAN-based Collaborative Models (KCMs), which reduce overall computational costs by minimizing large model invocations without compromising accuracy. Specifically, KCMs employ a small model to discriminate input data: relatively simple samples are processed by the small model, while complex samples are routed to the large model. Additionally, we propose bidirectional synergy between small and large models to enhance their respective performance. At the small model part, when samples are transferred to the large model, the reduced likelihood of belonging to the small model's proficient distribution is incorporated into prompts to augment the large model's decision-making. Conversely, when samples transition from the large model to the small model, knowledge distillation improves the small model's efficacy. To counter catastrophic forgetting, KAN—known for its robustness against this issue—is adopted as the foundational architecture for our small model. Experiments were conducted across three scenarios: language modality, visual modality, and vision-language multimodality. In ablation studies, KCMs demonstrated significant computational cost reduction compared to MLP-based Collaborative Models (MCMs), alongside slight accuracy improvements.

The primary contributions of this paper are as follows:

- KCM+PLM paradigm achieves comparable or minimally acceptable accuracy degradation relative to pure large model approaches, while consuming substantially fewer computational resources;
- KCM exhibits remarkable efficacy in alleviating catastrophic forgetting for long-tailed tasks;
- KCM outperforms MLP-based collaborative components when serving as small collaborative model.

2. RELATED WORKS

2.1. Large-Small Model Collaboration Methods

Large Language Models (LLMs) have exhibited remarkable proficiency across diverse NLP tasks in recent years, including text generation, question answering, and reasoning. Concurrently, Vision-Language Models (VLMs) have delivered unprecedented performance in vision-text multimodal tasks within the vision and multimodal domains. However, these large-scale models demand massive parameter scales and substantial computational resources, while demonstrating limited efficiency in domain-specific applications. This challenge has motivated researchers to leverage small

models as auxiliary components for large models, aiming to enhance scenario-specific performance by integrating outputs from computationally efficient and easily trainable small models with those of large models. Such approaches—wherein small models assist large models—have emerged in domains [7], including Retrieval-Augmented Generation (RAG) [1, 8], In-Context Learning (ICL) [4], language translation and summarization tasks [9].

2.2. KAN Introduction

Kolmogorov-Arnold Networks(KANs)[10] is a novel neural network architecture grounded in the Kolmogorov-Arnold representation theorem, fundamentally distinct from traditional Multi-Layer Perceptrons (MLPs). The difference between MLPs and KANs can be seen in fig.1. The defining characteristic of KANs is the placement of learnable activation functions on the network's edges, while summation operations occur at the nodes. In contrast, MLPs employ fixed activation functions at the nodes, with edges performing solely linear transformations. Algorithmically, this shift renders the activation functions in KANs learnable, unlike their fixed counterparts in MLPs. This learnability endows KANs with superior visualizability, interpretability, and interactivity. Furthermore, KANs can potentially mitigate the issue of catastrophic forgetting, offering a significant advantage over MLPs in several domains. However, these benefits come at the cost of increased computational expenses and reduced training efficiency.

Since its appearance, KAN has been extensively applied across

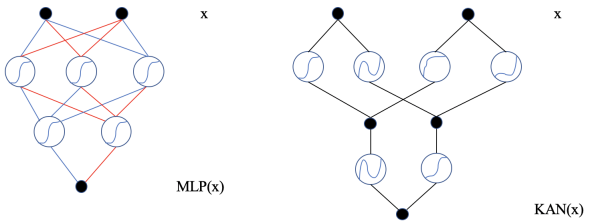

Fig. 1. The overview of MLP and KAN.MLP performs polynomial calculation at the node, and the activation function is the same; KAN is calculated on the edge, the activation functions may be different, and the nodes of KAN only play the role of addition.

diverse domains of deep learning [11], including time series forecasting [12], computer vision and image processing [13], graph neural networks (GNNs) [14], and collaborative filtering. Additionally, it has demonstrated significant strengths in interpretability and symbolic regression [15]. Across these research directions, KAN exhibits particularly powerful visualization and interpretability capabilities while inherently mitigating the issue of catastrophic forgetting.

3. METHODS

Our KCM-collaborate-PLM paradigm is shown in Fig.2. In this chapter, we will detail the structure of this paradigm, which includes: (1) Judge Kan Small Model classifies the input data, and for the data whose confidence C_x is greater than the threshold ϵ, we will input it into the large model as a knowledge aid for the small model to the large model; For the data with confidence C_x less than ϵ, we input it into the small model, thus reducing the call of the large model. (2) For the small model data of the input big model, we modify it to enhance the efficiency of the big model. (3) For small models, we use knowledge distillation to enhance the accuracy of small models. Below we will elaborate on these three methods.

3.1. KAN-Based Judgment Small Model

First, for each data x input to judgment model F_j, the confidence C_x is calculated by equation (1). For data with C_x less than epsilon, we input it into the large model, and for data with C_x greater than epsilon, we input it into the small model.

$$C_x = \frac{e^{y_i}}{\Sigma e^{y_n}}, y_i \in F_j(x) \tag{1}$$

3.2. Small Model Prompt Modification

This section mainly discusses how to help the judgment space of large model through the confidence judgment of small model, therefore we propose the method of prompt modification. For the data with low confidence C_x that are input into the large model, it is known that these data are not suitable for the classes that the small model is good at, so we can add a label similar to "the confidence of the small model is C_x" in the prompt, so that the large model pays less attention to these classes that the small model is good at, thus improving efficiency. Then we can calculate the result as follows:

$$R_l = F_l(x_i, prompt), y_i \in F_s(x) \tag{2}$$

F_l means the large model and R_l is the output result.

3.3. Large Model Distillation

This section mainly discusses the influence of large model on small model. Considering the extensive knowledge of large model, the ability boundary of small model can be expanded by knowledge distillation[16], so that more samples can be processed. In practice, for the input samples, when the reliability of a specific small model is low and the reliability of a large model is high, the learnable small model will obtain the prediction from the large model, so as to introduce additional knowledge and promote cost reduction. Such a process enables small models that can be learned to handle increasingly diverse samples. On the contrary, small models that can be learned will continue to learn high-confidence samples from specific small models to reduce the hallucination that knowledge from large models may produce.

$$C_l = \frac{e^{y_i}}{\Sigma e^{y_n}}, y_i \in F_l(x) \tag{3}$$

Similarly, we calculate the confidence of input data x the large model by Equation 3. For data with confidence greater than ϵ, we distill knowledge by calculating KL-divergence by equation 3 and 4, and iteratively calculate it to update Fs.

$$L_{ls} = KL(F_s(x), C_l), when C_l > \epsilon \tag{4}$$

$$L_{js} = KL(F_s(x), C_x), when C_x > \epsilon \tag{5}$$

3.4. Algorithm

To sum up, we can get the algorithm of KCM training process and inference process, and the whole process of the algorithm is shown separately in Algorithm 1 and Algorithm 2.

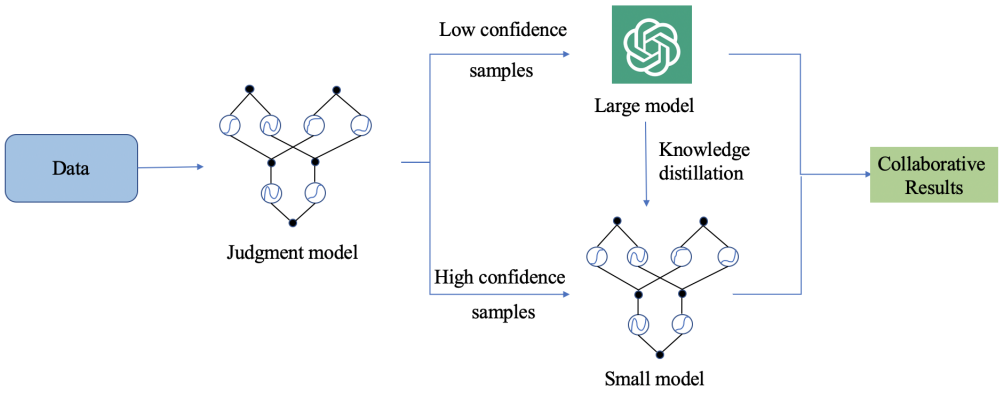

Fig. 2. The training process of KCM collaborate PLM. We input all the data into the judgment small model, input the samples with high confidence (representing suitable for small model prediction) into the small model, and input the samples with low confidence (representing suitable for large model prediction) into the large model, and combine the prompt modification of the small model with the knowledge distillation of the large model, and finally combine the prediction results of the two parts to get the result of the collaborative model.

Algorithm 1: KCM Training

Input: Input data x, Large model F_l, Judgment model F_j
Output: Small modelF_s
$F_s \leftarrow F_j$;
for *each sample* $x_i \in x$ **do**
  compute C_x and C_l;
  if $C_x > \epsilon$ **then**
    $x_1 \leftarrow x_i$;
  end
  else if $C_l > \epsilon$ **then**
    $x_2 \leftarrow x_i$;
  end
  else
    $x_3 \leftarrow x_i$;
  end
end
  Optimize F_s by KL-divergence;
return $\{F_s\}$

Algorithm 2: KCM Inference

Input: Input data x, Large model F_l, Judgment small model F_j, Small modelF_s
Output: Prediction result R
for *each sample* $x_i \in x$ **do**
  compute C_x and C_l;
  if $C_x > \epsilon$ **then**
    $R_i \leftarrow F_j(x_i)$;
  end
  else if $C_l > \epsilon$ **then**
    $R_i \leftarrow F_s(x_i)$;
  end
  else
    $R_i \leftarrow F_l(x_i)$;
  end
end
return $\{R\}$

4. EXPERIMENTS

Our experiments run on 2080ti GPU. It should be pointed out that considering the limitation of the experimental environment, the "large model" and "small model" used in the experiment are relative. For example, compared with ChatGPT, BERT is not a large language model, but it is undoubtedly a larger model than KCM as a collaboration, so we also regard BERT as large model in the experiment. We choose hypermeter $\epsilon = 0.98$.

4.1. Experiments on Language Task

Firstly, we apply KCM to the language modality, selecting sentiment analysis as the target task and utilizing the Amazon Product Data (APD) [17] as the experimental dataset. APD comprises 20 categories of product reviews annotated with positive or negative sentiment labels. We conduct experiments on 6 categories featuring balanced positive and negative samples, partitioning the dataset into training, validation, and test sets. We employ TKAN [12] as the KAN-based small model, while fine-tuned BERT [18] and ChatGPT serve as backbone large model . Experimental results using BERT and ChatGPT as backbone models are presented in Table 1 and Table 2 respectively.Table 1 demonstrates that the KCM-LLM collaboration framework achieves accuracy improvements across all categories except Kindle, with an average accuracy gain of 2.69(%). The Kindle category exhibits a minimal decrease of 0.21(%). Furthermore, the large model usage rate averages 81.83(%), indicating an approximate 18(%) reduction in invocation overhead. This reduction becomes more significant in Table 2, where ChatGPT serves as the backbone model. The KCM-augmented system achieves an average 67.85(%) reduction in invocation overhead, substantially decreasing computational resource consumption.
Building on this, the KCM-LLM collaboration system using ChatGPT shows slight accuracy decreases of 0.2(%) and 0.35(%) in the Games and Office categories, respectively, compared to standalone ChatGPT. Meanwhile, it achieves improvements (average about 0.72(%)) in the remaining four categories. This indicates that in the language modality, KCM significantly reduces large model invocation frequency—thereby lowering computational resource consumption—while maintaining comparable accuracy. This effect is more significant when employing larger backbone models (e.g. ChatGPT).

Table 1. BERT+KCM experiment accuracy on APD dataset.

Category	BERT(%)	KCM(%)	LM rate(%)
Games	90.39	95.82	84.02
Kindle	95.89	95.68	73.88
Baby	92.81	94.45	83.89
Movies	90.57	93.82	79.23
Electronics	91.76	93.66	82.12
Office	90.72	94.83	87.84

Table 2. ChatGPT+KCM experiment accuracy on APD dataset.

Category	ChatGPT (%)	KCM(%)	LM rate(%)
Games	96.22	96.02	33.75
Kindle	95.65	96.68	34.80
Baby	96.41	96.45	25.66
Movies	93.42	94.88	28.79
Electronics	95.41	95.76	36.68
Office	95.45	95.10	33.24

4.2. Experiments on Visual Task

In this section, we apply KCM to the visual modality, focusing on long-tail image classification—a challenging, prevalent vision task[19]. Following the setting of [20], we partition the CIFAR-100 dataset into head, med, and tail regions based on different numbers of samples. We employ CKAN [13] as the KAN-based small model and CLIP [21] as the large model, with results summarized in Table 3.Large Model rate(LM rate) mentioned in the table and later refers to the frequency with which the collaborative model invocate the large model. The KCM-assisted collaboration framework achieves: (1) a 3.29% accuracy improvement over the CKAN small model in the head region; (2) a 1.92% improvement over the CLIP large model in the medium region; (3) a 6.04% improvement over CLIP in the tail region. Notably, KCM assistance reduces the LM rate to 59.80% of the baseline, substantially lowering computational costs.

Table 3. Experiment accuracy on CIFAR-100-LT dataset.

Data	Small(%)	Large(%)	KCM(%)
head	70.25	60.00	71.54
med	48.21	57.28	59.20
tail	43.28	57.19	63.23
overall	53.25	58.18	64.66
LM rate	0.00	100.00	59.80

4.3. Experiments on Multimodal Task

We extend the application of KCM to visual-language multimodal tasks in this section, selecting image captioning as the target task and utilizing MSCOCO dataset[22] as our experiment dataset. We develop our small model using an encoder-decoder architecture, with CKAN serving as the encoder and TKAN as the decoder. Concurrently, BLIP-2 [23] as the backbone large model. Experimental results are summarized in Table 4. The results demonstrate that with KCM assistance, the frequency of large model invocation reduces to 62.32% , while achieving superior performance across BLEU-1 to BLEU-4 benchmarks[24]. This indicates that the KCM+BLIP-2 collaborative framework not only enhances task performance but also reduces computational cost by 37.68%.

Table 4. Experiment accuracy on MSCOCO dataset.

	Small	Large	KCM
BLEU-1	72.92	73.27	74.54
BLEU-2	55.73	60.04	60.89
BLEU-3	41.20	46.99	47.22
BLEU-4	30.28	36.11	36.42
LM rate	0.00%	100.00%	62.32%

4.4. KAN-Based Collabroative Model vs. MLP-Based Collabroative Model

In this section, we demonstrate the advantages of KAN-based collaborative small models (KCM) over MLP-based collaborative small models (MCM) through comparative analysis. Consistent with Section 4.2, we employ long-tail image classification on the CIFAR-100-LT dataset as the application scenario. The MCM architecture follows [6], utilizing a ResNet-101 model enhanced with prompt pruning and 2-stage confidence distillation as the collaborative model. Experimental results are summarized in Table 5. The data reveals that KCM achieves higher overall accuracy than MCM in the visual modality, with performance gaps widening toward the tail data. This superiority stems from KAN's enhanced capability in mitigating catastrophic forgetting, thereby yielding superior accuracy in tail data classification. KAN's edge-based learnable activations (Fig.1) enable adaptive feature reweighting for tail data, which mitigates catastrophic forgetting by preserving rare patterns during distillation—unlike MLPs' fixed node activations. Concurrently, KCM reduces large model invocation rate by 6.3 % compared with MCM. In summary, KCM achieves superior computational efficiency with reduced overhead.

Table 5. Ablation Experiment accuracy between MCM and KCM.

Data	MCM(%)	KCM(%)
head	70.99	71.54
med	57.34	59.20
tail	55.61	63.23
LM rate	66.10	59.80

5. CONCLUSION

With the continuous advancement of pre-trained large models, applications leveraging these models have become increasingly prevalent. However, direct utilization of pre-trained large models via APIs incurs substantial computational costs, constraining their broader adoption. Consequently, researchers have proposed auxiliary small models to collaborate with large models, reducing computational overhead by offloading partial data processing to small models. Our research extends this paradigm: to further harness the complementary strengths of small and large models, we introduce our KAN-Based Collabroative Model (KCM). Compared with MLPs, KANs exhibits marginally greater architectural complexity but delivers superior visualization capabilities, interpretability, and mitigation of catastrophic forgetting. At the scale of large models, the disadvantage of KAN's higher training costs relative to MLPs is circumvented. We validate our approach across diverse modalities and scenarios, demonstrating that it effectively reduces large model invocation frequency compared to standalone large model methods while enhancing task performance in certain modalities. Furthermore, we confirm that KAN, when employed as a collaborative small model, outperforms MLPs in both computational efficiency and task benchmark performance, demonstrating its superiority as an alternative.

6. REFERENCES

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *The Twelfth International Conference on Learning Representations*, 2024.

[2] Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen, "Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for LLMs," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar, Eds., Bangkok, Thailand, Aug. 2024, pp. 4420–4436, Association for Computational Linguistics.

[3] Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling, "Corrective retrieval augmented generation," 2024.

[4] Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley, "Small models are valuable plug-ins for large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar, Eds., Bangkok, Thailand, Aug. 2024, pp. 283–294, Association for Computational Linguistics.

[5] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu, "LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression," in *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

[6] Dong Chen, Yueting Zhuang, Shuo Zhang, Jinfeng Liu, Su Dong, and Siliang Tang, "Data shunt: Collaboration of small and large models for lower costs and better performance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, vol. 38, pp. 11249–11257.

[7] Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al., "A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness," *arXiv preprint arXiv:2411.03350*, 2024.

[8] Wenyu Huang, Guancheng Zhou, Hongru Wang, Pavlos Vougiouklis, Mirella Lapata, and Jeff Z. Pan, "Less is more: Making smaller language models competent subgraph retrievers for multi-hop KGQA," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, Eds., Miami, Florida, USA, Nov. 2024, pp. 15787–15803, Association for Computational Linguistics.

[9] Benjamin Bergner, Andrii Skliar, Amelie Royer, Tijmen Blankevoort, Yuki Asano, and Babak Ehteshami Bejnordi, "Think big, generate quick: Llm-to-slm for fast autoregressive decoding," *arXiv preprint arXiv:2402.16844*, 2024.

[10] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[11] Tianrui Ji, Yuntian Hou, and Di Zhang, "A comprehensive survey on kolmogorov arnold networks (kan)," *arXiv preprint arXiv:2407.11075*, 2024.

[12] Kunpeng Xu, Lifei Chen, and Shengrui Wang, "Kolmogorov-arnold networks for time series: Bridging predictive power and interpretability," *arXiv preprint arXiv:2406.02496*, 2024.

[13] Alexander Dylan Bodner, Antonio Santiago Tepsich, Jack Natan Spolski, and Santiago Pourteau, "Convolutional kolmogorov-arnold networks," *arXiv preprint arXiv:2406.13155*, 2024.

[14] Mehrdad Kiamari, Mohammad Kiamari, and Bhaskar Krishnamachari, "Gkan: Graph kolmogorov-arnold networks," *arXiv preprint arXiv:2406.06470*, 2024.

[15] Masoud Muhammed Hassan, "Bayesian kolmogorov arnold networks (bayesian_kans): A probabilistic approach to enhance accuracy and interpretability," *arXiv preprint arXiv:2408.02706*, 2024.

[16] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao, "Knowledge distillation: A survey," *International journal of computer vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[17] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[19] Yucan Zhou, Qinghua Hu, and Yu Wang, "Deep super-class learning for long-tail distributed image classification," *Pattern Recognition*, vol. 80, pp. 118–128, 2018.

[20] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang, "Trustworthy long-tailed classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6970–6979.

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

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19730–19742.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.