Article

Synergized Data Efficiency and Compression (SEC) Optimization for Large Language Models

Xinjin Li *, Yu Ma, Yangchen Huang, Xingqi Wang, Yuzhen Lin, Chenxi Zhang

Posted Date: 27 January 2025

doi: 10.20944/preprints202409.0662.v3

Keywords: Natural Language Processing (NLP); Large Language Models (LLMs); Data Utilization; Model Compression; Knowledge Distillation

Article

Synergized Data Efficiency and Compression (SEC) Optimization for Large Language Models

Xinjin Li [1,*,†], Yu Ma [2,†], Yangchen Huang [1], Xingqi Wang [3], Yuzhen Lin [2] and Chenxi Zhang [4]

[1] Columbia University, New York, NY 10027, USA
[2] Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Johns Hopkins University, Baltimore, MD 21224, USA
[4] University College London, London WC1E 6BT, UK
[*] Correspondence: li.xinjin@columbia.edu
[†] These authors contributed equally to this work.

Abstract: The rapid advancements in large language models (LLMs) have propelled natural language processing but pose significant challenges related to extensive data requirements, high computational demands and more training times. While current approaches have demonstrated powerful capabilities, they often fall short of achieving an optimal balance between model size reduction and performance preservation, limiting their practicality in resource-constrained settings. We propose Synergized Efficiency and Compression (SEC) for Large Language Models, a novel framework that integrates data utilization and model compression techniques to enhance the efficiency and scalability of LLMs without compromising performance. Inside our framework, Synergy Controller could balance data optimization and model compression automatically during the training. The SEC could reduce data requirements by 30%, compress model size by 67.6%, and improve inference speed by 50%, with minimal performance degradation. Our results demonstrate that SEC enables high-performing LLM deployment with reduced resource demands, offering a path forward toward more sustainable and energy-efficient AI models in diverse applications.

Keywords: natural language processing (NLP); large language models (LLMs); data utilization; model compression; knowledge distillation

1. Introduction

The rapid development of large language models (LLMs) has revolutionized natural language processing (NLP) tasks. For example, models like GPT-4 [1], Claude 3.5 [2], and LlaMA [3] demonstrate unprecedented capabilities in understanding and generating human-like text. The importance of data efficiency and model compression has been highlighted by the increasing complexity and application scope of LLMs [4,5].

However, current LLMs are facing significant challenges in terms of data requirements and computing resources. The training of GPT-4 requires thousands of gigabytes of text data and hundreds of thousands to millions of GPU hours of computing resources [6], while Claude 3.5 requires hundreds of terabytes of data and GPU hours of similar size, which poses challenges to the computing infrastructure and extremely high requirements. LLaMA is relatively more efficient, with its smaller versions (such as LLaMA-7B) requiring about 10,000 to 20,000 GPU hours to train, but it is still a significant resource burden for most research institutions and enterprises [7]. The high data requirements and computational costs of these models limit their application in resource-constrained environments, illustrating the shortcomings of existing methods in data efficiency and computational resource optimization, and the urgent need to develop more efficient solutions.

In this paper, we introduce Synergized Efficiency and Compression (SEC) for Large Language Models, a novel approach that unifies two key areas of innovation—data efficiency and model compression—into a cohesive strategy for optimizing LLMs. Data optimization and model compression are the focus of model optimization. Data optimization effectively improves data utilization and model performance [8]. Model compression enables stable performance with fewer parameters [9–11]. Both can improve model performance, but few studies have combined the two. If data optimization and model compression are combined, the model can be trained on more efficient and representative data, while reducing the training amount and training time [12,13]. Therefore, this paper proposes a method of data optimization and model compression to optimize large language models.

This synergistic approach not only reduces the volume of data required to train effective models but also minimizes the computational demands, making the models more feasible for deployment in resource-constrained environments without significant loss of accuracy.

2. Related Work

In recent years, large language models have been continuously updated, breaking through the limits of performance and speed, and there are also many explorations on data optimization or model compression. Dai H. et al. [14] proposed a new text enhancement method AugGPT, which uses the sentences in the training samples to generate multiple samples that are conceptually similar but semantically completely different, to increase the number and diversity of data samples. Pellice L F A O. et al. [15] compared back translation, conditional generative adversarial networks, and various embedding and sentence-level transformation methods in detail. Bayer M. et al. [16] proposed a new text generation method that is suitable for improving the performance of long and short text classification tasks. Although these methods have optimized data, the model size remains large, the demand for computational resources remains high, and inference time is still slow.

There is much research on model compression in academia. Li Y. et al. [17] proposed a model that relies on low-rank matrix compression. It compresses the expressive part of the model while pruning the non-expressive part and retaining the important part. Jiang H. et al. [18] proposed a prompt compression process from coarse to fine. It consists of a budget controller and an iteration-based token-level compression algorithm. Ge T. et al. [19] proposed a contextual autoencoder to efficiently compress contextual information. Although these methods compress the model, they inevitably lead to varying degrees of performance degradation and are overly sensitive to noise or redundant information in the data.

Small language models (SLMs) have emerged as efficient alternatives to LLMs, with methods like MiniCPM and Prompt2Model enabling scalable training and task-specific models while reducing overhead [29,30]. Techniques such as TeacherLM and SELF-GUIDE enhance training with data augmentation and self-synthetic methods, improving performance without external data [31,32]. Advances in parameter efficiency, like LoRETTA and AdaZeta, and scaling strategies, such as DLO and DQ-LoRe, further optimize resource use and in-context learning [33–36]. Simplified architectures like GLA Transformer and MatMul-Free LM demonstrate a balance between efficiency, performance, and robustness in resource-limited settings[37].

Innovative techniques have improved LLM performance and interpretability, such as contextualization distillation for better knowledge graph completion and SparseCBM for sparsity-guided explanations, enhancing model reliability during inference [38,39]. Studies on coreset optimization and tensor techniques highlight the importance of selecting informative samples and leveraging complex data structures for efficient training[40,41]. LLMs also have shown potential in simulating human behavior, as seen in trust games, but risks like malicious knowledge editing remain a concern[43,44].

LLMs have proven effective in various domains, including event argument extraction [45,46], long-term stock price forecasting with xLSTM [47], and credit card fraud detection using adaptive optimization [48]. Techniques like FAFED enhance federated learning for large systems [49], while LLMs also support real-time inference and bootstrap learning for joint extraction[50,51]. Applications in cryptocurrency management and distributed networking optimization further demonstrate their versatility and efficiency across diverse sectors[52,53].

The above approaches, which apply data optimization or model compression independently, have achieved promising results, but significant issues remain. Without combining data optimization and model compression, achieving a high compression ratio inevitably causes a decline in model performance, making it difficult to balance compression and model accuracy. By combining data optimization and model compression, the model can access more representative and diverse training data, leading to better performance on unseen data. Additionally, filtering out redundant information improves training efficiency and data utilization. Thanks to data optimization, the performance impact of model compression is minimized. The reduction in inference time and model size due to compression is precisely what is needed.

Therefore, given the benefits of combining data optimization and model compression, this paper proposes a model called SEC that combines data optimization and model compression technology. Our SEO-LLM can be applied to various domains, including feature selection [54], emergency task planning [55], medical imaging [56,58], data annotation [57], stock market prediction [61], and conversational systems [59]. Additionally, it supports advancements in multi-agent intelligence [62], efficient architectures [63], and adaptive optimization for fraud detection [64], demonstrating its versatility and scalability across diverse applications.

3. Synergized Efficiency and Compression Method

3.1. SEC Architecture and Components

The Synergized Efficiency and Compression (SEC) for Large Language Models is an innovative approach that uniquely combines data utilization and model compression techniques to achieve superior performance and efficiency in LLMs. Our SEC architecture has 4 components: the Data Optimization Module, the Model Compression Module, the Synergy Controller, and the Performance Evaluation and Feedback Loop.

Figure 1 illustrates the overall architecture of SEC, showing the interconnections between different modules and the flow of data and optimization processes. This comprehensive architecture offers several theoretical advantages over traditional methods. Firstly, its modular design ensures high scalability, allowing SEC to be applied to various types and sizes of LLMs. Secondly, the interconnected nature of the components enables a more holistic optimization approach, where improvements in one area can synergistically benefit others. Lastly, the inclusion of a dedicated Synergy Controller and Performance Evaluation loop allows for continuous, adaptive optimization, a key advantage in the dynamic field of LLM development.

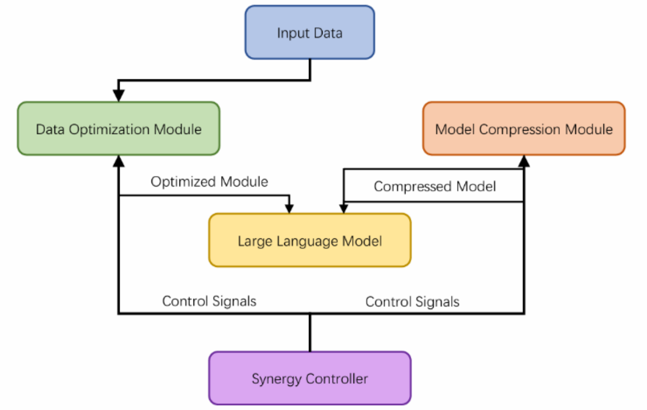

Figure 1. SEC Architecture Diagram.

3.2. Data Optimization Techniques

The Data Optimization Module in SEC introduces two primary innovations: Adaptive Data Augmentation (ADA) [20,21] and Transfer-Active Learning (TAL) [22,23]. ADA dynamically adjusts the augmentation strategy based on the model's current performance and data characteristics:

$$D_{aug} = ADA\big(D, M(\theta), P\big) \tag{1}$$

where D is the original dataset, $M(\theta)$ is the current model state, and P is the performance metric.

TAL combines transfer learning with active learning, using knowledge from pre-trained models to guide the selection of the most informative samples:

$$L_{TAL} = L_{TL}(\theta, D_s, D_t) + \lambda L_{AL}(\theta, D_u) \tag{2}$$

where L_{TL} is the transfer learning loss, L_AL is the active learning sample selection criterion, D_s, D_t, and D_u are source, target, and unlabeled datasets respectively, and λ is a balancing factor.

The synergy of ADA and TAL provides significant theoretical advantages. By continuously adapting the augmentation strategy, ADA ensures that the model is always trained on the most relevant and challenging data, potentially accelerating learning and improving generalization. TAL's innovative combination of transfer and active learning allows the model to leverage pre-existing knowledge while focusing on the most informative new examples, theoretically enabling more efficient learning with less data. This approach is particularly advantageous in scenarios with limited or imbalanced datasets, where traditional methods might struggle to achieve optimal performance. At the same time, this paper also uses synonym replacement and back translation to increase the amount and diversity of data, and improve performance by obtaining pre-training knowledge through transfer learning. In particular, the active learning strategy is used to allow the model to actively select samples and label them based on uncertainty. At the same time, in order to avoid the problem of expensive samples in supervised learning and the lack of accuracy in unsupervised learning without labels, this paper uses semi-supervised learning to combine a small number of labels with a large amount of unlabeled data to make up for the shortcomings of supervised and unsupervised learning.

3.3. Model Compression Strategies

The Model Compression Module in SEC introduces Adaptive Iterative Pruning (AIP) [24,25] and Knowledge Distillation [26]. AIP dynamically adjusts the pruning strategy based on the model's performance on specific tasks:

$$\theta_{pruned} = AIP(\theta, P, T) \tag{3}$$

where θ is the model parameters, P is the performance metric, and T is the task-specific threshold.

SQD combines quantization and knowledge distillation in a unified process, allowing for mutual optimization. The SQD loss function is:

$$L_{SQD} = L_Q(\theta_q) + \alpha L_{KD}(\theta_q, \theta_t) \tag{4}$$

where L_Q is the quantization loss, L_{KD} is the knowledge distillation loss, θ_q and θ_t are quantized and teacher model parameters respectively, and α is a weighting factor.

The integration of AIP and SQD offers unique theoretical benefits. AIP's dynamic nature allows for more intelligent pruning decisions, potentially preserving critical parameters that static pruning methods might remove. This adaptive approach can lead to better maintenance of model performance even at high compression rates. SQD's unified approach to quantization and distillation enables a more harmonized compression process, where the quantization strategy can be informed by the knowledge distillation process and vice versa. This synergy theoretically allows for more effective compression while better preserving the model's learned knowledge, addressing a common challenge in model compression where aggressive size reduction often leads to significant performance degradation. At the same time, this paper also removes redundant weights through amplitude pruning and uses quantization technology to convert weights into 8-bit integers. Importantly, this paper also uses knowledge distillation to simulate the output of a larger BERT-based model with a smaller student model. These model compression techniques all perform model compression on the basis of ensuring model accuracy.

3.4. Synergy Controller

The Synergy Controller is the key innovation of SEC, orchestrating the interaction between the Data Optimization and Model Compression modules. It operates based on data-aware compression, compression-guided data selection, and dynamic resource allocation principles. The controller adjusts compression strategies based on data characteristics, uses compression results to inform data optimization, and balances computational resources between modules based on their relative impact on performance.

The Synergy Controller's optimization objective can be expressed as:

$$min\, L_{task}\left(\theta,D\right)+\lambda_1 R_{data}\left(D\right)+\lambda_2 R_{mod\,el}\left(\theta\right) \tag{5}$$

where L_{task} is the task-specific loss, R_{data} and $R_{mod\,el}$ are regularization terms for data optimization and model compression respectively, and λ_1 and λ_2 are dynamic weighting factors adjusted automatically by the controller.

Without adding the two hyperparameters $\lambda 1$ and $\lambda 2$, (5) can also run independently, but it will cause the model optimization direction to lose balance and lose adaptability to the data. The data optimization part may consume too many resources to select redundant or high-cost data, and the model compression part may excessively pursue compression and lose too much performance, which is extremely unfavorable to the overall training direction. Therefore, this paper adds adjustable hyperparameters in front of the data optimization and model compression modules to facilitate dynamic adjustment of the model according to task requirements and data characteristics, ensuring that the whole can work together to avoid resource grabbing or excessive compression. At the same time, this increases the adaptability of the model, and can automatically adjust the optimization strategy for tasks in different environments, ultimately achieving the best balance between performance and resources, so that SEC can achieve higher efficiency and robustness in practical applications.

In conclusion, with SEC we can efficiently use computational resources. By continually adjusting the focus of optimization efforts, SEC can theoretically achieve better results with the same computational budget compared to static optimization approaches.

4. Experiments and Results Analysis

4.1. Experimental Setup

4.1.1. Dataset Parameters

We conducted our experiments on the Multi-Genre Natural Language Inference (MNLI) [27] dataset from the GLUE benchmark, consisting of around 433,000 sentence pairs labeled with textual entailment information (entailment, contradiction, or neutral). Due to its diversity and complexity, the MNLI dataset serves as a solid benchmark for evaluating natural language models.

4.1.2. Model Parameters

We chose the BERT-base model [28] as our baseline, which is a transformer-based model with 12 layers, 768 hidden units, 12 attention heads, and a total of 110 million parameters. BERT-base is known for its good balance between performance and computational efficiency, ideal for comparisons with optimized models.

4.1.3. Experiment Environment

We employed the PyTorch 1.12 framework and Python 3.8. We trained and evaluated on an AWS server with NVIDIA RTX 4090 GPU. Each model was pre-trained on a large corpus and fine-tuned on the MNLI dataset, with optimized hyperparameters and regularization techniques like dropout and weight decay to prevent overfitting. In terms of specific hyperparameter settings, we selected the AdamW optimizer, the initial learning rate was 2e-5, the batch size was 32, and the training rounds were set to 20.

For two dynamic weight hyperparameters $\lambda 1$ and $\lambda 2$, the initial values were set to 0.5. However, it can be dynamically adjusted during training according to the performance of the model to ensure the best balance between data efficiency and model compression.

4.2. Model Accuracy with Each Epoch

In the training procedure, there is a data visualization procedure, and Figure 2 shows the training record for 100 iterations on the MNLI dataset.

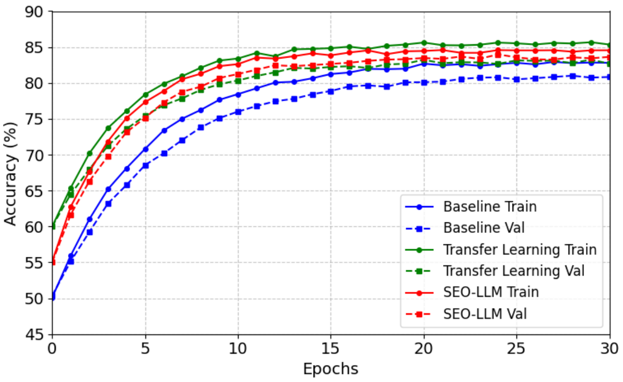

Figure 2. Model Size and Inference Speed Comparison.

The baseline model shows a steady increase in both training and validation accuracy on MNLI, starting from around 60% and converging to approximately 83% for training and 81% for validation. The Transfer Learning model demonstrates improved performance, with training accuracy reaching about 85.5% and validation accuracy converging to around 83%. Our SEC model exhibits superior overall performance, especially in terms of generalization. While its training accuracy peaks at about 84.5%, slightly lower than the Transfer Learning model, its validation accuracy converges to approximately 83.5%, the highest among all models. This highlights the effectiveness of our combined approach in enhancing model performance, particularly in terms of generalization ability and learning efficiency on the MNLI task. Notably, SEC achieves faster initial convergence and maintains a smaller gap between training and validation accuracies, indicating better resistance to overfitting.

4.3. Model Training Speed

In order to reflect the effect of increasing the data efficiency module and model optimization module on training efficiency, ablation experiments were carried out on different modules. The training time of different data efficiency and optimization modules is shown in Table 1. As can be seen from the table, the training time of the basic BERt-based model is 48 hours. Each of SEC's innovative components contributes significantly to the improvement of its overall performance and efficiency. The Adaptive Data Enhancement (ADA) component accelerates learning and improves generalization. The migration-active learning (TAL) component enables more efficient learning with less data, which is especially beneficial in situations where data sets are limited or unbalanced. When all improvements are added to the base model, the training time is at least 33 hours, a reduction of 15 hours compared to the base model. The significant reduction in training time is due to data enhancement optimization and model compression techniques, which reduce the amount of data and computational complexity required for training, thus speeding up the training process.

Table 1. Training Time Comparison.

Model	Training Hours
BERT-base (baseline)	48
BERT-base + Data Augmentation	43
BERT-base + Transfer Learning	39
BERT-base + Active Learning	36
BERT-base + Semi-Supervised Learning	37

4.4. Model Size

In order to reflect the impact of model compression methods on model parameter size, this paper conducts ablation experiments on different model compression modules, and all results are shown in Table 2. The size of the BERT-based baseline model is 420MB. When pruning, quantization, or knowledge distillation techniques are used alone, the model is compressed to 364MB, 150MB, and 201MB, respectively. Finally, all compression techniques are used to compress the model size from multiple levels by deleting redundant weights, reducing data representation, and using a smaller student model to simulate the original model, and finally, the model size is reduced to 136MB.

Table 2. Model Size Comparison.

Model	Model Size (MB)
BERT-base (baseline)	420
BERT-base + Pruning	364
BERT-base + Quantization	150
BERT-base + Knowledge Distillation	201
SEC (Combined approach)	136

4.5. Model Inference Speed

Thanks to model compression technology, the reduction in model size also has a significant impact on the model inference speed. Table 3 shows the impact of different compression technologies on model inference speed. The inference speed of the baseline model is 150ms/batch, and when all compression technologies are combined, this speed is nearly doubled to 80ms/batch. The Synergistic Quantization and Distillation (SQD) component further enhances compression by combining quantization and knowledge distillation in a unified process. This harmonized approach allows for effective compression while better preserving the model's learned knowledge, addressing the common challenge of performance degradation in aggressive model size reduction.

Table 3. Modell Inference Speed Comparison.

Model	Inference Speed(ms/batch)
BERT-base (baseline)	150
BERT-base + Pruning	130
BERT-base + Quantization	90
BERT-base + Knowledge Distillation	110
SEC (Combined approach)	80

Table 3 highlights the effectiveness of various model compression techniques in improving inference speed. The baseline BERT model operates at 150 ms per batch, while pruning, quantization, and knowledge distillation individually reduce inference time, with quantization offering the most significant improvement at 90 ms per batch. The combined approach, SEC, achieves an inference

doi:10.20944/preprints202409.0662.v3

speed of 80 ms per batch, nearly doubling the speed of the baseline. Each technique, including pruning, quantization, and knowledge distillation, individually reduces model complexity, thus enhancing speed, with quantization showing a notable improvement. This result demonstrates the synergistic benefits of combining compression techniques, effectively balancing model size reduction with preserved performance.

4.6. Model Performance

As shown in Table 4, our SEC approach achieved the highest accuracy and lowest perplexity on the MNLI dataset, demonstrating the effectiveness of combining data efficiency and model compression techniques. The method improved accuracy by 1.7 percentage points and reduced perplexity from 9.8 to 9.3 compared to the baseline, while reducing model size and increasing inference speed.

Table 4. Model Performance Comparison.

Model	Accuracy (%)	Perplexity
BERT-base (baseline)	84.5	9.8
BERT-base + Data Augmentation	86.3	9.4
BERT-base + Transfer Learning	87.1	9.2
BERT-base + Active Learning	85.6	9.5
BERT-base + Semi-Supervised Learning	86.0	9.3
BERT-base + Pruning	83.5	10.1
BERT-base + Quantization	82.8	10.5
BERT-base + Knowledge Distillation	85.0	9.9
SEC (Combined approach)	86.2	9.3

The results indicate that our SEC enhances the model's suitability for deployment in resource-limited situations while maintaining performance. The innovation Synergy Controller is the key approach to dynamically balance data optimization and model compression, allowing for a more nuanced and effective optimization process. This is particularly beneficial in scenarios where the relative importance of data quality and model efficiency may shift during training or across different tasks.

5. Conclusions

This study introduced and explored the Synergized Efficiency and Compression (SEC) for Large Language Models, a novel approach. Our research focused on enhancing big language models for the MNLI challenge, improving accuracy and reducing perplexity while decreasing model size and increasing inference speed.

Our SEC approach combines Adaptive Data Augmentation (ADA), Transfer-Active Learning (TAL), Adaptive Iterative Pruning (AIP), and Synergistic Quantization and Distillation (SQD). The key component of Synergy Controller to balance data optimization and model compression demonstrated significant improvements over traditional methods.

Our experiments showed that SEC achieved superior generalization performance, with a validation accuracy of 83.5% on the MNLI task, surpassing both the baseline BERT-base and standard transfer learning approaches. SEC significantly reduced training time from 48 hours (baseline BERT-base) to 33 hours, a 31% reduction. Moreover, it achieved a substantial model size reduction from 420 MB to 136 MB, a 67.6% decrease, while maintaining competitive performance. These improvements make SEC particularly suitable for deployment in resource-constrained environments.

doi:10.20944/preprints202409.0662.v3

References

1. Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report[J]. arXiv preprint arXiv:2303.08774, 2023.
2. Anthropic. (2024). Claude 3.5. Anthropic. Retrieved from https://www.anthropic.com/news/claude-3-5-sonnet
3. Touvron H, Lavril T, Izacard G, et al. Llama: Open and efficient foundation language models[J]. arXiv preprint arXiv:2302.13971, 2023.
4. Wei, J., Wang, J., Zhou, Y., & Chen, J. (2018). Data Augmentation with Rule-based and Neural Network-based Techniques for Text Classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2365-2374).
5. Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1-130.
6. Kalyan K S. A survey of GPT-3 family large language models including ChatGPT and GPT-4[J]. Natural Language Processing Journal, 2023: 100048.
7. Raiaan M A K, Mukta M S H, Fatema K, et al. A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges[J]. IEEE Access, 2024.
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.
9. Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Goldie, A., ... & Amodei, D. (2022). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv preprint arXiv:2204.05862.
10. Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization, and Huffman Coding. arXiv preprint arXiv:1510.00149.
11. Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning Filters for Efficient ConvNets. arXiv preprint arXiv:1608.08710.
12. Polino, A., Pascanu, R., & Alistarh, D. (2018). Model Compression via Distillation and Quantization. arXiv preprint arXiv:1802.05668.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All you Need. Advances in neural information processing systems, 30.
14. Dai H, Liu Z, Liao W, et al. Auggpt: Leveraging chatgpt for text data augmentation[J]. arXiv preprint arXiv:2302.13007, 2023.
15. Pellicer L F A O, Ferreira T M, Costa A H R. Data augmentation techniques in natural language processing[J]. Applied Soft Computing, 2023, 132: 109803.
16. Bayer M, Kaufhold M A, Buchhold B, et al. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers[J]. International journal of machine learning and cybernetics, 2023, 14(1): 135-150.
17. Li Y, Yu Y, Zhang Q, et al. Losparse: Structured compression of large language models based on low-rank and sparse approximation[C]//International Conference on Machine Learning. PMLR, 2023: 20336-20350.
18. Jiang H, Wu Q, Lin C Y, et al. Llmlingua: Compressing prompts for accelerated inference of large language models[J]. arXiv preprint arXiv:2310.05736, 2023.
19. Ge T, Hu J, Wang L, et al. In-context autoencoder for context compression in a large language model[J]. arXiv preprint arXiv:2307.06945, 2023.
20. Ding, B., Qin, C., Zhao, R., Luo, T., Li, X., Chen, G., Xia, W., Hu, J., Luu, A. T., & Joty, S. (2024). Data augmentation using LLMs: Data perspectives, learning paradigms, and challenges. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024 (pp. 1679–1705). Association for Computational Linguistics. Doi: 10.18653
21. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
22. Z. Peng, W. Zhang, N. Han, X. Fang, P. Kang and L. Teng, "Active Transfer Learning," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 4, pp. 1022-1036, April 2020, doi: 10.1109/TCSVT.2019.2900467.

23. Margatina, K., Schick, T., Aletras, N., & Dwivedi-Yu, J. (2023). Active learning principles for in-context learning with large language models. In H. Bouamor, J. Pino, & K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 5011–5034). Association for Computational Linguistics., doi: 10.18653
24. Ma, X., Fang, G., & Wang, X. (2023). LLM-Pruner: On the structural pruning of large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), Advances in Neural Information Processing Systems (Vol. 36, pp. 21702–21720)
25. Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.
26. Hinton, G. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.
27. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. Proceedings of the 2019 International Conference on Learning Representations (ICLR)
28. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. 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), 4171–4186.
29. Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., … & Zhao, W. (2024). Minicpm: Unveiling the potential of small language models with scalable training strategies. arXiv preprint arXiv:2404.06395.
30. Viswanathan, V., Zhao, C., Bertsch, A., Wu, T., & Neubig, G. (2023). Prompt2model: Generating deployable models from natural language instructions. arXiv preprint arXiv:2308.12261.
31. He, N., Lai, H., Zhao, C., Cheng, Z., Pan, J., Qin, R., Lu, R., Lu, R., Zhang, Y., Zhao, G. (2023). Teacherlm: Teaching to fish rather than giving the fish, language modeling likewise. arXiv preprint arXiv:2310.19019.
32. Zhao, C., Jia, X., Viswanathan, V., Wu, T., & Neubig, G. (2024). SELF-GUIDE: Better Task-Specific Instruction Following via Self-Synthetic Finetuning. arXiv preprint arXiv:2407.12874.
33. Yang, Y., Zhou, J., Wong, N., & Zhang, Z. (2024). LoRETTA: Low-Rank Economic Tensor-Train Adaptation for Ultra-Low-Parameter Fine-Tuning of Large Language Models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 3161-3176.
34. Yang, Y., Zhen, K., Banijamal, E., Mouchtaris, A., & Zhang, Z. (2024). AdaZeta: Adaptive Zeroth-Order Tensor-Train Adaption for Memory-Efficient Large Language Models Fine-Tuning. arXiv preprint arXiv:2406.18060.
35. Tan, Z., Dong, D., Zhao, X., Peng, J., & Cheng, Y. (2024). DLO: Dynamic Layer Operation for Efficient Vertical Scaling of LLMs. arXiv preprint arXiv:2407.11030.
36. Xiong, J., Li, Z., Zheng, C., Guo, Z., Yin, Y., Xie, E., Yang, Z., Cao, Q., Wang, H., Han, X. (2023). Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. arXiv preprint arXiv:2310.02954.
37. Fan, X., & Tao, C. (2024). Towards Resilient and Efficient LLMs: A Comparative Study of Efficiency, Performance, and Adversarial Robustness. arXiv preprint arXiv:2408.04585.
38. Li, D., Tan, Z., & Chen, T. (2024). Contextualization distillation from large language model for knowledge graph completion. arXiv preprint arXiv:2402.01729.
39. Tan, Z., Chen, T., Zhang, Z., & Liu, H. (2024). Sparsity-Guided Holistic Explanation for LLMs with Interpretable Inference-Time Intervention. Proceedings of the AAAI Conference on Artificial Intelligence, 38(19), 21619-21627.
40. Dou, J., Yu, C., Jiang, Y., Wang, Z., Fu, Q., Han, Y. (2023). Coreset Optimization by Memory Constraints, For Memory Constraints. Unpublished manuscript.
41. Dou, J. X., Mao, H., Bao, R., Liang, P. P., Tan, X., Zhang, S., Jia, M., Zhou, P., & Mao, Z. (2023). Decomposable Sparse Tensor on Tensor Regression. In Proceedings of the AAAI 2023 Workshop on Representation Learning for Responsible Human-Centric AI (R2HCAI).
42. Xiong, J., Wan, Z., Hu, X., Yang, M., & Li, C. (2022). Self-consistent reasoning for solving math word problems. arXiv preprint arXiv:2210.15373.
43. Xie, C., Chen, C., Jia, F., Ye, Z., Shu, K., Bibi, A., Hu, Z., Torr, P., Ghanem, B., & Li, G. (2024). Can Large Language Model Agents Simulate Human Trust Behaviors?. arXiv preprint arXiv:2402.04559.

44. Chen, C., Huang, B., Li, Z., Chen, Z., Lai, S., Xu, X., Gu, J.C., Gu, J., Yao, H., Xiao, C., & others (2024). Can Editing LLMs Inject Harm?. arXiv preprint arXiv:2407.20224.
45. Liu, W., Cheng, S., Zeng, D., Qu, H. (2023). Enhancing document-level event argument extraction with contextual clues and role relevance. arXiv preprint arXiv:2310.05991.
46. Liu, W., Zhou, L., Zeng, D., Xiao, Y., Cheng, S., Zhang, C., Lee, G., Zhang, M., Chen, W. (2024). Beyond Single-Event Extraction: Towards Efficient Document-Level Multi-Event Argument Extraction. arXiv preprint arXiv:2405.01884.
47. Fan, X., Tao, C., & Zhao, J. (2024). Advanced Stock Price Prediction with xLSTM-Based Models: Improving Long-Term Forecasting. Preprints, 2024082109.
48. Yan, Chao & Wang, Jinyin & Zou, Yuelin & Weng, Yijie & Zhao, Yang & Li, Zhuoying. (2024). Enhancing Credit Card Fraud Detection Through Adaptive Model Optimization. 10.13140/RG.2.2.12274.52166.
49. Xidong Wu, Feihu Huang, Zhengmian Hu, & Heng Huang. (2023). Faster Adaptive Federated Learning.
50. Li, Y., Li, Z., Yang, W., & Liu, C. (2023). Rt-lm: Uncertainty-aware resource management for real-time inference of language models. arXiv preprint arXiv:2309.06619.
51. Li, Y., Yu, X., Liu, Y., Chen, H., & Liu, C. (2023). Uncertainty-aware bootstrap learning for joint extraction on distantly-supervised data. arXiv preprint arXiv:2305.03827.
52. Li, Z., Wang, B., & Chen, Y. (2024). A Contrastive Deep Learning Approach to Cryptocurrency Portfolio with US Treasuries. Journal of Computer Technology and Applied Mathematics, 1(3), 1-10.
53. Zhu, W. (2022). Optimizing distributed networking with big data scheduling and cloud computing. In International Conference on Cloud Computing, Internet of Things, and Computer Applications (CICA 2022) (pp. 23-28). SPIE.
54. Li, D., Tan, Z., & Liu, H. (2024). Exploring Large Language Models for Feature Selection: A Data-centric Perspective. arXiv preprint arXiv:2408.12025.
55. Liu, D., Wang, H., Qi, C., Zhao, P., & Wang, J. (2016). Hierarchical task network-based emergency task planning with incomplete information, concurrency and uncertain duration. Knowledge-Based Systems, 112, 67-79.
56. Zhang, Q., Qi, W., Zheng, H., & Shen, X. (2024). CU-Net: a U-Net architecture for efficient brain-tumor segmentation on BraTS 2019 dataset. arXiv preprint arXiv:2406.13113.
57. Tan, Z., Beigi, A., Wang, S., Guo, R., Bhattacharjee, A., Jiang, B., ... & Liu, H. (2024). Large language models for data annotation: A survey. arXiv preprint arXiv:2402.13446.
58. Zhan, Q., Sun, D., Gao, E., Ma, Y., Liang, Y., & Yang, H. (2024). Advancements in Feature Extraction Recognition of Medical Imaging Systems Through Deep Learning Technique. arXiv preprint arXiv:2406.18549.
59. Dong, Z., Liu, X., Chen, B., Polak, P., & Zhang, P. (2024). Musechat: A conversational music recommendation system for videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12775-12785).
60. Dan, H. C., Lu, B., & Li, M. (2024). Evaluation of asphalt pavement texture using multiview stereo reconstruction based on deep learning. Construction and Building Materials, 412, 134837.
61. Wei, Y., Gu, X., Feng, Z., Li, Z., & Sun, M. (2024). Feature Extraction and Model Optimization of Deep Learning in Stock Market Prediction. Journal of Computer Technology and Software, 3(4).
62. Chen, W., You, Z., Li, R., Guan, Y., Qian, C., Zhao, C., Yang, C., Xie, R., Liu, Z., Sun, M. (2024). Internet of Agents: Weaving a Web of Heterogeneous Agents for Collaborative Intelligence. arXiv preprint arXiv:2407.07061.
63. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
64. Liu, W., Cheng, S., Qu, H. (2024). Enhancing Credit Card Fraud Detection Through Adaptive Model Optimization. Unpublished manuscript.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.