International Journal of Artificial Intelligence and EvolveTech
ISSN: 0000-0000
Volume 1, Issue 1 | PP 27-34 | www.ijaiet.com |

Deep Learning for Low-Resource Settings: A Survey of Model Compression and Transfer Learning Strategies

Dr. Nitin Saraswat

Assistant professor, Dept of Management Studies, Jagan Institute of Management studies, 3 Near Rithala Metro Station Rohini Sector 5, Institutional Area, New Delhi, 110085 INDIA

*Corresponding author
Assistant professor, Dept of Management Studies, Jagan Institute of Management studies, 3 Near Rithala Metro Station Rohini Sector 5, Institutional Area, New Delhi, 110085 INDIA , Email: nitin.saraswat@jimsindia.org

Received: 18-06-2026 | Revised : 13-08-2026 | Accepted : 25-08-2026 | Published : 31-08-2026

Abstract

The models are becoming larger, and the data sets are larger, leading to greater success in deep learning, but this approach is not compatible with many contexts that are resource-constrained. There are two scarcity types that are prevalent: Limited compute, memory, and energy on edge, mobile, and embedded devices, and limited labelled data in specialised domains where it is costly to annotate it. The two main strategy families that enable deep learning under these constraints are discussed: model compression (compressing the trained network to make it more economical), and transfer learning (reusing knowledge from data-rich source tasks to require relatively little target data). It presents the field through four lenses: a taxonomy of compression techniques: pruning, quantization, knowledge distillation, and compact architectural design, as well as the transfer-learning techniques covered: few-shot learning, domain adaptation, parameter-efficient adaptation, fine-tuning, and feature extraction, which tackle data scarcity issues, a deployment pipeline that fuses the two: (few-shot learning, etc. to overcome limited data and compression, etc. to overcome limited compute) prior to on-device deployment, and open challenges, such as accuracy-efficiency trade-offs, automation, robustness, and evaluation. It shows that compression and transfer are complementary, both for orthogonal axes of scarcity: the combination of a large, general-purpose, pre-trained model and its compression for the target device and task is the most prevalent for practical low-resource deployment. The design principle is always to align the strategy to the binding constraint either transfer learning when data is scarce, or compression when compute is scarce, or both when both data and compute become scarce just like they do more often than not.

Keywords: deep learning; model compression; transfer learning; knowledge distillation; quantization; pruning; edge computing; low-resource machine learning

1. Introduction

The performance of deep learning has been improving rapidly thanks to the scale: bigger models, trained with bigger datasets, using more computations, have consistently outperformed. This recipe, however, assumes that a lot of resources are available – something that is not often true in the real world. On the other hand, the model sizes of the state-of-the-art accuracy are too large, slow and power consuming to be deployed on edge, mobile and embedded devices where inference is desired [1]. The other is that the assumption of training is that large labelled datasets are not present in the clinical, industrial or in a low-resource-language domain where annotation is either too expensive or insufficient [2]. The divide between the compute and data requirements of deep learning and the resources available in constrained settings inspires a vast literature on making it efficient in both compute and data requirements.

Two different axes of "low-resource" are useful. The first is limited compute: restrictions on memory, latency, and energy which decide whether a trained model can be deployed at all on a particular device. The second is limited labelled data: restrictions on the number of examples with labels that can be used for training or specializing a model for a target task. These are orthogonal axes a project may have to deal with, both of which require a different type of solution. The most common solution to the first is model compression, that is, compressing the size and expense of a trained network while maintaining as much accuracy as possible. Transfer learning is the key remedy for the latter, where transferring representations learned in a very well-defined source task allows for efficient solution of the target task with substantially fewer examples [3].

They are both rich and well-surveyed, but are generally treated separately, hiding from sight that they solve complementary problems and are often applied in combination: a large pretrained model is first adapted to the target task with transfer learning, then compressed for the target device. A review dealing with compression and transfer in one goes, for the explicit purpose of the low-resource setting, will help practitioners to reason about which strategy, or combination, is appropriate to their binding constraint.

This survey offers this treatment. It answers four questions: (RQ1) What model-compression techniques are available and how can they be organised; (RQ2) What transfer-learning techniques can be used to combat the data scarcity issue; (RQ3) How are the two combined into a deployment pipeline and what trade-offs do they involve; and (RQ4) What challenges remain. It includes a taxonomy of compression techniques (Figure 1, Table 3), a structured overview of transfer-learning strategies (Table 4), a low-resource deployment pipeline that brings these two together (Figure 2), and a synthesis of open challenges and design guidance.

2. Review Approach

2.1 Method. This is a structured narrative survey that was carried out following a predetermined protocol which pre-determined the research questions, the literature sources, the inclusion criteria and the literature extraction and synthesis process prior to the screening. The nature of the primary literature makes a narrative approach instead of meta-analysis appropriate, as the outcomes reported are incommensurable across tasks, architectures and hardware.

2.2 Scope and eligibility. This survey includes model compression techniques and transfer learning and the combination of both methods, that make deep learning feasible in low compute and low labelled data settings. Any general purpose work that relies on accuracy and has no factor of efficiency or data scarcity is out of scope. All the criteria are contained in Table 1.

2.3 Information sources and search strategy. The five databases – IEEE Xplore, the ACM Digital Library, Scopus, Web of Science, and arXiv – were searched, as were citation chasing techniques from key surveys, for the pre-publication work. The search consisted of three concept blocks – deep learning, efficiency/compression, and transfer/data-scarcity – which were combined using different logical operators for every database; representative terms are shown in Table 2.

2.4 Selection and extraction. Records retrieved were de-duplicated, screened by title/abstract and then screened by full text against the eligibility criteria, and the number retained at each screening should be noted on the final manuscript. For every work that was retained, a template documented the strategy family, the technique, the constraint the technique solved and the reported impact on model size, speed or data efficiency, and the impact on accuracy.

2.5 Synthesis. Findings were organised thematically around the four research questions. The taxonomy of Figure 1 grouped various compression techniques, and Table 3 compared the different techniques. Table 4 compared the various transfer-learning strategies; the deployment pipeline of Figure 2 demonstrated how the two families worked together.

Table 1. *Scope and eligibility criteria for the survey.*

Dimension	Inclusion criteria	Exclusion criteria

Dimension	Inclusion criteria	Exclusion criteria
Focus	Model compression or transfer learning for deep networks	Accuracy-oriented work with no efficiency / data-scarcity aim
Constraint	Addresses limited compute/memory/energy or limited labelled data	No resource constraint considered
Contribution	Technique, system, empirical study, or survey	Non-technical opinion pieces
Type	Peer-reviewed papers and well-cited preprints	Non-archival abstracts and posters
Language	Published in English	Full text unavailable in English
Window	Modern deep-learning era (2010–present)	Pre-dates the relevant literature

Table 2. *Search concept blocks and representative terms; syntax was adapted to each database.*

Concept block	Representative search terms (OR-combined within block; blocks AND-combined)
A · Deep learning	"deep learning" OR "neural network" OR CNN OR transformer OR "deep model"
B · Efficiency / compression	"model compression" OR pruning OR quantization OR "knowledge distillation" OR "low-rank" OR "efficient architecture" OR "neural architecture search"
C · Transfer / data scarcity	"transfer learning" OR fine-tuning OR "domain adaptation" OR "few-shot" OR "parameter-efficient" OR "low-resource" OR pretrained

3. Findings

3.1 Two axes of low-resource deep learning. A key theme of this survey is that the term “low resource” refers to two distinct problems. If a model is constrained to a few megabytes of memory, has to respond in a few milliseconds, or requires a battery, then the focus is not to sacrifice accuracy in order to reduce the cost of the model, and that is the job of model compression. If, on the other hand, labelled data is a limitation in a target task that only has a few annotated examples the aim is to reduce the data a model has to process by importing knowledge from a different source, which is the field of transfer learning. The remedies to the two constraints were independent, and so are each discussed separately below and then how they work together.

As deep learning has followed two tracks, the importance of this framing has been increasing. Models on the edge are now far larger in size, making the gap between what's at the cutting-edge of performance and what can be deployed on a phone or microcontroller even larger; meanwhile, deployment has expanded to ever more specialised and constrained environments such as wearables, sensors, clinical devices and low-connectivity areas where compute power and labelled data are scarce. Consequently, optimisation for efficiency has gone from being a sideline for a few embedded applications to a mainstream requirement for realising modern deep learning, and compression and transfer are thus no longer just after-thought optimisation considerations, but central.

3.2 RQ1 A taxonomy of model compression. There are four families of compression techniques, as shown in Figure 1. The lottery-ticket hypothesis indicated that dense subnetworks can contain dense subnetworks that are sparse and can be trained to full accuracy, which is an indication of actual speed-up for structured pruning and speeds for unstructured pruning for high sparsity [4], [5]. After training (post-training quantization) or during training (quantization-aware training), quantization reduces numerical precision from 32-bit floats to 8-bit integers or lower, resulting in significant memory and energy savings and the ability to run quantization on commodity hardware directly using integer arithmetic [6], [7]. Knowledge distillation involves training a small model (the student) to emulate the behaviour or internal representations of a large model (the teacher) and transfer these into a compact representation [8], [9]. Efficient designs are achieved from the outset, using low rank factorization, efficient architectures designed by hand like depthwise-separable convolutions [10], [11] and efficient architectures discovered with neural architecture

International Journal of Artificial Intelligence and EvolveTech
ISSN: 0000-0000
Volume 1, Issue 1 | PP 27-34 | www.ijaiet.com |

search [12]. These techniques have been shown to compress the order of magnitude with a deep-compression work by Han et al. [13], which combines pruning, quantization and coding. The properties of their are compared in Table 3.

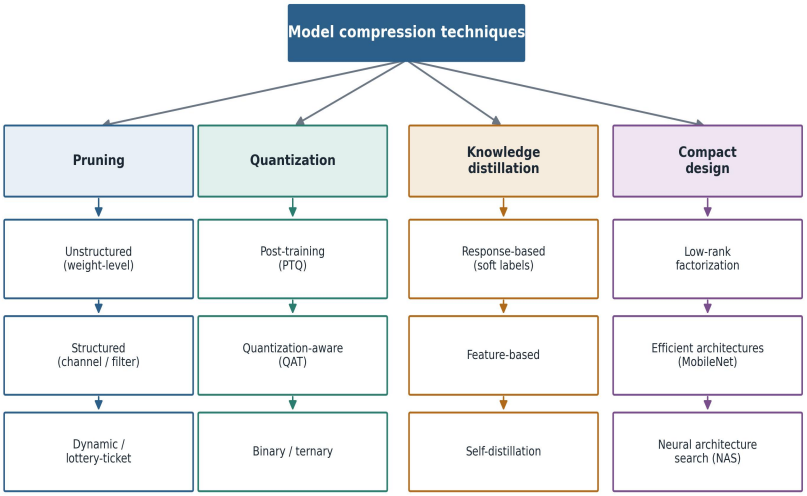

Figure 1. *A taxonomy of model compression techniques for low-resource deep learning.*

Table 3. *Comparison of model compression techniques on the dimensions relevant to low-resource deployment.*

Technique	Mechanism	Size reduction	Accuracy impact	Key consideration
Unstructured pruning	Zero individual weights	High (sparse)	Low if fine-tuned	Needs sparse-aware hardware for speedup
Structured pruning	Remove channels / filters	Moderate	Low–moderate	Direct speedup on standard hardware
Post-training quantization	Lower precision after training	High	Low–moderate	Fast; no retraining
Quantization-aware training	Learn under low precision	High	Low	Retraining required
Knowledge distillation	Train small student on teacher	Model-dependent	Low	Needs teacher and training data
Low-rank factorization	Factor weight matrices	Moderate	Low–moderate	Layer-dependent gains
Efficient architecture / NAS	Design compact models	By design	Varies	Search cost (NAS) can be high

3.3 RQ2 Transfer-learning strategies. In the case of limited data compared to compute, the knowledge of a source task, usually a large model trained on abundant data, is transferred to a target task with limited labels [2], [14]. As shown in Table 4, the strategies lie on a continuum between reusing and retraining the source model. Feature extraction involves fixing the weights of the pretrained network and only training a new task head, which is based on the assumption that the early layers of the network are general, transferable features and the later layers are task-specific [15]. The method is analogous to fine-tuning, which has become a standard practice for fine-tuning large pretrained models like BERT for various tasks in the natural-language domain [16]. Parameter-efficient methods update only a small number of added (or selected) parameter adapters and low rank updates while providing comparable accuracy and also significantly lower resource cost, compared to tuning every parameter of a large model

International Journal of Artificial Intelligence and EvolveTech
ISSN: 0000-0000
Volume 1, Issue 1 | PP 27-34 | www.ijaiet.com |

[17], [18]. Few-shot learning aims at the extreme scenario of limited target samples (e.g., only a few target examples) [19], [20] and domain adaptation focuses on the changes between the source and target domain. In particular, parameter-efficient fine-tuning also lightens the compute axis, previewing the combination that is developed later.

Table 4. *Comparison of transfer-learning strategies by target-data regime and reuse pattern.*

Strategy	Data regime	What is reused / updated	Trade-off
Feature extraction	Very limited target data	Frozen backbone; new head only	Cheapest; limited target fit
Full fine-tuning	Moderate target data	All weights updated	Strong fit; compute-heavy, overfitting risk
Parameter-efficient (adapters, LoRA)	Limited–moderate data	Small added / low-rank parameters	Near full-fine-tune accuracy at low cost
Domain adaptation	Shifted target distribution	Align source & target features	Handles distribution shift; added complexity
Few-/zero-shot	Handful or no examples	Prompting / metric learning	Broadest data saving; less reliable

3.4 RQ3 Combining strategies: a deployment pipeline. Real low-resource deployments typically involve compression and transfer at the same time, since they are orthogonal constraints, as shown in the Figure 2 below. It starts with a large model trained on lots of source data, then fine-tunes it using limited labelled data, generally with the transfer learning technique, and often in combination with other techniques like parameter-efficient adaptation; then, it compresses the adapted model to fit the target device's memory, latency, and energy budget, typically using model pruning, quantization, and distillation, frequently in a sequence, before deploying it on the target device. Ordering matters: adapting before compressing allows the specialised model to be shrunk into the device to the specific task and distillation in particular can transfer the adapted teacher's behaviour into a compact student. All the while, the same goals of accuracy retention, latency, memory footprint, and energy are used to guide trade-offs, with each technique trading one of the goals for another in the trade-off; the pipeline is therefore adapted to the binding constraint and is not applied in the same way across the board. The following are two practical points. Compression applied after adaptation can also be driven by the target task, by pruning the part of the network that is not used by that task, or by the native capacity of the deployment hardware, by quantizing to the precision native to the hardware, leading to a shrinking that is task- and device-specific instead of generic. Compression is generally iterated, as the gains from careful adaptation can be lost, and a small fine-tuning pass is performed after compression to aid in recovering the accuracy that was lost due to pruning and/or quantization.

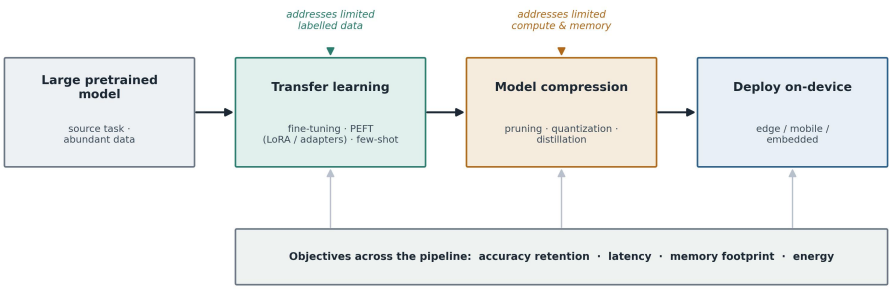

Figure 2. *A low-resource deployment pipeline combining transfer learning and model compression.*

3.5 RQ4 Open challenges. Four challenges recur. The accuracy–efficiency frontier: each compression or adaptation decision comes with a tradeoff between accuracy and cost, and defining the frontier of reliability above the difficultly

of reporting a single point of optimality. Even with the selection and configuration of techniques, expertise is required, which is why automated compression and automated neural architecture search [12] have emerged, but both require large compute during the search phase. Robustness and fairness: robust and fair compression and aggressive adaptation, which are not reflected in headline accuracy figures. Evaluation: Results are often presented using non-standard hardware and benchmarks, and sometimes using a weak baseline, making it difficult to compare across studies [5]. The major challenge between promising evaluation methods and robust low resource systems is to overcome these evaluation methods that are so standardised and hardware dependent.

4. Discussion

4.1 Principal findings. Three conclusions emerge. The first is that low-resource means two orthogonal constraints – compute and data – each with a remedy – and when they're confused, the wrong tool is applied. Second, model compression and transfer learning are not mutually exclusive; they can be used simultaneously in the same model with no conflict.Second, model compression and transfer learning are not competing; the former reduces the cost of a model, the latter reduces the data required by the model and they can be used together in the same system. Third, the most prevalent practice is precisely that – adapt a large pre-trained model to the target task and compress it for the target device, so that both families are best learned and utilized as a single pipeline and not as a choice of alternatives.

4.2 Convergence of the two families. There is a growing blurring in the boundaries between compression and transfer. Knowledge distillation is also a compression method, but can also be considered as a form of knowledge transfer as it takes a teacher's learnt behaviour into a student. Parameter-efficient fine-tuning, which is designed to make the transfer less expensive, lowers the compute and storage of adaptation at the same time as it does to reduce the number of parameters [17], [18]. The fact that this convergence implies that the best performing low resource approaches are those that do not choose between data and compute, but rather economies both, shows that computing and data economy are not mutually exclusive

.**4.3 Implications for practice.** A straightforward rule of thumb for practitioners: Know what is the limiting factor first and then choose a technique. In the event of limited labelled data but a capable deployment target, invest in the transfer learning part of the pipeline; in the event of abundant data but a constrained deployment target, invest in the compression part of the pipeline; and in the case of both, as often is the case in the edge domain on a targeted device in a specialised application, invest in the combined pipeline, adapting the pipeline first, and compressing the pipeline second, tailored to the budget of the device. Each technique gives up some accuracy for efficiency, so the operating point must be selected based on application-specific requirements, rather than based exclusively on the criterion of optimising just one value.

4.4 The foundation-model era. These worries have been exacerbated, not mitigated, by the advent of very large pretrained "foundation" models. On the data axis, these models transfer the default: adaptation of a general-purpose, pretrained model, often parameter-efficiently, is the standard approach to get to a specialised task, using little labelled data. On the compute side, they are so large that it is becoming increasingly important to compress them, and distilling smaller versions, “students”, is now a commonplace component of their deployment. As models grow, combined pipeline this survey describes adapt, then compress thus becomes even more central, with the former being parameter-efficient and the latter making lots of work to squeeze the frontier capability into the limited hardware.

4.5 Limitations of this survey. This review is a narrative synthesis, which implies that the text selected and arranged is based on the editorial judgment upon which a very large and rapidly developing literature was the source and that the search was limited to English-language publications. The taxonomy, comparison tables and pipeline are not intended to be a complete list of all primary results; instead, they should be viewed as analytical scaffolds to organize the field rather than as guarantees of the quality of the primary results, as these are reported under a variety of conditions. The constraints should be clearly documented in the final manuscript.

5. Conclusion

This survey has explored the two main strategy families that balance scale and limited compute or labelled data, but a reliance on scale is not always compatible with the many settings where compute and/or labelled data are limited. It classified model compression pruning, quantization, knowledge distillation and compact design into a taxonomy, identified the various strategies for transfer learning that tackle the issue of data scarcity (feature extraction, fine-tuning, parameter-efficient adaptation, domain adaptation and few-shot learning) and combined them into a deployment pipeline to adapt a large pretrained model before compressing it for the target device, and surveyed the challenges of the accuracy–efficiency frontier, automation, robustness, and evaluation. The over-arching principle is that compression and transfer are orthogonal, that they are most effective together, and that the appropriate strategy is to determine the binding constraint. The emphasis in deployment is moving towards devices that are constrained and data that is scarce, and methods that keep both computations and data usage minimal will be pivotal to true "low resource" deep learning.

Declarations

Conflict of Interest

There are no conflicts of interest related to this research to declare.

Funding Statement

No funding was specifically sought for this research from any public, commercial or not-for-profit funding agencies.

Ethical Approval

This study had no human or animal subjects and as such did not require ethical approval.

References

[1] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “Model compression and acceleration for deep neural networks: The principles, progress, and challenges,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018, doi: 10.1109/MSP.2017.2765695.

[2] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[3] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, Art. no. 9, 2016, doi: 10.1186/s40537-016-0043-6.

[4] J. Frankle and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” in *International Conference on Learning Representations*, 2019.

[5] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, “What is the state of neural network pruning?” *Proceedings of Machine Learning and Systems*, vol. 2, pp. 129–146, 2020.

[6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2018, doi: 10.1109/CVPR.2018.00286.

[7] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, “A survey of quantization methods for efficient neural network inference,” *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.13630

[8] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021, doi: 10.1007/s11263-021-01453-z.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[11] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97, pp. 6105–6114, 2019.

[12] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.

[13] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *International Conference on Learning Representations*, 2016.

[14] F. Zhuang, Z. Qi, K. Duan, X. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, pp. 3320–3328, Curran Associates, 2014.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. 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*, pp. 4171–4186, ACL, 2019, doi: 10.18653/v1/N19-1423.

[17] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97, pp. 2790–2799, 2019.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[19] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, Art. no. 63, 2020, doi: 10.1145/3386252.

[20] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, pp. 270–279, Springer, 2018, doi: 10.1007/978-3-030-01424-7_27.