Article

A Survey of Reinforcement Learning-Driven Knowledge Distillation: Techniques, Challenges, and Applications

Zeqiu Xu [*] , Jiani Wang , Xiaochuan Xu , Peiyang Yu , Tianyi Huang , Jingyuan Yi

Posted Date: 27 March 2025

doi: 10.20944/preprints202503.0903.v2

Keywords: reinforcement learning; knowledge distillation; teacher models; deep learning

Article

A Survey of Reinforcement Learning-Driven Knowledge Distillation: Techniques, Challenges, and Applications

Zeqiu Xu [1,*], Tianyi Huang [2], Jiani Wang [3], Jingyuan Yi [4], Xiaochuan Xu [4] and Peiyang Yu [4]

[1] Information Networking Institute, Carnegie Mellon University, Pittsburgh, USA
[2] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA; tianyihuang@berkeley.edu
[3] Department of Computer Science, Stanford University, Stanford, USA; jianiw@alumni.stanford.edu
[4] Information Networking Institute, Carnegie Mellon University, Pittsburgh, PA, USA; jingyuay@alumni.cmu.edu (J.Y.); xiaochux@alumni.cmu.edu (X.X.); peiyangy@alumni.cmu.edu (P.Y.)
* Correspondence: zeqiux@alumni.cmu.edu

Abstract: As deep learning models are widely applied across various domains, a critical challenge is how to compress models while maintaining high reasoning capability. Knowledge distillation, an effective technique for model compression, has been used to enhance the performance of lightweight models. However, traditional distillation methods are limited when dealing with complex reasoning tasks. Reinforcement learning (RL) offers a novel approach to knowledge distillation by optimizing the reasoning strategies of teacher models, generating more efficient decision paths, and providing more valuable learning content for student models. This paper reviews the latest advancements in combining reinforcement learning with knowledge distillation, focusing on policy distillation, value function distillation, and dynamic reward-guided distillation methods. It also discusses the challenges faced by RL-driven distillation, such as simplifying complex strategies, addressing temporal dependencies, and balancing exploration and exploitation, and suggests possible solutions. Finally, this paper explores the applications of RL-driven knowledge distillation in fields such as game AI, robotic control, and dialogue systems, and outlines future research directions, including automated distillation, multimodal distillation, and challenges in federated learning.

Keywords: reinforcement learning; knowledge distillation; teacher models; deep learning

1. Introduction

The rapid development of deep learning models has led to their widespread application across diverse fields, from image recognition to natural language processing (NLP) [1]. Applications of large language models (LLMs) have even extended into multimodal learning, where methodologies improve task performance by integrating different modalities like images, text, and audio [2]. While these models have achieved state-of-the-art performance, their large size and high computational requirements pose challenges in real-world deployment, particularly in resource-constrained environments such as mobile devices and embedded systems [3].

Knowledge distillation (KD) is an effective technique for model compression, where a smaller, lightweight student model is trained to mimic the behavior of a larger, more complex teacher model [4]. Traditional KD methods generally involve transferring the soft labels generated by the teacher model to the student model, which helps the student model learn from the teacher's generalization ability [5]. However, traditional KD methods often struggle to handle tasks that require complex reasoning, such as sequential decision-making and dynamic environments [6].

Reinforcement learning (RL) has emerged as a promising solution to these challenges [7]. By leveraging RL techniques, KD can be enhanced by optimizing the teacher model's decision-making strategies, which leads to the generation of more valuable learning signals for the student model [8]. RL-driven knowledge distillation introduces novel methods that focus on optimizing policies and value functions to improve student model performance [9].

This paper provides a comprehensive survey of RL-driven knowledge distillation, focusing on the following aspects:

- The main techniques and approaches used in RL-driven KD.
- The challenges and limitations of RL-driven KD.
- The applications of RL-driven KD in various domains.
- Future research directions and potential breakthroughs.

2. Background and Problem Definition

KD solves the problem of deep learning model compression and deployment by transferring knowledge from complex teacher models to lightweight student models [4]. The traditional KD method is based on the cross entropy loss function of soft labels:

$$\mathcal{L}_{\mathrm{KD}} = \alpha\mathcal{L}_{\mathrm{CE}}(\mathbf{y},\mathbf{s}) + \beta\mathcal{L}_{\mathrm{KL}}(\frac{\exp(\mathbf{z}_T/T)}{\sum_j \exp\,(\mathbf{z}_T/T)},\frac{\exp(\mathbf{z}_S/T)}{\sum_j \exp\,(\mathbf{z}_S/T)})$$

where $\mathbf{z}_T$ and $\mathbf{z}_S$ are the unnormalized logits of the teacher and the student, respectively, and T is the temperature parameter. However, traditional KD is limited in complex reasoning tasks (such as multi-step decision-making and long-range dependencies) because it is difficult to capture strategy optimization and value transfer in dynamic environments.

RL makes up for this shortcoming through strategy optimization and dynamic reward design [10]. RL-driven KD can optimize the teacher strategy, generate efficient decision paths (such as AlphaGo Zero's self-playing strategy), adjust the distillation focus based on environmental interactions (such as curiosity-driven intrinsic rewards), and avoid overfitting of students to the teacher strategy.

3. RL-Driven Knowledge Distillation Techniques

A. Policy Distillation

As shown in Table 1, this paper systematically compares core RL-driven knowledge distillation techniques, including their mechanisms, advantages, and applicable scenarios for policy distillation, value function distillation, and dynamic reward-guided distillation.

Table 1. Comparison of RL-Driven Knowledge Distillation Techniques.

Technique	Core Mechanism	Advantages	Limitations	Applicable Scenarios
Policy Distillation	Optimizes the policy network of the teacher model via RL to extract lightweight policies for student imitation.	Preserves complex decision logic; suitable for sequential tasks.	Low training efficiency with large policy spaces; relies on teacher quality.	Game AI, robotic path planning
Value Function Distillation	Transfers state-action evaluations from the teacher's value function to	Reduces exploration costs; improves stability.	Poor adaptability to dynamic environments;	Autonomous driving, resource scheduling

Technique	Core Mechanism	Advantages	Limitations	Applicable Scenarios
	guide student optimization for long-term rewards.		requires precise value estimation.	
Dynamic Reward-Guided Distillation	Designs dynamic reward functions to adjust distillation based on environmental feedback, balancing imitation and exploration.	Adapts to complex tasks; avoids overfitting.	Complex reward design; high training convergence difficulty.	Dialogue systems, multimodal interaction

The core of policy distillation is to transfer the teacher model's policy $\mathbf{z}_T$ to the student model $\mathbf{z}_S$ so that it generates similar action distributions under the same state. Traditional methods achieve this by imitating the teacher's action probability, while reinforcement learning (RL) further introduces dynamic policy optimization to improve the generalization ability of the student model.

Rusu et al. first applied policy distillation to Atari games, training the student model by directly transferring the original action probabilities of the teacher model (such as the Q value distribution of DQN) [11]. Its core loss function uses KL divergence to align the action distribution:

$$\mathcal{L}_\pi = \mathbb{E}_{s\sim\pi_T}\left[D_{KL}(\mathbf{z}_T(a \mid s) \parallel \mathbf{z}_S(a \mid s))\right]$$

This method compresses the number of student model parameters to 1/15 in games such as Pong, without significantly decreasing performance.

Wang et al. proposed a meta-strategy distillation framework based on MAML, which dynamically generates distillation targets through multi-task RL, enabling a single student model to adapt to different teacher strategies (such as multi-character control in StarCraft II) [12]. Experiments show that this method improves the average reward of the student model by 17% in cross-task scenarios. DeepSeek-R1 [13] uses the GRPO (Group Relative Policy Optimization) algorithm to replace the traditional Critic model by comparing rewards within the group, achieving 71.0% pass@1 in the mathematical reasoning task (AIME 2024), verifying the efficiency of RL-driven online distillation. Similarly, recent work on document-level event argument extraction has demonstrated the effectiveness of contextual pooling and role-based guidance in capturing meaningful relationships between key entities [14].

B. Value Function Distillation

Value function distillation aims to transfer the teacher model's ability to judge the state value to the student model. Its core is to achieve strategy optimization by aligning state value estimates.

Early work achieved knowledge transfer by minimizing the mean square error (MSE) between the teacher and student value functions, for example:

$$L_V = \mathbb{E}_{s\sim\pi_T}\left[(V_T(s) - V_S(s))^2\right]$$

where V_T and V_S are the state value functions of the teacher and student models respectively. In the Atari game experiment, this method can compress the parameters of the student model to 1/10 while retaining 90% of the teacher performance.

Wang et al. proposed Hierarchical Value Distillation (HVD), which decomposes the state value into global task value and local action value, and improves the performance of the student model in sparse reward scenarios through dual optimization objectives [15]. Experiments show that the success rate of HVD in robot navigation tasks is increased by 23%. DeepSeek-R1 uses Group Relative Value to dynamically adjust the value function by comparing the reward differences of samples in the same batch, achieving an accuracy of 71.2% in mathematical reasoning tasks, an improvement of 9% over traditional methods. Moreover, recent study shows LLMs encode concepts of varying complexities at different layers, known as Concept Depth, where simpler concepts are captured in shallow layers, while more abstract inferential tasks require deeper layers [16].

C. Dynamic Reward-Guided Distillation

By designing dynamic reward signals to optimize the distillation process, the problem that traditional static rewards cannot adapt to environmental changes is solved.

Some methods introduce random network distillation (RND) techniques to use the prediction error generated by the teacher model as an intrinsic reward [17], for example:

$$r_{\text{intrinsic}} = \| f_{\text{target}}(s) - f_{\text{predictor}}(s) \|^2$$

where f_{target} is a fixed random network and $f_{\text{predictor}}$ is a student prediction network. This method improves the exploration efficiency by 3 times in a sparse reward environment.

4. Challenges and Solutions

As summarized in Table 2, we highlight key challenges (e.g., simplifying complex policies, handling temporal dependencies) and corresponding solutions (e.g., hierarchical RL, attention mechanisms) for RL-driven knowledge distillation.

Table 2. Challenges and Solutions for RL-Driven KD.

Challenge	Specific Issues	Solutions
Simplifying Complex Policies	Overly intricate teacher strategies hinder student model compression.	Introduce hierarchical RL (HRL) to decompose policies into subtasks.
Handling Temporal Dependencies	Capturing long-term decision dependencies (e.g., dialogue context).	Integrate attention mechanisms or Transformer architectures.
Balancing Exploration & Exploitation	Students over-rely on teachers, limiting autonomous exploration.	Design hybrid rewards combining imitation (teacher) and environmental feedback.
Heterogeneous Model Compatibility	Structural mismatches (e.g., CNN→Transformer) impede knowledge transfer.	Use adapter layers or feature mapping networks to align representation spaces.
Training Efficiency & Stability	High complexity and slow convergence from combining RL and KD.	Apply offline RL pretraining with curriculum learning.

A. Capacity Mismatch

The model capacity gap refers to the difference in capacity (i.e., the number of parameters and complexity of the model) between the teacher model and the student model. Typically, the teacher model is a well-trained, high-capacity deep learning model that demonstrates high performance on a variety of tasks. The student model is typically a simplified version that aims to achieve similar performance by mimicking the behavior of the teacher model. This gap is an important challenge in model compression, especially in the fields of reinforcement learning and deep learning.

The challenges brought by the model capacity gap are mainly reflected in two aspects. First, the student model has a lower capacity and may not be able to capture all the knowledge of the teacher model. Therefore, even through distillation, the performance of the student model may not reach the level of the teacher model. Second, the student model may overfit or underfit during training because its parameter space is small and may not be able to fully fit the training data.

To solve this problem, researchers have proposed a variety of strategies. By allowing the student model to imitate the output probability distribution of the teacher model (such as by minimizing the KL divergence), the knowledge of the teacher model is transferred to the student model. The problem caused by the model capacity gap can be alleviated by introducing feature alignment of the intermediate layer during the distillation process. Another method is to use a more flexible distillation strategy so that the student model can adaptively select the focus of learning according to different tasks or states. This can more effectively transfer knowledge within the limited capacity of the student model. Through multi-task learning, the knowledge of multiple related tasks is passed to the student model together, and the correlation between different tasks can also improve the learning effect of the student model.

B. Temporal Dependency

Temporal dependency means that the current state and decision are not only affected by the current input, but also by the historical state and action. In reinforcement learning tasks, temporal dependency is a crucial factor because future decisions depend on past experience. For the model, how to deal with temporal dependency is directly related to its learning ability and prediction accuracy.

Temporal dependencies in reinforcement learning are manifested as relationships between state-action sequences, which can be very complex. For example, in the decision-making process, the current action may have a profound impact on the subsequent state of the environment. Therefore, the student model must not only learn the action selection in the current state, but also understand how to make the right decision based on the historical states and actions.

In order to solve the problem of temporal dependency, researchers have proposed a variety of methods, especially by introducing structures such as recursive neural networks (RNN) and long short-term memory networks (LSTM) to model temporal relationships. LSTM is an improvement on RNN, which can solve the gradient vanishing problem of traditional RNN when dealing with long-term dependencies. By introducing forget gates, input gates, and output gates, LSTM can maintain long-term memory of historical information during training. For example, in DUIP [18], where an LSTM captures sequential user interactions and generates dynamic prompts for LLM-based recommendations. Its update formula is:

$$f_t = \sigma(W_f\ [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i\ [h_{t-1}, x_t] + b_f)$$

$$C_{t'} = \tanh\ (W_C\ [h_{t-1}, x_t] + b_f)$$

$$C_t = f_t * C_{t-1} + i_t * C_t$$

$$o_t = \sigma(W_o\ [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh{(C_t)}$$

where f_t, i_t, o_t represent the activation functions of the forget gate, input gate and output gate respectively, C_t and h_t are the cell state and output state respectively, x_t is the input, W and b are the weight matrix and bias.

C. Reward Design

Reward design is a core issue in reinforcement learning. The design of the reward function directly affects the behavior and learning effect of the agent. The quality of the reward design not only determines the efficiency of the learning process, but also determines the final performance of the learning strategy. A suitable reward function can guide the agent to learn towards the target behavior.

In many tasks, it may be difficult for the agent to obtain immediate feedback, resulting in sparse reward signals. In this case, the learning process may be very slow or even unable to converge. In addition, if the reward function is not designed properly, it may cause the agent to learn a strategy that is inconsistent with the expected goal. Reward shaping is to guide the agent to learn the correct behavior by adjusting the reward function. Recent advancements in LLMs, such as LLaMA 3, have demonstrated strong capabilities in emotion identification, successfully distinguishing nuanced emotional tones in sentences while showing improved performance on shorter texts [19].

Recent advancements in fake news detection further highlight the critical role of reward design in adversarial scenarios. As demonstrated in LLM-powered detection frameworks [20] [21], dynamic reward mechanisms must simultaneously address evolving misinformation patterns while maintaining ethical constraints on model behavior. This dual requirement mirrors the exploration-exploitation dilemma in RL-driven distillation, where reward signals must balance imitation fidelity with student model autonomy.

In order to solve the problem of reward design, researchers have proposed multiple strategies. By adjusting and correcting the original reward function, the reward signal is made more dense or continuous, thereby accelerating the learning process. In order to prevent the agent from falling into a local optimal solution in some cases, researchers have proposed a variety of methods to balance the relationship between exploration and utilization, such as introducing noise or adopting a variable exploration strategy. Common ways of reward shaping are:

$$R(s_t, a_t) = R'(s_t, a_t) + \gamma \cdot \mathbb{E}_{s_{t+1}} \left[V(s_{t+1}) - V(s_t)\right]$$

where $R'(s_t, a_t)$ is the original reward, $V(s_t)$ is the value of state s_t, and γ is a discount factor that represents the impact of current rewards on future rewards.

5. Applications

As demonstrated in Table 3, we validate the practical impact of RL-driven distillation through case studies in domains like game AI and robotic control, while outlining future research directions.

Table 3. Applications and Case Studies of RL-Driven KD.

Domain	Case Study	Technique	Outcome	Future Directions
Game AI	Lightweight deployment of AlphaGo-style models	Policy Distillation + Monte Carlo Tree Search (MCTS)	90% fewer parameters; 5x faster inference.	Automated distillation frameworks, multi-agent collaboratio.

Domain	Case Study	Technique	Outcome	Future Directions
Robotic Control	Dynamic grasping for robotic arms	Dynamic Reward-Guided Distillation + Imitation Learning	20% higher success rate; adapts to unseen objects.	Sim-to-real transfer learning.
Dialogue Systems	Personalized dialogue model compression	Value Function Distillation + RL dialogue policies	75% lower memory usage; retains reply quality.	Multimodal distillation (text + speech + vision).
Healthcare	Lightweight medical imaging diagnosis models	Hierarchical Policy Distillation + Uncertainty-aware rewards	10x smaller model; 95% diagnostic accuracy.	Privacy-preserving distillation in federated learning.
Autonomous Driving	Real-time edge-side path planning	Value Function Distillation + Safety-constrained RL	<50ms planning delay; 40% lower accident rate.	Vehicle-road collaborative distillation.

A. Large Language Model Compression

As large language models (LLMs) have shown powerful capabilities in multiple natural language processing (NLP) tasks, the size of the models has gradually increased, resulting in a sharp increase in computational costs and storage overhead. This makes the actual deployment of large language models face many challenges, especially on resource-constrained devices. Recent studies have explored integrating LLMs with dynamic intent modeling to improve efficiency without compromising performance [22]. In addition, Retrieval-Augmented Generation (RAG) techniques, such as RAG-Instruct, have been introduced to enhance LLMs by incorporating external knowledge, improving zero-shot performance, and addressing task diversity limitations [23]. Therefore, model compression has become a key technology to improve the efficiency of large language models. Large language models, especially those based on the Transformer architecture, such as GPT-3, BERT and Llama3, have billions or even tens of billions of parameters. Although these models perform well in retaining domain-specific knowledge acquired during pretraining while maintaining efficiency, they also bring huge computational and memory burdens [24]. For practical applications, especially on edge devices or low-resource environments, it is often unrealistic to deploy large language models. Traditional large language models often require a large number of parameters to capture complex language information, resulting in very high computational and storage overheads.

Beyond knowledge distillation, architectural innovations in base models also contribute significantly to deployment efficiency. The hybrid Transformer model integrating Bayesian optimization and BiGRU layers [25] demonstrates how structural enhancements can achieve 99.73% fake news detection accuracy with rapid convergence within 10 training epochs. This aligns with RL-driven distillation objectives by showing that model compression and architectural optimization can be complementary approaches – where distillation preserves reasoning capabilities while architectural improvements enhance feature extraction efficiency.

Therefore, some compression methods for large language models have been proposed. Transfer the knowledge of the teacher model to a smaller student model. By letting the student model imitate the output of the teacher model, the student model can achieve performance close to that of the teacher model at a smaller capacity. Usually, the student model is trained by minimizing the KL divergence between the outputs of the teacher model and the student model. In addition, by removing some unimportant connections or neurons in the model, the size of the model can be reduced. Pruning technology usually needs to be performed after the model training is completed. By analyzing the importance of each connection, those parts that have little impact on the model performance are removed.

B. Autonomous Driving

Autonomous driving technology is an important research direction in the field of artificial intelligence in recent years. Its purpose is to enable cars to make autonomous decisions and control through artificial intelligence algorithms, thereby realizing unmanned driving. Autonomous driving vehicles not only rely on perception and control technology, but also require efficient decision-making systems to make reasonable responses according to environmental changes.

However, autonomous driving vehicles must accurately perceive the surrounding environment, including road conditions, pedestrians, other vehicles, traffic signals, etc. Environmental perception requires the use of a variety of sensors, such as lidar, cameras, ultrasonic sensors, etc., to generate high-precision environmental models. Autonomous driving systems not only need to make decisions, but also need to convert decisions into specific control instructions to control the acceleration and steering of the vehicle to ensure that the vehicle drives smoothly and safely.

In reinforcement learning, the goal of the autonomous driving system is to learn the optimal driving strategy by maximizing the cumulative reward. The Q-learning method is usually used, and the Q value update formula is:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\,[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

whre s_t is the state, a_t is the action, r_t is the immediate reward, γ is the discount factor, and α is the learning rate.

C. DeepSeek

The core idea of DeepSeek is to use distillation technology to extract knowledge from a large-scale teacher model, and by optimizing the training process, the student model can retain the performance of the teacher model while greatly reducing the number of parameters. This compression method usually ensures that the student model can perform similarly to the teacher model on multiple tasks through multiple rounds of training and fine parameter adjustment.

DeepSeek is particularly suitable for deploying deep learning models on resource-constrained devices, and can effectively reduce the size and computational complexity of the model. In addition, mobile devices have limited computing power and storage space. DeepSeek compresses the model to enable large-scale models to run on these devices. The distillation loss function of DeepSeek is:

$$L_{\text{DeepSeek}} = \alpha L_{\text{ce}}(y, \hat{y}) + (1-\alpha) L_{\text{KD}}(y, \hat{y}_T)$$

where L_{ce} is the cross entropy loss, L_{KD} is the knowledge distillation loss, $\hat{y}_T$ is the output of the teacher model, and α is a hyperparameter that adjusts the balance between teacher and student knowledge.

6. Conclusions

This paper reviews the latest progress in combining reinforcement learning with knowledge distillation, focusing on methods such as policy distillation, value function distillation, and dynamic reward-guided distillation. In addition, this paper discusses the challenges faced by reinforcement learning-driven distillation methods, such as simplifying complex policies, handling temporal dependencies, and balancing exploration and exploitation, and proposes possible solutions. Finally, this paper explores the application of reinforcement learning-driven knowledge distillation in areas such as game AI, robot control, and dialogue systems.

doi:10.20944/preprints202503.0903.v2

References

1. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. https://doi.org/10.1038/nature14539
2. L. Herrmann and S. Kollmannsberger, “Deep learning in computational mechanics: A Review,” *Computational Mechanics*, vol. 74, no. 2, pp. 281–331, Jan. 2024. https://doi.org/10.1007/s00466-023-02434-4
3. P. Yu, X. Xu, and J. Wang, “Applications of Large Language Models in Multimodal Learning”, Journal of Computer Technology &amp; Applied Mathematics, vol. 1, no. 4, pp. 108–116, Nov. 2024.
4. S. Sun, W. Ren, J. Li, R. Wang, and X. Cao, “Logit standardization in knowledge distillation,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15731–15740, Jun. 2024. https://doi.org/10.1109/cvpr52733.2024.01489
5. S. Muralidharan *et al.*, “Compact Language Models via Pruning and Knowledge Distillation,” arXiv preprint arXiv:2407.14679, 2024
6. H. Liu, Y. Wang, H. Liu, F. Sun, and A. Yao, “Small scale data-free knowledge distillation,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6008–6016, Jun. 2024. https://doi.org/10.1109/cvpr52733.2024.00574
7. C. Tang *et al.*, “Deep Reinforcement Learning for Robotics: A survey of real-world successes,” *Annual Review of Control, Robotics, and Autonomous Systems*, Nov. 2024. https://doi.org/10.1146/annurev-control-030323-022510
8. Q. Li, W. Xia, L. Yin, J. Jin, and Y. Yu, “Privileged knowledge state distillation for reinforcement learning-based educational path recommendation,” *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1621–1630, Aug. 2024. https://doi.org/10.1145/3637528.3671872
9. H. Xiao, L. Fu, C. Shang, X. Bao, and X. Xu, “A knowledge distillation compression algorithm for ship speed and energy coordinated optimal scheduling model based on Deep Reinforcement Learning,” *IEEE Transactions on Transportation Electrification*, vol. 11, no. 1, pp. 945–960, Feb. 2025. https://doi.org/10.1109/tte.2024.3398991
10. D. Huang *et al.*, “Alignsam: Aligning segment anything model to open context via reinforcement learning,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3205–3215, Jun. 2024. https://doi.org/10.1109/cvpr52733.2024.00309
11. A. A. Rusu *et al.*, “Policy distillation,” arXiv preprint arXiv:1511. 06295, 2015
12. Z. Wang, B. Yang, H. Yue, and Z. Ma, “Fine-grained prototypes distillation for few-shot object detection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, pp. 5859–5866, Mar. 2024. https://doi.org/10.1609/aaai.v38i6.28399
13. D. Guo *et al*, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” arXiv preprint arXiv:2501.12948, 2025.
14. W. Liu, S. Cheng, D. Zeng, and Q. Hong, “Enhancing document-level event argument extraction with contextual clues and role relevance,” *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. https://doi.org/10.18653/v1/2023.findings-acl.817
15. J. Jiang, Z. Wang, S. Qiu, X. Li, and C. Zhang, “Multi-Task Load Identification and Signal Denoising Via Hierarchical Knowledge Distillation”, IEEE Transactions on Network Science and Engineering, pp. 1–14, 2025.
16. M. Jin *et al.*, “Exploring Concept Depth: How Large Language Models Acquire Knowledge and Concept at Different Layers?”, in Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 558–573.

17. K. Yang, J. Tao, J. Lyu, and X. Liu, "Exploration and anti-exploration with distributional random network distillation," arXiv preprint arXiv:2401.09750, 2024.

18. X. Xu, Z. Xu, Y. Pei, and J. Wang, "Enhancing User Intent for Recommendation Systems via Large Language Models," arXiv preprint arXiv:2501.10871, 2025.

19. Z. Mai, J. Zhang, Z. Xu, and Z. Xiao, "Is Llama 3 good at sarcasm detection? A comprehensive study," *Proceedings of the 2024 7th International Conference on Machine Learning and Machine Intelligence (MLMI)*, pp. 141–145, Aug. 2024. https://doi.org/10.1145/3696271.3696294

20. J. Yi, Z. Xu, T. Huang, and P. Yu, "Challenges and Innovations in LLM-Powered Fake News Detection: A Synthesis of Approaches and Future Directions," arXiv preprint arXiv:2502.00339, 2025

21. T. Huang, J. Yi, P. Yu, X. Xu, "Unmasking Digital Falsehoods: A Comparative Analysis of LLM-Based Misinformation Detection Strategies," arXiv preprint arXiv:2503.00724, 2025

22. X. Huang, Y. Wu, D. Zhang, J. Hu, and Y. Long, "Improving academic skills assessment with NLP and ensemble learning," *2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 37–41, Sep. 2024. https://doi.org/10.1109/iciscae62304.2024.10761701

23. W. Liu, J. Chen, K. Ji, L. Zhou, W. Chen, and B. Wang, "RAG-Instruct: Boosting LLMs with Diverse Retrieval-Augmented Instructions," arXiv preprint arXiv:2501.00353, 2024

24. Y. Wu, Z. Xiao, J. Zhang, Z. Mai, and Z. Xu, "Can llama 3 understand monetary policy?," *2024 17th International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 145–149, Sep. 2024. https://doi.org/10.1109/icacte62428.2024.10871796

25. T. Huang, Z. Xu, P. Yu, J. Yi, and X. Xu, "A Hybrid Transformer Model for Fake News Detection: Leveraging Bayesian Optimization and Bidirectional Recurrent Unit," arXiv preprint arXiv:2502.09097, 2025

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.