Reinforced Self-Supervised Training for Few-Shot Learning

Zhichao Yan, Yuexuan An, *Graduate Student Member, IEEE*, and Hui Xue, *Member, IEEE*

***Abstract*—Few-shot learning is an open problem to learning a new concept with little supervision from limited labeled data. As an alternative knowledge for few-shot learning, self-supervised learning can extract supervisory signals directly from unlabeled data. However, existing self-supervised few-shot methods which directly take the summation of two tasks, have two fundamental bottlenecks: 1) representation bias: how to extract efficacious supervisory signals in self-supervision and eliminate the disturbance of undesirable shortcuts with limited examples and 2) objective conflict: how to adaptively trade-off the self-supervision and supervision to achieve the optimal model performance. To address the above problems, in this paper, we propose a novel approach named ReInforced SElf-supervised training (RISE) for few-shot learning. RISE leverages agent-relative supervision to eliminate the undesirable shortcut learning of self-supervised training. Meanwhile, it dynamically explores the balance between supervisory signals from self-supervised tasks and inherent supervision from few-shot tasks to avoid the trade-off dilemma. Therefore, the new pattern for training self-supervision can be resilient to few-shot learning and enhance the performance for few-shot identification. Extensive experiments on several public benchmark datasets verify the effectiveness of our approach.**

***Index Terms*—Few-shot learning, reinforcement learning, self-supervised learning.**

I. Introduction

THE data-driven deep learning has achieved significant success with large-scale labeled datasets [17], [21]. However, large-scale labeled datasets can not be guaranteed in many areas such as medical image analysis [22], drug discovery [12], remote sensing [13], and robotics [16]. Therefore, few-shot learning comes into being, which aims to mimic humans to rapidly learn new concepts with little supervision from limited labeled data [2], [42], [43], [44], and to solve the problem of unavailability of access to large datasets [3], [20], [38]. However, training models with sparsely labeled data can easily lead to overfitting, severely degrading the generalization of the model [23]. On the contrary, self-supervised learning constructs proxy tasks from unlabeled data, without human annotation [14], [15], to extract supervisory signals [24], [25]. Meanwhile, in image processing, image-level supervised information usually simplifies the rich visual information contained in an image, while self-supervised learning can provide richer learning signals [18]. Therefore, supplementing the traditional few-shot main task with a self-supervised auxiliary task motivates scholars to enhance the generalization of models [26], [40]. Gidaris et al. [40] first proposed a multi-task learning paradigm by combining few-shot learning with self-supervised learning. Rong et al. [27] proposed a novel self-supervised Episodic Spatial Pretext Task (ESPT) to capture the local spatial features of different images and their inter-relational structural information in each input episode. Lim et al. [11] proposed a Self-supervised Contrastive Learning (SCL) that enriched the model representation with multiple self-supervision objectives for few-shot classification. An et al. [28] proposed conditional self-supervised learning (CSS) which equally treated self-supervised and supervised learning to fully extract semantic information contained in the dataset.

However, the methods mentioned above simply merge two tasks, which have two fundamental bottlenecks: 1) representation bias. The training of self-supervised learning typically requires a significant amount of data. Once under circumstance of constrained data, traditional self-supervised learning might learn an undesirable shortcut, which will raise the risk of learning a biased representation. 2) objective conflict. There is usually an ill-posed problem between self-supervised learning and the few-shot learning objective. Therefore, an adaptive trade-off between the supervisory signals and supervision should be adopted to achieve the optimal model performance.

We propose ReInforced SElf-supervised training (RISE) to tackle the issues in few-shot learning. RISE leverages inherent supervision as a guide to mitigate biased representation risks caused by data limitations. It incorporates reinforcement learning in self-supervision, dynamically balancing the two to avoid conflicts. RISE employs two action networks for policy updates to utilize limited trajectory data. Importance sampling assesses policy differences, reducing distributional disparities. Additionally, the objective function is pruned to correct policy ratios. RISE's agent adapts a balanced policy for supervised and self-supervised tasks in a continuous action space.

The contributions of our work are:

- We propose a reinforced conditional self-supervised learning approach. The approach can dynamically guide self-supervised learning to eliminate the undesirable shortcut

Manuscript received 26 October 2023; revised 7 February 2024; accepted 11 February 2024. Date of publication 27 February 2024; date of current version 12 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62076062, in part by the Social development Science and Technology Project of Jiangsu Province under Grant BE2022811, and in part by the Big Data Computing Center of Southeast University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xuesong Wang. *(Corresponding author: Hui Xue.)*

The authors are the with School of Computer Science and Engineering, Southeast University, Nanjing 211189, China, and also with the Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing 210096, China (e-mail: zhichao_yan@seu.edu.cn; yx_an@seu.edu.cn; hxue@seu.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LSP.2024.3370488, provided by the authors.

Digital Object Identifier 10.1109/LSP.2024.3370488

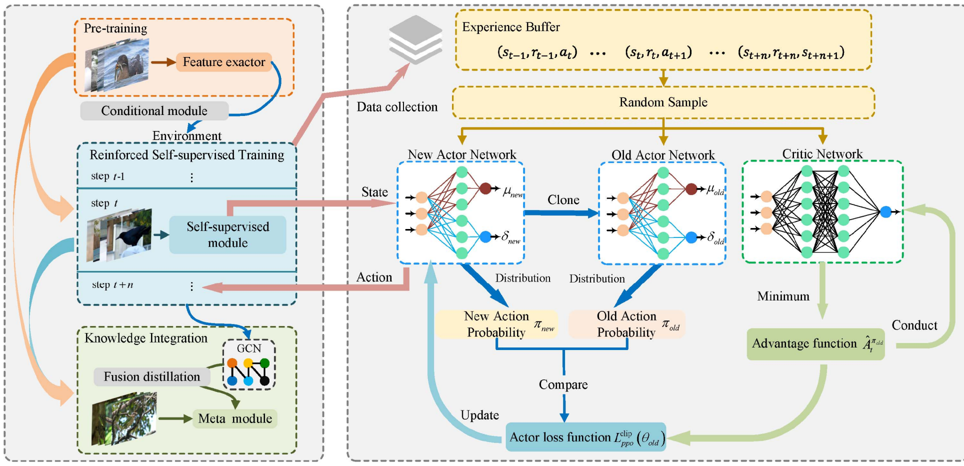


Fig. 1. Architecture of reinforced self-supervised training for few-shot learning.

of limited examples and the trade-off dilemma between self-supervision and supervision.

- We design a data-limited self-supervised environment to update the reinforcement learning paradigm by acquiring Markovian decision sequences through the agent. The agent of reinforcement learning is introduced to guide the trade-off between supervisory signals from self-supervised training and inherent supervision from pre-training.
- We systematically experiment with self-supervised few-shot learning on four benchmarks datasets, demonstrating that our method enhances model performance.

II. Method

A. Preliminary

Generally, the dataset of typical few-shot learning task based on meta-learning is described as $\mathcal{D} = (\mathcal{D}_{trn}, \mathcal{D}_{tst})$, corresponding to the meta train dataset and test dataset, respectively. Note that $\mathcal{D}_{trn}$ has a disjoint label space with $\mathcal{D}_{tst}$. Meanwhile, $\mathcal{D}_{trn} = \{\mathcal{T}_i\}_{i=1}^{N_{trn}}, \mathcal{D}_{tst} = \{\mathcal{T}_i^*\}_{i=1}^{N_{tst}}$, where $\mathcal{T}$ denotes the task. Each task is randomly sampled episodes from a base class set, which consists of a support set and a query set. In meta train phase, the dataset $\mathcal{D}_{trn} = \{\mathcal{T}_i\}_{i=1}^{N_{trn}} = \{\mathcal{S}_i, \mathcal{Q}_i\}_{i=1}^{N_{trn}}$. The support set $\mathcal{S}_i = \{(x_i, y_i) \mid i = 1,\ldots, N \times K\}$ is created by randomly selecting N classes from the training dataset $\mathcal{D}_{trn}$ and extracting K samples from each of the selected classes, resulting in a total of $N \times K$ samples in the support set. The query set $\mathcal{Q}_i = \{(x_i, y_i) \mid i = 1,\ldots, N \times M\}$ consists of N classes and M samples per class, where none of the M samples in each class are present in the support set $\mathcal{S}_i = \{(x_i, y_i) \mid i = 1,\ldots, N \times K\}$. In the testing phase, the meta test dataset $\mathcal{D}_{tst}$ with the same settings [1].

B. Overview

The overview of RISE framework is depicted in Fig. 1. Our RISE consists of three stages:

1) Pre-training stage: the inherent supervisory is extracted in this stage to guide the reinforced self-supervised training.
2) Reinforced self-supervised training stage: the agent of RL is introduced to guide the trade-off between supervisory signals from self-supervised training and inherent supervision from pre-training.
3) Knowledge integration stage: the knowledge extracted for the above two stages is integrated into comprehensive knowledge with graph convolution network (GCN) [29].

It is worth noting that our proposed method does not introduce an external dataset for model training.

C. The Training Process of RISE

Traditional self-supervised few-shot learning methods often suffer from biased feature representations due to the limited training data, and the complex and unknown relationships between objectives can lead to conflicts during model training. CSS relies on manual experience to optimize these relationships and cannot find a suitable optimization balance. In contrast, our RISE method utilizes adaptive agent-relative supervision to guide self-supervised training and reduce the impact of biased representations. The agent in RISE interacts with the reinforced self-supervision environment to dynamically adjust the balance between supervision and self-supervision, making the optimal equilibrium point less dependent on actual experience. This agent-relative supervision approach is more resilient to self-supervision and effectively improves the model's generalization performance.

1) Pre-Training Stage: The pre-training process of RISE adopts the supervised few-shot task to extract inherent supervisory signals effectively, which are used as a prerequisite for reinforced self-supervised training. In this stage, a good feature extractor $f(\cdot)$ is acquired, which is trained by minimizing the distance between samples and prototypes. Other methods are equally acceptable to achieve the supervised feature extractor.

2) Reinforced Self-Supervised Training Stage: The reinforced self-supervised training aims to learn good self-supervised feature extractor $g(\cdot)$ with the guide of pre-training feature extractor $f(\cdot)$ without a contradiction of two different objectives. The problem settings for RISE are shown below:

State: This research designs a module to adjust the balance of self-supervised and supervised dynamically when training the reinforced self-supervised stage. As shown in Fig. 1, the training of the self-supervised backbone is modeled as the environment interacting with a reinforcement learning component. As the loss function value has a strong relationship with the update of the feature network, the state is defined as a loss to ensure the agent satisfies the Markov condition at times step [4], [39]. The state function is proposed as:

$$s_t = \left\{ s_t = L_{RISE}^{train,\mathrm{t}} \right\} \tag{1}$$

where $t \in \{0, 1, 2, \ldots, T\}$. T is the episode of the self-supervised module. $L_{RISE}^{train,\mathrm{t}}$ denotes the loss function L_{RISE} obtained on the training set.

Action: The loss function of reinforced self-supervised training stage can be proposed as:

$$L_{RISE}(x) = \mathop{\mathbb{E}}_{x \in \mathcal{D}} [L_{self}(x) + \gamma_{RISE} L_{cond}(x)] \tag{2}$$

x is an input image, γ_{RISE} is the action of agent. The pre-train acquired feature extractor is $f(\cdot)$. The second stage learned is $g(\cdot)$. L_{self} is obtained by an arbitrary self-supervised method. L_{cond} adopts the minimization of the negative cosine similarity between $f(\cdot)$ and $g(\cdot)$. Besides γ_{RISE} is a coefficient of the self-supervised and the conditional terms of the loss.

Thereby, the γ_{RISE} is computed at episode t for balancing self-supervised and conditional terms of the loss as the action a_t which is drawn from the distribution:

$$\pi(a_t|s_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a_t - \mu)^2}{2\sigma^2} \right) \tag{3}$$

where μ is the mean, σ is the standard deviation. As illustrated in Fig. 1, given the state s_t, the agent can decide which coefficient along the action space should be chosen.

Reward: After agent executes action a_t over continuous action space, the environment based on self-supervised module can return the reward r_t of the evaluation a_t. In our method, we expect the self-supervised module can effectively fuse the self-supervised representation with condition information. Since the accuracy provides the most intuitive indication of the effect of different actions on model classification. Thereby, the reward r_t is defined as the accuracy $acc_{val,t}$ of the module validation process. The reward function is proposed as:

$$r_t = \{r_t = acc_{val,t}\} \tag{4}$$

Proximal Policy Optimization (PPO) is an on-policy RL algorithm for tasks with time-varying dynamics in discrete or continuous workspace [5], which belongs to Actor-Critic framework [6]. As shown in Fig 1. The parameters of the actor network are defined as θ and the corresponding policy is π_θ. Among the process of learning, contains two value functions in Actor-Critic framework: action-value function Q^{π_θ} and state-value function V^{π_θ} [9], [30]. The mathematical expressions are defined as follows respectively:

$$V^{\pi_\theta}(s) = E\left[\sum_{t=1}^{\infty} \kappa_t r_t \mid s_1 = s, \pi_\theta \right] \tag{5}$$

$$Q^{\pi_\theta}(s, a) = E\left[\sum_{t=1}^{\infty} \kappa_t r_t \mid s_1 = s, a_1 = a, \pi_\theta \right] \tag{6}$$

where κ_t is the discount factor of t step.

The goal of RL is to maximize the cumulative reward interacting with the environment [7].

$$\underset{\pi_\theta}{\operatorname{argmax}} \sum_{t=1}^{N} E[\kappa^{t-1} r_t] = E_{s \sim \rho_\pi, a \sim \pi_\theta}[Q^{\pi_\theta}(s, a)] \tag{7}$$

where ρ_π is the distribution function of state generated by π_θ.

As the agent can adjust the balance of self-supervised representation and condition information. The training process of RISE is defined as PPO. In the training of RISE, the trajectory $\tau = \{s_1, a_1, \ldots, s_T, a_T\}$ is stored in experience replay buffer [10]. During the actor-critic network's strategy update for the agent, PPO randomly chooses trajectories as input. To effectively use the data, importance sampling [31] is introduced. Meanwhile, PPO contains a clipped surrogate clip$(\cdot)$. Thereby, the final optimization objective function of PPO based on policy gradient (PG) method [30] is:

$$\begin{aligned} L_{ppo}^{\text{clip}}(\theta_{old}) = E_{(s_t, a_t) - \pi_\theta} \Bigg[\min \Bigg(\frac{\pi_{\theta_{new}}(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}, \\ \text{clip}\left(\frac{\pi_{\theta_{new}}(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t))}, 1 - \epsilon, 1 + \epsilon \right) \Bigg) \hat{A}_t^{\pi_{\theta_{old}}}(s_t, a_t) \Bigg] \end{aligned} \tag{8}$$

where ϵ is a hyperparameter. $\hat{A}_t^{\pi_{\theta_{old}}} = Q^{\pi_{\theta_{old}}}(s_t, a_t) - V^{\pi_{\theta_{old}}}(s_t)$ is the advantage function at timestep t. After that, the balance between supervision and self-supervised learning can be adjusted by PPO to obtain the optimal reinforced self-supervised model.

3) Knowledge Integration Stage: After completing the above two stages of training, we are able to obtain the embedding vectors $f(\cdot)$ and $g(\cdot)$ which is used for augmented representation with GCN [28], [29]. Subsequently, for a support set S_i and a query sample x_q, the classification score for class k is calculated based on a similarity measure between the query sample's augmented representation and the prototype, which is defined as the centroid of class k's support set S_k. For an x, the final loss of knowledge integration stage is given by:

$$\begin{aligned} L_{KI} = \mathop{\mathbb{E}}_{(x_q, y_q) \in Q_i} [&- \log p^*(y_q \mid x_q, S_i) \\ &- \eta \log p(y_q \mid x_q, S_i)] \end{aligned} \tag{9}$$

where η is a trade-off constant. $p^*(y_q \mid x_q, S_i)$ calculates the probability between a classification score of x_q and corresponding label y_q with augmented representation. $p(y_q \mid x_q, S_i)$ directly calculates that without augmented representation.

4) Inference Stage: During the inference stage, RISE employs a feature embedding network in the knowledge integration stage to extract features from original images. Concomitantly, RISE averages the embeddings of the support samples associated with each class as a representative prototype. Utilizing a similarity metric to measure the distance between the query sample and the prototypes of each class, RISE deduces the class with the highest similarity as the predicted classification output.

The time complexity of RISE is: $\mathcal{O}((N^2F^2K)(B_1E_1 + B_2E_2 + B_3E_3) + (MB_{RL}B_2E_2))$, where N and F denote the sizes of the input image and convolution kernel respectively, K and M are the numbers of output channels and a hidden layer dimension respectively, B_1, B_2, and B_3 are the batch sizes of the three stages, B_{RL} is the batch size of PPO, and E_1, E_2 and E_3 are the epoch of the three stages.

TABLE I
CLASSIFICATION ACCURACIES WITH 95% CONFIDENCE INTERVALS OF COMPARED ALGORITHMS ON 5-WAY FEW-SHOT TASKS. IN ADDITION, •/○ INDICATES WHETHER RISE IS STATISTICALLY SUPERIOR/INFERIOR TO COMPARED ALGORITHMS ON EACH DATASET (PAIRWISE T-TEST AT 0.05 SIGNIFICANCE LEVEL)

Method	CIFAR-FS		CUB-200		*mini*ImageNet		*tiered*ImageNet	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
FEAT[37]	44.13±0.87 •	65.43±0.79 •	50.58±0.89 •	75.78±0.68 •	43.49±0.82 •	57.65±0.69 •	39.68±0.85 •	52.27±0.79 •
ProtNet[19]	43.65±0.86 •	68.78±0.77 •	51.47±0.88 •	74.68±0.70 •	44.42±0.75 •	64.69±0.69 •	40.47±0.80 •	61.50±0.74 •
RelationNet[42]	46.76±0.86 •	65.01±0.79 •	62.14±0.94 •	77.45±0.64 •	48.11±0.82 •	63.96±0.69 •	44.82±0.81 •	60.35±0.75 •
CC+rot[40]	55.16±0.84 •	70.77±0.72 •	62.43±0.89 •	79.04±0.62 •	47.87±0.76 •	64.56±0.68 •	46.58±0.80 •	64.51±0.72 •
ProtNet+rot[26]	51.54±0.91 •	73.64±0.72 •	54.67±0.89 •	78.30±0.62 •	48.55±0.81 •	66.81±0.67 •	42.51±0.80 •	64.31±0.74 •
SLA[41]	52.22±0.94 •	72.09±0.75 •	52.99±0.87 •	72.56±0.69 •	45.23±0.79 •	64.71±0.67 •	42.26±0.83 •	60.21±0.82 •
ESPT[27]	52.18±0.23 •	70.49±0.18 •	62.62±0.23 •	79.42±0.17 •	45.91±0.17 •	66.18±0.17 •	42.58±0.21 •	61.29±0.19 •
CSS[28]	56.49±0.93 •	74.59±0.72 •	66.01±0.90 •	81.84±0.59 •	50.85±0.84 •	68.08±0.73 •	48.22±0.87 •	67.79±0.79 •
RISE (*ours*)	**60.05±0.96**	**78.44±0.64**	**67.42±0.96**	**82.86±0.59**	**53.22±0.81**	**69.41±0.67**	**51.73±0.90**	**69.65±0.74**

III. EXPERIMENTS

A. Datasets

The proposed RISE is evaluated on the four widely adopted few-shot classification datasets (CIFAR-FS, CUB-200, *mini*ImageNet and *tiered*ImageNet [8], [32], [33], [34]) to verify that the feasibility of our approach.

B. Implemention Details

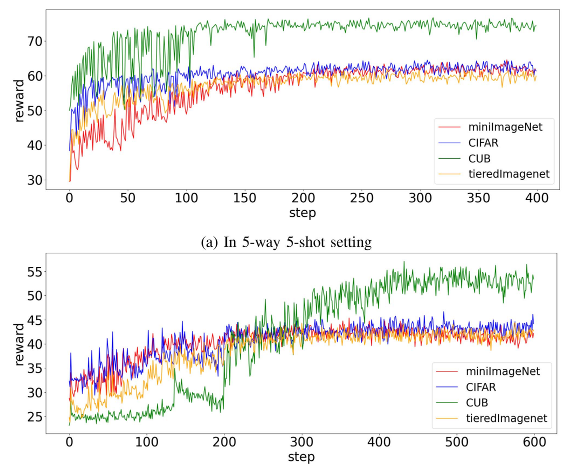
For a fair comparison with the proposed methods, we employ a Conv-4 backbone structure for feature extractor whose embedding dimension is 1600. In the reinforced self-supervised training, the settings are given below: AdamOptimizer [35] is applied, and the batch size is 100. The embedding dimension of fully connected MLP is 256 [36] in reinforcement module, containing two networks: actor network and critic network, where the ReLU function is used in the hidden layer of actor network to output the positive-terrestrial distribution. Meanwhile, the hidden layer in critic network uses Leaky ReLU function and ReLU function to output the value under the corresponding action of the agent. In the knowledge integration stage, we train 1200 epochs for 5-way 5-shot and 1800 epochs for 5-way 1-shot.

C. Results and Comparisons

To verify the effectiveness of RISE, we compare our method with several state-of-the-art methods including FEAT, ProtNet, and RelationNet [19], [37], [42], all of which are typical few-shot learning methods. Meanwhile, CC+rot, SLA, ProtoNet+rot, ESPT, and CSS as few-shot methods combined with self-supervised learning will also be compared.

1) Comparison With SOTA Methods: From the results in Table I, we can observe that the performance of the proposed RISE is superior to the compared SOTA methods. For instance, under the 5-way 5-shot setting, our method averagely improves by 3.85%, 1.02% and 1.33%, respectively, compared to CSS, while it has 3.56%, 1.41% and 2.73% performance improvements under the 5-way 1-shot setting, which means the agent can learn properly supervised information to guide the process of training self-supervised stage.

2) Convergence of RISE: The convergence curves of the reward achieved by RISE on 5-way 5-shot and 5-way 1-shot are shown in Fig. 2(a) and (b). We observe that RISE with the pre-designed reward exhibits good convergence. With the training process, the accuracy of reinforced self-supervised training stage keeps improving until convergence. According to the convergence curve, it becomes apparent that the convergence process of RL differs across various datasets.



(a) In 5-way 5-shot setting

(b) In 5-way 1-shot setting

Fig. 2. Reward curves of RISE

As a result, the agent of RL dynamically adapts the self-supervised model training paradigm as it interacts with the environment, while the most available balancing factor is selected by rating the explored policy through the training process. According to the convergence curve of the agent in different datasets, it can be observed that the reward values obtained by the agent module in different datasets can converge as the reinforced self-supervised training period increase. This effectively demonstrates the resilience of reinforcement learning in different datasets.

IV. CONCLUSION

In our letter, RISE utilizes agent-relative supervision to guide self-supervised learning so as to mine semantic information effectively and mitigate representation bias during the self-supervised phase. It enables RISE to dynamically learn a more diverse distribution, effectively improving the performance of the model. Extensive experiments reveal that the proposed RISE can optimize self-supervised learned representations, and our proposed method can outperform self-supervised few-shot methods. Meanwhile, under different datasets, the reward value obtained by the agent can effectively converge, which shows that the proposed pattern can be flexible and stable in few-shot learning.

REFERENCES

[1] W. Fu, L. Zhou, and J. Chen, "Bidirectional matching prototypical network for few-shot image classification," *IEEE Signal Process. Lett.*, vol. 29, pp. 982–986, 2022.

[2] C. Xiong, W. Li, Y. Liu, and M. Wang, "Multi-dimensional edge features graph neural network on few-shot image classification," *IEEE Signal Process. Lett.*, vol. 28, pp. 573–577, 2021.

[3] B. Jiang, K. Zhao, and J. Tang, "RGTransformer: Region-graph transformer for image representation and few-shot classification," *IEEE Signal Process. Lett.*, vol. 29, pp. 792–796, 2022.

[4] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[6] V. R. Konda and N. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1008–1014.

[7] L. Wen, X. Li, and L. Gao, "A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12890–12900, Dec. 2021.

[8] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," 2011.

[9] J. Christopher, "Q-learning," *Mach. Learn.*, vol. 3, pp. 279–292, 1992.

[10] L. J. Lin, "Reinforcement learning for robots using neural networks," Ph.D. dissertation Carnegie Mellon Univ., Pittsburgh, PA, USA, 1992.

[11] J. Y. Lim, K. M. Lim, C. P. Lee, and Y. X. Tan, "Self-supervised contrastive learning for few-shot image classification," *Neural Netw.*, vol. 165, pp. 19–30, 2023.

[12] Z. Zeng, Y. Yao, Z. Liu, and M. Sun, "A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 862.

[13] J. Zhang, L. Liu, R. Zhao, and Z. Shi, "A Bayesian meta-learning-based method for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote. Sens*, vol. 61, 2023, Art. no. 5500613.

[14] M. Li, L. Zhang, L. Cui, L. Bai, Z. Li, and X. Wu, "BLoG: Bootstrapped graph representation learning with local and global regularization for recommendation," *Pattern Recognit.*, vol. 144, 2023, Art. no. 109874.

[15] Y. Liu and B. Jiang, and J. Xu, "Axial assembled correspondence network for few-shot semantic segmentation," *IEEE J. Automatica Sinica*, vol. 10, no. 3, pp. 711–721, Mar. 2023.

[16] Y. Li et al., "Incremental few-shot object detection for robotics," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 8447–8453.

[17] Q. Xu, J. Chen, J. Tang, Q. Kang, and M. Zhou, "Performing effective generative learning from a single image only," in *Proc. 32nd Wireless Opt. Commun. Conf.*, 2023, pp. 1–5.

[18] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9630–9640.

[19] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.

[20] Z. Yang, J. Wang, and Y. Zhu, "Few-shot classification with contrastive learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 293–309.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[22] Y. Wang, S. Wang, Y. Li, and D. Dou, "Recognizing medical search query intent by few-shot learning," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 502–512.

[23] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Representations*, 2019.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[26] Jong-Chyi Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 645–666.

[27] Y. Rong, X. Lu, Z. Sun, Y. Chen, and S. Xiong, "ESPT: A self-supervised episodic spatial pretext task for improving few-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 9596–9605.

[28] Y. An, H. Xue, X. Zhao, and L. Zhang, "Conditional self-supervised learning for few-shot classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 2140–2146.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[30] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1057–1063.

[31] Y. Bengio and J.-S. Senecal, "Quick training of probabilistic neural nets by importance sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 17–24.

[32] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proc. Int. Conf. Learn. Representations*, 2019.

[33] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.

[34] M. Ren et al., "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Representations*, 2018.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[36] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[37] H.-J. Ye, H. Hu, D. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8805–8814.

[38] W.-H. Chu, Y.-J. Li, J.-C. Chang, and Y.-C. F. Wang, "Spot and learn: A maximum-entropy patch sampler for few-shot image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6251–6260.

[39] J. Hong et al., "Reinforced attention for few-shot learning and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 913–923.

[40] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8058–8067.

[41] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5714–5724.

[42] F. Sung et al., "Learning to compare: Relation network for few-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1199–1208.

[43] Y. An, H. Xue, X. Zhao, and J. Wang, "From instance to metric calibration: A unified framework for open-world few-shot learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, 2023, pp. 9757–9773.

[44] Y. Che, Y. An, and H. Xue, "Boosting Few-shot open-set recognition with multi-relation margin loss," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, Macao, SAR, China, 2023, pp. 3505–3513.