Enhancing Low-Rank Adaptation with Recoverability-Based Reinforcement Pruning for Object Counting

Haojie Guo[1], **Junyu Gao**[1,2], **Yuan Yuan**[1*]

[1]Northwestern Polytechnical University
[2]Institute of Artificial Intelligence (TeleAI), China Telecom
guohaojie@mail.nwpu.edu.cn, junyugao@nwpu.edu.cn, y.yuan.ieee@gmail.com

Abstract

Object counting is crucial for understanding the distribution of objects in different scenarios. Recently, many object counting networks have been designed to be more complex to achieve marginal improvements, leading to excessive time spent on model design. With the development of large models (LMs), various visual tasks can be accomplished by transferring pre-trained weights from LMs and fine-tuning them. However, tens of millions of training data make the pre-training parameters of LMs not entirely necessary. Moreover, if unnecessary parameters in the large model are not removed, it may lead to decreased performance on the tasks to be transferred. Motivated by this, this paper proposes an **E**nhancing low-**R**ank adaptation with **R**ecoverability-based **R**einforcement **P**runing (E3RP) method to balance the complexity of large model and the accuracy of counting tasks. Firstly, we design a new reward mechanism based on the feature similarity of large model before and after globally unstructured pruning of specific parameters. Additionally, we design a Patch Query Flip Attention (PQFA) mechanism to align multi-scale features through bidirectional interaction of features. Finally, the parameters of large model are pruned utilizing the pruning rate autonomously determined by the reinforcement learning network, and the large model is fine-tuned to counting tasks by a simple decoding head. Extensive experiments on four cross-scenario datasets demonstrate that the proposed method can remove redundant network parameters while ensuring network performance, with a maximum reduction of up to 63%.

Introduction

Object counting aims to analyze the quantity of objects in various scenarios and achieve image-level density estimation (Gao et al. 2021; Yang et al. 2020). Recent advances in deep neural networks and Transformers (Zhang, Gao, and Yuan 2024a,b; Chi et al. 2024) have led to a proliferation of algorithms for object counting (Yuan, Guo, and Gao 2025). However, the diminishing returns in counting accuracy brought by the complexity of networks have become more apparent. Moreover, due to critical challenges like variations in object sizes, the time needed for network design has extended. Meanwhile, open-source large models (LMs) are emerging as potential solutions to the protracted network design process. Consequently, many researchers aim to leverage the rich prior knowledge in LMs as pre-trained weights and transfer them to various downstream tasks such as object counting (Wan et al. 2025; Wan, Wu, and Chan 2023). Nevertheless, large models exhibit highly generalized multi-category feature extraction capabilities, resulting in a considerable redundancy of weights for counting tasks.

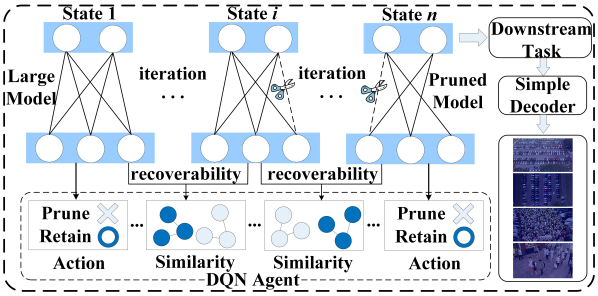

Figure 1: The illustration of the proposed reinforcement pruning method (E3RP). The redundant parameters of SAM are adaptively pruned in E3RP and then transferred to downstream tasks by fine-tuning the decoder.

In this situation, fine-tuning all weights of LMs for downstream tasks demands substantial computational resources, which is inconvenient and time-consuming. To expedite the transfer of LMs, the Low-Rank Adapter (LoRA) method (Hu et al. 2022) is proposed to fine-tune the entire large language model with reduced learning costs. LoRA trains additional parameters on top of the original pre-trained weights to serve as supplementary information, achieving weight fine-tuning and transfer with little cost. Fine-tuning redundant parameters in large models does not enhance their adaptability to downstream tasks. Additionally, significant differences exist in target distributions across scenarios, and simply using the prior knowledge of large models is insufficient to ensure generalization capability.

Building on the previous discussion, transferring priors from LMs to counting tasks faces two main challenges: (1) the parameter redundancy inherent in large models, and (2) the distribution gap between different scenarios. To tackle these issues, this paper introduces the Enhancing low-

*Corresponding Author

Rank adaption with Reinforcement-based Recoverability-based Pruning (E3RP) approach. This approach maintains counting accuracy without adding extra parameters to LMs by adaptively pruning the model's redundant parameters. Specifically, E3RP calculates the similarity of fused features before and after applying global unstructured pruning using a recoverability reward, gradually determining the most suitable model pruning ratio in the deep Q-learning process. Additionally, E3RP designs a scale-aware counting head for cross-scenario object counting task based on the proposed Patch Query Flip Attention (PQFA) mechanism.

Fig. 1 illustrates the E3RP workflow, where a DQN agent iteratively searches for the optimal pruning ratio, utilizing recoverability measurement to reduce model parameters while preserving key adaptation parameters for low-rank adaptation. In summary, the contributions of this paper are as follows:

- We introduce a large model transfer method that integrates reinforcement pruning with fine-tuning named E3RP. E3RP adaptively determines the optimal pruning ratio for large models, facilitating both adaptive pruning and fine-tuning.
- We propose a reinforcement learning reward mechanism based on recoverability measurement to effectively accelerate the model pruning.
- A scale-aware decoding head is designed to enable bidirectional interaction of multi-scale object features, thereby enhancing the model's generalization ability.

Related Work

Fine Tuning of Large Models

Fine-tuning large models involves transferring the knowledge of a large model to downstream tasks while minimizing the number of training parameters. For example, adapter-based fine-tuning methods repeatedly integrate simple adaptation layers into the large model's structure (Chen et al. 2023; Cheng et al. 2024). Compared to standard adapter methods (Yao et al. 2024; Ren et al. 2024), LoRA supplements large models with low-rank adaptation information while minimizing the number of training parameters (Hu et al. 2022; Wang et al. 2023). Additionally, Prompt-based fine-tuning methods focus on using task-specific prompts without training additional parameters in the large model (Nie et al. 2024; Zhu et al. 2023). However, prompt-based methods often struggle to ensure the relevance of the prompt to the downstream task. In contrast, prefix-based methods train the prefix of the input sequence, resulting in more reliable outcomes (Maleki, Yang, and Burtscher 2016; Saito et al. 2023).

Reinforcement Learning with Vision Tasks

Compared to deep learning methods, reinforcement learning excels at tasks requiring greater autonomy due to its unique action agents. In recent years, numerous studies have focused on integrating reinforcement learning with visual tasks (Zeng et al. 2021; Allioui et al. 2022). For example, well-designed agents can focus on key areas of an image to perform object segmentation tasks (Vecchio et al. 2020; Wang et al. 2020). Additionally, classification agents can evaluate the fit between the current input and various decoders to enhance object detection results (Dong, Xia, and Peng 2021; Uzkent, Yeh, and Ermon 2020). Furthermore, some unique weighting agents can assess the number of objects at different scales within an image, thereby enabling crowd counting (Liu et al. 2020).

Unstructured Pruning of Models

Model pruning seeks to balance network parameter redundancy with performance (Wang and Zhang 2024). Unstructured pruning makes the global parameters of the model sparse, without requiring the network structure to be rebuilt as in structured pruning methods. For example, an image super-resolution reconstruction network can achieve better upsampling results through unstructured pruning (Zhang, Yin, and Zhang 2024). Additionally, unstructured pruning is more suitable for transferring large model weights to target tasks. In recent years, some studies have also focused on reducing the layers of deep neural networks, such as activation layers, through unstructured pruning (Liao et al. 2023). However, the criteria for model pruning significantly impact the final pruning outcomes. Consequently, pruning methods based on weight representation and dynamic probabilities have been gradually proposed (Kwon et al. 2020; Gonzalez-Carabarin et al. 2022).

Cross-Scenario Object Counting

In surveillance scenarios, significant differences in object scales impose high demands on the model's ability to align cross-scale features (Lin and Chan 2024, 2023). Conversely, the target scale differences in remote sensing perspectives are smaller, but the objects are generally smaller in size (Gao, Zhao, and Li 2024). Balancing feature extraction across these two scenarios places significant demands on the model's generalization ability (Gao, Yuan, and Wang 2020). Consequently, some researchers have embedded dilated convolutions in models to capture object size variations through a larger receptive field (Gao, Liu, and Wang 2020). Moreover, treating objects with scale differentiation before fusion is a common approach (Gao et al. 2022; Yi et al. 2023; Wang et al. 2024). Alternatively, enhancing object counting regression tasks through segmentation can balance the numerical disparities in Gaussian regression kernels (Guo, Gao, and Yuan 2024).

Proposed Method

Overall Framework

Fig. 2 illustrates the main framework of the proposed Reinforcement Pruning (E3RP) method. The goal of E3RP is to find the optimal pruning ratio for the large model using deep Q-learning and fine-tune the decoder head for the target counting task. In E3RP, we first define the agent's actions and states based on different pruning ratios and then design a recoverability reward mechanism. Specifically, we prune the large model's decoder based on the pruning ratio and compare features before and after pruning to calculate

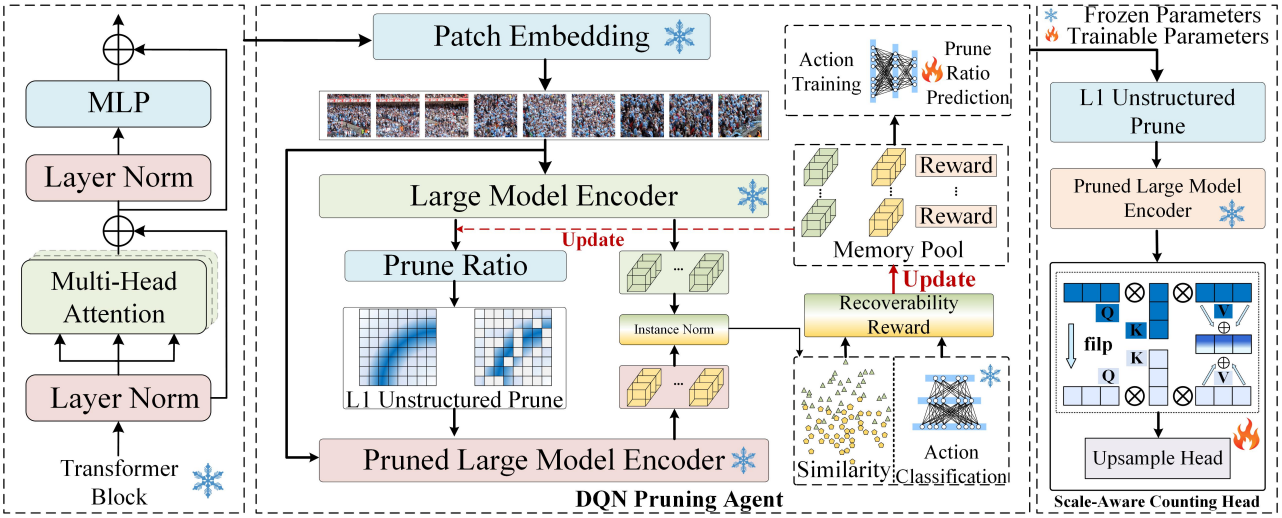

Figure 2: The framework of the proposed method. The flame and ice icons are utilized to represent trainable and frozen parameters, respectively. Note that during the iterative process of the reinforcement pruning agent, only the action classification network is being trained.

a recoverability reward, combining it with the output of the action classification network.

During each iteration of deep Q-learning, the agent's state includes the large model before and after pruning at various ratios. The agent's action is whether to keep or discard network weights. We store the network's output features and rewards in a memory pool to train the action classifier. Feature similarity acts as prior knowledge, while the action classification network serves as another learnable expert. Using the memory pool as training data ensures the action classifier aligns with the prior knowledge of feature similarities. Notably, no parameters are trained during the memory pool storage process. The total time cost is about 80 epochs, which is reasonable compared to nearly 500 training periods required by most counting algorithms.

Adaptive Pruning with Recoverability Reward

Since the main purpose of pruning methods is to reduce the redundancy of a large model's parameters, we can define the large model as a set of parameters, denoted as $P \in \{p_0, p_1, ..., p_n\}$. Given an initial pruning rate p_r, we can obtain the new pruned large model P' after pruning. In this paper, L1 global unstructured pruning is employed to eliminate the redundant parameters in P_0. L1 global unstructured pruning determines the importance of parameters by evaluating their absolute values. The calculation process of P' is in (1):

$$P' = \{p_i \in P_0 | \, |p_i| \geq \sigma(|P_0|)_{n \times p_r}\}, \tag{1}$$

where σ represents the sorting process from smallest to largest, and $\sigma(|P_0|)_{n \times p_r}$ denotes the $(n \times p_r)_{th}$ parameter after sorting operation based on their absolute values. Assuming the input images to the model are $\{x_1, x_2, \ldots, x_n\}$, and the output features obtained from the large model before pruning are represented as F_b, then the features of the image before pruning can be expressed by the following equation:

$$F_b = \sum_{i=1}^{n} P_0(x_i), \tag{2}$$

where x_i represents the input image. Similarly, the features calculated by P' after pruning can be obtained as F_a. Subsequently, the network features before and after pruning are evaluated through action classification and similarity calculation to determine whether the current pruning operation is reasonable. Specifically, we firstly calculate the cosine similarity between features F_a and F_b, as shown in the following equation:

$$S_{cos} = \frac{\mathbf{F}_a \cdot \mathbf{F}_b}{\|\mathbf{F}_a\| \|\mathbf{F}_b\|}. \tag{3}$$

Although cosine similarity can evaluate the similarity of network output features before and after pruning, it is not a learnable process. As pruning increases, network sparsity will reduces the cosine similarity reliability. To address this, we design a simple but effective action classification network that evaluates pruning rationality post-training and corrects cosine similarity inaccuracies when parameters are sparse. The action classification is represented by the following equation:

$$\hat{y} = \arg\max_i f_c([\mathbf{F}_a; \mathbf{F}_b]; \mathbf{W}_c)_i \in \{0, 1\}, \tag{4}$$

where $\hat{y}$ stands for the final binary classification result, taking a value of 0 or 1 to indicate whether pruning is necessary or not. f_c denotes the classification network used to determine whether there is a significant difference between features F_a and F_b. $\mathbf{W}_c$ represents the parameters of the classification network f_c.

The action classification network and cosine similarity can serve as complementary experts to evaluate the pruning operation. Based on this, the recoverability reward can

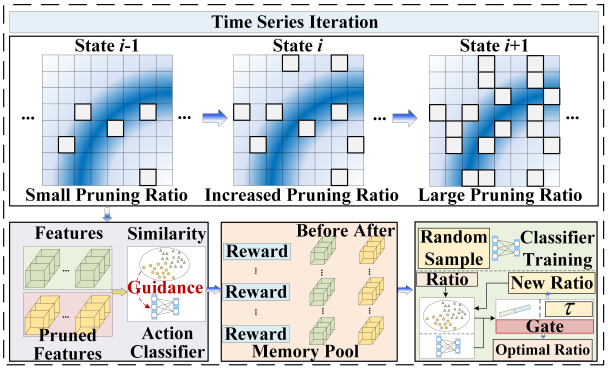

Figure 3: The iterative inference process of the reinforcement pruning agent.

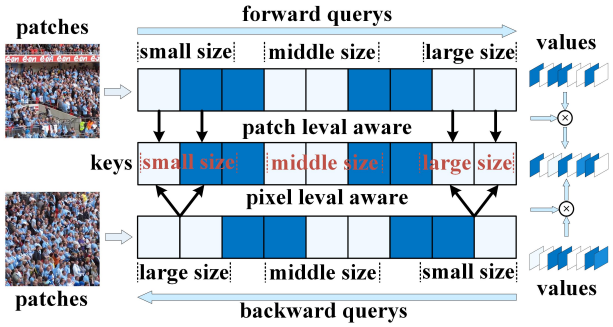

Figure 4: The Demonstration of Patch Qury Flip Attention (PQFA). The input of PQFA consists of stretched patches.

be calculated utilizing the action classification results and the similarity of the features. Assume that the results from the action classification network and the cosine similarity produce the following four forms: The specific calculation process is as follows: $I_1 = \mathbb{I}(\hat{y} = 1, s \geq \tau), I_2 = \mathbb{I}(\hat{y} = 1, s < \tau), I_3 = \mathbb{I}(\hat{y} = 0, s \geq \tau), I_4 = \mathbb{I}(\hat{y} = 0, s < \tau)$, where $\mathbb{I}(\cdot)$ is an indicator function, which takes the value 1 when the condition $(\cdot)$ is true, and 0 otherwise. τ stands for the threshold for cosine similarity. Hence, the reward can be calculated as the following equation:

$$r^* = r(I_1 + I_4 - I_2 - I_3), \tag{5}$$

where r denotes a predetermined reward value, which is leveraged to penalize or reward different action classifications.

Fig. 3 illustrates the adaptive pruning process under the recoverability-based reward, detailing how the action classification network and cosine similarity collectively determine the rationality of pruning.

Fine-Tuning with Scale-Aware Counting Head

After adaptive pruning of large mdoel, the optimal pruning ratio can be calculated in the previous section. The pruned model effectively ensures the effectiveness of extracted features while minimizing the number of parameters. However, dense objects in surveillance scenarios often exhibit significant scale variations, even within a single patch of the transformer network structure. To enhance feature interaction for objects of varying sizes within a single patch, we propose a Patch Query Flip Attention (PQFA) mechanism. Specifically, the calculation process of PQFA is as follows:

$$F_{mix} = \left(\mathrm{S}\left(\frac{Q_i K^T}{\sqrt{d_k}} \right) + \lambda \cdot \mathrm{S}\left(\frac{(Q_i^f) K^T}{\sqrt{d_k}} \right) \right) V, \tag{6}$$

where Q_i^f represents the flipped tokens, and λ denotes the weight that controls the flipped query tokens. $\mathrm{S}(\cdot)$ stands for the softmax operation. As shown in Fig. 4, the token sequence in the query from the surveillance perspective includes objects of various sizes. By flipping the tokens, direct interaction between objects of various sizes within the tokens can be obtained, resulting in pixel-level feature alignment.

The pruned features must be upsampled to match the sizes of the input images. In this paper, the output feature of SAM's encoder is progressively upsampled through a combination of deconvolution and convolution operations. Because deconvolution operations often cause a checkerboard effect, necessitating a 1×1 convolution to fill in the missing features pixel by pixel. Specifically, the feature upsampling process is represented by the following formula:

$$F_{\text{out}} = f_1^{-1}\left(f_2^{-1}\left(\cdots f_m^{-1}(F_{mix}) * W_m \right) * W_2 \right) * W_1, \tag{7}$$

where f_i^{-1} denotes the i_{th} deconvolution operation, and W_i stands for the 1×1 convolution kernel of the i_{th} layer. The variable m represents the number of upsampling layers and is determined by the size of patch.

Note that the network structure in this part needs to be trained as a decoder in the proposed E3RP method. Additionally, because decoders that come with large models are usually not directly suitable for counting tasks, the proposed decoder is also used in other large model fitting methods in comparative experiments such as LoRA. Therefore, the proposed E3RP method can be seen as only additionally training an action classification network to iteratively find the optimal pruning ratio of the large model.

Training and Optimization

The training process involves three steps: transfer and fine-tuning of the large model, iterative deep Q-learning, and training the action classification network. Firstly, the transfer and fine-tuning include freezing the large model's parameters and transferring training to the Scale-Aware Counting Head (SACH) decoder. For example, in SAM, only the decoder's parameters are trained during transfer. The action classification network doesn't join the fine-tuning since it's used to determine the optimal pruning ratio. Meanwhile, the final result of the designed decoder SACH constrains the convergence of the decoder through the Mean Squared Error (MSE) loss. This part of the loss can be expressed by the following equation:

$$L_{dm} = \frac{1}{n} \sum_{i=1}^{n} \left(D_p - D_{gt} \right)^2, \tag{8}$$

where L_{dm} represents the regression counting loss.

Secondly, the iterative steps of deep Q-learning mainly include the storage of memory pool data and the state transition of Q-values. After obtaining the features before and after pruning, the action classification network and the cosine similarity calculation module respectively provide parameter retention opinions. Specifically, the action classification network utilizes a binary vector to indicate whether the pruning operation is reasonable, and cosine similarity serves as the judgment guide for the output of the action classification network, particularly when it exceeds the threshold τ. If there is a difference between the two prediction results, the reward value is negative, and this negative reward is stored in the memory pool as a true label to supervise the training of the action classifier. Additionally, a negative reward triggers the agent's greedy strategy, which influences the prediction network's expected output in the next state. The specific greedy strategy of Q-values can be described by the following equation:

$$R(s) = R(s, p_s) + \gamma \cdot E[R(s', p_{s'})], \text{s.t.} \quad p_{s'} = p_s + 1, \quad (9)$$

where $R(s)$ denotes the reward value of state s calculated by equation (5). p_s and $p_{s'}$ stand for the pruning ratios of current state and the next state, respectively. $E[R(s', p_{s'})]$ represents the expected reward of the next state s'. γ means the greed coefficient. Given that cosine similarity is an intuitive physical prior, this process can be regarded as the guidance of the cosine similarity for the action classification network.

Finally, the action classification network is trained leveraging the features and Q-values corresponding to different pruning rates stored in the memory pool. The size of the Q-values is consistent with the output size of the action classification network, and the calculation process is as follows:

$$Q = \begin{cases} [R(s), f_{\text{c}}([\mathbf{F}_a; \mathbf{F}_b]; \mathbf{W}_{\text{c}})[1 - \hat{y}]], \text{if } \hat{y} = 0 \\ [f_{\text{c}}([\mathbf{F}_a; \mathbf{F}_b]; \mathbf{W}_{\text{c}})[1 - \hat{y}], R(s)], \text{if } \hat{y} = 1 \end{cases}, \quad (10)$$

where $\hat{y}$ stands for the binary classification result computed by (4). Specifically, this process does not require the involvement of cosine similarity, as the guidance information of cosine similarity is already stored in the different Q-values corresponding to the reward values. To ensure the predicted values of the action classification network closely match the stored Q-values, the loss function can be defined as follows:

$$L_{\text{c}} = \frac{1}{n} \sum_{i=1}^{n} \left(f_{\text{c}}([\mathbf{F}_a; \mathbf{F}_b]; \mathbf{W}_{\text{c}}) - Q \right)^2, \quad (11)$$

Up to this point, the entire proposed E3RP training process has been thoroughly exhibited.

Experiments

Data Preparation

Drone-View Datasets The CAR Parking lot (CARPK) and Pontifical Catholic University of Parana (PUCPR) datasets contain images of vehicle targets in parking lots from a drone's perspective, making them suitable for validating remote sensing target counting methods (Xia et al. 2018). Specifically, CARPK contains 1448 images and PUCPR contains 125 images. Although PUCPR has fewer images, it includes scenes under different weather conditions, making it more challenging to predict.

Surveillance Datasets Shanghai Tech_A (SHHA) and Shanghai Tech_B (SHHB) are obtained by collecting images of densely populated crowds from the internet and images from fixed surveillance perspectives, respectively (Zhang et al. 2016). These two datasets contain rich information on target scale variations, which can effectively validate the algorithm's ability to handle objects with scale differences.

Implementation Details

Experimental Setting During training, utilizing SAM as an example, the pre-trained model only loads the feature encoding and upsampling modules. Additionally, no extra parameters are added to the SAM feature extraction network, which remains frozen throughout the training process. All experiments are conducted on a server with four RTX 3090 GPUs. The input images are uniformly resized to 720×720 pixels.

Hyperparameters Setting Due to the use of pre-trained weights, the learning rate for the proposed E3RP network decoder is set to 0.0001. The optimizers for both the action classification network and the counting network are AdamW. The cosine similarity threshold τ in Equation (5) is set to 0.99, and the base reward value is 10. The memory pool capacity of the deep Q-learning network is set to 100.

Evaluation Metrics

Object counting aims to calculate the number of objects in an image and display their distribution. Therefore, the proposed E3RP method is evaluated leveraging two metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). The calculation processes for MAE and MSE are represented by the following formulas:

$$MAE = |C_{gt} - C_{pred}|, \quad (12)$$

$$MSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (C_{pred} - C_{gt})^2}, \quad (13)$$

where C_{gt} represents the ground truth count of instances in different crowd scenes, and C_{pred} represents the predicted count of instances.

Results and Discussion

Overall Results In this section, we quantitatively compare the proposed Enhancing low-Rank Recoverability-based Reinforcement Pruning (E3RP) with recently introduced methods on cross-scenario datasets. As shown in Tab. 1, E3RP achieves state-of-the-art (SOTA) results on two remote sensing UAV datasets, demonstrating its efficacy in achieving competitive experimental results while removing redundant parameters of large models like SAM. Due to the scenario differences between UAV and surveillance perspectives, many algorithms struggle with cross-scenario generalization. For example, ASPDNet utilizes a large number of dilated convolution structures, enhancing its ability to

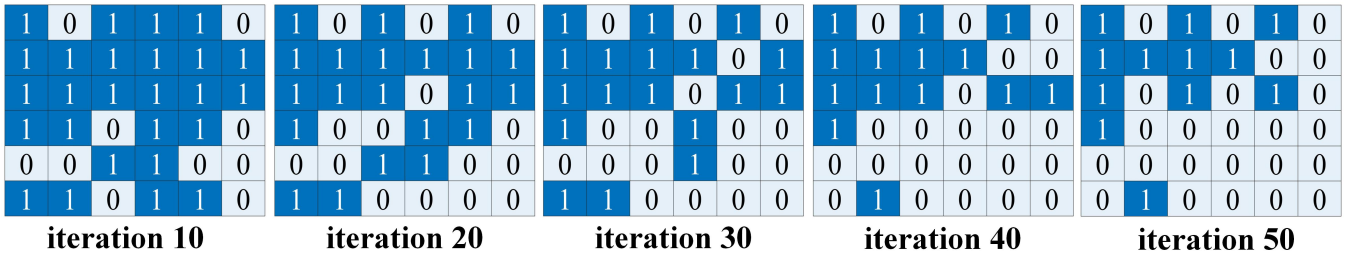

Figure 5: The visualization of network weights during the iterative process of the reinforcement pruning agent. We selected a 6×6 patch of the LayerNorm layer weights for visualization. The numbers 0 and 1 indicate whether the weights are retained.

Methods	CARPK↓		PUCPR↓		SHHA↓		SHHB↓	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN (Zhang et al. 2016)	39.10	43.30	21.86	29.53	110.20	173.20	26.40	41.30
CSRNet (Li, Zhang, and Chen 2018)	11.48	13.32	8.65	10.24	68.20	115.00	10.60	16.00
BL (Ma et al. 2019)	9.58	11.38	6.54	8.13	62.80	101.80	7.70	12.70
ASPDNet (Gao, Liu, and Wang 2020)	7.81	10.16	2.97	3.78	60.80	96.20	7.20	10.50
PSGCNet (Gao et al. 2022)	8.15	10.46	5.24	7.36	**56.10**	95.60	6.60	**9.70**
LMSFFNet (Yi et al. 2023)	7.05	9.03	4.49	6.21	85.85	139.90	9.20	15.10
BDRNet (Guo, Gao, and Yuan 2024)	5.94	8.59	1.91	2.49	64.31	102.98	9.04	12.78
HKINet (Wang et al. 2024)	4.89	7.15	2.77	3.62	61.00	94.30	8.00	11.50
E3RP (Ours)	**4.18**	**5.51**	**1.40**	**1.85**	58.37	**93.69**	**6.46**	10.39

Table 1: Comparison results with different algorithms of E3RP on four cross-scenario datasets.

extract objects with significant scale differences, particularly from a surveillance perspective. However, the extensive utilization of dilated convolutions leads to high memory consumption and suboptimal performance on remote sensing datasets. In contrast, E3RP achieves comparable results on both surveillance perspective datasets (SHHA and SHHB) and remote sensing UAV datasets (CARPK and PUCPR). Although E3RP does not achieve the best results on each dataset, it remains competitive compared to other algorithms, particularly given that it was trained on fewer parameters. Overall, the experimental results across various scenarios indicate that E3RP exhibits high generalization ability, attributed to the proposed Patch Query Filp Attention (PQFA) mechanism, which enhances feature interaction of targets of different sizes within a single patch in the Transformer structure.

Ablation of DQN-Pruning In this section, we conduct ablation experiments on key hyperparameters within the deep reinforcement pruning framework (DQN-P). Specifically, we focuses on two critical hyperparameters: the similarity threshold (τ) and the reward value r. Although the proposed reward mechanism is newly designed, the magnitude of the single reward value during the agent's iterations can affect network convergence. As shown in Tab. 2, we conduct ablation experiments on the hyperparameters τ and the reward value r under three different levels. As the similarity threshold τ decreases, the model's pruning rate increases. This occurs because a smaller τ value allows for greater variability between network parameters. Conversely, the reward value r have the most significant impact on the convergence of DQN when τ is 0.98. Additionally, we visualize the iterative process of reinforcement pruning through the change of

r \ τ	$\tau = 0.99$			$\tau = 0.98$			$\tau = 0.97$		
	PR	MAE	MSE	PR	MAE	MSE	PR	MAE	MSE
1	53%	1.47	2.01	58%	2.33	2.92	63%	1.94	3.02
10	53%	1.40	1.85	59%	2.05	2.88	63%	2.32	3.06
20	54%	1.50	2.09	54%	1.99	3.03	63%	2.31	3.02

Table 2: Ablation results with different reinforcement pruning hyperparameters on PUCPR. PR stands for the Pruning Ratio.

SAM-B	DQN-P	PQFA	MAE	MSE
✔	✘	✘	6.96	11.79
✔	✔	✘	6.90	10.61
✔	✘	✔	6.61	10.56
✔	✔	✔	**6.46**	**10.39**

Table 3: Ablation results with different module settings in E3RP on SHHB.

Models	Total Params	Training Params	MAE	MSE
SAM-B	93.31M	3.73M	2.02	2.78
SAM-B+LoRA	93.83M	4.25M	1.90	2.64
SAM-B+DQN-P	55.89M	3.73M	1.67	2.38
SAM-B+DQN-P+LoRA	56.41M	4.25M	**1.40**	**1.85**

Table 4: Ablation Results with Different Module Settings concerning the training parameters on PUCPR.

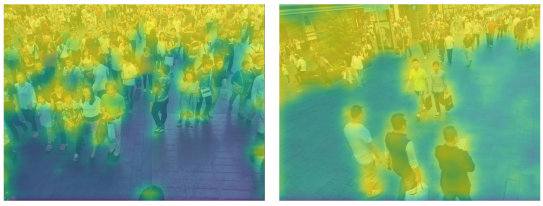

Figure 6: The visualization of the attention regions in PQFA.

network weights. As shown in Fig. 5, the model's weights become increasingly sparse with more iterations.

Ablation of Modules and Cost We compare the impact of various proposed modules on the network's final experimental results in this section. As shown in Tab. 3, DQN-P represents the reinforcement pruning module, and PQFA represents the Patch Query Flip Attention module. Due to minor differences in object sizes in the PUCPR and CARPK datasets, we choose the SHHB dataset for ablation experiments. In Tab. 3, proper model pruning evidently filters out redundant parameters in the network. Additionally, we visualize the attention regions of PQFA in Fig. 6. Combining quantitative and qualitative analyses from Fig. 6 and Tab. 3, E3RP effectively balances network parameters and downstream task transfer effectiveness.

In Tab. 4, the trainable parameters of the proposed E3RP are fewer than those of LoRA. Although LoRA has a smaller parameter size due to low-rank adaptation, it is non-adaptive and requires the decoder to adapt to downstream tasks. In contrast, E3RP only trains the decoder during the training process and the parameters of pruned SAM is frozen. Because the action classification network isn't involved in counting inference, its trained parameters are not included in E3RP. Generally, E3RP does not increase the parameter size compared to LoRA and achieves better experimental results, demonstrating the effectiveness of the proposed E3RP.

Ablation of Datasets As shown in Tab. 5, the pruned network, combined with PQFA and LoRA, achieves competitive results on different datasets from remote sensing drones and surveillance perspectives. Additionally, compared to the SAM-base model, E3RP prunes up to 59% of network parameters with hyperparameters τ set to 0.99 and r to 10. Notably, E3RP can achieve a higher pruning rate by reducing τ, but excessive reduction may lead to a decline in model performance. Therefore, an appropriate τ can balance pruning rate and counting ability. Furthermore, we visualize the generalization capabilities of E3RP and other methods across the four datasets. As shown in Fig. 7, E3RP outperforms on the remote sensing UAV and surveillance perspective datasets, demonstrating PQFA's effective attention to object size differences and validating the effectiveness of removing redundant parameters for performance improvement.

Datasets	Model	PR	MAE	MSE
SHHA	SAM-B	0%	63.67	94.50
	DQN-P	**52%**	**58.37**	**93.69**
SHHB	SAM-B	0%	6.96	11.79
	DQN-P	**59%**	**6.46**	**10.39**
PUCPR	SAM-B	0%	2.02	2.78
	DQN-P	**53%**	**1.40**	**1.85**
CARPK	SAM-B	0%	4.49	6.09
	DQN-P	**48%**	**4.18**	**5.51**

Table 5: Comparison results of pruning ratios under reinforcement pruning on cross-scenario datasets.

Figure 7: The visualization and comparison of object counting results on cross-scenario datasets. The four input images are from PUCPR, CARPK, SHHA, and SHHB, respectively.

Conclusion

This paper proposes a reinforcement pruning method named E3RP to enhance the low-rank adaptation of fine-tuning large models. we conduct experiments on four cross-scenario datasets to validate the generalization capability of E3RP. Experimental results demonstrate that the proposed E3RP effectively balances parameter redundancy and accuracy in downstream counting tasks.

In future work, we will focus on transferring multimodal large models to downstream tasks, particularly in modalities like image and text, to reduce model design costs.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 62306241.

References

Allioui, H.; Mohammed, M. A.; Benameur, N.; Al-Khateeb, B.; Abdulkareem, K. H.; Garcia-Zapirain, B.; Damaševičius, R.; and Maskeliūnas, R. 2022. A multi-agent deep reinforcement learning approach for enhancement of COVID-19 CT image segmentation. *Journal of personalized medicine*, 12(2): 1–23.

Chen, T.; Zhu, L.; Deng, C.; Cao, R.; Wang, Y.; Zhang, S.; Li, Z.; Sun, L.; Zang, Y.; and Mao, P. 2023. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3367–3375.

Cheng, Z.; Wei, Q.; Zhu, H.; Wang, Y.; Qu, L.; Shao, W.; and Zhou, Y. 2024. Unleashing the potential of SAM for medical adaptation via hierarchical decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3511–3522.

Chi, K.; Li, J.; Jing, W.; Li, Q.; and Wang, Q. 2024. Neural Implicit Fourier Transform for Remote Sensing Shadow Removal. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–10.

Dong, S.; Xia, Y.; and Peng, T. 2021. Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. *IEEE Transactions on Network and Service Management*, 18(4): 4197–4212.

Gao, G.; Liu, Q.; Hu, Z.; Li, L.; Wen, Q.; and Wang, Y. 2022. PSGCNet: A pyramidal scale and global context guided network for dense object counting in remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–12.

Gao, G.; Liu, Q.; and Wang, Y. 2020. Counting from sky: A large-scale data set for remote sensing object counting and a benchmark method. *IEEE Transactions on geoscience and remote sensing*, 59(5): 3642–3655.

Gao, J.; Han, T.; Yuan, Y.; and Wang, Q. 2021. Domain-adaptive crowd counting via high-quality image translation and density reconstruction. *IEEE transactions on neural networks and learning systems*, 34(8): 4803–4815.

Gao, J.; Yuan, Y.; and Wang, Q. 2020. Feature-aware adaptation and density alignment for crowd counting in video surveillance. *IEEE transactions on cybernetics*, 51(10): 4822–4833.

Gao, J.; Zhao, L.; and Li, X. 2024. NWPU-MOC: A Benchmark for Fine-grained Multi-category Object Counting in Aerial Images. *IEEE Transactions on Geoscience and Remote Sensing*.

Gonzalez-Carabarin, L.; Huijben, I. A.; Veeling, B.; Schmid, A.; and van Sloun, R. J. 2022. Dynamic probabilistic pruning: A general framework for hardware-constrained pruning at different granularities. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1): 733–744.

Guo, H.; Gao, J.; and Yuan, Y. 2024. Balanced Density Regression Network for Remote Sensing Object Counting. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–13.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 1–13.

Kwon, S. J.; Lee, D.; Kim, B.; Kapoor, P.; Park, B.; and Wei, G.-Y. 2020. Structured compression by weight encryption for unstructured pruning and quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1909–1918.

Li, Y.; Zhang, X.; and Chen, D. 2018. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1091–1100.

Liao, Z.; Quétu, V.; Nguyen, V.-T.; and Tartaglione, E. 2023. Can Unstructured Pruning Reduce the Depth in Deep Neural Networks? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1402–1406.

Lin, W.; and Chan, A. B. 2023. Optimal transport minimization: Crowd localization on density maps for semi-supervised counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21663–21673.

Lin, W.; and Chan, A. B. 2024. A Fixed-Point Approach to Unified Prompt-Based Counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3468–3476.

Liu, L.; Lu, H.; Zou, H.; Xiong, H.; Cao, Z.; and Shen, C. 2020. Weighing counts: Sequential crowd counting by reinforcement learning. In *Proceedings of European Conference on Computer Vision*, 164–181.

Ma, Z.; Wei, X.; Hong, X.; and Gong, Y. 2019. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6142–6151.

Maleki, S.; Yang, A.; and Burtscher, M. 2016. Higher-order and tuple-based massively-parallel prefix sums. *ACM SIGPLAN Notices*, 51(6): 539–552.

Nie, X.; Ni, B.; Chang, J.; Meng, G.; Huo, C.; Xiang, S.; and Tian, Q. 2024. Pro-Tuning: Unified Prompt Tuning for Vision Tasks. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6): 4653–4667.

Ren, B.; Qian, Z.; Sun, Y.; Gao, C.; and Zhang, C. 2024. WebSAM-Adapter: Adapting Segment Anything Model for Web Page Segmentation. In *European Conference on Information Retrieval*, 439–454.

Saito, K.; Sohn, K.; Zhang, X.; Li, C.-L.; Lee, C.-Y.; Saenko, K.; and Pfister, T. 2023. Prefix conditioning unifies language and label supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2861–2870.

Uzkent, B.; Yeh, C.; and Ermon, S. 2020. Efficient object detection in large images using deep reinforcement learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1824–1833.

Vecchio, G.; Palazzo, S.; Giordano, D.; Rundo, F.; and Spampinato, C. 2020. MASK-RL: Multiagent video object segmentation framework through reinforcement learning. *IEEE transactions on neural networks and learning systems*, 31(12): 5103–5115.

Wan, J.; Wu, Q.; and Chan, A. B. 2023. Modeling noisy annotations for point-wise supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 15065–15080.

Wan, J.; Wu, Q.; Lin, W.; and Chan, A. 2025. Robust Zero-Shot Crowd Counting and Localization With Adaptive Resolution SAM. In *European Conference on Computer Vision*, 478–495.

Wang, A.; Islam, M.; Xu, M.; Zhang, Y.; and Ren, H. 2023. Sam meets robotic surgery: an empirical study on generalization, robustness and adaptation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–244.

Wang, H.; Zeng, X.; Zhang, T.; Wei, J.; Hou, X.; Wu, H.; and Zhang, K. 2024. Hierarchical Kernel Interaction Network for Remote Sensing Object Counting. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–13.

Wang, H.; and Zhang, W.-Q. 2024. Unstructured Pruning and Low Rank Factorisation of Self-Supervised Pre-Trained Speech Models. *IEEE Journal of Selected Topics in Signal Processing*, 1–14.

Wang, Y.; Dong, M.; Shen, J.; Wu, Y.; Cheng, S.; and Pantic, M. 2020. Dynamic face video segmentation via reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6959–6969.

Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; and Zhang, L. 2018. DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3974–3983.

Yang, Y.; Li, G.; Wu, Z.; Su, L.; Huang, Q.; and Sebe, N. 2020. Reverse perspective network for perspective-aware object counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4374–4383.

Yao, B.; Deng, Y.; Liu, Y.; Chen, H.; Li, Y.; and Yang, Z. 2024. SAM-Event-Adapter: Adapting Segment Anything Model for Event-RGB Semantic Segmentation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 9093–9100.

Yi, J.; Shen, Z.; Chen, F.; Zhao, Y.; Xiao, S.; and Zhou, W. 2023. A lightweight multiscale feature fusion network for remote sensing object counting. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–13.

Yuan, Y.; Guo, H.; and Gao, J. 2025. Distance-aware network for physical-world object distribution estimation and counting. *Pattern Recognition*, 157: 110896.

Zeng, N.; Li, H.; Wang, Z.; Liu, W.; Liu, S.; Alsaadi, F. E.; and Liu, X. 2021. Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip. *Neurocomputing*, 425: 173–180.

Zhang, B.; Gao, J.; and Yuan, Y. 2024a. A Descriptive Basketball Highlight Dataset for Automatic Commentary Generation. In *ACM Multimedia*, 10316–10325.

Zhang, B.; Gao, J.; and Yuan, Y. 2024b. Center-enhanced video captioning model with multimodal semantic alignment. *Neural Networks*, 180: 106744.

Zhang, S.; Yin, R.; and Zhang, M. 2024. Dynamic Unstructured Pruning Neural Network Image Super-resolution Reconstruction. *Informatica*, 48(7): 11–22.

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 589–597.

Zhu, B.; Niu, Y.; Han, Y.; Wu, Y.; and Zhang, H. 2023. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15659–15669.