GRPO-RM: FINE-TUNING REPRESENTATION MODELS VIA GRPO-DRIVEN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Group Relative Policy Optimization (GRPO), a reinforcement learning method used to fine-tune large language models (LLMs), has proved its effectiveness in practical applications such as DeepSeek-R1. It raises a question whether GRPO can be generalized to representation learning models. In this paper, we propose Group Relative Policy Optimization for Representation Model (GRPO-RM), and investigate the performance of GRPO-like policy in post-training representation models. Specifically, our method treats categories as a predefined output set to functionally replace token sequence sampling in LLMs, thereby generating an output group, which is essential for the probability-driven optimization of GRPO. In addition, a specialized reward function is designed to accommodate the properties of representation models. Extensive experiments are conducted on various real-world datasets to validate the effectiveness of our proposed method.

1 INTRODUCTION

Representation Learning has been a fundamental and crucial research topic Chen et al. (2020); Grill et al. (2020); He et al. (2020); Caron et al. (2021) in computer vision. The representation models aim to encode comprehensive visual semantics and generate feature embeddings that are invariant with tasks. These robust image representations are valid for various downstream tasks, including image classification Dai et al. (2021); Pham et al. (2021), semantic segmentation Ding et al. (2019); Zhang et al. (2019), instance segmentation Liu et al. (2018), action recognition Ji et al. (2012), and autonomous driving systems Levinson et al. (2011). Generally, despite its outstanding generalization performance, an additional fine-tuning phase is needed to perform downstream tasks, which can also be regarded as the post-training set in Large Language Models Bai et al. (2023); Touvron et al. (2023). Specifically, a task-specific head network is employed to map the extracted features to downstream task objectives. The overall network is then trained in a supervised manner to improve the performance.

Since DeepSeek-R1 DeepSeek-AI (2025) was published, a new optimization method, namely Group Relative Policy Optimization (GRPO), has become popular Shao et al. (2024); Yu et al. (2025). As a reinforcement learning method, GRPO has demonstrated strong capabilities in fine-tuning Large Language Models (LLMs), which naturally leads to a question: ***Can GRPO be generalized to representation learning models?*** In this paper, we are going to answer the above question and develop a GRPO adaption for representation models.

The backbones of representation models vary in different methods. ResNet He et al. (2016), U-Net Ronneberger et al. (2015), and Vision Transformers (ViTs) Dosovitskiy et al. (2020) are representative network architectures of visual models. Given that both LLMs and ViTs predominantly utilize transformer architectures, they are likely to exhibit inherent architectural similarities. Therefore, we attempt to adapt GRPO for post-training on ViT-based representation models. In this paper, we employ DINOv2 Oquab et al. (2023), a well-known self-supervised ViT model that can generate robust features for various downstream tasks, as the pre-trained base model.

Specifically, we post-train DINOv2 for two representative downstream tasks: image classification and semantic segmentation. As a cornerstone computer vision task, image classification provides an effective benchmark for assessing a model's ability to extract and leverage global semantic representations. Semantic segmentation, on the other hand, provides a direct evaluation of local repre-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

sentation quality. These two tasks align precisely with the DINOv2 output structure, that is, class tokens and patch tokens as well.

Our contributions can be summarized as follows:

- Based on the properties of representation learning, we redesign the reward functions in the objective of GRPO. To be specific, we employ accuracy rewards and uniformity rewards, encouraging the model to generate correct predictions while adaptively discouraging the model to generate wrong ones.
- With the novel reward functions, we propose GRPO-RM to post-train DINOv2, which is the first reinforcement post-training method for representation learning models to our best knowledge.
- Extensive experiments are conducted on various datasets for different downstream tasks to validate the effectiveness of our method. As a result, our method significantly outperforms the standard fine-tuning method. In particular, GRPO-RM achieves an average 4.26% accuracy improvement in out-of-distribution datasets.

2 Related Work

2.1 Visual Representation Learning

Representation learning, which focuses on deriving robust feature embeddings from raw data, is widely explored in the field of computer vision Chen et al. (2020); Grill et al. (2020); Chen & He (2021); Radford et al. (2021); Oquab et al. (2023). With techniques like contrastive loss van den Oord et al. (2018) and self-distillation Caron et al. (2021), representation learning methods manage to eliminate the reliance on labels and demonstrate strong generalization.

Generally, the extracted representations are then fed to a supervised predictor for downstream tasks. In this paper, we attempt to use a predictor that shares the same architecture but employs a different objective. Specifically, we replace the standard cross-entropy loss with a novel reinforcement-learning-inspired objective function to optimize representation models.

Moreover, DINOv2 Oquab et al. (2023), which presents strong capability to extract features, is adopted as the pre-trained base model, as mentioned in Section 1.

2.2 Downstream Tasks of Visual Representation Learning

Computer vision, serving as a foundational interface for intelligent systems to interpret visual data, plays a crucial role in artificial intelligence. Therefore, the semantic understanding extracted through the representation models is applicable in various types of downstream tasks, such as image and video classification Karpathy et al. (2014), semantic segmentation Yang et al. (2018), and object detection He et al. (2017); Carion et al. (2020).

In this paper, we investigate the performance of our proposed method in image classification and semantic segmentation. Generally, the pre-trained models are fine-tuned with a fully connected layer - based neural network and a Softmax layer for classification. While for segmentation, we follow the setting of DINOv2, projecting and upsampling the patch tokens to obtain a pixel-level probabilistic distribution.

2.3 Reinforcement Learning Methods for Post-Training

Reinforcement learning methods are widely used in the post-training of LLMs to align the model with its users Ouyang et al. (2022); Lee et al. (2024). Inspired by the idea of TRPO Schulman et al. (2015), Proximal Policy Optimization (PPO) Schulman et al. (2017), which can be implemented much simpler and is more general, is proposed to post-train LLMs based on reinforcement learning. Another policy that is widely used in post-training LLMs, Direct Preference Optimization (DPO), directly optimizes the model with a classification objective to obtain the best policy that satisfy the preferences, thus avoiding an explicit reward function Rafailov et al. (2023).

In recent years, Group Relative Policy Optimization, namely GRPO, was proposed as a reinforcement learning method to post-train DeepSeek-R1 DeepSeek-AI (2025), and has demonstrated exceptional performance in practical applications. Compared with PPO, GRPO samples multiple outputs for each input question, and computes the group-wise advantages to optimize the policy model. Without the generalized advantage estimation in PPO, GRPO is able to reduce computational overhead during the training. Moreover, similar to GRPO, Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO), an open-source model that dynamically samples output prompts, also achieve success in post-training LLMs Yu et al. (2025).

Several recent works, such as VLM-R1 Shen et al. (2025) and TinyLLaVA-Video-R1 Zhang et al. (2025), have shown the advantages of GRPO on the fine-tuning of multimodal LLMs, indicating that GRPO is a more effective fine-tuning paradigm with low computational cost. In this paper, we introduce GRPO-RM, extending GRPO to the post-training of representation model. This generalization faces some core challenges stemming from architectural differences. To be specific, the outputs of LLMs are different from those of representation models formally. Moreover, the advantage computation of GRPO relies on token-level reasoning traces, which lack equivalents in visual feature extractors.

2.4 Downstream Tasks Based on DINO

As one of the state-of-the-art representation learning methods, DINO is widely employed for all kinds of downstream tasks, which also indicates its generalizability. Grounding DINO Liu et al. (2024), which combine DINO with grounded pre-training, aims to detect arbitrary objects, i.e. open-set object detection. dino.txt Jose et al. (2025) employs DINO to align self-supervised visual features with language, and achieves great success in zero-shot classification and open-vocabulary semantic segmentation. Moreover, DINOv2 are also extended to medical image classification and segmentation area, and performs well in public datasets Huang et al. (2024); Ayzenberg et al. (2024). Generally, DINO is extended to a certain task of Computer Vision (CV), and is verified to be effective and generalizable. In this paper, we aim to post-train DINOv2 in two representative CV tasks, showing the advantages of our method over the standard fine-tuning method.

3 The Proposed Method

In this section, we first introduce the GRPO adaptation for the post-training of representation models. Then, with a specially designed reward function, the GRPO-RM method is elaborated. The framework of our method is illustrated in Fig. 1.

3.1 Preliminaries

For a question q and a group of corresponding outputs $\{o_1, o_2, \cdots, o_G\}$, the GRPO method with hyper-parameters ϵ and β aims to maximize the objective as follows:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)] \frac{1}{G} \sum_{i=1}^{G} \\ [\min(\frac{\pi_\theta(O|q)}{\pi_{\theta_{old}}(O|q)} A_i, \text{clip}(\frac{\pi_\theta(O|q)}{\pi_{\theta_{old}}(O|q)}, 1-\varepsilon, 1+\varepsilon) A_i) - \beta \mathbb{D}_{KL}(\pi_\theta || \pi_{ref})], \tag{1}$$

where π_θ, $\pi_{\theta_{old}}$, and π_{ref} represent the policy model, the old policy model, and the reference model, respectively, and A_i indicates the advantages calculated from the rewards. As mentioned in Section 2.3, to generalize the GRPO method to post-training of representation models, there are some issues to be addressed:

- $\mathcal{I}1$ Unlike LLMs that support probabilistic sampling of diverse outputs, representation models produce deterministic embeddings. Consequently, the explicit probability distribution (e.g. $\pi_\theta(o_i|q)$ and $\pi_{\theta_{old}}(o_i|q)$) in the objective is not available.
- $\mathcal{I}2$ The reward functions of GRPO are designed for LLMs. It aims to improve accuracy as well as enforce the model to put a thinking process between "<think>" and "</think>"

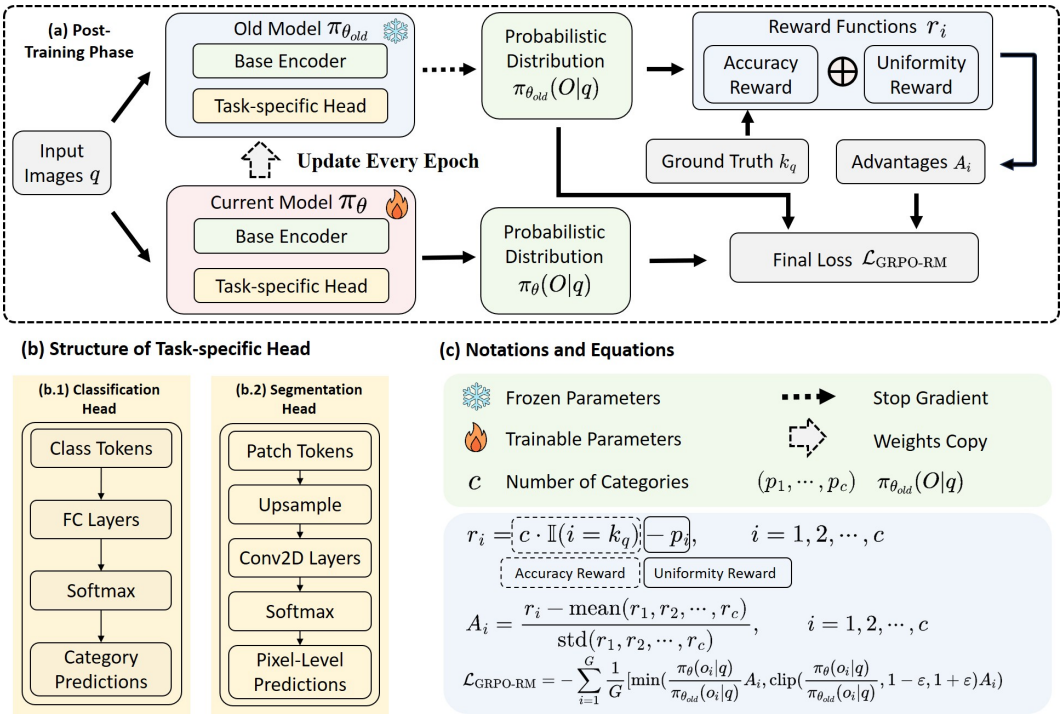


Figure 1: Framework of GRPO-RM: (a) The post-training architecture of GRPO-RM comprises a base encoder and a task-invariant head. At each epoch, the parameters of the old model (θ_{old}) are updated and incorporated into the loss computation without gradient propagation. Advantages are subsequently computed using ground-truth annotations and the probabilistic distributions generated by the old model. Finally, the loss is derived from the opposite number of Eq. (1) with hyper-parameter β fixed to 0. (b) Specifically, the network of task-specific heads and the tokens used for post-training vary in different tasks. For image classification, we simply feed the class tokens to a full-connected layer-based neural network. For semantic segmentation, the patch tokens are upsampled and projected to obtain a pixel-level prediction.

tags DeepSeek-AI (2025). However, this mechanism is incompatible with representation models.

$\mathcal{I}3$ No task-specific head is available to produce output distributions. Hence, the KL divergence term may not be computable.

$\mathcal{I}1$ and $\mathcal{I}2$ will be addressed in Section 3.2 and Section 3.3, respectively. For $\mathcal{I}3$, we remove the KL divergence term from the objective, which is equivalent to setting hyperparameter β to 0. Further discussion is available in Appendix C.2.

3.2 Outputs Sampling for Single Input

As mentioned in $\mathcal{I}1$, an outputs group with explicit probabilistic distribution is needed in the objectives. Moreover, the outputs group $\{o_1, o_2, \cdots, o_G\}$ also directly informs the computation of advantages A_i, which is the key idea of GRPO. To encompass the two properties, the output-sampling scheme requires the model to generate different outputs with probabilistic distribution for a single input. Moreover, the outputs should also support the computation of rewards. For LLMs, the token sequences meet both needs well, while the outputs of representation models are not. Intuitively, alternatives for the "question" and various "responses" (i.e., token sequences) of LLMs are essential to generalize GRPO.

Considering that the post-training of representation models is relative to the downstream tasks, the task itself can serve as the questions, take image classification as an example. The input image can be treated as a question asking about its class, and the whole set of categories can be considered the

set of all possible responses. Thus, a simple Softmax layer yields the probabilistic distribution, and a definite 'response' set facilitates the estimation of advantages.

Inspired by this insight, our method fixes the "response" set and employs a Softmax layer to generate probabilistic distributions over candidates, thereby converting the open-ended Q&A into a choice selection task.

3.3 Advantages for Representation Learning

As mentioned in $\mathcal{I}2$, the advantages in GRPO is incompatible with representation learning models. To be specific, the advantages A_i in Eq. (1) is computed using the rewards $\{r_1, r_2, \cdots, r_G\}$ corresponding to the group of outputs:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}. \tag{2}$$

However, the reward functions of GRPO are limited to the post-training for LLMs. To adapt the rewards to representation models, the functions should be redesigned to accommodate their properties. Intuitively, we require that representations of similar inputs cluster closely in the embedding space, while the overall distribution maintains near-uniform coverage throughout the representation space, i.e., alignment and uniformity Wang & Isola (2020).

Inspired by this, we decompose the reward function into two components:

Accuracy Rewards
Similar to the original GRPO methodology, accuracy rewards are adopted, directly encouraging the model to generate correct results. As the possible output set is fixed, the accuracy reward can be obtained through the corresponding target of the input.

For example, for a dataset with c classes $\{C_1, C_2, \cdots, C_c\}$, given an input x_k of class C_k, a group of outputs that exactly stand for c classes will be sampled. The accuracy reward for each output can be formulated as

$$\begin{cases} r_{\text{acc}_i} = 0 & i \neq k, \\ r_{\text{acc}_i} = c & i = k. \end{cases} \tag{3}$$

It is worth noting that the value of r_{acc_k} is set to c, ensuring that $\mathbb{E}r_{\text{acc}_i}$ is equal to 1 for simplification of calculation.

Uniformity Rewards
Given input x_k, the sampled output group $\{o_1, o_2, \cdots, o_c\}$ is corresponding to the c classes of the dataset. Moreover, a probabilistic distribution $\{p_1, p_2, \cdots, p_c\}$ is obtained through a Softmax layer. The uniformity rewards are designed to suppress the probability of wrong outputs. Specifically, it can be formulated as

$$r_{\text{uni}_i} = -p_i, \qquad j \in \{1, 2, \cdots, c\}. \tag{4}$$

Intuitively, for an output that is higher possible to be generated, its uniformity reward will be lower. Thus, the model is encouraged to output a more uniform probabilistic distribution during the post-training.

Note that the reward of o_k, which is the correct output, is also negative to its corresponding probability, which may discourage the model to generate correct output. As a contrast, another uniformity reward function scheme is as follows:

$$\begin{cases} r'_{\text{uni}_i} = \dfrac{1 - p_k}{c - 1} - p_i & i \neq k, \\ r'_{\text{uni}_i} = p_i & i = k. \end{cases} \tag{5}$$

In this case, $\sum_{i=1}^{c} r'_{\text{uni}_i} = 1 - p_k - \sum_{i \neq k} p_i + p_k = -p_k < 1$, ensuring that the accuracy rewards are dominant in the total rewards. Furthermore, Eq. (5) encourages the uniformity of the probability of all wrong outputs and is adaptive to each output. However, there are some issues remaining:

- To realize the function in Eq. (5), more computations are needed because of its asymmetry of different indexes.

Algorithm 1 Pseudo code of GRPO-RM

Input: Dataset $\mathcal{D}$, batch size m, train epoch e, pretrained model f_{pre}.
Hyper-parameters: $\beta = 0, \varepsilon$

```
build the post-train model π_θ with f_pre and a projection head.
for k =1 to e do
    π_θold ← π_θ // copy old model
    for each sampled batch {x_i, y_i}_{i=1}^m do
        Generate probabilistic distribution: π_θold(y|x).
        Compute Rewards according to Eq. (3), Eq. (4), and Eq. (6) with π_θold(y|x) and y_i.
        Compute advantages according to Eq. (2).
        Compute loss L according to Eq. (1) with advantages, β, ε, π_θ, and π_θold.
        Update π_θ to maximize L.
    end for
end for
return π_θ
```

- In Eq. (4), the reward of $o_k \in (-1, 0)$, which is far smaller than the corresponding reward in Eq. (3). Hence, there is no need to specifically modify the reward of o_k.

Based on the above concerns, we employ Eq. (4) as the uniformity rewards in GRPO-RM. Detailed comparisons between Eq. (4) and Eq. (5) will be shown in Section 4.6.

After both components are determined, the final rewards is obtained through addition, i.e.,

$$r_i = r_{\text{acc}_i} + r_{\text{uni}_i}. \tag{6}$$

Additional Punishment for Segmentation
It is worth noting that in segmentation datasets, quite a lot of pixels belong to class 1, which stands for the background. To deal with the imbalanced problem, an additional punishment is applied to the class 1. For each rewards $\{r_1, r_2, \cdots, r_c\}$ obtained through Eq. (6), r_1 is replaced by $r_1 - c/2$, which is simple for calculation and effectively reduces the impact of the imbalance.

3.4 Post-Training Models with GRPO-RM

With all issues mentioned in Section 3.1 addressed, we propose Group Relative Policy Optimization for Representation Model (GRPO-RM), which aims to post-train representation models using reinforcement learning method.

To be specific, GRPO-RM employs the categories of dataset as the output group and the redesigned reward function to compute advantages. Then, the objective is computed through Eq. (1) with hyper-parameter β fixed to 0. The procedure of GRPO-RM is summarized in Algorithm 1. Note that the copied model $\pi_{\theta_{old}}$ does not propagate gradient during the training.

4 Experiments

In this section, we perform experiments to investigate the following questions:

- **Q1** Does GRPO-RM outperform the standard fine-tuning method?
- **Q2** What is the time and memory burden of GRPO-RM?
- **Q3** How do the uniformity reward and the KL diversity term influence the post-training results (as is mentioned in Section 3)?

We will discuss **Q1** in Section 4.2 and Section 4.3, and then **Q2** in Section 4.4. Due to space constraints, **Q3** along with other ablation studies are detailed in Appendix C.

4.1 Datasets

Eight benchmark vision datasets are utilized for experimental study, including five image classification datasets **CIFAR-10** Torralba et al. (2008), **CIFAR-100** Torralba et al. (2008), **STL-10**

Table 1: Accuracy (in percent ± standard deviation) on image classification datasets over 5 runs. The best result on each dataset is highlighted with **bold**.

Dataset	Method	SR	kNN
CIFAR-10	DINOv2	47.26	43.26
	Fine-tuning	94.55 ± 0.27	94.39 ± 0.29
	GRPO-RM	**95.78 ± 0.26**	**95.81 ± 0.30**
CIFAR-100	DINOv2	20.68	25.92
	Fine-tuning	68.87 ± 2.70	71.32 ± 1.85
	GRPO-RM	**74.41 ± 2.28**	**78.48 ± 0.92**
STL-10	DINOv2	15.71	37.57
	Fine-tuning	43.44 ± 6.89	**89.76 ± 3.50**
	GRPO-RM	**46.42 ± 5.86**	89.03 ± 4.43
Tiny-ImageNet	DINOv2	29.24	21.10
	Fine-tuning	58.67 ± 0.61	58.59 ± 1.25
	GRPO-RM	**65.96 ± 0.40**	**66.36 ± 0.32**
ImageNet	DINOv2	74.34	**72.61**
	Fine-tuning	74.48 ± 0.85	71.33 ± 0.45
	GRPO-RM	**76.21 ± 0.92**	72.16 ± 0.73

Table 2: Semantic segmentation results in terms of pixel accuracy, Intersection over Union (IoU), and mean IoU on COCO-stuff. The best result on each dataset is highlighted with **bold**.

Feature Type	Method	COCO-stuff		
		Pixel Acc	IoU	mIoU
Linear	DINOv2	63.43	46.45	30.79
	Fine-tuning (10%)	63.71	46.67	30.94
	Fine-tuning	64.08	47.72	31.67
	GRPO-RM (10%)	63.72	47.51	31.07
	GRPO-RM	**64.10**	**48.01**	**32.25**
Multi Scale	DINOv2	64.57	47.68	31.75
	Fine-tuning (10%)	64.87	47.98	32.77
	Fine-tuning	64.99	48.26	32.88
	GRPO-RM (10%)	64.94	48.08	32.22
	GRPO-RM	**65.02**	**48.53**	**33.06**

Table 3: Semantic segmentation results (in percent ± standard deviation) in terms of pixel accuracy, Intersection over Union (IoU), and mean IoU on PASCAL-VOC and ADE20k over 5 runs. The setting of feature types is consistent with the setting in Oquab et al. (2023). The best result on each dataset is highlighted with **bold**.

Dataset	Feature	Method	Pixel Acc	IoU	mIoU
Pascal VOC 2012	**Linear**	DINOv2	93.06	87.02	66.77
		Fine-tuning (10%)	93.46 ± 0.01	88.03 ± 0.07	69.62 ± 0.18
		Fint-tuning	93.66 ± 0.04	88.24 ± 0.02	69.78 ± 0.11
		GRPO-RM (10%)	93.67 ± 0.02	88.57 ± 0.03	70.38 ± 0.08
		GRPO-RM	**93.98 ± 0.02**	**88.64 ± 0.03**	**70.44 ± 0.12**
	Multi Scale	DINOv2	94.04	88.75	70.20
		Fine-tuning (10%)	94.33 ± 0.01	89.06 ± 0.05	70.98 ± 0.04
		Fine-tuning	94.34 ± 0.02	89.19 ± 0.04	71.45 ± 0.05
		GRPO-RM (10%)	**94.38 ± 0.01**	89.35 ± 0.01	71.66 ± 0.06
		GRPO-RM	**94.38 ± 0.01**	**89.38 ± 0.01**	**71.71 ± 0.06**
ADE20k	**Linear**	DINOv2	**70.19**	54.07	31.52
		Fine-tuning (10%)	69.72 ± 0.02	54.02 ± 0.01	32.03 ± 0.04
		Fine-tuning	69.74 ± 0.02	54.44 ± 0.02	32.23 ± 0.03
		GRPO-RM (10%)	69.74 ± 0.01	54.55 ± 0.01	32.19 ± 0.05
		GRPO-RM	69.76 ± 0.01	**55.34 ± 0.02**	**32.62 ± 0.01**
	Multi Scale	DINOv2	71.02	55.07	32.26
		Fine-tuning (10%)	71.18 ± 0.02	55.32 ± 0.01	32.19 ± 0.05
		Fine-tuning	71.29 ± 0.02	55.80 ± 0.02	33.19 ± 0.03
		GRPO-RM (10%)	71.26 ± 0.01	55.35 ± 0.01	33.12 ± 0.02
		GRPO-RM	**71.52 ± 0.01**	**56.01 ± 0.03**	**33.58 ± 0.02**

Coates et al. (2011), **Tiny-ImageNet** Russakovsky et al. (2015), and **ImageNet-1k** Russakovsky et al. (2015) as well as three semantic segmentation datasets **Pascal VOC** Everingham et al. (2010), **ADE20k** Zhou et al. (2017), and **COCO-stuff** Caesar et al. (2018). These datasets cover a wide range of scales and category numbers. The classification datasets span 10 to 1,000 classes, while the segmentation datasets span 21 to 171 classes. The details of the datasets are summarized in Appendix A. Note that for classification datasets, ImageNet-1k is treated as the in-distribution benchmark, and the remaining datasets serve as out-of-distribution benchmarks.

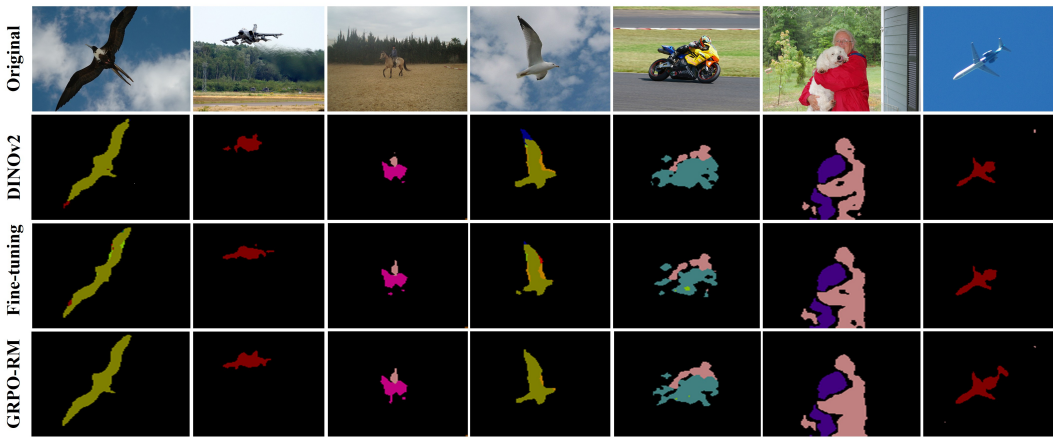


Figure 2: Visualization of DINOv2, Fine-tuning, and GRPO-RM on PASCAL-VOC. As shown in the figure, GRPO-RM has more advantages in segmenting fine details. For example, in the last column, GRPO-RM more completely segments the aircraft tail, whereas other methods miss this region. In images with multiple objects, GRPO-RM also delineates the boundaries between different objects more clearly.

4.2 Performance Analysis

To evaluate the effectiveness of GRPO-RM, we conduct a performance analysis on two representative downstream tasks, image classification and semantic segmentation. We also visualize the semantic segmentation results to provide an intuitive comparison between different methods.

Image Classification
For classification accuracy, we measure the performance through Softmax Regression and kNN accuracy with $k = 5$. The classification accuracy is reported in Table 1. Our method achieves an average 3.754% Softmax Regression (SR) improvement and 3.29% kNN improvement than standard fine-tuning method (denoted by "Fine-tuning" in table), which is more significant in out-of-distribution datasets (i.e., the datasets expect for ImageNet), with an average 4.26% SR improvement and 3.905% kNN improvement. For example, in Tiny-ImageNet, GRPO-RM outperforms the standard fine-tuning method by 7.29% in SR and 7.77% in kNN. Moreover, GRPO-RM also has a lower standard deviation in most datasets, demonstrating the stability of our method.

Semantic Segmentation
For segmentation datasets, we measure the performance through three metrics, the pixel accuracy, Intersection over Union (IoU), and mean IoU. As both approaches take a long time, we conduct further experiments that use a random 10% of the dataset during the training for comparison, corresponding to a quick and small-scale post-training. The results of the standard fine-tuning method and GPRO-RM are reported in Table 3 and Table 2. As a result, our method performs better on all datasets. Notably, even with only 10% of the training data, GRPO-RM achieves performance comparable to the standard fine-tuning method trained on the full dataset.

Visualization
To further demonstrate the effectiveness of GRPO-RM in semantic segmentation tasks, we visualize the segmentation results of DINOv2, standard fine-tuning, and GRPO-RM in Fig. 2. As shown in Figure 2, GRPO-RM exhibits greater advantages in fine-grained segmentation. First, GRPO-RM more accurately segments the terminal structures of objects, such as the aircraft tail in the second and seventh columns, whereas other methods often miss or incompletely segment them. Second, GRPO-RM better delineates and classifies object boundaries, such as the bird wings in the first and fourth columns, which are more clearly and completely outlined. In addition, for images containing multiple objects, GRPO-RM also distinguishes the boundaries between different objects more clearly.

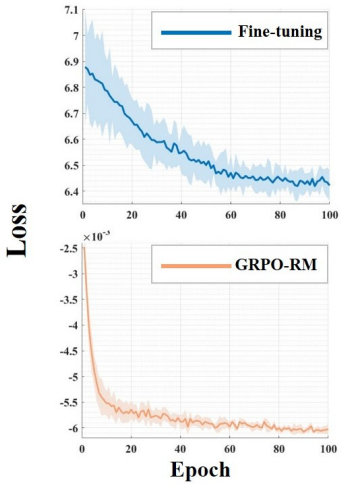


Figure 3: Training loss curves for GRPO-RM versus baseline on ImageNet (in-distribution). It can be easily derived from the figure that GRPO-RM converge much faster than standard fine-tuning method.

Table 4: Comparison of computational costs in different datasets. For classification datasets, the batch size is set to 1024, while for segmentation datasets, the batch size is set to 256. Moreover, the patch tokens are upsampled to 128×128. All experiments are run on 4 H100 vGPUs with 80GB memory.

Dataset	Method	Time	FLOPs	Memory
STL10	Fine-tuning	0.38s	8.65e7	14.30G
	GRPO-RM	0.44s	1.73e8	14.73G
Tiny-ImageNet	Fine-tuning	8.62s	8.67e7	14.30G
	GRPO-RM	9.63s	1.73e8	14.73G
ImageNet	Fine-tuning	223.67s	8.75e7	31.02G
	GRPO-RM	272.72s	1.75e8	31.33G
Pascal VOC	Fine-tuning	1.75s	8.61e7	115.55G
	GRPO-RM	1.86s	1.72e8	112.99G
ADE20k	Fine-tuning	25.43s	8.63e7	116.23G
	GRPO-RM	29.72s	1.73e8	134.26G
COCO-stuff	Fine-tuning	144.14s	8.64e7	117.61G
	GRPO-RM	170.94s	1.73e8	137.80G

4.3 Convergence

Considering that the reinforcement learning method is never used to post-train a representation learning model, it is essential to verify whether the proposed GRPO-RM can converge through training. To be specific, we investigate the convergence rate of GRPO-RM through the loss curves on different datasets, and draw the corresponding curve of the standard fine-tuning method as a comparison. The results are shown in Fig. 3. It is easy to find that GRPO-RM exhibits rapid loss reduction rate during post-training, achieving near-convergence within 20 epochs. For the standard fine-tuning method, the loss keeps decreasing in terms of overall trends until 100 epochs are finished. More results can be found in Appendix D.

4.4 Computational Burden

The computational burden of GRPO-RM is shown in Table 4. Three metrics are adopted to evaluate the costs, including training time for each epoch, floating point operations (FLOPs), and GPU memory required. Note that although GRPO-RM requires higher FLOPs, roughly twice those of standard fine-tuning, its runtime overhead and space burden is modest because a portion of the computation does not involve gradient propagation. This relatively small extra burden is acceptable given its accuracy improvement and faster convergence.

5 Conclusion

In this paper, we propose a reinforcement post-training framework, namely GRPO-RM, that adapts Group Relative Policy Optimization to representation models, which is the first reinforcement learning method for post-training of the visual representation models, to our best knowledge. Inspired by the success of GRPO in post-training LLMs, we reconfigure its objective for visual representations. To be specific, we regard the predictions with probabilities as the output group and design reward functions that align with the properties of representation learning for them, encouraging the model when the probability of correct predictions is high while discouraging it for the opposite. Extensive experiments conducted on various datasets with different tasks validate the effectiveness of our method over standard post-training methods. Although the computational burden experiment shows that GRPO-RM requires more training time and memory, GRPO-RM display a better performance

in accuracy and a much faster convergence rate, which is an acceptable trade-off. We also validate the effectiveness of our designed reward functions through the ablation study.

6 AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks.

7 REPRODUCIBILITY STATEMENT

The relevant training parameter settings are provided in Appendix B, including network architecture, training hyper-parameters, and evaluation settings. The code for this paper has been submitted as supplementary material.

REFERENCES

Lev Ayzenberg, Raja Giryes, and Hayit Greenspan. Dinov2 based self supervised learning for few shot medical image segmentation. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 215–223. JMLR.org, 2011.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8885–8894, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9726–9735, 2020.

Yuning Huang, Jingchen Zou, Lanxi Meng, Xin Yue, Qing Zhao, Jianqiang Li, Changwei Song, Gabriel Jimenez, Shaowu Li, and Guanghui Fu. Comparative analysis of imagenet pre-trained deep learning models and dinov2 in medical imaging classification. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 297–305. IEEE, 2024.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24905–24916, 2025.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. RLAIF: Scaling reinforcement learning from human feedback with AI feedback, 2024.

Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)*, pp. 163–168. IEEE, 2011.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11557–11568, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11): 1958–1970, 2008.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3684–3692, 2018.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 548–557, 2019.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Tinyllava-video-r1: Towards smaller lmms for video reasoning. *arXiv preprint arXiv:2504.09641*, 2025.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Table 5: Statistics of datasets used in experiments. "Classification" stands for image classification, and "Segmentation" stands for semantic segmentation.

Dataset	Type	Size	Classes
CIFAR-10	Classification	60,000	10
CIFAR0-100	Classification	60,000	100
STL-10	Classification	113,000	10
Tiny-ImageNet	Classification	100,000	200
ImageNet-1k	Classification	1,331,167	1,000
Pascal VOC 2012	Segmentation	1,464	21
ADE20k	Segmentation	20,210	150
COCO-stuff	Segmentation	118,287	171

A Datasets

Eight benchmark vision datasets are utilized for experimental study. The details of the datasets are summarized in Table 5.

B Experiments Settings

The experiments are carried out on 4 NVIDIA A100 vGPUs with 40 GB memory, 2 NVIDIA H200 vGPUs with 140 GB memory, or 4 NVIDIA H100 vGPUs with 80GB memory.

B.1 Backbone

Due to the limitation of GPU, all experiments are conducted with a pretrained ViT-S/14 DINOv2 model Oquab et al. (2023). For image classification datasets, a two-layer fully connected neural network is adopted as the projection head. For semantic segmentation datasets, we resize the input image size to 448×448 and obtain 32×32 patch tokens. The tokens are then upsampled to 128×128 as the segmentation map for the segmentation datasets due to the limitation of GPU.

B.2 Baselines

We fine-tune the DINOv2 model in a standard way for comparison. The settings of networks align with those of GRPO-RM. Moreover, we also conduct experiments on frozen features of the backbone once to directly show the improvement.

B.3 Hyper-parameters

Unless otherwise specified, all experiments adopt the default hyper-parameters as follows. For GRPO-RM, the hyper-parameter ε is set to 0.2. For experiments on image classification datasets, the dimension of the hidden layer is fixed to 256, and the batch size is 1024 for each dataset. For experiments on semantic segmentation datasets, the hidden dimension of Pascal VOC is 64 as there are only 21 classes, for the rest datasets, the hidden dimension is also 256. For all semantic segmentation datasets, the batch size is set to 256.

B.4 Training Settings

For all datasets, an Adam optimizer Kingma & Ba (2017) is employed to train the model. The weight decay factor of the optimizer is fixed to 0.0. For the learning rate factor in classification tasks, it is initially set to $10^{-3} \times m/256$, where m denotes the batch size. During the training, the learning rate factor gradually decreases to $10^{-5} \times m/256$. As for segmentation tasks, the learning rate starts with $10^{-5} \times m/256$ and ends with $10^{-7} \times m/256$ The DINOv2 model is trained for 100 epochs in each dataset.

Table 6: Comparison in terms of accuracy and training time with different uniformity reward function settings. "set to 0" means only an accuracy reward is applied, and the rest two items denotes the reward function with Eq. (5) and Eq. (4) as the uniformity rewards, respectively.

Dataset	Uniformity Reward	SR	Time
STL-10	set to 0	34.90	1.11s
	computed through Eq. (5)	36.27	3.56s
	Eq. (4) (GRPO-RM)	46.42	0.95s
Tiny-ImageNet	set to 0	62.16	9.40s
	computed through Eq. (5)	60.32	18.57s
	Eq. (4) (GRPO-RM)	65.96	10.37s
ImageNet	set to 0	77.04	232s
	computed through Eq. (5)	74.70	251.3s
	Eq. (4) (GRPO-RM)	76.21	238.2s
Pascal VOC 2012	set to 0	93.61	138.27s
	computed through Eq. (5)	-	157min
	Eq. (4) (GRPO-RM)	94.67	174.29s

B.5 Evaluation Settings

After post-training, we employ the frozen features of the trained backbone to train a randomly initialized head with an Adam optimizer Kingma & Ba (2017) for 50 epochs. The learning rate and weight decay factor of the optimizer is set in the same way as the training stage.

C Ablation Study

C.1 Uniformity Reward Setting

As mentioned in Section 3.3, further experiments are conducted to compare the effect while using Eq. (4) or Eq. (5) as uniformity reward functions. We also employ the accuracy rewards as the only reward function to compute advantages, as another comparison. The different uniformity settings are denoted as "set to 0", "computed through Eq. (5)", and "Eq. (4) (GRPO-RM)" in Table 6, respectively.

As a result, when Eq. (5) is employed as the uniformity reward function, the method shows relatively poor performance in both SR accuracy and training time. The reason might be that the punishment in Eq. (5) of wrong output is not strong enough, and may have an encouraging effect in some occasions. The disadvantages of Eq. (5) is more significant in segmentation datasets, as the formula requires much more computation, resulting in a much longer training time, which is over 50 times of GRPO-RM.

On the other hand, compared with the method that only employs accuracy reward, GRPO-RM requires an average 6.16% extra training time while exhibits stronger performance for an average 3.88% SR accuracy improvement, which demonstrates an optimal efficiency-performance tradeoff. Moreover, the results also validate the effectiveness of the uniformity rewards, especially for out-of-distribution datasets like STL-10 and Tiny-ImageNet, achieving an average 7.66% SR accuracy improvement.

C.2 KL Divergence Term

In Section 3.1, we set $\beta = 0$ to eliminate the KL term and thus address $\mathcal{I}3$. In this section, we add the KL term to GRPO-RM. Since the reference model lacks a task-specific head, this term cannot be computed based on the output distribution as in standard GRPO. Instead, it is obtained by comparing the representations of the reference model π_{ref} and the current model π_θ. We compare this variant with other fine-tuning approaches, and the results are listed in Table 7.

Table 7: Accuracy (in percent ± standard deviation) on image classification datasets over 5 runs, with ViT as the backbone.. The best result on each dataset is highlighted with **bold**.

Dataset	Method	SR	kNN
CIFAR-10	Fine-tuning	94.55 ± 0.27	94.39 ± 0.29
	GRPO-RM w/ KL term	92.50 ± 2.06	93.23 ± 1.03
	GRPO-RM w/o KL term	**95.78 ± 0.26**	**95.81 ± 0.30**
CIFAR-100	Fine-tuning	68.87 ± 2.70	71.32 ± 1.85
	GRPO-RM w/ KL term	55.06 ± 3.03	68.59 ± 1.83
	GRPO-RM w/o KL term	**74.41 ± 2.28**	**78.48 ± 0.92**
STL-10	Fine-tuning	43.44 ± 6.89	**89.76 ± 3.50**
	GRPO-RM w/ KL term	30.25 ± 2.56	85.07 ± 0.75
	GRPO-RM w/o KL term	**46.42 ± 5.86**	89.03 ± 4.43
Tiny-ImageNet	Fine-tuning	58.67 ± 0.61	58.59 ± 1.25
	GRPO-RM w/ KL term	54.17 ± 1.65	58.52 ± 0.98
	GRPO-RM w/o KL term	**65.96 ± 0.40**	**66.36 ± 0.32**
ImageNet	Fine-tuning	74.48 ± 0.85	71.33 ± 0.45
	GRPO-RM w/ KL term	74.68 ± 0.09	**75.68 ± 0.51**
	GRPO-RM w/o KL term	**76.21 ± 0.92**	72.16 ± 0.73

Table 8: Accuracy (in percent ± standard deviation) on image classification datasets over 5 runs, with ResNet50 as the backbone. The best result on each dataset is highlighted with **bold**.

Dataset	Method	SR	kNN
CIFAR-10	pretrained	70.98	81.33
	Fine-tuning	90.93 ± 0.61	88.03 ± 2.02
	GRPO-RM w/ KL term	89.26 ± 1.43	88.30 ± 0.56
	GRPO-RM w/o KL term	**91.12 ± 0.22**	**89.77 ± 1.06**
CIFAR-100	pretrained	29.15	34.05
	Fine-tuning	**53.82 ± 0.34**	55.73 ± 0.54
	GRPO-RM w/ KL term	52.80 ± 0.04	55.13 ± 0.09
	GRPO-RM w/o KL term	53.54 ± 0.25	**55.76 ± 0.11**
STL-10	pretrained	67.35	74.80
	Fine-tuning	92.49 ± 0.48	92.01 ± 0.23
	GRPO-RM w/ KL term	92.62 ± 0.18	91.82 ± 0.17
	GRPO-RM w/o KL term	**92.65 ± 0.21**	**92.14 ± 0.10**
Tiny-ImageNet	pretrained	**55.63**	**32.84**
	Fine-tuning	49.76 ± 0.23	29.98 ± 0.46
	GRPO-RM w/ KL term	49.92 ± 0.04	29.78 ± 0.13
	GRPO-RM w/o KL term	50.08 ± 0.08	30.52 ± 0.39
ImageNet	pretrained	74.56	**69.02**
	Fine-tuning	74.22 ± 0.13	66.96 ± 0.13
	GRPO-RM w/ KL term	**74.67 ± 0.18**	67.43 ± 0.09
	GRPO-RM w/o KL term	74.42 ± 0.03	67.19 ± 0.10

The performance of GRPO-RM with the KL term degrades on most datasets, which may be attributed to the specificity of our reward design. Since the rewards are directly derived from ground-truth labels, the additional KL constraint likely hinders convergence. This conjecture is further supported by the observation that better performance on the in-distribution dataset (i.e., ImageNet) is achieved when the KL term is included.

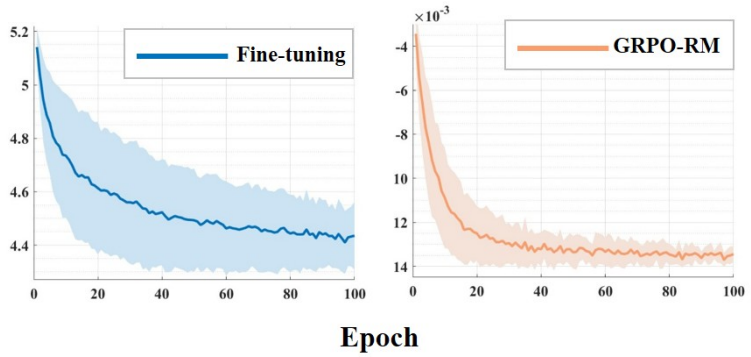


Figure 4: Training loss curves for GRPO-RM versus baseline on Tiny-ImageNet (out-of-distribution). It can be easily derived from the figure that GRPO-RM can also converge much faster than standard fine-tuning method on out-of-distribution dataset.

Table 9: Comparison of GRPO-RM and standard fine-tuning method. The running time is obtained by multiplying the per-epoch training time by the number of training epochs.

Dataset	Method	epoch	Time	SR Accuracy
Tiny-ImageNet	Fine-tuning	100	14.37min	58.67 ± 0.61
	GRPO-RM	50	8.03min	$\mathbf{63.88 \pm 1.08}$
ImageNet	Fine-tuning	100	372.78min	74.48 ± 0.85
	GRPO-RM	50	227.27min	$\mathbf{75.26 \pm 0.94}$

C.3 Other Backbone

To evaluate the generalization of GRPO-RM across different backbones, we apply it to fine-tuning with ResNet50He et al. (2016). We adopt the pretrained ResNet50_Weights.IMAGENET1K_V1 from torchvision, fine-tune the model on various datasets, and present the accuracy results in Table 8. Regarding hyper-parameter settings, we use a training learning rate of $10^{-4} \times m/256$ for CIFAR-100 and Tiny-ImageNet. All other hyper-parameters follow the default settings.

For the ResNet50 backbone, the relative accuracy is largely similar to that of the ViT backbone. GRPO-RM still holds a relative advantage, whereas GRPO-RM with the KL term only shows some advantage on in-distribution datasets. Moreover, the performance gain of GRPO-RM is relatively limited, suggesting that its compatibility with ResNet may be suboptimal.

D Efficiency Validation

We have preliminarily verified the convergence speed of GRPO-RM in Section 4.3. In this section, We first further compare GRPO-RM with standard fine-tuning on the out-of-distribution dataset Tiny-ImageNet, as shown in Fig. 4. It can be derived that GRPO-RM also shows a significant advantage in convergence speed on out-of-distribution data. The figure also shows that GRPO-RM is more stable during training.

To further explore the training efficiency of GRPO-RM, we also compare GRPO-RM under a shorter runtime (trained for 50 epochs) with the standard fine-tuning method, and the results are shown in Table 9.