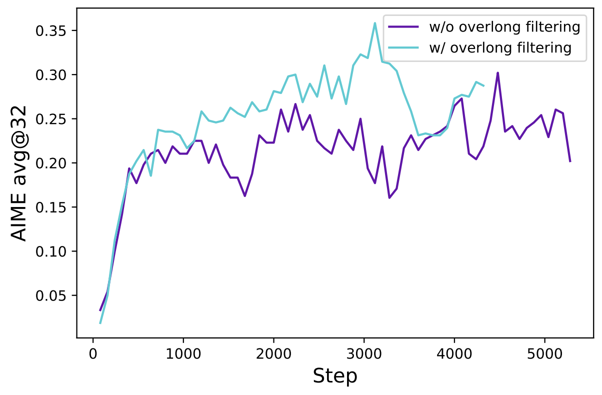
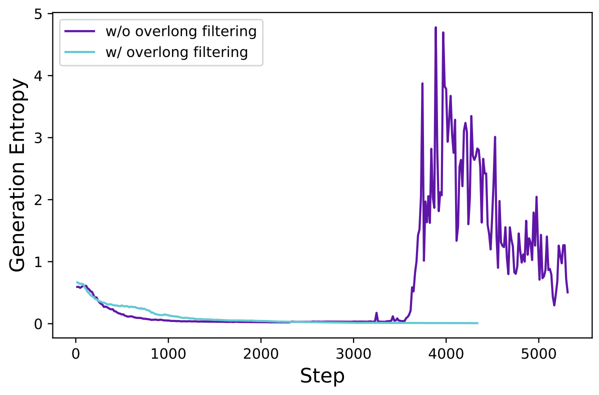


(a) Performance on AIME.



(b) Entropy of actor model.

Figure 5 The accuracy of the actor model on AIME and the entropy of its generation probabilities, both before and after applying **Overlong Reward Shaping** strategy.

By default, we assign a punitive reward to truncated samples. This approach may introduce noise into the training process, as a sound reasoning process can be penalized solely due to its excessive length. Such penalties can potentially confuse the model regarding the validity of its reasoning process.

To investigate the impact of this reward noise, we first apply an **Overlong Filtering** strategy which masks the loss of truncated samples. We find that this approach significantly stabilizes training and enhances performance, as demonstrated in Figure 5.

Algorithm 1 **DAPO**: **D**ecoupled Clip and **D**ynamic s**A**mpling **P**olicy **O**ptimization

```
Input initial policy model $\pi_\theta$; reawrd model $R$; task prompts $\mathcal{D}$; hyperparameters $\varepsilon_{\text{low}}, \varepsilon_{\text{high}}$
 1: for step = 1,...,M do
 2:     Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
 3:     Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
 4:     Sample $G$ outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)$ for each question $q \in \mathcal{D}_b$
 5:     Compute rewards $\{r_i\}_{i=1}^G$ for each sampled output $o_i$ by running $R$
 6:     Filter out $o_i$ and add the remaining to the dynamic sampling buffer (Dynamic Sampling Equation (11))
 7:     if buffer size $n_b < N$:
 8:         continue
 9:     For each $o_i$ in the buffer, compute $\hat{A}_{i,t}$ for the $t$-th token of $o_i$ (Equation (9))
10:     for iteration = 1, ..., $\mu$ do
11:         Update the policy model $\pi_\theta$ by maximizing the DAPO objective (Equation (8))
Output $\pi_\theta$
```

Furthermore, we propose **Soft Overlong Punishment** (Equation 13), a length-aware penalty mechanism designed to shape the reward for truncated samples. Specifically, when the response length exceeds the predefined maximum value, we define a punishment interval. Within this interval, the longer the response, the greater the punishment it receives. This penalty is added to the original rule-based correctness reward, thereby signaling to the model to avoid excessively long responses.

$$R_{\text{length}}(y) = \begin{cases} 0, & |y| \le L_{\max} - L_{\text{cache}} \\ \frac{(L_{\max} - L_{\text{cache}}) - |y|}{L_{\text{cache}}}, & L_{\max} - L_{\text{cache}} < |y| \le L_{\max} \\ -1, & L_{\max} < |y| \end{cases} \tag{13}$$

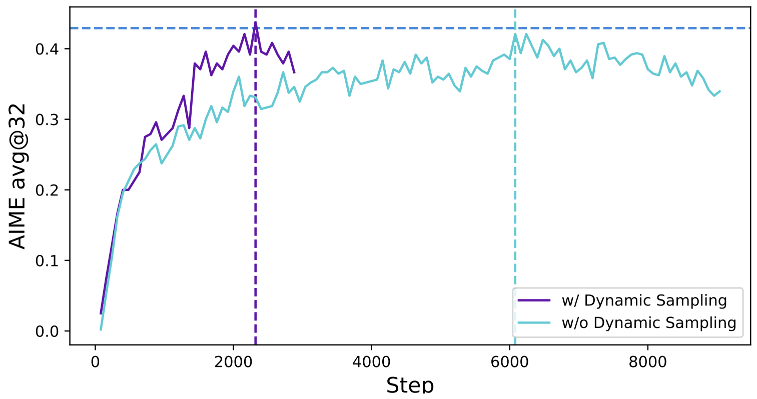


Figure 6 The training progress before and after applying dynamic sampling on a baseline setting.

3.5 Dataset Transformation

Our dataset is sourced from the web and official competition homepages through a combination of web scraping and manual annotation. The answers of math dataset typically come in a variety of formats, such as expression, formula and number, which makes it challenging to design comprehensive rules to parse them. To provide accurate reward signals using rules and minimize errors introduced by formula parsers, inspired by AIME, we select and transform the answers into integers, which are easy to parse. For example, if the original answer is expressed in the form of $\frac{a+\sqrt{b}}{c}$, we instruct the LLM to modify the question so that the expected answer becomes $a + b + c$. After selection and transformation, we obtained the **DAPO-Math-17K** dataset, which consists of 17K prompts, each paired with an integer as the answer.

4 Experiments

4.1 Training Details

In this work, we focus specifically on mathematical tasks to evaluate our algorithm, which can be readily transferred to other tasks. We adopt the verl framework [20] for training. We use naive GRPO [38] as our baseline algorithm and estimate advantages using group reward normalization.

For hyper-parameters, we utilize the AdamW [39] optimizer with a constant learning rate of 1×10^{-6}, incorporating a linear warm-up over 20 rollout steps. For rollout, the prompt batch size is 512 and we sample 16 responses for each prompt. For training, the mini-batch size is set to 512, i.e., 16 gradient updates for each rollout step. For **Overlong Reward Shaping**, we set the expected maximum length as 16,384 tokens and allocate additional 4,096 tokens as the soft punish cache. Therefore, the maximum number of tokens for generation is set to 20,480 tokens. As for the **Clip-Higher** mechanism, we set the clipping parameter ε_{low} to 0.2 and $\varepsilon_{\text{high}}$ to 0.28, which effectively balance the trade-off between exploration and exploitation. For evaluation on AIME, we repeat the evaluation set for 32 times and report avg@32 for results stability. The inference hyperparameters of evaluation are set to temperature 1.0 and topp 0.7.

4.2 Main Results

Experiments on AIME 2024 demonstrate that **DAPO** has successfully trained the Qwen-32B Base model into a powerful reasoning model, achieving performance superior to DeepSeek's experiments on Qwen2.5-32B using the R1 approach. In Figure 1, we observe a substantial improvement of performance on AIME 2024, with accuracy increasing from near 0% to 50%. Notably, this improvement is achieved with only 50% of the training steps required by DeepSeek-R1-Zero-Qwen-32B.

We analyze the contributions of each training technique in our methodology, as detailed in Table 1. The observed improvements demonstrate the effectiveness of these techniques in RL training, each contributing

Table 1 Main results of progressive techniques applied to **DAPO**

Model	**AIME24**$_{\text{avg@32}}$
DeepSeek-R1-Zero-Qwen-32B	47
Naive GRPO	30
+ Overlong Filtering	36
+ Clip-Higher	38
+ Soft Overlong Punishment	41
+ Token-level Loss	42
+ Dynamic Sampling (**DAPO**)	**50**

several accuracy points in AIME 2024. Notably, given the vanilla GRPO setting, only 30% accuracy can be reached by training from a Qwen2.5-32B base model.

For token-level loss, although it brings less performance improvement, we find it enhances training stability and makes the length increase more healthily.

When applying **Dynamic Sampling**, although more data needs to be sampled due to the filtering out of zero-gradient data, the overall training time is not significantly affected. As shown in Figure 6, although the number of sampling instances increases, the model's convergence time is even reduced, due to fewer training steps required.

4.3 Training Dynamics

Reinforcement learning on large language models is not only a cutting-edge research direction but also an intrinsically complex systems engineering challenge, characterized by the interdependence of its various subsystems. Modifications to any single subsystem can propagate through the system, leading to unforeseen consequences due to the intricate interplay among these components. Even seemingly minor changes in initial conditions, such as variations in data and hyperparameters, can amplify through iterative reinforcement learning processes, yielding substantial deviations in outcomes. This complexity often confronts researchers with a dilemma: even after meticulous analysis and well-founded expectations that a modification will enhance specific aspects of the training process, the actual results frequently diverge from the anticipated trajectory. Therefore, monitoring of key intermediate results during experimentation is essential for swiftly identifying the sources of discrepancies and, ultimately, for refining the system.

- **The Length of Generated Responses** is a metric closely related to training stability and performance, as shown in Figure 7a. The increase in length provides the model with a larger space for exploration, allowing more complex reasoning behaviors to be sampled and gradually reinforced through training. However, it is important to note that length does not always maintain a continuous upward trend during training. In some considerable periods, it can exhibit a trend of stagnation or even decline, which has also been demonstrated in [2]. We typically use length in conjunction with validation accuracy as indicators to assess whether an experiment is deteriorating.

- **The Dynamics of Reward** during training has always been one of the crucial monitoring indicators in reinforcement learning, as shown in Figure 7b. In the majority of our experiments, the trend of reward increase is relatively stable and does not fluctuate or decline significantly due to adjustments in experimental settings. This indicates that, given a reliable reward signal, language models can robustly fit the distribution of training set. However, we find that the final reward on the training set often exhibits little correlation with the accuracy on the validation set, which indicates overfitting to the training set.

- **The Entropy of the Actor Model and Generation Probability** are related to the model's exploration capability and are key metrics that we closely monitor in our experiments. Intuitively, the model's entropy needs to be maintained within an appropriate range. An excessively low entropy indicates that the probability distribution is overly sharp, leading to a loss of exploration capability. Conversely, an