

Supplementary Materials for MRO

A Limitations and Ethics Statement

A.1 Limitations

We propose the step-wise group reward optimization (SGRO) approach, which can effectively reduce the reward variance during the reward optimization. However, this introduces new hyperparameters to our MRO, including the size of denoising steps w per group and the size of groups chosen for optimizing DLMs in the rejection sampling and reinforcement learning. The selection of these hyperparameters requires careful consideration. To address this limitation, we conduct comprehensive ablation experiments, as shown in Figure 7. These experimental results provide valuable guidance for selecting the optimal hyperparameters. Additionally, in future work, we will consider designing techniques to determine these hyperparameters automatically.

A.2 Ethics Statement

This work does not involve any ethical concerns. All data collected for training our DLMs through rejection sampling and reinforcement learning are sourced exclusively from open-source materials. Additionally, this paper may reference certain case study content. However, these references are presented in an elliptical manner, and any potentially harmful content will not be explicitly presented.

B Proofs for Theoretical Results

In this section, we provide the proofs for two theoretical results. The first result shows that introducing a potential-based shaping in reward optimization leads to a higher reward variance. The second result demonstrates that using SGRO can mitigate this issue.

Property 1. *Under potential-based reward shaping, the expected reward remains the same, i.e., $\mathbb{E}[R(s, a)] = \mathbb{E}[\hat{R}(s, a)]$, but $\text{Var}(R(s, a))$ is higher than $\text{Var}(\hat{R}(s, a))$.*

Proof: Let’s denote the original reward as $\hat{R}(s_t, a_t)$ and the potential function as $\Phi(s_t)$. The potential-based reward shaping is given by

$$R(s_t, a_t) = \hat{R}(s_t, a_t) + \lambda\Phi(s_{t+1}) - \Phi(s_t) \quad (10)$$

where λ is the discount factor. The expected reward under the potential-based shaping is

$$\mathbb{E}[R(s_t, a_t)] = \mathbb{E}[\hat{R}(s_t, a_t) + \lambda\Phi(s_{t+1}) - \Phi(s_t)] \quad (11)$$

$$= \mathbb{E}[\hat{R}(s_t, a_t)] + \lambda\mathbb{E}[\Phi(s_{t+1})] - \mathbb{E}[\Phi(s_t)] \quad (12)$$

Since the expectation of the potential function at the next state $\mathbb{E}[\Phi(s_{t+1})]$ is equal to the expectation of the potential function at the current state $\mathbb{E}[\Phi(s_t)]$ (because the potential function is a function of the state and the state transitions are Markovian) [46], we have

$$\mathbb{E}[R(s_t, a_t)] = \mathbb{E}[\hat{R}(s_t, a_t)] + \lambda\mathbb{E}[\Phi(s_t)] - \mathbb{E}[\Phi(s_t)] \quad (13)$$

$$= \mathbb{E}[\hat{R}(s_t, a_t)] + (\lambda - 1)\mathbb{E}[\Phi(s_t)] \quad (14)$$

For the variance, we have

$$\text{Var}(R(s_t, a_t)) = \text{Var}(\hat{R}(s_t, a_t) + \lambda\Phi(s_{t+1}) - \Phi(s_t)) \quad (15)$$

$$\begin{aligned} &= \text{Var}(\hat{R}(s_t, a_t)) + \\ &\quad \text{Var}(\lambda\Phi(s_{t+1})) + \text{Var}(-\Phi(s_t)) + 2\text{Cov}(\hat{R}(s_t, a_t), \lambda\Phi(s_{t+1})) + \\ &\quad \underbrace{2\text{Cov}(\hat{R}(s_t, a_t), \Phi(s_t)) - 2\text{Cov}(\lambda\Phi(s_{t+1}), \Phi(s_t))}_{\text{additional variance introduced by the potential function terms}} \end{aligned} \quad (16)$$

Since $\Phi(s_t)$ and $\Phi(s_{t+1})$ are not independent, the covariance terms are not zero. However, the variance of the potential function terms could add to the variance of the original reward, leading to a

higher overall variance. In the general reinforcement learning, [47] has also proven that this result holds when $\Phi(s) > 2\hat{R}(s_t, a_t)$ for all a . Furthermore, [51] suggests that such a condition is indeed possible, even when using an optimal baseline reward technique. Therefore, we have

$$\text{Var}(R(s_t, a_t)) > \text{Var}(\hat{R}(s_t, a_t)) \quad (17)$$

This completes the proof of Property 1.

Property 2. *Using step-wise group reward optimization can reduce the reward variance, i.e., $\text{Var}(R(s, a)) < \text{Var}(R^{(w)}(s, a))$, where $R^{(w)}(\cdot)$ represents the reward after applying step-wise group reward optimization.*

Proof: SGRO groups w denoising steps together and provides a reshaped reward for each group. This means that the number of times the potential function is computed and added to the reward is reduced. Let's denote the reward for a group of w steps as

$$R^{(w)}(s_t, a_t) = \sum_{i=0}^{w-1} \hat{R}(s_{t+i}, a_{t+i}) + \lambda\Phi(s_{t+w}) - \Phi(s_t) \quad (18)$$

The variance of this group reward is

$$\text{Var}(R^{(w)}(s_t, a_t)) = \text{Var}\left(\sum_{i=0}^{w-1} \hat{R}(s_{t+i}, a_{t+i}) + \lambda\Phi(s_{t+w}) - \Phi(s_t)\right) \quad (19)$$

Following the derivation in Eq. 16, we obtain

$$\begin{aligned} \text{Var}(R^{(w)}(s_t, a_t)) &= \text{Var}(\hat{R}(s_t, a_t)) + \\ &\quad \lambda^2 \text{Var}(\Phi(s_{t+w})) + \text{Var}(\Phi(s_t)) + 2\lambda \text{Cov}(\hat{R}(s_t, a_t), \Phi(s_{t+w})) - \\ &\quad 2\text{Cov}(\hat{R}(s_t, a_t), \Phi(s_t)) - 2\lambda \text{Cov}(\Phi(s_{t+w}), \Phi(s_t)) \end{aligned} \quad (20)$$

Compared to Eq. 16, we can observe that due to the larger interval w between s_{t+w} and s_t , the correlation between $\Phi(s_{t+w})$ and $\Phi(s_t)$ is typically weaker than the correlation between $\Phi(s_{t+1})$ and $\Phi(s_t)$. Therefore, we have

$$|\text{Cov}(\Phi(s_{t+w}), \Phi(s_t))| < |\text{Cov}(\Phi(s_{t+1}), \Phi(s_t))| \quad (21)$$

Here, the negative term $-2\lambda \text{Cov}(\Phi(s_{t+w}), \Phi(s_t))$ in the variance expression of $R^{(w)}(s_t, a_t)$ has a larger absolute value compared to $-2\lambda \text{Cov}(\Phi(s_{t+1}), \Phi(s_t))$ in the variance expression of $R(s_t, a_t)$, as the correlation between $\Phi(s_{t+w})$ and $\Phi(s_t)$ is weaker. Therefore, we can obtain that this results in a smaller overall variance for $R^{(w)}(s_t, a_t)$:

$$\text{Var}(R^{(w)}(s_t, a_t)) < \text{Var}(R(s_t, a_t)) \quad (22)$$

This completes the proof of Property 2. Here, one problem may arise when only a single group provides a potential-based reward, as the reward that could guide the model toward better token correlation may be lost. Therefore, in this approach, we strike a balance between the reward signal with the reward variance for our MRO.

C Experiments

In this section, we provide additional experimental details and present the experimental results of MRO with reinforcement learning.

C.1 Experimental Details

Training Setups. For training LLaDA-s1, we used a pre-trained version of LLaDA. The learning rate was set to 2e-5. We trained this model on the s1 dataset for 3 epochs. In contrast to [26], we found that training for more epochs on the s1 dataset did not result in further performance improvements. For rejection sampling and reinforcement learning, we set the learning rate to 2e-6. During training, we performed model validation every 50 steps and selected the best model based on performance on the validation set as our final model. All of our experiments were conducted on eight A800 GPUs.

Model/Length	GSM8K			MATH500			GPQA			Countdown		Sudoku
	128	256	512	128	256	512	128	256	512	64	128	64
LLaDA	74.3	78.8	79.4	28.0	33.2	34.4	22.2	29.2	30.3	13.8	14.1	11.2
LLaDA-MRO-RS	76.9	79.6	82.6	31.0	34.2	36.2	26.3	32.1	34.3	21.4	22.0	17.2
LLaDA-MRO-RL	77.1	80.9	81.8	33.4	35.2	37.4	28.8	33.8	33.8	24.6	27.2	20.2
LLaDA-s1	70.7	76.4	78.7	25.0	26.0	28.8	21.7	25.3	28.3	10.2	12.4	8.4
LLaDA-s1-MRO-RS	73.3	77.7	80.1	27.8	28.0	29.0	24.2	29.3	32.8	17.3	17.8	15.2
LLaDA-s1-MRO-RL	71.8	75.9	78.0	26.6	27.2	29.4	23.7	28.3	32.3	17.1	16.2	13.8

Table 2: Results of the MRO in reinforcement learning. The suffixes “-RS” and “-RL” denote the results obtained using rejection sampling with $k = 4$ and reinforcement learning, respectively.

Training Datasets. For both rejection sampling and reinforcement learning, we utilized DeepScaleR [52] in conjunction with the 10k Countdown² and Sudoku³ datasets. These datasets were randomly shuffled to ensure a well-balanced data distribution.

Evaluation. For the evaluation, we focus on five reasoning tasks: GSM8K [53], MATH500 [54], GPQA [55], Countdown⁴, and Sudoku⁵. During testing, we set the sampling temperature to 0.25.

C.2 Reinforcement Learning

We explore the use of reinforcement learning to implement our MRO. The experimental setup and results are presented below.

Task Setup. We employed the REINFORCE [56] to perform this optimization. Specifically, during the optimization process, we used a temperature-based sampling to obtain a denoising sequence $\{\hat{r}_T, \hat{r}_{T-1}, \dots, \hat{r}_0\}$. Subsequently, we use a REINFORCE to train our DLMs through a cumulative reward. The loss function can be given by

$$\mathcal{L}_{\text{rl}}(\theta) = -\mathbb{E}_{o_t, p_0, \{\hat{r}_T, \hat{r}_{T-1}, \dots, \hat{r}_0\}} \left[\frac{1}{O_t} \sum_{i=1}^{L_{\hat{r}}} \mathbf{1}[\hat{r}_t^i = M] \log \Pr_{\theta}(\hat{r}_{t-1}^i | p_0, \hat{r}_t) \right] R_{\text{acc}} \quad (23)$$

where R_{acc} is the cumulative reward, computed as: $R_{\text{acc}} = \sum_{t=T}^0 R(s_t, a_t)$. Similarly, we apply the SGRO in reinforcement learning, where we group the sampled sequence, with each group containing w steps. For each group, we provide a reshaped reward. Additionally, during the optimization, similar to rejection sampling, instead of using all groups, we sample only a subset of the groups. Building upon this, we integrated quality evaluation scores into the reinforcement learning training process through a shaping mechanism. Specifically, we selected a group $\hat{r}_{i:i+w}$. Then, we computed the quality rewards R_i^q and R_{i+w}^q for the first and last steps of the group, respectively. The final quality reward for the group was determined by the difference between these two quality rewards. Although we implemented MRO in the DLM using REINFORCE, any reinforcement learning algorithm (such as PPO [37] or GRPO [57]) could be used to achieve MRO, as outlined in the modeling framework presented in Section 4.2.3. The primary focus of this work is to demonstrate the effectiveness of our MRO in enhancing token correlation, rather than to explore the performance of different reinforcement learning algorithms in the DLM training process. Therefore, we did not conduct tests of these algorithms one by one.

Results. We compare the performance of our DLMs trained via reinforcement learning to those trained with rejection sampling and the LLaDA instruction model. The results are listed in Table 2. From the results, we observe that with reinforcement learning, our MRO still achieves significant improvements across various benchmarks. However, when compared to rejection sampling, our reinforcement learning approach does not show a substantial advantage. We identify two main reasons for this observation. First, our rejection sampling approach has been enhanced compared to the original; it combines offline data construction with online sampling techniques. Second, our

²<https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-3to4>

³<https://huggingface.co/datasets/Ritvik19/Sudoku-Dataset>

⁴<https://github.com/HKUNLP/diffusion-vs-ar>

⁵https://github.com/dllm-reasoning/d1/blob/main/dataset/4x4_test_sudoku.csv

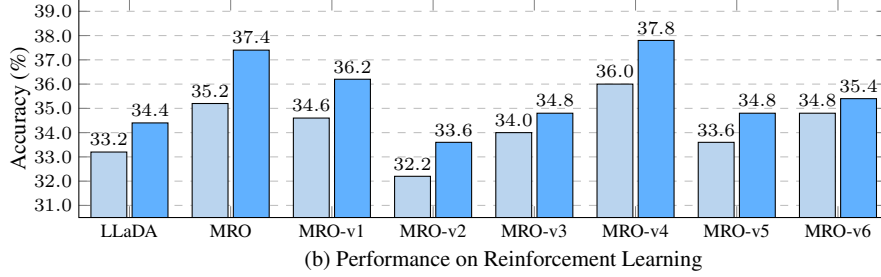
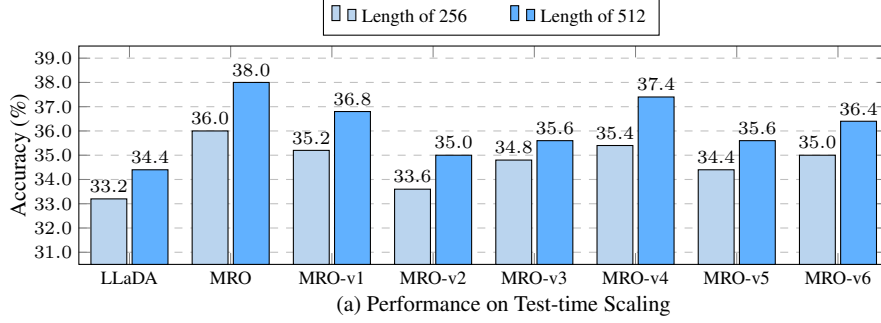


Figure 6: Performance comparison of different MRO variants on the MATH500 benchmark for both test-time scaling and reinforcement learning.

reinforcement learning approach is relatively basic and does not incorporate advanced modifications or improvements.

C.3 Ablation Study

In this subsection, we design several MRO variants to further describe the functionality of intra-sequence and inter-sequence rewards, as well as SGRO.

Reward Design. As shown in Table 3, we design six MRO variants to investigate the impacts of intra-sequence and inter-sequence rewards.

Reward\Variant	MRO-v1	MRO-v2	MRO-v3	MRO-v4	MRO-v5	MRO-v6
R_t^{iv}	✓			✓		✓
R_t^{ppi}		✓			✓	✓
R_0^q			✓	✓	✓	

Table 3: Description of MRO variants.

We conduct experiments on test-time scaling and reinforcement learning to evaluate the performance of these variants. As illustrated in Figure 3, the results reveal that achieving superior performance with a single reward combination is challenging. For example, MRO-v2 and MRO-v3 demonstrate relatively poor performance compared to other variants in both test-time scaling and reinforcement learning. Furthermore, the token verification reward proves to be highly effective. We can see that MRO-v2, MRO-v3, and MRO-v5 perform worse in comparison. However, when compared to LLaDA, it is clear that all of our reward designs are effective, except for MRO-v2 in reinforcement learning, which exhibits some performance degradation. Other variants, to varying degrees, lead to performance improvements. This further supports the correctness of our design approach, centered around enhancing token correlation through tailored reward strategies. Interestingly, results in reinforcement learning show that MRO-v4 outperforms MRO itself. This improvement could be attributed to sampling and the potential reward variance caused by the perplexity reward. Nevertheless, since other experiments have confirmed the usefulness of this reward, we chose not to discard it.

Step-wise Group Reward Optimization. We also conduct an ablation study on our SGRO. Specifically, we test the case where SGRO is not applied, meaning that each denoising step receives a

Model/Length	MATH500		GPQA	
	256	512	256	512
LLaDA-MRO	34.2	36.2	32.1	34.3
LLaDA-MRO w/o SGRO	32.8	35.4	31.3	33.3

Table 4: Ablation study of SGRO.

977 shaping reward in rejection sampling. As shown in Table 4, we find that SGRO is effective and helps
978 the MRO achieve better performance.

979 D Analysis

980 D.1 Comparison of MRO with Other Reasoning-Enhanced DLM Approaches

981 We compare MRO with other existing reasoning-
982 enhanced models and approaches. These include
983 DiffuLLaMA [34], which adapts the LLaMA
984 model for DLMs; EDLM [35], which introduces
985 an energy function to enhance sequence-level cor-
986 relation; Dream [24], which utilizes the Qwen-2.5-
987 7B model for initialization; and d1-LLaDA [26],
988 which trains the LLaDA-8B-Instruct model using
989 GRPO. The results are presented in Table 5. Note
990 that, except for EDLM, the results for the other
991 baselines are taken directly from the original pa-
992 pers. For EDLM, we replicate its autoregressive
993 energy function version in our codebase. First,
994 compared to DiffuLLaMA and Dream, our MRO
995 achieves competitive results. Moreover, we ob-
996 serve that although EDLM incorporates sequence-
997 level correlation (i.e., inter-sequence correlation
998 described in this paper), our MRO still outperforms it. We attribute this to the lack of consideration
999 for intra-sequence correlation in EDLM. However, we observe that our model performs slightly worse
1000 than the d1-LLaDA model. We argue that this comparison is not entirely fair, as d1-LLaDA is trained
1001 using a task-specific training set for GRPO. This potentially gives it an advantage by benefiting from
1002 task-specific training. To validate this, we also conduct reinforcement learning using a task-specific
1003 training set. More specifically, during the reinforcement learning training, we use the GSM8K⁶ and
1004 MATH500⁷ training sets to perform the MRO, respectively. We find that the results from this training
1005 approach surpass the performance of d1-LLaDA.

Model/Length	GSM8K		MATH500	
	256	512	256	512
DiffuLLaMA	63.1		-	-
Dream-7B-Instruct	77.2		-	-
EDLM	78.1	80.0	34.4	35.4
d1-LLaDA	81.1	82.1	38.6	40.2
LLaDA-MRO-RS	79.6	82.6	34.2	36.2
LLaDA-MRO-RL	80.9	81.8	35.2	37.4
LLaDA-MRO-TS	82.5	82.9	39.4	42.6

Table 5: Performance comparison of MRO with other reasoning-enhanced models and approaches. “-TS” indicates that we use task-specific training data to optimize the DLM in reinforcement learning.

1006 D.2 Scaling Training with Different Group Sizes

1007 We scale training in MRO by using different
1008 group sizes. Specifically, we keep the total
1009 number of training samples constant while
1010 testing various group sizes {1, 2, 4, 8, 16, 32,
1011 64} during both rejection sampling and re-
1012 inforcement learning. The results are shown
1013 in Figure 7. From the results, we can find
1014 that increasing the group size provides some
1015 benefits, but these benefits diminish after a
1016 group size of 2, with minimal improvements
1017 beyond that. Notably, performance becomes
1018 unstable after a group size of 16. Considering both performance and training costs, we choose a
1019 group size of 2 for our experiments.

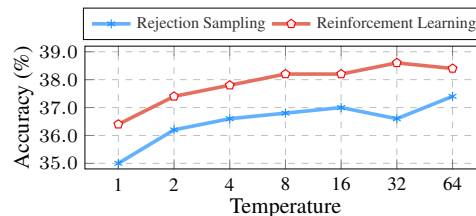


Figure 7: Performance of MRO with different temperature settings, evaluated on the MATH500 benchmark.

⁶<https://huggingface.co/datasets/openai/gsm8k>

⁷https://github.com/openai/prm800k/tree/main/prm800k/math_splits

Prompt	Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?
Answer	64
LLaDA	<p>To determine how much Kylar needs to pay for 16 glasses, we need to consider the cost of the first glass and the discounted of every second glass.</p> <ol style="list-style-type: none"> 1. The first glass costs \$5. 2. Every second glass costs 60% of the price price of the first glass, which is $16 \times 0.60 = 9.60$ dollars. <p>Let's calculate the total cost:</p> <ul style="list-style-type: none"> - The first glass costs \$5. - The second glass costs \$9.60. - The third glass costs \$5. - The fourth glass costs \$9.60. - . . . - The sixteenth glass costs \$5. <p>So, the total cost is: $5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 + 9.60 + 5 = 120$</p> <p>The answer is \$120.</p>
LLaDA + MRO	<p>To determine how much Kylar needs to pay for 16 glasses, we need to consider the cost of the first glass and the discounted of the second glass.</p> <ol style="list-style-type: none"> 1. The cost of the first glass is \$5. 2. The cost of the second glass is 60% of the first glass's price, which is $0.6 \times 5 = 3$ dollars. <p>Since Kylar wants to buy 16 glasses, he can pair them up follows:</p> <ul style="list-style-type: none"> - First glass: \$5 - Second glass: \$3 - Third glass: \$5 - Fourth glass: \$3 - . . . - Sixteenth glass: \$5 <p>Each pair of glasses consists of $5 + 3 = 8$.</p> <p>There are 8 pairs of glasses (since 16 divided by 2 is 8).</p> <p>So, the total cost is: $8 \text{ pairs} \times 8 \text{ per pair} = 64$</p> <p>Therefore, Kylar needs to pay \$64 for 16 glasses.</p> <p>The answer is \$64.</p>

Table 6: A case from the test-time scaling experiments. MRO provides accurate guidance for token correlation and helps identify a more precise CoT-style reasoning path. Note that we remove the “<think></think>” and “<answer></answer>” format tags to improve the readability of this case.

Prompt	Two sides of a triangle are each 8 units long. If the third side has a whole number length, what is the greatest possible perimeter, in units, for the triangle?
Answer	31
LLaDA	<p>x is a positive number, this condition is always satisfied</p> <p>From conditions 1 and 2, we get:</p> $16 > x$ $x < 0$ <p>Since x must be a whole number, the largest possible value for x is satisfies both conditions is $x = 7$.</p> <p>Now, we calculate the perimeter:</p> $\text{Perimeter} = 8 + 8 + 7 = 23$ <p>Thus, the greatest possible perimeter of the triangle is 23.</p>
	<p>To solve:</p> <ol style="list-style-type: none"> 1. We know to form a triangle, the sum of the lengths of any two sides must be greater than the length of the third side. 2. Let the denote the length of the third side as x. 3. We to satisfy the triangle inequality theorem: <ul style="list-style-type: none"> - $8 + 8 > x$ - $8 + x > 8$ - $x + 8 > 8$ 4. Simplifying these inequalities: <ul style="list-style-type: none"> - $16 > x$ - $x > 0$ - $x > 0$ 5. Combining these inequalities, we get: <ul style="list-style-type: none"> - $0 < x < 16$ 6. Since x must be a whole number, the possible values for x are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15. 7. The find the greatest possible perimeter, we choose the largest possible value for x: <ul style="list-style-type: none"> - $x = 15$ 8. The perimeter of the triangle is the sum of its sides: <ul style="list-style-type: none"> - $\text{Perimeter} = 8 + 8 + 15 = 31$ <p>Therefore, the greatest possible perimeter for the triangle is 31.</p>

Table 7: A case from the rejection sampling experiments. We find that using MRO to search for better and more accurate reasoning paths, and then fine-tuning the DLM with these paths, helps the model tend toward better reasoning paths during the decoding process.

Prompt	Artie has a flower stand at the Farmers Market. He sells three kinds of flowers: marigolds, petunias and begonias. He usually sells marigolds for \$2.74 per pot, petunias for \$1.87 per pot and begonias for \$2.12 per pot. Artie has no change today, so he has decided to round all his prices to the nearest dollar. If Artie sells 12 pots of marigolds, 9 pots of petunias and 17 pots of begonias, how much will he make?
Answer	88
LLaDA	<p>To determine how much Artie will make, we need to calculate the total revenue from each type of flower and then sum these amounts.</p> <p>1. **Marigolds:** - Price per pot: \$2.74 - Number of pots sold: 12 - Revenue from marigolds: $2.74 \times 12 = 32.88$</p> <p>2. **Petunias:** - Price per pot: \$1.87 - Number of pots sold: 9 - Revenue from petunias: $1.87 \times 9 = 16.83$</p> <p>3. **Begonias:** - Price per pot: \$2.12 - Number of pots sold: 17 - Revenue from begonias: $2.12 \times 17 = 36.04$</p> <p>Now, we sum the revenues:</p> $32.88 + 16.83 + 36.04 = 85.75$ <p>The answer is \$86.</p>
LLaDA-MRO-RL	<p>First, we's round the prices to the nearest dollar: - Marigolds: \$2.74 rounds to \$3 - Petunias: \$1.87 rounds to \$2 - Begonias: \$2.12 rounds to \$2</p> <p>Next, we calculate the total revenue for each type of flower: - Marigolds: 12 pots * \$3/pot = \$36 - Petunias: 9 pots * \$2/pot = \$18 - Begonias: 17 pots * \$2/pot = \$34</p> <p>Finally, we add up the revenues together: $\\$36 + \\$18 + \\$34 = \\88</p> <p>The answer is \$88.</p>

Table 8: A case from the reinforcement learning experiments. Interestingly, after applying reinforcement learning for MRO optimization, we find that the DLM generates shorter and more accurate CoT-style reasoning paths. We conjecture that this improvement is enabled by the optimization against the inter-sequence correlation reward, which encourages the model to achieve better format and accuracy rewards with fewer denoising steps.