Curriculum Group Policy Optimization: Adaptive Sampling for Unleashing the Potential of Text-to-Image Generation

Supplementary Material

6. Derivation of the Category Calibration Formula

In this section, we present the detailed derivation process for our category calibration.

To derive the analytical solution for the optimization problem in Eq. (6), we first expand the KL divergence term. The original problem is:

$$\begin{aligned} \max_{q} \quad & \sum_{i=1}^{c} \log(q_i) - \lambda \cdot \mathrm{KL}(v\|q), \\ \text{s.t.} \quad & \forall q_i \geq 0 \,, \sum_{i=1}^{c} q_i = 1. \end{aligned} \tag{9}$$

The KL divergence between (v) and (q) can be written as:

$$\mathrm{KL}(v\|q) = \sum_{i=1}^{c} v_i \log \frac{v_i}{q_i} = \sum_{i=1}^{c} v_i \log v_i - \sum_{i=1}^{c} v_i \log q_i. \tag{10}$$

Substituting this expression into the objective, we obtain:

$$\begin{aligned} &\sum_{i=1}^{c} \log q_i - \lambda \left(\sum_{i=1}^{c} v_i \log v_i - \sum_{i=1}^{c} v_i \log q_i \right) \\ &\quad = -\lambda \sum_{i=1}^{c} v_i \log v_i + \sum_{i=1}^{c} (1 + \lambda v_i) \log q_i. \end{aligned} \tag{11}$$

Since the term $-\lambda \sum_i v_i \log v_i$ is constant with respect to (q), maximizing the objective is equivalent to maximizing:

$$\sum_{i=1}^{c} (1 + \lambda v_i) \log q_i. \tag{12}$$

We now apply the method of Lagrange multipliers to incorporate the normalization constraint. The Lagrangian is given by:

$$\mathcal{L}(q, \mu) = \sum_{i=1}^{c} (1 + \lambda v_i) \log q_i + \mu \left(\sum_{i=1}^{c} q_i - 1 \right). \tag{13}$$

Taking the derivative with respect to each q_i and setting it to zero yields:

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{1 + \lambda v_i}{q_i} + \mu = 0, \tag{14}$$

which implies:

$$q_i = -\frac{1 + \lambda v_i}{\mu}. \tag{15}$$

Using the normalization constraint $\sum_{i=1}^{c} q_i = 1$, we have:

$$-\frac{1}{\mu} \sum_{i=1}^{c} (1 + \lambda v_i) = -\frac{1}{\mu} (c + \lambda \sum_{i=1}^{c} v_i) = 1. \tag{16}$$

Since (v) is a normalized distribution, we have $\sum_{i=1}^{c} v_i = 1$, and therefore:

$$\mu = -(c + \lambda). \tag{17}$$

Substituting this result back into the expression for q_i gives the closed-form solution:

$$q_i = \frac{1 + \lambda v_i}{c + \lambda}. \tag{18}$$

This completes the derivation.

7. Further Details on the Experimental Setup

Our CGPO framework builds upon the Flow-GRPO architecture. The configuration uses a sampling timestep $T = 10$ during training and $T = 40$ for evaluation, with an image group size $G = 24$, noise level $a = 0.8$, and image resolution 256. The KL ratio is set to 0.004 (0.04 for the fast variant). LoRA parameters are configured with $\alpha = 64$, $r = 32$.

8. Extended Experimental Results

8.1. Multiple Rewards Experimental

For a controlled comparison, the reproduced 8-GPU Flow-GRPO uses the same batch size, rollout configuration, reward model, and training steps as CGPO, with the training framework being the only difference. Following the evaluation protocol of Flow-GRPO, we train three separate models using GenEval, OCR, and PickScore as the training reward, respectively, and report their corresponding results. Although larger computational budgets may further improve performance, all of our experiments are conducted under the practical constraint of eight H100 GPUs. Under this setting, CGPO consistently outperforms both the reproduced 8-GPU Flow-GRPO and the 24-GPU result reported in the original work, achieving stronger performance on GenEval, OCR, and PickScore, as shown in Table 5.

8.2. Proxy Indicators Experimental

Table 6 presents a preliminary comparison of different proxy indicators. For each indicator, the original training set is filtered into a corresponding subset, on which models

Table 5. Comparison Experiments with Multiple Rewards.

Method	GPU	RL Method	GenEval	OCR	Pickscore
SD3.5	–	–	0.63	0.59	21.72
Visual-CoG	–	PPO	0.84	–	–
DanceGRPO	32	GRPO	0.69	–	23.00
GRPO-Guard	–	GRPO	0.95	0.93	23.30
Flow-GRPO	8	GRPO	0.94	0.92	23.31
Flow-GRPO	24	GRPO	0.95	0.92	23.31
CGPO	8	CGPO	**0.96**	**0.95**	**23.43**

Table 6. Comparison of Multiple Proxy Indicators.

Method	Counting	Colors	Position	Attribute	Overall
SD3.5	0.50	0.81	0.24	0.52	0.63
Reward variance	**0.83**	**0.85**	**0.72**	0.63	**0.83**
Advantage magnitude	0.83	0.83	0.57	0.63	0.80
Multi-criteria	0.82	0.83	0.67	**0.65**	0.82

are trained using Flow-GRPO and evaluated on GenEval. Among the compared indicators, reward variance achieves the best overall performance. We adopt variance as the proxy indicator because it arises naturally from the group-based reward structure of GRPO without introducing additional computational overhead, while also showing stronger empirical effectiveness than the alternative choices. In particular, variance is more sensitive to rare but important deviations than frequency- or mean-based indicators.

8.3. Hyperparameter Study

Table 7 presents a comprehensive analysis of the hyperparameter λ in our category calibration framework. Experimental results demonstrate a clear performance peak at $\lambda = 10$, with measurable performance degradation observed at both lower and higher values. This pattern reveals the critical balance struck by our calibration method in managing category-level sampling distributions.

At lower λ values ($\lambda < 10$), the calibration term exerts insufficient influence on the sampling distribution, failing to adequately address the inter-category imbalance. The limited regularization effect results in suboptimal allocation of training resources across categories, particularly hindering the model's ability to improve on underperforming categories.

Conversely, at higher λ values ($\lambda > 10$), the calibration term dominates the sampling process. This over-amplification of category weights leads to numerous prompts approaching the maximum sampling probability, effectively negating the probabilistic nature of our sampling strategy. The resulting near-uniform distribution undermines the adaptive advantages of our curriculum learning approach, diminishing both training efficiency and final performance.

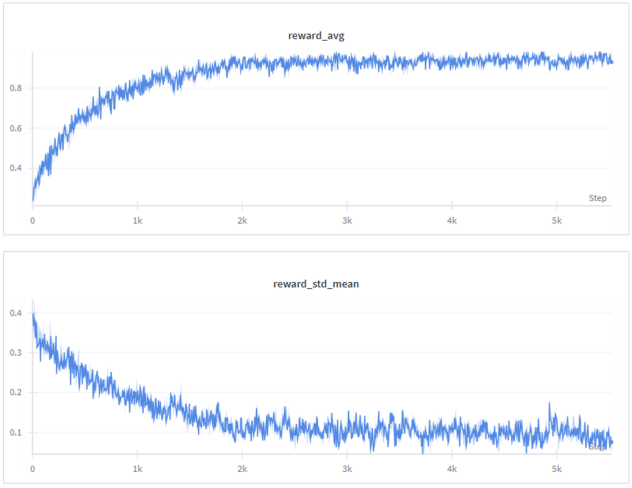

Figure 6. **Model Training Curves on Weights & Biases.** This figure displays the changes in rewards during our model's training process. Here, reward_avg represents the average reward, while reward_std_mean indicates the mean standard deviation of rewards.

The optimal configuration at $\lambda = 10$ achieves an equilibrium where the calibration effectively balances sampling across categories while preserving the probabilistic discrimination between individual prompts. This balance ensures that both prompts that remain actively learnable and underrepresented categories receive appropriate attention during training, contributing to the overall performance gains

Table 7. **Hyperparameter Study.** Effect of the hyperparameter λ in Category Calibration.

λ	Single object	Two object	Counting	Colors	Position	Attribute	Overall
5	1.00	0.98	0.90	0.94	0.98	0.88	0.95
8	1.00	0.98	0.87	0.93	0.97	0.87	0.94
10	1.00	0.99	0.96	0.94	0.99	0.89	0.96
12	1.00	1.00	0.88	0.92	0.99	0.87	0.94
15	1.00	0.97	0.95	0.91	0.97	0.84	0.94

demonstrated in our experiments.

8.4. Reward During Training

As shown in the training curves in Figure 6, the model's reward (reward_avg) demonstrates a continuous and significant upward trend throughout training, eventually stabilizing at a high level. Concurrently, the standard deviation of rewards (reward_std_mean) steadily decreases to a low value. This pattern of "increasing reward with decreasing variance" indicates that the model gradually shifts from inconsistent partial mastery to more stable and reliable performance on the training prompts. The temporal alignment between reward stabilization and variance reduction provides evidence of convergence to a well-regularized solution, confirming our method's effectiveness in achieving both strong performance and training stability.

8.5. Additional Visualizations

Figures 7 and 8 provide comprehensive qualitative comparisons between our proposed CGPO framework and established baseline methods across diverse compositional scenarios. The visualizations distinctly showcase our model's enhanced capability in handling complex instructions involving multiple objects, attributes, and spatial relationships. Specifically, in tasks requiring precise attribute binding, our method consistently produces correct object-property associations, whereas baseline models frequently exhibit color-object dissociation or positional errors.

This performance advantage stems from our adaptive sampling mechanism, which effectively prioritizes prompts that remain actively learnable during training, enabling the model to develop more robust representations of complex visual concepts.

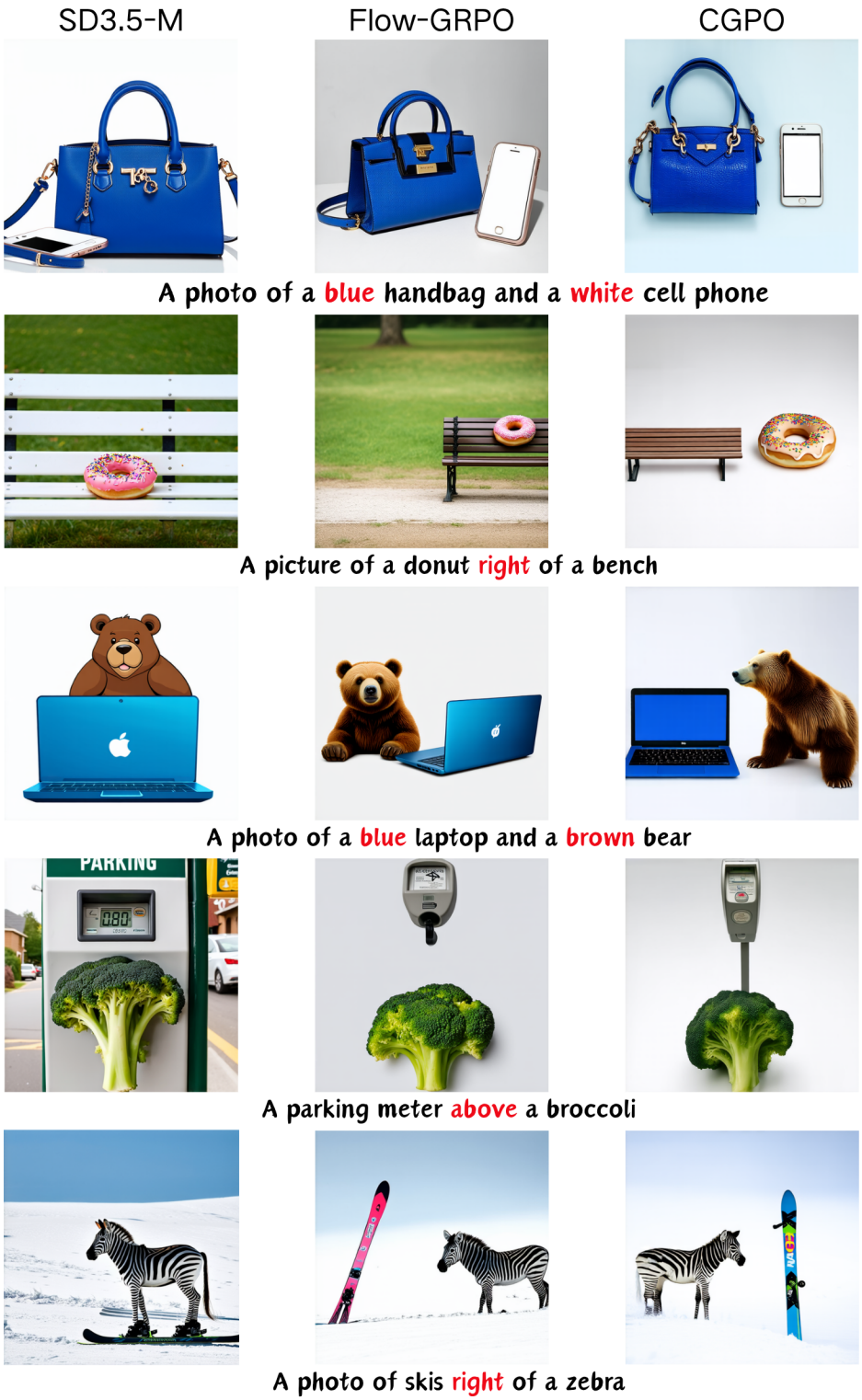

Figure 7. **Additional Visualizations.** Our method outperforms SD3.5-M and Flow-GRPO in key areas including Attribute Binding, Color, Spatial, and Counting.

Figure 8. **Additional Visualizations.** Our method outperforms SD3.5-M and Flow-GRPO in key areas including Attribute Binding, Color, Spatial, and Counting.