Training Diffusion Models Towards Diverse Image Generation with Reinforcement Learning

Supplementary Material

A. Discussions on Diversity Reward

We discuss two reward choices in Section 4.1, which are based on MMD [13] and Gaussian Process mutual information [38]. Both of them conclude with a kernel computation in the deep feature space. We also explore other options to quantify the diversity of the diffusion model, including one that is non-differentiable, i.e., the recall reward r_D^{recall}.

Recall Reward As proposed in [20], Recall directly measures the coverage of the generative distribution P_g. Intuitively, optimizing the recall will encourage the diffusion model to generate images that have a larger coverage. Thus, we adopt Recall as another diversity reward for RL fine-tuning. Specifically, given $\mathbf{Z_g}$ and $\mathbf{Z_r}$, we define the recall reward can be formulated as,

$$r_D^{\text{recall}}(\mathbf{X_g}; \mathbf{X_r}, \phi) = \text{recall}(\mathbf{Z_g}, \mathbf{Z_r}) = \frac{1}{|\mathbf{Z_r}|} \sum_n l(Z_r^n, \mathbf{Z_g}), \tag{21}$$

where $l(Z_r^n, \mathbf{Z_g})$ is a binary function that returns 1 if Z_r^n falls in any k-nearest neighbors spanned by $\mathbf{Z_g}$. It can be described as,

$$l(Z_r^n, \mathbf{Z_g}) = \begin{cases} 1, \text{if } ||Z_r^n - Z_g^m||_2 < ||Z_g^m < NN_k(Z_g^m)||_2, \exists Z_g^m \in \mathbf{Z_g} \\ 0, \text{otherwise.} \end{cases} \tag{22}$$

where $NN_k(Z_g^m)$ returns the k-th nearest neighbor of Z_g^m.

Intuitively, r_D^{recall} enlarges the NN-manifold's coverage, eventually enriching the generated images' diversity.

B. Reward Evaluation Experiments

In this section, we provide more details about the post-sampling selection experiment, which is used for evaluating the effectiveness of proposed *Diversity Reward*.

Table 4. More results for Reward Evaluation Experiments.

Method	Recall	Precision	FID
Baseline	40.22	85.77	8.10
Recall (max)	44.07	83.41	7.20
MMD (max)	**47.66**	83.26	**6.22**
GP-MI (max)	45.31	84.14	6.81
Recall(min)	36.12	87.53	9.22
MMD (min)	27.69	92.39	14.94
GP-MI (min)	31.24	88.26	10.32

B.1. More Details on Settings

We adopt the implementation in https://github.com/openai/guided-diffusion as the pre-trained class-conditional diffusion model. Specifically, with guidance scale $s = 4.0$, we sample 100K images with $O = 10$ images as a basic set that are sampled from the same class. For each basic set, we select a subset with $M = 5$ images, thus we evaluate $\binom{10}{5} = 252$ subsets' reward values, and select both the one with the maximum *Diversity Reward* and the one with the minimum reward value. By conducting this process by $100K/10 = 10K$ times, we construct one $50K$ set selected with the maximum reward criterion, and another $50K$ set with the minimum reward criterion. As for the reward function. As for the reward function, we test all three designs of *Diversity Reward*.

B.2. More Experimental results

We provide more experimental results along with the Recall reward in Table 4. All the proposed rewards can help select diverse subsets, as indicated by the Recall and FID values. Besides, we also provide visualizations in Figure 7 and Figure 8, where visually diverse & similar subsets can be selected with the proposed reward functions.

C. RL Fine-tuning for class-conditional diffusion models

C.1. ImageNet Experiments

Implementation Details. We also adopt the implementation in https://github.com/openai/guided-diffusion as the pre-trained class-conditional diffusion model. For LoRA fine-tuning [15], we add trainable LoRA weights to all the attention layers in both the diffusion model and the guided classifier with rank $R = 4$. We adopt the InceptionV3 [35] as ϕ in the *Diversity Reward*. For RL tuning, we adopt the implementation in https://github.com/kvablack/ddpo-pytorch, and set the learning rate to 0.0002, and set $M = 5$, which is also adopted as both the sampling batch size. We sample 16 batches before the policy gradient update. We conduct 1 epoch of policy gradient update after sampling images for reward computation, and we set the total number of epochs to 150. We adopt the same hyperparameters for all three designs of *Diversity Reward*

More Results. We provide results using all of the proposed *Diversity Reward*, including the non-differentiable

Recall reward, in Table 5. We also provide visualizations in Figure Figs. 9 to 13.

C.2. CIFAR Experiments

Following [27], we adopt the implementation in DDPM [14] as the pre-trained CIFAR diffusion model. We apply LoRA to attention layers with $R = 4$. We also adopt the InceptionV3 [35] as ϕ in the *Diversity Reward*. For RL set the learning rate to 0.0001, and set $M = 5$, which is also adopted as both the sampling batch size. We sample 16 batches before the policy gradient update. We conduct 1 epoch of policy gradient update after sampling images for reward computation, and we set the total number of epochs to 100. We adopt the same hyperparameters for all three designs of *Diversity Reward*

D. RL Fine-tuning for StableDiffusion

Implementation Details. We utilize the StableDiffusion v1-4 checkpoint as in https://huggingface.co/CompVis/stable-diffusion. For LoRA fine-tuning [15], we add trainable LoRA weights to all the attention layers in both the diffusion model and the guided classifier with rank $R = 4$. We adopt the visual encoder of CLIP-B/16 [28] as ϕ in the *Diversity Reward*. For RL tuning, we adopt the implementation in https://github.com/kvablack/ddpo-pytorch, and set the learning rate to 0.0001, and set $M = 6$, which is also adopted as both the sampling batch size. We sample 4 batches before the policy gradient update. We conduct 1 epoch of policy gradient update after sampling images for reward computation, and we set the total number of epochs to 80. We adopt the same hyperparameters for all three designs of *Diversity Reward*

Table 5. More RL Fine-tuning Results on ImageNet-128x128 with different rewards.

Method	Recall	Precision	FID
Baseline $s = 4.0$	36.15	82.96	24.49
MMD	**47.66**	83.26	**23.42**
GP-MI	45.31	81.72	23.81
Recall	42.13	81.63	24.05
Baseline $s = 3.0$	40.35	82.10	23.00
MMD	**49.31**	81.26	**22.08**
GP-MI	46.31	81.74	22.31
Recall	45.23	82.01	22.49
Baseline $s = 2.0$	45.95	79.20	21.19
MMD	**49.30**	78.82	**20.48**
GP-MI	47.83	79.05	20.95
Recall	46.12	79.18	21.03
Baseline $s = 1.0$	55.10	74.16	19.86
MMD	**59.63**	**74.84**	**19.19**
GP-MI	56.13	73.52	19.49
Recall	56.01	74.01	19.63

More Visualizations. We provide more visualization results of unbiased image generation in Figure Figs. 14 to 18.

E. Ablation Experiments

We adopt the all the implementation and hyperparameters in Section C.1, with the different number of reference images N and the number of images for reward computation M.

(a) Generated images

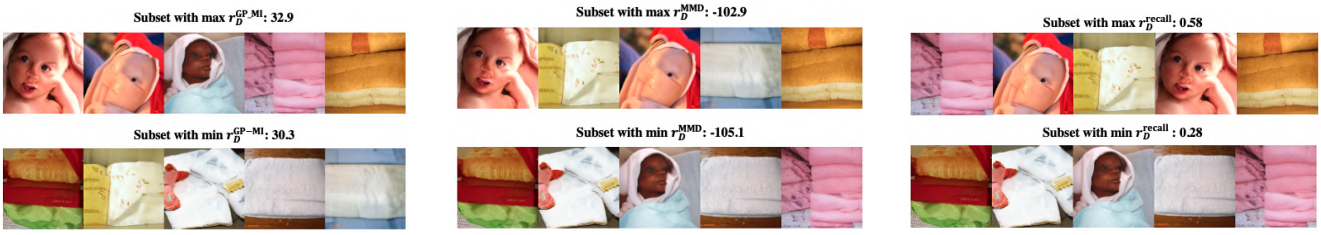
(b) Diverse reference images

(c) Subsets Selected with $r_D^{\text{GP-MI}}$ (d) Subsets Selected with r_D^{MMD} (e) Subsets Selected with r_D^{recall}

Figure 7. Illustration of the post-sampling experiments to show the effectiveness of our reward functions on class 761, 'Remote Control'.

(a) Generated images

(b) Diverse reference images



(c) Subsets Selected with $r_D^{\text{GP-MI}}$ (d) Subsets Selected with r_D^{MMD} (e) Subsets Selected with r_D^{recall}

Figure 8. Illustration of the post-sampling experiments to show the effectiveness of our reward functions on class 434, 'Bath Towel'.

(a) Baseline

(b) Reference Images $\mathbf{X_r}$

(c) RL Fine-tuning with GP-MI *Diversity Reward*

(d) RL Fine-tuning with MMD *Diversity Reward*

Figure 9. RL fine-tuning results with different *Diversity Reward* on ImageNet ($s = 4.0$) for class 31, 'Tree Frog'.

(a) Baseline

(b) Reference Images $\mathbf{X_r}$

(c) RL Fine-tuning with GP-MI *Diversity Reward*

(d) RL Fine-tuning with MMD *Diversity Reward*

Figure 10. RL fine-tuning results with different *Diversity Reward* on ImageNet ($s = 4.0$) for class 46, 'Green Lizard'.

(a) Baseline

(b) Reference Images $\mathbf{X_r}$

(c) RL Fine-tuning with GP-MI *Diversity Reward*

(d) RL Fine-tuning with MMD *Diversity Reward*

Figure 11. RL fine-tuning results with different *Diversity Reward* on ImageNet ($s = 4.0$) for class 0, 'Tench'.

(a) Baseline

(b) Reference Images $\mathbf{X_r}$

(c) RL Fine-tuning with GP-MI *Diversity Reward*

(d) RL Fine-tuning with MMD *Diversity Reward*

Figure 12. RL fine-tuning results with different *Diversity Reward* on ImageNet ($s = 4.0$) for class 3, 'Tiger Shark'.

(a) Baseline

(b) Reference Images $\mathbf{X_r}$

(c) RL Fine-tuning with GP-MI *Diversity Reward*

(d) RL Fine-tuning with MMD *Diversity Reward*

Figure 13. RL fine-tuning results with different *Diversity Reward* on ImageNet ($s = 4.0$) for class 11, 'Goldfinch'.

(a) w/ 'Eyeglasses'

(b) w/o 'Eyeglasses'

(c) Reference images from CelebA.

Figure 14. RL fine-tuning results for StableDiffusion on unbiased face generation for attribute, 'Eyeglasses'.

(a) 'Male'

(b) 'Female'

(c) Reference images from CelebA.

Figure 15. RL fine-tuning results for StableDiffusion on unbiased face generation for attribute, 'Male'.

(a) 'Young'

(b) 'Old'

(c) Reference images from CelebA.

Figure 16. RL fine-tuning results for StableDiffusion on unbiased face generation for attribute, 'Young'.

(a) w/ 'Smiling'

(b) w/o 'Smiling'

(c) Reference images from CelebA.

Figure 17. RL fine-tuning results for StableDiffusion on unbiased face generation for attribute, 'Smiling'.

(a) w/ 'Pale Skin'

(b) w/o 'Pale Skin'

(c) Reference images from CelebA.

Figure 18. RL fine-tuning results for StableDiffusion on unbiased face generation for attribute, 'Pale Skin'.