

Ranking-based Preference Optimization for Diffusion Models from Implicit User Feedback

This is the official implementation of the paper, *Ranking-based Preference Optimization for Diffusion Models from Implicit User Feedback*.

Requirements

Install the required dependencies using the following command:

```
pip install -r requirements.txt
```

Note: The results in the paper were obtained using **Python 3.9.20** and **torch==2.3.1** with **cuda-12.1**.

Datasets

► Pick-a-Pic v2

- Download **Pick-a-Pic v2** and use the PickScore model to score all images, selecting 500 high-scoring images as training data:

```
accelerate launch --multi_gpu --gpu_ids all --num_processes 8 -m
tools.pickapic \
  --cache ./data/cache \
  --version v2 \
  --split train \
  --score hpsv2 \
  --top 500 \
  --batch_size 64 \
  --no-preferred \
  --output ./data/pickapicv2_hpsv2_preferred_500
```

- For each test caption, select the top 1 scoring image for visualization:

```
accelerate launch --multi_gpu --gpu_ids all --num_processes 8 -m
tools.pickapic \
  --cache ./data/cache \
  --version v2 \
  --split test \
  --score pickscore \
  --top 1 \
  --per_caption \
  --batch_size 64 \
```

```
--no-preferred \  
--output ./data/pickapicv2_pickscore_test
```

► HPDv2

- Download benchmark prompts and expert images.

```
export HPDv2_BENCHMARK_DIR="./data/hpdv2/benchmark"  
mkdir -p ${HPDv2_BENCHMARK_DIR}  
for file in concept-art.json anime.json paintings.json photo.json; do  
  wget  
  "https://huggingface.co/datasets/ymhao/HPDv2/resolve/main/benchmark/${  
  file}" -O "${HPDv2_BENCHMARK_DIR}/${file}"; done  
  
export  
HPDv2_BENCHMARK_IMGS_DIR="${HPDv2_BENCHMARK_DIR}/benchmark_imgs"  
mkdir -p ${HPDv2_BENCHMARK_IMGS_DIR}  
for file in CM.tar.gz Cog2.tar.gz DALLE-mini.tar.gz DALLE.tar.gz DF-  
IF.tar.gz DL.tar.gz Deliberate.tar.gz ED.tar.gz FD.tar.gz LDM.tar.gz  
Laf.tar.gz MM.tar.gz OJ.tar.gz RV.tar.gz SDXL-base-0.9.tar.gz SDXL-  
refiner-0.9.tar.gz VD.tar.gz VQD.tar.gz VQGAN.tar.gz glide.tar.gz  
sdv1.tar.gz sdv2.tar.gz; do wget  
  "https://huggingface.co/datasets/ymhao/HPDv2/resolve/main/benchmark/be  
  nchmark_imgs/${file}" -O "${HPDv2_BENCHMARK_IMGS_DIR}/${file}"; tar -  
  zxvf "${HPDv2_BENCHMARK_IMGS_DIR}/${file}" -C  
  "${HPDv2_BENCHMARK_IMGS_DIR}/"; done
```

- For each style (anime, concept-art, paintings, photo), select the top 1 scoring image per prompt:

```
accelerate launch --multi_gpu --gpu_ids all --num_processes 8 -m  
tools.hpdv2_benchmark \  
  --cache ./data/cache \  
  --batch_size 64 \  
  --output ./data/hpdv2_{style}
```

Training

Hardware Settings

The model was trained using 4 RTX 3090 GPUs over approximately 20 ~ 25 hours.

- See our `default_config.yaml`

The `default_config.yaml` for `accelerate launch` is as follows:

```
compute_environment: LOCAL_MACHINE
debug: false
distributed_type: 'NO'
downcast_bf16: 'no'
gpu_ids: all
machine_rank: 0
main_training_function: main
mixed_precision: 'no'
num_machines: 1
num_processes: 1
rdzv_backend: static
same_network: true
tpu_env: []
tpu_use_cluster: false
tpu_use_sudo: false
use_cpu: false
```

Start Training

- SD 1.5

```
accelerate launch --multi_gpu --gpu_ids 0,1,2,3 --num_processes 4
train.py \
  --pretrained_model_name_or_path stable-diffusion-v1-5/stable-
diffusion-v1-5 \
  --train_dataset ./data/pickapicv2_hpsv2_nopreferred_500 \
  --validation_dataset ./data/pickapicv2_pickscore_test \
  --resolution 512 \
  --random_flip \
  --random_drop_prompt_probability 0.2 \
  --num_steps 25600 \
  --batch_size 4 \
  --gradient_accumulation_steps 16 \
  --learning_rate 1e-4 \
  --hinge \
  --margin 0.001 \
  --validation_steps 256 \
  --validation_scheduler DDPM \
  --validation_num_inference_steps 50 \
  --validation_guidance_scale 7.5 \
  --buffer_batch_size 4 \
  --buffer_batch_accumulation 1 \
  --buffer_scheduler DPMSolver++ \
  --buffer_num_inference_steps 20 \
  --buffer_guidance_scale 1.0 \
  --buffer_sample_steps 1 \
  --buffer_update_steps 16 \
  --buffer_size 4 \
  --buffer_perturb_timesteps \
  --no-buffer_sync \
  --checkpointing_steps 1280 \
```

```
--use_ema \  
--offload_ema \  
--mixed_precision bf16 \  
--logdir ./logs/sd15_pickapicv2hpsv2nopreferredd500_bs256_hinge_1
```

Perform Inference

- Pick-a-Pic v2 test:

```
accelerate launch --gpu_ids 0,1,2,3 --multi_gpu --num_processes 4  
inference.py \  
  --pretrained_model_name_or_path stable-diffusion-v1-5/stable-  
diffusion-v1-5 \  
  --test_dataset_root ./data/pickapicv2_pickscore_test \  
  --batch_size 4 \  
  --num_images_per_prompt 5 \  
  --mixed_precision bf16 \  
  --scheduler DDPM \  
  --num_inference_steps 50 \  
  --guidance_scale 7.5 \  
  --seed 0 \  
  --UNET  
./logs/sd15_pickapicv2hpsv2nopreferredd500_bs256_hinge_1/ckpt-25600/ema  
\  
  --output ./output/pickapicv2_pickscore_test
```

- HPDv2 Benchmark

Replace `--test_dataset_root` with `./data/hpdv2_{style}` and `--output` with `./output/hpdv2_{style}`. The style can be `anime`, `concept-art`, `paintings`, or `photo`.

Quantitative Evaluation

Calculate PickScore, HPSv2, Aesthetic Score, CLIP Score, and ImageReward for the generated images:

```
accelerate launch --gpu_ids 0,1,2,3 --multi_gpu --num_processes 4 score.py  
\  
  --batch_size 32 \  
  --pickscore \  
  --hpsv2 \  
  --aestheticv2 \  
  --clip \  
  --imagereward \  
  --dir ./output/pickapicv2_pickscore_test \  
  --dir ./output/hpdv2_anime \  
  --dir ./output/hpdv2_concept-art \  
  --dir ./output/hpdv2_paintings \  
  --dir ./output/hpdv2_photo
```

