

## Scripts on GSM: Reasoning

*Use VLLM to start white-box model server*

```
bash ./scripts/vllm.sh 8000 0
```

*Run Inference using gpt-3.5-turbo*

```
python gsm_pipeline.py \  
  --whitebox meta-llama/Meta-Llama-3-8B-Instruct \  
  --port 8000 \  
  --few_gpt_shot 1 \  
  --gpt-engine gpt3.5_0125
```

## IGO Training

### Step1. Sample Data On training set

```
python gsm_generate_data.py \  
  --few_gpt_shot 1
```

Suppose generate data is saved at ./data/gsm8k/gpt3.5/train.jsonl

### Step2. DPO Training

```
set -x  
  
read -r -d '' training_commands <<EOF  
train_dpo \  
  --save_path ./gsm/llama3-8b-instruct-dpo-lora \  
  --save_steps -1 \  
  --logging_steps 1 \  
  --eval_steps -1 \  
  --train_batch_size 64 \  
  --micro_train_batch_size 1 \  
  --pretrain meta-llama/Meta-Llama-3-8B-Instruct \  
EOF
```

```

--bf16 \
--max_epochs 1 \
--max_len 8192 \
--zero_stage 3 \
--learning_rate 5e-6 \
--beta 0.1 \
--dataset json@./data/gsm8k/gpt3.5 \
--apply_chat_template \
--chosen_key chosen \
--rejected_key rejected \
--load_checkpoint \
--gradient_checkpointing \
--label_smoothing 0.1 \
--lora_rank 8 \
--lora_alpha 16 \
--lora_dropout 0.05
EOF

# --use_wandb

if [[ ${1} != "slurm" ]]; then
    deepspeed --master_port=29400 --include localhost:0,1,2,3 --module $training_commands
fi

```

Suppose DPO trained Lora model is saved at ./gsm/llama3-8b-instruct-dpo-lora

### Step3. Use trained Lora to do inference

```

# open white-box vllm server
bash ./scripts/vllm_peft.sh 8000 0

# run inference using lora model
python gsm_pipeline.py \
    --whitebox lora \
    --port 8000 \
    --few_gpt_shot 1 \
    --gpt-engine gpt3.5_0125

```

## Scripts on LLaMP: Personalization

Similar to the scripts on GSM8K, simply replace the 'gsm\*.py' files with 'llamp\*.py' files.

# Scirps on Alfworld: Planning

*Run Inference using gpt-3.5-turbo*

```
python alfworld_pipeline.py \  
  --whitebox meta-llama/Meta-Llama-3-8B-Instruct \  
  --port 8000 \  
  --use_gpt
```

## *IGO Training*

### Step1. Sample Data On training set

```
python alfworld_generate_data.py \  
  --disable_cloeloop
```

Suppose generate data is saved at ./data/alfworld/gpt3.5/train.jsonl

### Step2. DPO Traning

```
set -x  
  
read -r -d '' training_commands <<EOF  
train_dpo \  
  --save_path ./alfworld/llama3-8b-instruct-dpo-lora \  
  --save_steps -1 \  
  --logging_steps 1 \  
  --eval_steps -1 \  
  --train_batch_size 64 \  
  --micro_train_batch_size 1 \  
  --pretrain meta-llama/Meta-Llama-3-8B-Instruct \  
  --bf16 \  
  --max_epochs 1 \  
  --max_len 8192 \  
  --zero_stage 3 \  
  --learning_rate 5e-6 \  
  --beta 0.1 \  
  --dataset json@./data/alfworld/gpt3.5 \  
  --apply_chat_template \  
  --chosen_key chosen \  
  --rejected_key rejected \  
  --load_checkpoint \  
  --gradient_checkpointing \  
EOF
```

```

--label_smoothing 0.1 \
--lora_rank 8 \
--lora_alpha 16 \
--lora_dropout 0.05
EOF

# --use_wandb

if [[ ${1} != "slurm" ]]; then
    deepspeed --master_port=29400 --include localhost:0,1,2,3 --module $training_commands
fi

```

Suppose DPO trained Lora model is saved at ./alfworld/llama3-8b-instruct-dpo-lora

### Step3. Use trained Lora to do inference

```

# open white-box vllm server
bash ./scripts/vllm_peft.sh 8000 0 meta-llama/Meta-Llama-3-8B-Instruct alfworld/llama3-8b-instruct-dpo-lora

# run inference using lora model
python alfworld_pipeline.py \
    --whitebox lora \
    --port 8000 \
    --use_gpt

```