Dreambooth | LoRA | Textual Inversion | Finetuning | Utilities | About

Training | Tools | Guides

Train a custom model using kohya train network LoRA python code...

Configuration

Accelerate launch

Resource Selection

Mixed precision
Whether or not to use mixed precision training.
no

Number of processes
The total number of processes to be launched in parallel.
1

Number of machines
The total number of machines used in this training.
1

Number of CPU threads per core
The number of CPU threads per process.
10

Dynamo backend
The backend to use for the dynamo JIT compiler.
no

Dynamo mode
Choose a mode to optimize your training with dynamo.
default

Whether to use full graph mode for dynamo or it is ok to break model into several subgraphs
Dynamo use fullgraph

Whether to enable dynamic shape tracing.
Dynamo use dynamic

Hardware Selection

Whether or not this should launch a distributed GPU training.
Multi GPU

Distributed GPUs

GPU IDs
What GPUs (by id) should be used for training on this machine as a comma-separated list
example: 0,1

Main process port
The port to use to communicate with the machine of rank 0.
0

Extra accelerate launch arguments
List of extra parameters to pass to accelerate launch
example: --same_network --machine_rank 4

Model

Pretrained model name or path
stabilityai/stable-diffusion-xl-base-1.0

Trained Model output name
last

Image folder (containing training images subfolders)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/out/img

Dataset config file (Optional. Select the toml configuration file to use for the dataset)

Training comment
(Optional) Add training comment to be included in metadata

Save trained model as
ckpt | safetensors

Save precision
float | fp16 | bf16

Metadata

Folders

Output directory for trained model
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/out/model

Regularisation directory (Optional. containing regularisation images)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/REGULARISATION

Logging directory (Optional. to enable logging and output Tensorboard log)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/out/log

Dataset Preparation

This section provide Dreambooth tools to help setup your dataset...

Dreambooth/LoRA Folder preparation | Dreambooth/LoRA Dataset balancing

This utility will create the necessary folder structure for the training images and optional regularization images needed for the kohys_ss Dreambooth/LoRA method to function correctly.

Instance prompt
heysalora3

Class prompt
woman

Training Images (directory containing the training images)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3

Repeats
20

Regularisation images (Optional. directory containing the regularisation images)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/REGULARISATION

Repeats
1

Destination training directory (where formatted training and regularisation folders will be placed)
/Volumes/Akos-700-NVMD-Scratchdisk/heysalora3/out

Prepare training data

Copy info to respective fields

Parameters

Presets
none

Basic

LoRA type
Standard

Network weights
Path to an existing LoRA network weights to resume training from
(Optional)

Automatically determine the dim(rank) from the weight file.
DIM from weights

Train batch size
1

Epoch
10

Max train epoch
training epochs (overrides max_train_steps). 0 = no override
0

Max train steps
Overrides if training steps. 0 = no override
1600

Save every N epochs
1

Caption file extension
.txt

Seed
Set to 0 to make random
0

Cache latents

Cache latents to disk

LR Scheduler
cosine

Optimizer
AdamW

Max grad norm
1

LR scheduler extra arguments
(Optional) eg: milestones=[1,10,30,50] gamma=0.1

Optimizer extra arguments
(Optional) eg: relative_step=True scale_parameter=True warmup_init=True

Learning rate
Set to 0 to not train the Unet
0.0001

LR warmup (% of total steps)
0

LR # cycles
Number of restarts for cosine scheduler with restarts
10

LR power
Polynomial power for polynomial scheduler
1

Max resolution
1024,1024

Stop TE (% of total steps)
0

Enable buckets

Minimum bucket resolution
Minimum size in pixel a bucket can be (>= 64)
256

Maximum bucket resolution
Maximum size in pixel a bucket can be (>= 64)
2048

Text Encoder learning rate
(Optional)
0.0001

Unet learning rate
(Optional)
0.0001

SDXL Specific Parameters

Cache the outputs of the text encoders. This option is useful to reduce the GPU memory usage. This option cannot be used with options for shuffling or dropping the captions.
Cache text encoder outputs

Disable the half-precision (mixed-precision) VAE. VAE for SDXL seems to produce NaNs in some cases. This option is useful to avoid the NaNs.
No half VAE

Network Rank (Dimension)
256

Network Alpha
alpha for LoRA weight scaling
1

Scale weight norms
Max Norm Regularization is a technique to stabilize network training by limiting the norm of network weights. It may be effective in suppressing overfitting of LoRA and improving stability when used with other LoRAs. See PR #545 on kohya_ss/sd_scripts repo for details. Recommended setting: 1. Higher is weaker. Lower is stronger.
0

Network dropout
Is a normal probability dropout at the neuron level. In the case of LoRA, it is applied to the output of down. Recommended range 0.1 to 0.5
0

Rank dropout
can specify `rank_dropout`. Is dropout each rank with specified probability. Recommended range 0.1 to 0.3
0

Module dropout
can specify `module_dropout`. Is dropout each rank with specified probability. Recommended range 0.1 to 0.3
0

Advanced

Weights | Blocks | Conv

Down LR weights
Specify the learning rate weight of the down blocks of U-Net.
(Optional) eg: 0,0,0,0,0.5,0.5,1,1,1,1,1,1

Mid LR weights
Specify the learning rate weight of the mid block of U-Net.
(Optional) eg: 0.5

Up LR weights
Specify the learning rate weight of the up blocks of U-Net. The same as down_lr_weight.
(Optional) eg: 0,0,0,0,0.5,0.5,1,1,1,1,1,1

Blocks LR zero threshold
If the weight is not more than this value, the LoRA module is not created. The default is 0.
(Optional) eg: 0.1

Gradient accumulate steps
Number of updates steps to accumulate before performing a backward/update pass
1

Weighted captions

Prior loss weight
1

VAE (Optional: Path to checkpoint of vae for training)

Additional parameters
(Optional) Use to provide additional parameters not handled by the GUI. Eg: --some_parameters "value"

Scheduled Huber Loss

Loss type
The type of loss to use and whether it's scheduled based on the timestep
l2

Huber schedule
The type of loss to use and whether it's scheduled based on the timestep
snr

Huber C
The huber loss parameter. Only used if one of the huber loss modes (huber or smooth l1) is selected with loss_type
0.1

Save every N steps
(Optional) The model is saved every specified steps
0

Save last N steps
(Optional) Save only the specified number of models (old models will be deleted)
0

Save last N steps state
(Optional) Save only the specified number of states (old models will be deleted)
0

Keep n tokens
0

Clip skip
1

Max Token Length
max token length of text encoder
75

U-Net and Text Encoder can be trained with fp8 (experimental)
fp8 base training (experimental)

Full fp16 training (experimental)

Required bitsandbytes >= 0.36.0
Full bf16 training (experimental)

Gradient checkpointing

Shuffle caption

Persistent data loader

Memory efficient attention

CrossAttention
sdpa

Enable weak color augmentation
Color augmentation

Enable horizontal flip augmentation
Flip augmentation

Apply mask for calculating loss. conditioning_data_dir is required for dataset
Masked loss

Only for SD v2 models. By scaling the loss according to the time step, the weights of global noise prediction and local noise prediction become the same, and the improvement of details may be expected.
Scale v prediction loss

Min SNR gamma
Recommended value of 5 when used
0

Automates the processing of noise, allowing for faster model fitting, as well as balancing out color issues. Do not use if Min SNR gamma is specified.
Debiased Estimation loss

Don't upscale bucket resolution

Bucket resolution steps
64

Random crop instead of center crop

V Pred like loss
Recommended value of 0.5 when used
0

Min Timestep
0

Max Timestep
Values lower than 1000 will make the model more img2img focussed. 1000 = noise only
1000

Noise offset type
Original

Noise offset
Recommended values are 0.05 - 0.15
0

Use random strength between 0~noise_offset for noise offset.
Noise offset random strength

IP noise gamma
enable input perturbation noise. used for regularization. recommended value: around 0.1
0

Use random strength between 0~ip_noise_gamma for input perturbation noise
IP noise gamma random strength

Adaptive noise scale
Add `latent mean absolute value * this value` to noise_offset
0

Dropout caption every n epochs
0

Rate of caption dropout
0

VAE batch size
0

Save training state (including optimizer states etc.) when saving models
Save training state

Save training state (including optimizer states etc.) on train end
Save training state at end of training

Resume from saved training state (path to "last-state" state folder)
Saved state to resume training from

Max num workers for DataLoader
Override number of epoch. Default: 0
0

Logging
Loggers to use, tensorboard will be used as the default.

WANDB API Key
Users can obtain and/or generate an api key in the their user settings on the website: https://wandb.ai/login
(Optional)

WANDB run name
The name of the specific wandb session
(Optional)

Log tracker name
Name of tracker to use for logging, default is script-specific default name
(Optional)

Log tracker config
Path to tracker config file to use for logging

Samples

HuggingFace

Stop training

Print training command

Open tensorboard