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

Training | Tools | Guides

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

Configuration ▼

Load/Save Config file

/home/Ubuntu/Desktop/New Folder/48 ▼

🔄 📂 💾 ↩️

Accelerate launch ◀

Model ▼

Pretrained model name or path

/home/Ubuntu/Downloads/flux1-dev.safetenso ▼

🔄 📄 📂

Trained Model output name

test1

Image folder (containing training images subfolders)

/home/Ubuntu/Documents/train_data_dir ▼

🔄 📂

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

▼

🔄 📄

□ v2 □ v_parameterizat □ SDXL □ SD3 □ Flux.1

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

/home/Ubuntu/ap| ▾

🔄 📂

Regularisation directory (Optional. containing regularisation images)

▾

🔄 📂

Logging directory (Optional. to enable logging and output Tensorboard log)

/home/Ubuntu/apps/kohya_ss/dataset/images/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

fnfng

Class prompt

man

Training images (directory containing the training images)

/home/Ubuntu/Docu ▾

🔄 📂

Repeats

40

Regularisation images (Optional. directory containing the regularisation images)

▾

🔄 📂

Repeats

1

Destination training directory (where formatted training and regularisation folders will be placed)

/home/Ubuntu/apps/kohya_ss/dataset/images/img ▾

🔄 📂

Prepare training data

Copy info to respective fields

Parameters ▼

Presets

none ▾

Basic ▼

LoRA type

Flux1 ▾

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

200

Max train epoch

training epochs (overrides max_train_steps). 0 = no override

Max train steps

Overrides # training steps. 0 = no override

0

0

Save every N epochs

25

Caption file extension

.txt ▾

Seed

Set to 0 to make random

0

□ Cache latents

□ Cache latents to disk

LR Scheduler

constant ▾

LR Scheduler type

(Optional) custom scheduler module name

▾

Optimizer

Adafactor ▾

Max grad norm 0

○

LR scheduler extra arguments

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

Optimizer extra arguments

scale_parameter=False relative_step=False warmup_init=False weight_decay=0.01

Learning rate

Set to 0 to not train the Unet

0.00005

LR warmup (% of total steps) 0

Can't use LR warmup with LR Scheduler constant... setting to 0 and disabling field...

○

LR # cycles

Number of restarts for cosine scheduler with restarts

LR power

Polynomial power for polynomial scheduler

1

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

Unet learning rate

(Optional)

0.00005

Network Rank (Dimension)

128

Network Alpha

alpha for LoRA weight scaling

128

Flux.1 ▼

VAE Path

/home/Ubuntu/Downloads/ae.safetensors

CLIP-L Path

/home/Ubuntu/Downloads/clip_l.safetensor

T5-XXL Path

/home/Ubuntu/Downloads/t5xxl_fp16.safetensors

Discrete Flow Shift

Discrete flow shift for the Euler Discrete Scheduler, default is 3.0

3

Model Prediction Type

raw ▾

Timestep Sampling

sigmoid ▾

Apply attention mask to T5-XXL encode and FLUX double blocks

☐ Apply T5 Attention Mask

Split mode for Flux1

☐ Split Mode

Train Blocks

all ▾

Split the projection layers of q/k/v/txt in the attention

☐ Split QKV

Guidance Scale

Guidance scale for Flux1

1

T5-XXL Max Token Length

Max token length for T5-XXL

512

Cache text encoder outputs to speed up inference

☐ Cache Text Encoder Outputs

Cache text encoder outputs to disk to speed up inference

☐ Cache Text Encoder Outputs to Disk

[Experimentsl] Enable memory efficient save. We do not recommend using it unless you are familiar with the code.

☐ Memory Efficient Save

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 0 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 0 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 1

☐ Weighted captions

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

○

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 ◀

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 ▾

Use fp8 for base model

□ fp8 base

□ Full fp16 training (experimental)

Required bitsandbytes >= 0.36.0

□ Full bf16 training (experimental)

Disable low VRAM optimization. e.g. do not clear CUDA cache after each latent caching (for machines which have bigger VRAM)

□ highvram

Enable low RAM optimization. e.g. load models to VRAM instead of RAM (for machines which have bigger VRAM than RAM such as Colab and Kaggle)

□ lowvram

□ 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 0

Recommended value of 5 when used

○

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 0

Recommended value of 0.5 when used

○

Min Timestep 0

Values greater than 0 will make the model more img2img focussed. 0 = image only

○

Max Timestep 1000

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

○

Noise offset type

Original ▾

Noise offset 0

Recommended values are 0.05 - 0.15

○

Use random strength between 0~noise_offset for noise offset

□ Noise offset random strength

Adaptive noise scale 0

Add `latent mean absolute value * this value` to noise_offset

○

IP noise gamma 0

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

○

Use random strength between 0~ip_noise_gamma for input perturbation noise

□ IP noise gamma random strength

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 training parameter to WANDB

□ Log config

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 ◀

Stop training

Print training command

Start tensorboard

v24.2.0

Use via API · Built with Gradio