AdaCLIP finetune in xtune

1.Data prepare:

Here we use **ActivityNet** dataset as an example, you can also create and use your own dataset.

For more dataset preparation, please check in: https://github.com/jilongW/GenAIComps/blob/main/comps/finetuning/src/integrations/xtune/doc/Prepare_dataset.md

1.1 ActivityNet Dataset download

Download the videos from the official website. The authors have made the videos available on Google and Baidu drives.

1.2 Frame Extraction

`Run GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune/utils/frame_extraction.py` to extract frames after having downloaded the dataset videos and annotations from the website:

```
python utils/frame_extraction.py /path/to/videos /path/to/frames
--parallel
```

Make sure that all the videos are in the same directory (no sub-directories allowed).

The frames from each video will be saved under: `/path/to/frames/video_name`

1.3 Dataset JSON prepare

We need to prepare dataset json which includes data annotations.

- For ActivityNet, you can use the annotated data under: https://github.com/SamsungLabs/AdaCLIP/tree/main/annots/activitynet
- If you want to create your own dataset json. Please follow the below format:

```
"video_name": {
    "sentences": [
    "sentence 1",
    "sentence 2",
    ...
    "sentence n"
    ]
}
```

Each data needs video name and sentences that describe the video content.

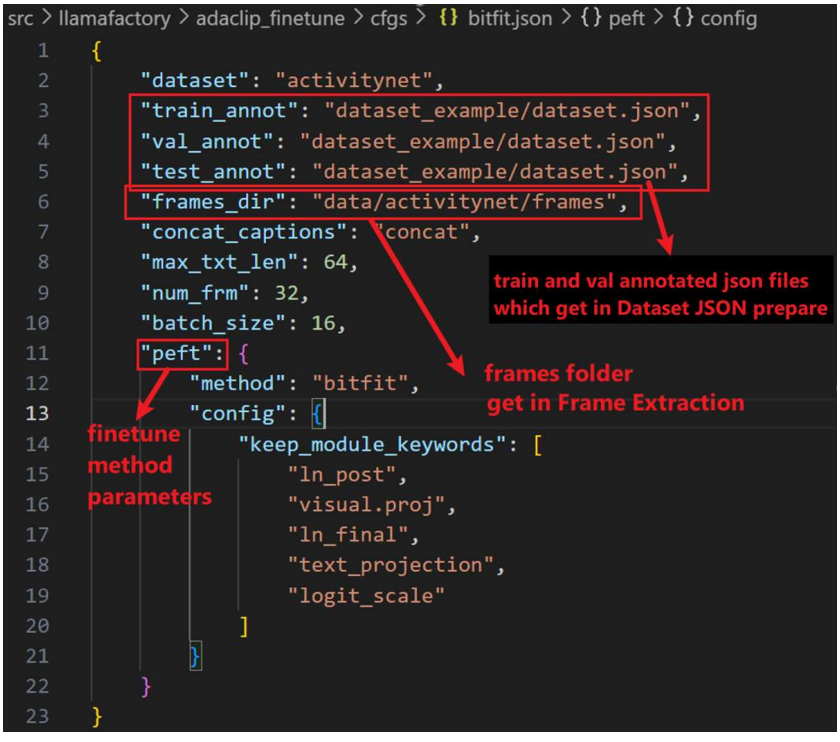
- An example json file is provided in: `src/llamafactory/adaclip_finetune/dataset_example/dataset.json`

2. AdaCLIP Finetune methods and their configurations

We have implemented the **BitFit** and **IBS** and **FULL** fine-tuning methods.

Each method has a configure json file to help run finetune process, you can utilize the corresponding configuration files under: `GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune/cfgs.`

Below is an example configuration json file which use bitfit algorithm.




```
{
    "dataset": "activitynet",
    "train_annot": "dataset_example/dataset.json",
    "val_annot": "dataset_example/dataset.json",
    "test_annot": "dataset_example/dataset.json",
    "frames_dir": "data/activitynet/frames",
    "concat_captions": "concat",
    "max_txt_len": 64,
    "num_frm": 32,
    "batch_size": 16,
    "peft": {
        "method": "bitfit",
        "config": {
            "keep_module_keywords": [
                "ln_post",
                "visual.proj",
                "ln_final",
                "text_projection",
                "logit_scale"
            ]
        }
    }
}
```

You can change these parameters as you require.

3. Use optuna to automatic get the best param

You can enable optuna to automatic get the best param by adding `optuna_cfg` configs to config files like:

```
    "optuna_cfg": {
        "n_trials": 30,
        "n_warmup_steps":10,
        "sampler": {
            "name": "TPESampler"
        },
        "opt_params": {
            "coef_lr": {
                "range": [0.02,0.5],
                "log": false
            },
            "weight_decay": {
                "range": [0.01,0.5],
                "log": false
            }
        }
    }
```

Optuna Parameter description:

Config name	Description
n_trials	The max number of trials. Must be set to an integer.
n_warmup_steps	The pruning is disabled until the trial exceeds the given number of step(epochs). Note that this feature assumes that step starts at zero.
sampler	Choose samplers which optuna uses. now support `TPESampler`, `CmaEsSampler` and `GPSampler`.
opt_params	The parameters you want to optimize.

Configs of opt_params	Description
range	The min and max value of the parameter.
log	A flag to sample the value from the log domain or not. If log is true, the value is sampled from the range in the log domain. Otherwise, the value is sampled from the range in the linear domain.

The config example is: GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune/cfgs/bitfit-optuna.json

If you want to continue finetune models with the best parameters after optuna optimization, add “do_training_af_optuna”: true into the configuration file.

4.Fine-Tuning AdaCLIP with LLaMA Board GUI (powered by Gradio)

4.1 data preparation

- If you want to run AdaCLIP finetune with **LLaMA Board GUI,** you need a few steps to get your dataset displayed in the UI interface:
  1. Create **ActivityNet.json** under `GenAIComps/comps/finetuning/src/integrations/xtune/data`:

     Just save `[]` in the json is ok.

  2. Add data information into: `GenAIComps/comps/finetuning/src/integrations/xtune/data/dataset_info.json`:

4.2 Start llamafactory UI:

Run with A100:

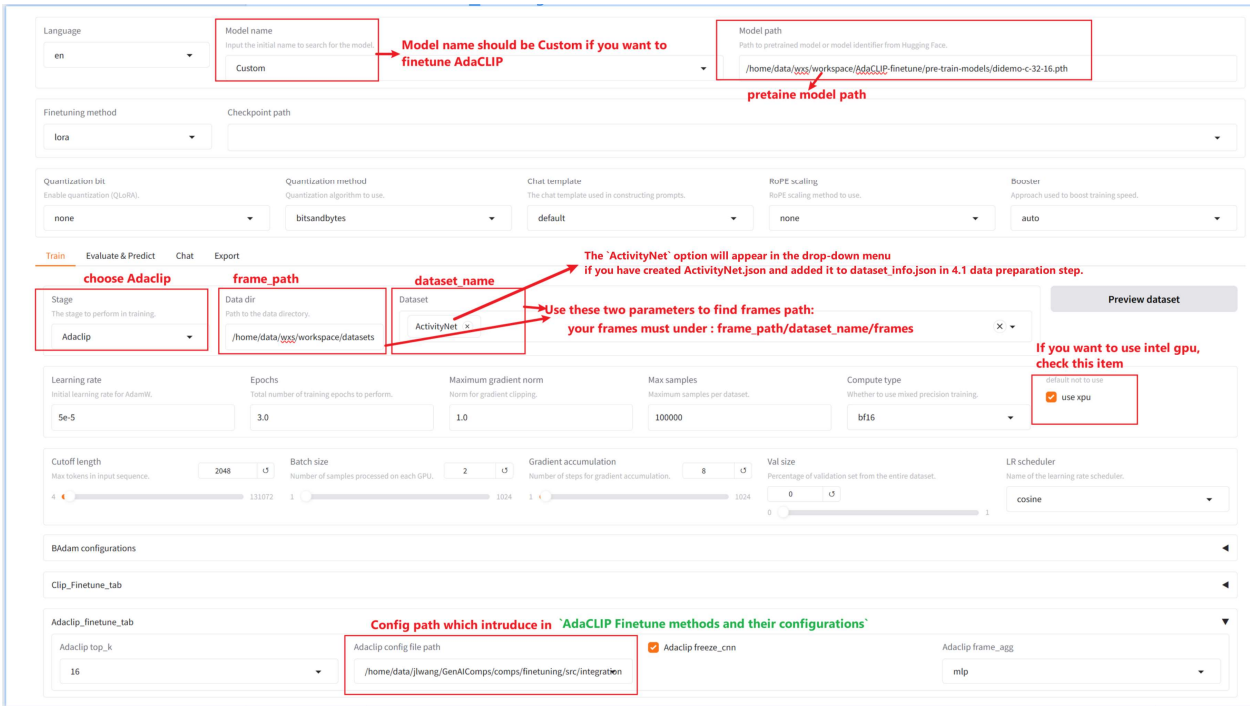
```
CUDA_VISIBLE_DEVICES=0 llamafactory-cli webui
```

Run with ARC770:

```
ZE_AFFINITY_MASK=0 llamafactory-cli webui
```

Then access in web through http://localhost:7860/

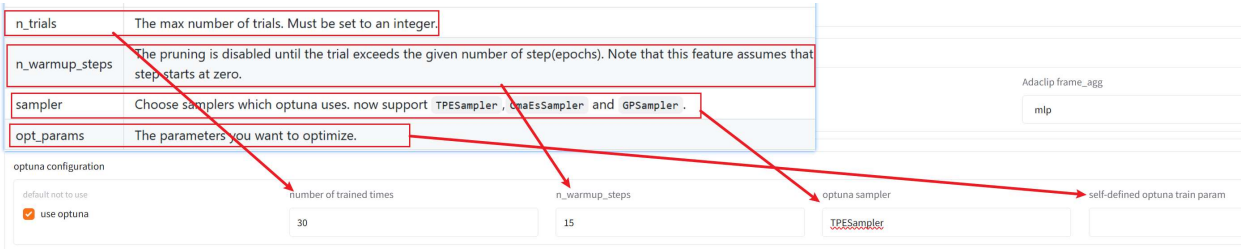
4.3 Config in UI page:



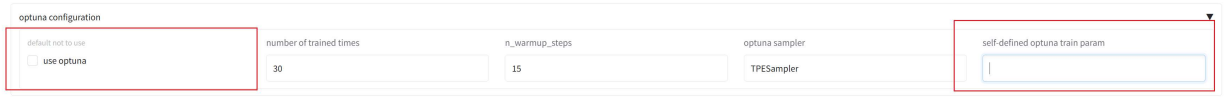
4.4 (Optional) Use Optuna to do parameters tuning:

If you want to use Optuna to tune some parameters, check `use optuna ` and set other optuna parameters.

The parameters correspond as follows

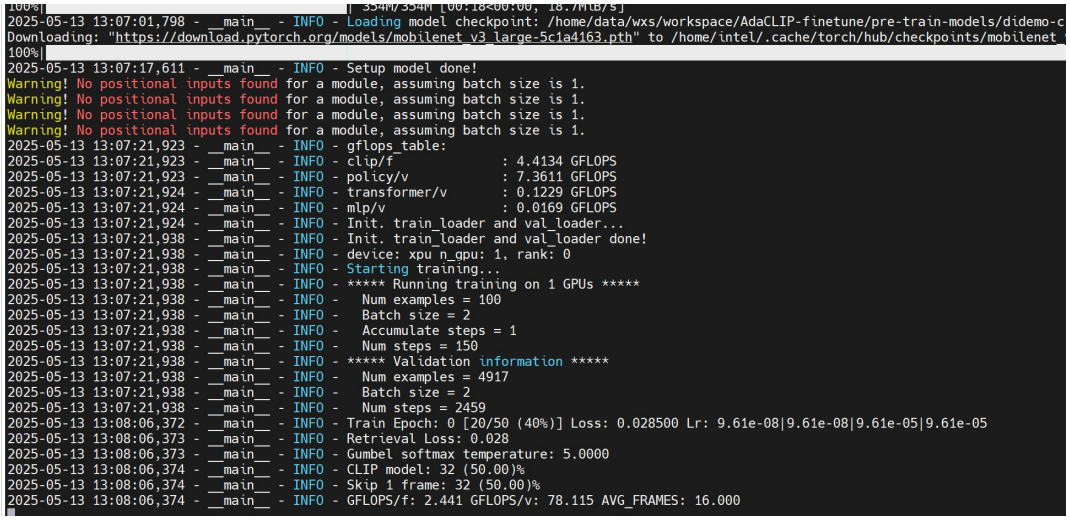


If you don't want to use Optuna parameters tuning, please *clear* `self-defined optuna train param` and do not check the `use optuna` box.



4.5 Check `Start` to start finetune:

You can check the finetune status in the terminal which running **llamafactory-cli webui**:



5. Fine-Tuning with Shell instead of GUI

You can find more details in:
GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune/README.md

5.1 How to Finetune

You can finetune AdaCLIP by using configs under :GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune/cfgs.

You can modify the information in config jsons to meet your requirements, like `train_annot`,`val_annot` and `test_annot` in the configs according to your own dataset.

5.2 Finetune on NVIDIA

5.2.1 Enter project folder:

```
cd GenAIComps/comps/finetuning/src/integrations/xtune/src/llamafactory/adaclip_finetune
```

5.2.1 Finetune AdaCLIP with bitfit:

```
python  train.py --config src/llamafactory/adaclip_finetune/cfgs/bitfit.json --frames_dir /path/to/frames --top_k 16 --freeze_cnn --frame_agg mlp --resume /path/to/pre-train/model --batch_size 8
```

5.2.3 Finetune AdaCLIP with ibs

```
python  train.py --config
src/llamafactory/adaclip_finetune/cfgs/ibs.json --frames_dir
/path/to/frames --top_k 16 --freeze_cnn --frame_agg mlp --resume
/path/to/pre-train/model --batch_size 8
```

5.2.4 Full finetune

```
python  train.py --config src/llamafactory/adaclip_finetune/cfgs/full-
finetune.json --frames_dir  /path/to/frames --top_k 16 --freeze_cnn --
frame_agg mlp --resume /path/to/pretrain/model --batch_size 8
```

5.3 Finetune on Arc A770

5.3.1 Specify the XPU :

Currently only single card finetune is supported, you can specify the XPU with the following command:

```
export ZE_AFFINITY_MASK=the_card_number
```

5.3.2 Enter the AdaCLIP folder:

```
cd src/llamafactory/adaclip_finetune
```

5.3.3 Finetune AdaCLIP with bitfit

```
python  train.py --config
src/llamafactory/adaclip_finetune/cfgs/bitfit.json --frames_dir
/path/to/frames --top_k 16 --freeze_cnn --frame_agg mlp --resume
/path/to/pretrain/model --xpu --batch_size 8
```

5.3.4 Finetune AdaCLIP with ibs

```
python  train.py --config
src/llamafactory/adaclip_finetune/cfgs/ibs.json --frames_dir
/path/to/frames --top_k 16 --freeze_cnn --frame_agg mlp --resume
/path/to/pretrain/model --xpu --batch_size 8
```

5.3.5 Full finetune

```
python  train.py --config src/llamafactory/adaclip_finetune/cfgs/full-
finetune.json --frames_dir  /path/to/frames --top_k 16 --freeze_cnn --
frame_agg mlp --resume /path/to/pretrain/model --xpu --batch_size 8
```

The finetune output will locate in `src/llamafactory/adaclip_finetune/output`

5.4 Using Optuna

5.4.1 Run Optuna

```
cd src/llamafactory/adaclip_finetune
```

```
python train.py --config ./cfgs/bitfit-optuna.json --frames_dir
/path/to/frames --top_k 16 --freeze_cnn --frame_agg mlp --resume
/path/to/pre-train/model --xpu --batch_size 8
```

5.4.2 Visualization

You can review optuna tuning results by:

```
sudo ufw allow 8084
```

```
optuna-dashboard --host 0.0.0.0 --port 8084 sqlite:///optuna.db
```

Open in the website:

```
http://<serverIP>:8084/dashboard
```

You can see finetune curves for different parameters and other information in the website: