torchdistill Meets Hugging Face Libraries for Reproducible, Coding-Free Deep Learning Studies: A Case Study on NLP

Yoshitomo Matsubara*

(University of California, Irvine)

*This work was done prior to joining Amazon.

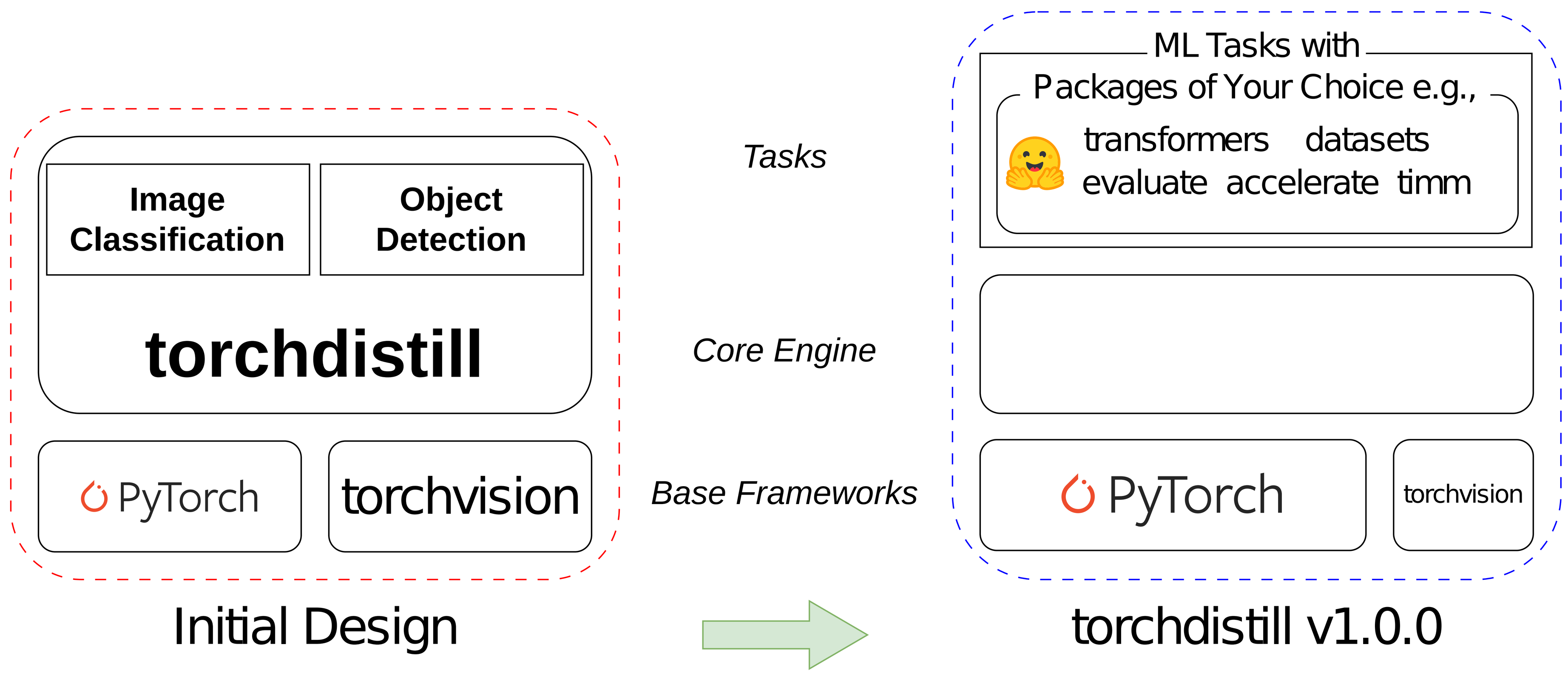
Paper | Code | Models | Documentation

torchdistill v1.0.0

```
$ pip3 install torchdistill
```

- Machine Learning Open Source Software (ML OSS) build on PyTorch
- Lower barriers to **reproducible**, **coding-free** deep learning/knowledge distillation studies
- Less dependent on torchvision and support more tasks
- PyYAML configuration-driven framework
  i.e., a config file defines an experiment

```
$ python3 your_script.py --config your_config.yaml --run_log your_log.txt
```



Initial Design (Matsubara 2021) → torchdistill v1.0.0

PyYAML config file defines the following:

- Dependencies
- Datasets
- Preprocessing
- Data loaders
- Models
- Model wrappers
- Forward hooks
- Forward interfaces
- Loss functions
- Optimizer
- LR scheduler
- Stage-wise configs
- and more!

Pipeline w/ abstracted modules

- Package imports
- Datasets & Loaders
- Model initialization
- Training + Epoch-wise validation
- Evaluation (Test)

Evaluation result, training log, model weights

Google Colab Demos

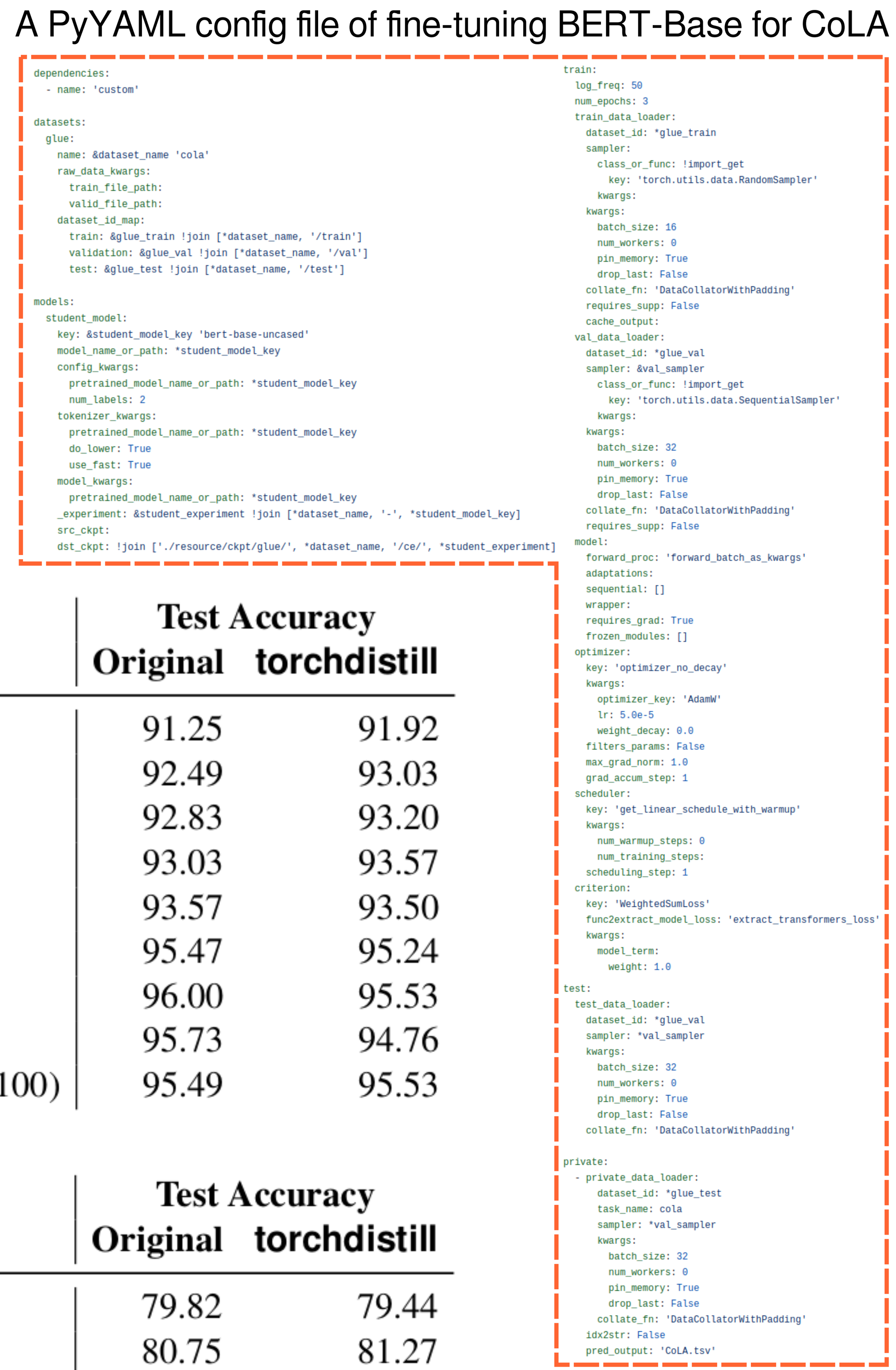
(Compatible with Amazon SageMaker Studio Lab)

Reproducing BERT GLUE Results + Knowledge Distillation

- Fine-tuned BERT-Large & Base, following (Devlin et al., 2019)
- Used the fine-tuned BERT-Large models as teachers for KD
- Achieved the reported test results and published models at Hugging Face
- Hinton et al. (2014)'s KD outperformed fine-tuned BERT-Base

Model (Method, Reference)	MNLI-(m/mm) Acc./Acc.	QQP F1	QNLI Acc.	SST-2 Acc.	CoLA M Corr.	STS-B P-S Corr.	MRPC F1	RTE Acc.	WNLI Acc.
BERT-Large (FT, Devlin et al. (2019))	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	N/A
BERT-Large (FT, Ours)	86.4/85.7	72.2	92.4	94.6	61.5	85.0	89.2	68.9	65.1
BERT-Base (FT, Devlin et al. (2019))	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	N/A
BERT-Base (FT, Ours)	84.2/83.3	71.4	91.0	94.1	51.1	84.4	86.8	66.7	65.8
BERT-Base (KD, Ours)	85.9/84.7	72.8	90.7	93.7	57.0	85.6	87.5	66.7	65.1

A PyYAML config file of fine-tuning BERT-Base for CoLA



Reproducing CIFAR-10/100 Results

- Reproduced the test results, following the original studies
- Pretrained models are available in torchdistill e.g.,

```
>>> from torchdistill.models.classification.resnet import resnet20
>>> cifar10_resnet20 = resnet20(num_classes=10, pretrained=True)
```

CIFAR-10 Model	Test Accuracy Original	Test Accuracy torchdistill
ResNet-20	91.25	91.92
ResNet-32	92.49	93.03
ResNet-44	92.83	93.20
ResNet-56	93.03	93.57
ResNet-110	93.57	93.50
WRN-40-4	95.47	95.24
WRN-28-10	96.00	95.53
WRN-16-8	95.73	94.76
DenseNet-BC (k=12, depth=100)	95.49	95.53

CIFAR-100 Model	Test Accuracy Original	Test Accuracy torchdistill
WRN-40-4	79.82	79.44
WRN-28-10	80.75	81.27
WRN-16-8	79.57	79.26
DenseNet-BC (k=12, depth=100)	77.73	77.14

More Experiments

- More experiments are available in this paper e.g., ILSVRC 2012, PASCAL VOC 2012, COCO 2017
- All the trained models and PyYAML configurations in this paper are publicly available