Using the '[tta-conjugate](#)' repo.

Nick Swetlin

Problem Statement.

TTA: Given a source model and unlabeled test data, adapt the source model to fit the test data (produce the lowest error that you can).

Some factors that impact results:
- Loss function that trained source model.
- Loss function used in TTA.

Data From The Paper.

Bolded results mean the best (lowest) average error per specific (source model, TTA loss) combination.

5.2 Results on classifiers trained with cross-entropy

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (**ENT**)
CIFAR-10-C	✗	13.95 (±0.06)	13.97 (±0.04)	**12.60** (±0.04)	13.07 (±0.05)
	✓	13.95 (±0.06)	12.85 (±0.04)	**12.51** (±0.01)	**12.51** (±0.03)
CIFAR-100-C	✗	45.22 (±0.4)	39.80 (±0.18)	**38.52** (±0.16)	41.15 (±0.25)
	✓	45.22 (±0.4)	36.37 (±0.10)	37.38 (±0.06)	**36.10** (±0.07)
ImageNet-C	✗	**45.43** (±0.05)	45.68 (±0.01)	48.91(±0.03)	45.82(±0.01)
	✓	45.43 (±0.05)	45.61 (±0.01)	48.91(±0.04)	**45.36**(±0.01)

Data From The Paper.

Bolded results mean the best (lowest) average error per specific (source model, TTA loss) combination.

1. Source Model Loss (table)

5.2 Results on classifiers trained with cross-entropy

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (**ENT**)
CIFAR-10-C	✗	13.95 (±0.06)	13.97 (±0.04)	**12.60** (±0.04)	13.07 (±0.05)
	✓	13.95 (±0.06)	12.85 (±0.04)	**12.51** (±0.01)	**12.51** (±0.03)
CIFAR-100-C	✗	45.22 (±0.4)	39.80 (±0.18)	**38.52** (±0.16)	41.15 (±0.25)
	✓	45.22 (±0.4)	36.37 (±0.10)	37.38 (±0.06)	**36.10** (±0.07)
ImageNet-C	✗	**45.43** (±0.05)	45.68 (±0.01)	48.91(±0.03)	45.82(±0.01)
	✓	45.43 (±0.05)	45.61 (±0.01)	48.91(±0.04)	**45.36**(±0.01)

Data From The Paper.

Bolded results mean the best (lowest) average error per specific (source model, TTA loss) combination.

1. Source Model Loss (table)

2. Dataset (row)

5.2 Results on classifiers trained with cross-entropy

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (**ENT**)
CIFAR-10-C	✗	13.95 (±0.06)	13.97 (±0.04)	**12.60** (±0.04)	13.07 (±0.05)
	✓	13.95 (±0.06)	12.85 (±0.04)	**12.51** (±0.01)	**12.51** (±0.03)
CIFAR-100-C	✗	45.22 (±0.4)	39.80 (±0.18)	**38.52** (±0.16)	41.15 (±0.25)
	✓	45.22 (±0.4)	36.37 (±0.10)	37.38 (±0.06)	**36.10** (±0.07)
ImageNet-C	✗	**45.43** (±0.05)	45.68 (±0.01)	48.91(±0.03)	45.82(±0.01)
	✓	45.43 (±0.05)	45.61 (±0.01)	48.91(±0.04)	**45.36**(±0.01)

Data From The Paper.

Bolded results mean the best (lowest) average error per specific (source model, TTA loss) combination.

1. Source Model Loss (table)

2. Dataset (row)

3. Loss type used for TTA (column)

5.2 Results on classifiers trained with cross-entropy

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (ENT)
CIFAR-10-C	✗	13.95 (±0.06)	13.97 (±0.04)	**12.60** (±0.04)	13.07 (±0.05)
	✓	13.95 (±0.06)	12.85 (±0.04)	**12.51** (±0.01)	**12.51** (±0.03)
CIFAR-100-C	✗	45.22 (±0.4)	39.80 (±0.18)	**38.52** (±0.16)	41.15 (±0.25)
	✓	45.22 (±0.4)	36.37 (±0.10)	37.38 (±0.06)	**36.10** (±0.07)
ImageNet-C	✗	**45.43** (±0.05)	45.68 (±0.01)	48.91(±0.03)	45.82(±0.01)
	✓	45.43 (±0.05)	45.61 (±0.01)	48.91(±0.04)	**45.36**(±0.01)

Data From The Paper.

Bolded results mean the best (lowest) average error per specific (source model, TTA loss) combination.

1. Source Model Loss (table)

2. Dataset (row)

3. Loss type used for TTA (column)

5.2 Results on classifiers trained with cross-entropy

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (ENT)
CIFAR-10-C	✗	13.95 ($\pm$0.06)	13.97 ($\pm$0.04)	**12.60** ($\pm$0.04)	13.07 ($\pm$0.05)
	✓	13.95 ($\pm$0.06)	12.85 ($\pm$0.04)	**12.51** ($\pm$0.01)	**12.51** ($\pm$0.03)
CIFAR-100-C	✗	45.22 ($\pm$0.4)	39.80 ($\pm$0.18)	**38.52** ($\pm$0.16)	41.15 ($\pm$0.25)
	✓	45.22 ($\pm$0.4)	36.37 ($\pm$0.10)	37.38 ($\pm$0.06)	**36.10** ($\pm$0.07)
ImageNet-C	✗	**45.43** ($\pm$0.05)	45.68 ($\pm$0.01)	48.91($\pm$0.03)	45.82($\pm$0.01)
	✓	45.43 ($\pm$0.05)	45.61 ($\pm$0.01)	48.91($\pm$0.04)	**45.36**($\pm$0.01)

Results!

How To Recreate The Experiment

Workflow:

- **Edit config files (.yaml)**
- **Run commands!**

Workflow - Config Files

1. Source Model Loss (table)
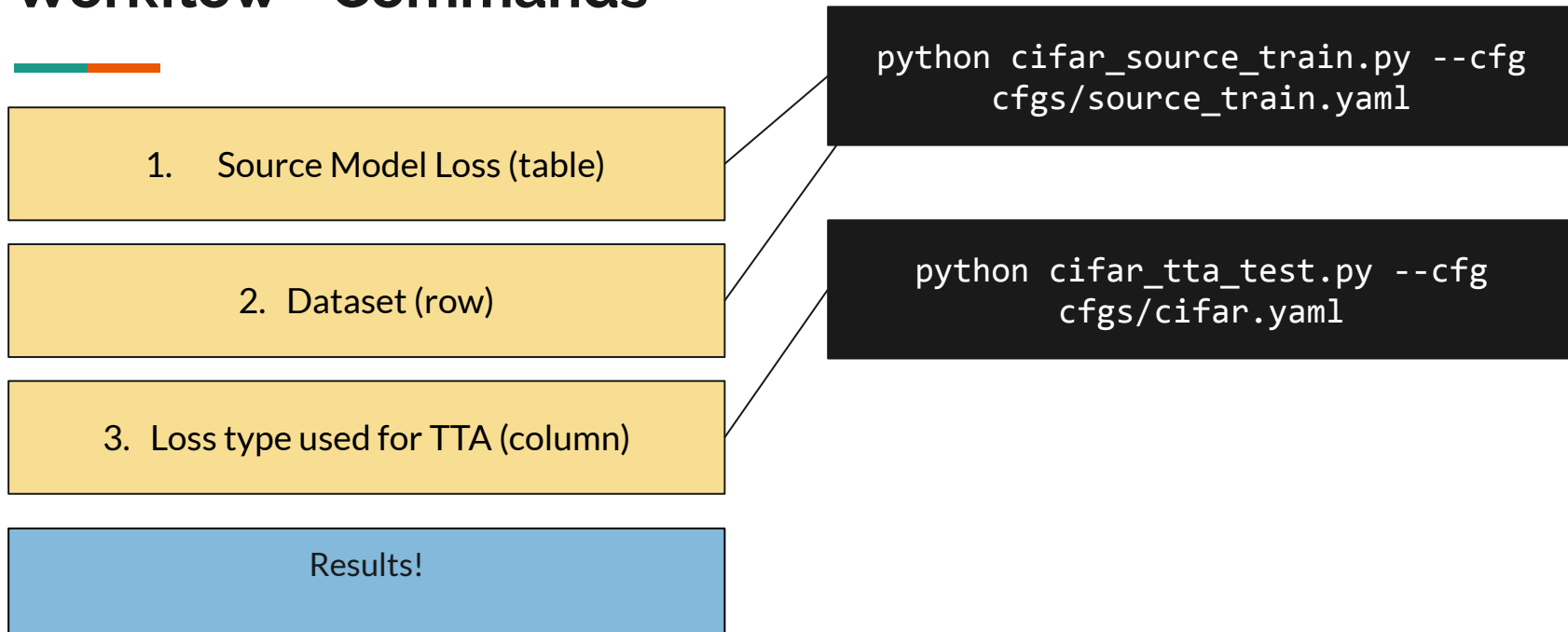
2. Dataset (row)

3. Loss type used for TTA (column)

*Results!

*make sure SAVE_PATH and CKPT_PATH are the same location

```yaml
source_train.yaml                          ×
1  MODEL:
2    LOSS: cross-entropy
3    DATASET: cifar10
4    SAVE_PATH: saved_models/pretrained/nick.pth
5    EPS: 6.0
```

```yaml
cifar.yaml
1   MODEL:
2     ADAPTATION: tent
3     ARCH: ResNet-18
4     CKPT_PATH: saved_models/pretrained/nick.pth
5     EPS: 6.0
6   TEST: ⋯
   CORRUPTION: ⋯
38  OPTIM:
39    METHOD: SGD
40    ADAPT: conjugate
41    STEPS: 1
42    BETA: 0.9
43    LR: 1e-3
44    TEMP: 1.0
45    WD: 0.
```

Workflow - Commands

1. Source Model Loss (table)

2. Dataset (row)

3. Loss type used for TTA (column)

Results!

```
python cifar_source_train.py --cfg
        cfgs/source_train.yaml
```

```
python cifar_tta_test.py --cfg
        cfgs/cifar.yaml
```

*These commands might differ depending on your source dataset, consult the [github](#) for more info.

My Experience

Ideal Situation: Run in parallel on DSMLP! By far the quicker option!

My Situation: Run locally on laptop overnight. It's a long job (10+ hours for single GPU)

Running gradient descent…

```
  PROBE_LAYERS: 1
Files already downloaded and verified
Files already downloaded and verified
C:\Users\nicks\anaconda3\lib\site-packages\torch\nn\_reduction.py:42:
deprecated, please use reduction='none' instead.
  warnings.warn(warning.format(ret))
Epoch : 0 : Acc : 0.4521999955177307
Epoch : 1 : Acc : 0.510699987411499
Epoch : 2 : Acc : 0.6006999611854553
```

```
Meta test begin!
[24/10/11 11:28:59] [cifar_tta_test.py:  218]: error % [gaussian_noise5]: 31.20%
Meta test begin!
[24/10/11 11:29:49] [cifar_tta_test.py:  218]: error % [shot_noise5]: 29.61%
Meta test begin!
[24/10/11 11:30:40] [cifar_tta_test.py:  218]: error % [impulse_noise5]: 37.25%
Meta test begin!
[24/10/11 11:31:30] [cifar_tta_test.py:  218]: error % [defocus_blur5]: 23.48%
Meta test begin!
[24/10/11 11:32:21] [cifar_tta_test.py:  218]: error % [glass_blur5]: 37.08%
Meta test begin!
[24/10/11 11:33:13] [cifar_tta_test.py:  218]: error % [motion_blur5]: 29.19%
Meta test begin!
[24/10/11 11:34:04] [cifar_tta_test.py:  218]: error % [zoom_blur5]: 22.25%
Meta test begin!
[24/10/11 11:34:55] [cifar_tta_test.py:  218]: error % [snow5]: 31.65%
Meta test begin!
[24/10/11 11:35:47] [cifar_tta_test.py:  218]: error % [frost5]: 29.94%
Meta test begin!
[24/10/11 11:36:39] [cifar_tta_test.py:  218]: error % [fog5]: 32.12%
Meta test begin!
[24/10/11 11:37:31] [cifar_tta_test.py:  218]: error % [brightness5]: 23.04%
Meta test begin!
[24/10/11 11:38:23] [cifar_tta_test.py:  218]: error % [contrast5]: 31.12%
Meta test begin!
[24/10/11 11:39:15] [cifar_tta_test.py:  218]: error % [elastic_transform5]: 30.89%
Meta test begin!
[24/10/11 11:40:07] [cifar_tta_test.py:  218]: error % [pixelate5]: 24.98%
Meta test begin!
[24/10/11 11:41:00] [cifar_tta_test.py:  218]: error % [jpeg_compression5]: 27.03%
Meta test begin!
```

Average Error on Severity 5 Distribution Shifts:
-   **29.83%**

Significantly better than chance alone after only ~10 iterations of grad. descent.

Comparing To The Paper.

Paper produced significantly better results due to:
- More iterations of grad. Descent!
- Temperature hyperparameter

- -

5.2 Results on classifiers trained with cross-entropy

- **29.83 vs 12.51** percent error.

Dataset	Temperature (T)	Hard PL	Robust PL	MEMO	Conjugate PL (**ENT**)
CIFAR-10-C	✗	13.95 (±0.06)	13.97 (±0.04)	**12.60** (±0.04)	13.07 (±0.05)
	✓	13.95 (±0.06)	12.85 (±0.04)	**12.51** (±0.01)	**12.51** (±0.03)
CIFAR-100-C	✗	45.22 (±0.4)	39.80 (±0.18)	**38.52** (±0.16)	41.15 (±0.25)
	✓	45.22 (±0.4)	36.37 (±0.10)	37.38 (±0.06)	**36.10** (±0.07)
ImageNet-C	✗	**45.43** (±0.05)	45.68 (±0.01)	48.91(±0.03)	45.82(±0.01)
	✓	45.43 (±0.05)	45.61 (±0.01)	48.91(±0.04)	**45.36**(±0.01)

Takeaways

- **You should run this code.**
    - **Teaches terminal behavior with machine learning.**
    - **Familiarizes PyTorch architecture.**
    - **Live observation of each step.**
- **Always preferable to <u>run on DSMLP, background Kubernetes pod with multiple GPUS.</u>**
    - Much faster than locally.
    - If you'd like to run it locally, ask me for tweaks I made.
- **Commands are listed openly on the <u>github</u> for any table entry from the experiment you want to recreate.**