Revisiting Realistic Test-Time Training: Sequential Inference and Adaptation by Anchored Clustering

Yongyi Su, Xun Xu, Kui Jia

Motivation

Confused definitions of Test-Time Training (TTT):

- TENT [1] & BN [2] make predictions **instantly** during test time, while SHOT [3] & TTT++ [4] need to **train multiple epochs** before making predictions.
- TTT-R [5] & TTT++ [4] need to **train an additional self-supervised branch**.

Definition of a realistic TTT setting (sTTT):

- **One-pass adaptation**: test samples are sequentially streamed and prediction must be made instantly.
- **No modifications to training objective**, e.g. self-supervised branch.
- Not restricting access to a **light-weight information** from the source domain.

Contribution

- ✓ In light of the confusions within TTT works, we provide **a categorization of TTT protocols** by two key factors. Comparison of TTT methods is now fair within each category.
- ✓ We adopt **a realistic TTT setting**, namely sTTT. To improve test-time feature learning, we propose TTAC by matching the statistics of the target clusters to the source ones. The target statistics are updated through moving averaging with filtered pseudo labels.
- ✓ The proposed method **is complementary to existing TTT method** and is demonstrated on **six TTT datasets**, achieving the state-of-the-art performance under all categories of TTT protocols.

Methodology

Anchored Clustering for Test-Time Training (Category-wise alignment)

- We allocate the same number of clusters in both source and target domains and each target cluster is assigned to one source cluster. In general, one cluster corresponds to one semantic category.
- Minimizing the KL-Divergence between each pair of clusters

$$\mathcal{L}_{ac} = \sum_k D_{KL}(\mathcal{N}(\mu_{sk}, \Sigma_{sk}) || \mathcal{N}(\mu_{tk}, \Sigma_{tk})) = \sum_k -H(\mathcal{N}(\mu_{sk}, \Sigma_{sk})) + H(\mathcal{N}(\mu_{sk}, \Sigma_{sk}), \mathcal{N}(\mu_{tk}, \Sigma_{tk}))$$

Clustering through Pseudo Labeling

- Temporal consistency filtering

$$F_i^{TC} = \mathbb{1}((P_{i\hat{k}}^t - \widetilde{P}_{i\hat{k}}^{t-1}) > \tau_{TC}), \quad s.t. \quad \hat{k} = \arg\max_k (P_{ik}^t)$$

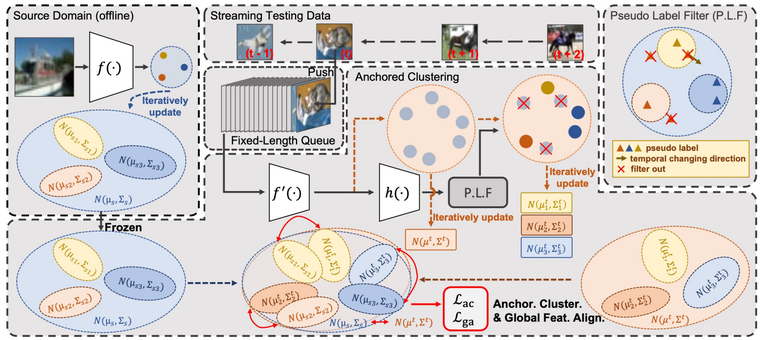
- Posterior probability filtering

$$F_i^{PP} = \mathbb{1}(\widetilde{P}_{i\hat{k}}^t > \tau_{PP})$$

To update the component Gaussian

$$\mu_{tk} = \frac{\sum_i F_i^{TC} F_i^{PP} \mathbb{1}(\hat{y}_i = k) f(x_i)}{\sum_i F_i^{TC} F_i^{PP} \mathbb{1}(\hat{y}_i = k)}, \quad \Sigma_{tk} = \frac{\sum_i F_i^{TC} F_i^{PP} \mathbb{1}(\hat{y}_i = k)(f(x_i) - \mu_{tk})^\top (f(x_i) - \mu_{tk})}{\sum_i F_i^{TC} F_i^{PP} \mathbb{1}(\hat{y}_i = k)}$$

Overview



Experiment

We categorize test-time training based on two key factors:

- Whether the training objective must be changed during training on the source domain. **Y**/**N** indicates modifying source domain training objective or not.
- Whether testing data is sequentially streamed and predicted. **M**/**O** indicates **m**ultiple passes or **o**ne pass test-time training.

Method	TTT Protocol	Assum. Strength	C10-C	C100-C	MN40-C
TEST	-	-	29.15	60.34	34.62
BN [13]	N-O	Weak	15.49	43.38	26.53
TENT [33]	N-O	Weak	14.27	40.72	26.38
T3A [14]	N-O	Weak	15.44	42.72	24.57
SHOT [21]	N-O	Weak	13.95	39.10	19.71
TTT++ [22]	N-O	Weak	13.69	40.32	-
TTAC (Ours)	N-O	Weak	**10.94**	36.64	22.30
TTAC+SHOT (Ours)	N-O	Weak	10.99	**36.39**	**19.21**
TTT++ [22]	Y-O	Medium	13.00	35.23	-
TTAC (Ours)	Y-O	Medium	**10.69**	**34.82**	-
BN [13]	N-M	Medium	15.70	43.30	26.49
TENT [33]	N-M	Medium	12.60	36.30	21.23
SHOT [21]	N-M	Medium	14.70	38.10	15.99
TTAC (Ours)	N-M	Medium	**9.42**	33.55	16.77
TTAC+SHOT (Ours)	N-M	Medium	9.54	**32.89**	**15.04**
TTT-R [29]	Y-M	Strong	14.30	40.40	-
TTT++ [22]	Y-M	Strong	9.80	34.10	-
TTAC (Ours)	Y-M	Strong	**8.52**	**30.57**	-

Ablation Study

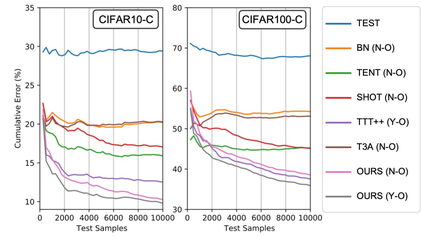
- Anchored Clustered combined with Pseudo Label Filtering makes a significant boost in performance. Error rate: 14.32% / 15.00% → 11.33%.
- Comparison on aligning global feature alone with KL-Divergence or L2 distance. Error rate: KL-Diver. vs L2 Dist.: 10.80% vs 11.87%.
- Combining all components yields the best performances consistently under all TTT protocols.

TTT Protocol	-	N-O				Y-O		N-M				Y-M	
Anchored Cluster.	-	✓	-	✓	-	✓	✓	✓	✓	-	-	✓	✓
Pseudo Label Filter.	-	-	✓	✓	-	✓	✓	-	✓	-	-	✓	✓
Global Feat. Align.	-	-	-	-	KLD	KLD	KLD	-	-	L2 Dist.[22]	KLD	KLD	KLD
Contrast. Branch [22]	-	-	-	-	-	-	✓	-	-	-	-	-	✓
Avg Acc	29.15	14.32	15.00	11.33	11.72	10.94	10.69	11.11	10.01	11.87	10.8	9.42	8.52

Additional Analysis

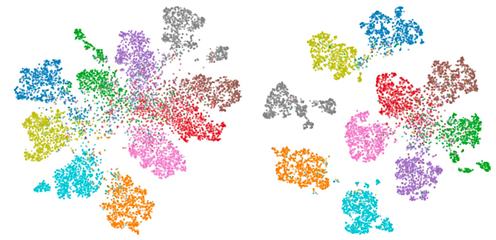
Cumulative performance under sTTT

TTAC outperforms all competing methods consistently throughout the TTT procedure.



TSNE Visualization of TTAC features

The features learned by TTAC have a better separation between classes.



(a) TTT++ Features (b) TTAC Features

Test Sample Queue and Update Epochs

- Maintaining a sample queue can substantially improve the performance of methods that estimate target distribution.
- Consistent improvement can be observed with increasing update epochs for SHOT[3] and TTAC.

	CIFAR10-C					ImageNet-C		
	w/ Queue				w/o Queue	w/ Queue		w/o Queue
#Epochs	1	2	3	4*	1	1	2*	1
BN	15.84	15.99	16.04	16.00	15.44	62.34	62.34	62.59
TENT	13.35	13.83	13.85	13.87	13.48	47.82	49.23	48.39
SHOT	13.96	13.93	13.83	13.75	15.18	46.91	46.09	51.46
TTAC	**10.88**	**10.80**	**10.58**	**9.96**	**11.91**	**45.44**	**44.56**	**46.64**

References

Dequan Wang et al. (2021). "Tent: Fully test-time adaptation by entropy minimization". ICLR 2021

Sergey Ioffe et al. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". ICML 2015

Jian Liang et al. (2020). "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation". ICML 2020

YuejiangLiu et al. (2021). "TTT++: When does self-supervised test-time training fail or thrive?". NeurIPS 2021

Yu Sun et al. (2020). "Test-time training with self-supervision for generalization under distribution shifts". ICML 2020