Neuro-Modulated Hebbian Learning for Fully Test-Time Adaptation

Yushun Tang[1,2] Ce Zhang[1] Heng Xu[1] Shuoshuo Chen[1]
Jie Chen[2] Luziwei Leng[2] Qinghai Guo[2*] Zhihai He[1,3*]

[1]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China

[2]Advanced Computing and Storage Laboratory, Huawei Technologies Co., Ltd., Shenzhen, China

[3]Pengcheng Laboratory, Shenzhen, China

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Poster Session TUE-AM-355

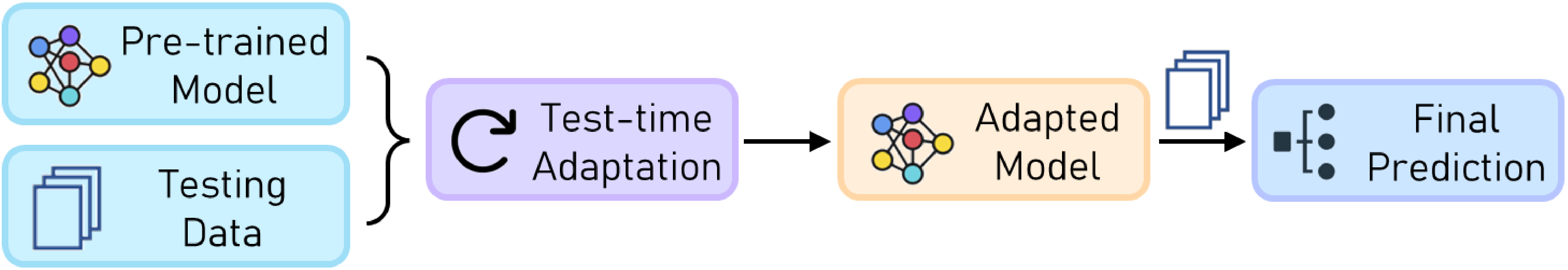
Highlights

- We identify that the major challenge in fully test-time adaptation lies in effective unsupervised learning of early layer representations, and explore neurobiology-inspired soft Hebbian learning for effective early layer representation learning and fully test-time adaptation.
- We develop a new neuro-modulated Hebbian learning method which combines unsupervised feed-forward Hebbian learning of early layer representation with a learned neuro-modulator to capture feedback from external responses. We analyze the optimal property of the proposed NHL algorithm based on free-energy principles.
- We evaluate our proposed NHL method on benchmark datasets for fully test-time adaptation, demonstrating its significant performance improvement over existing methods.

Problem Definition

- The performance of DNNs tends to degrade when there is data shift between the training data in the source domain and the testing data in the target domain.

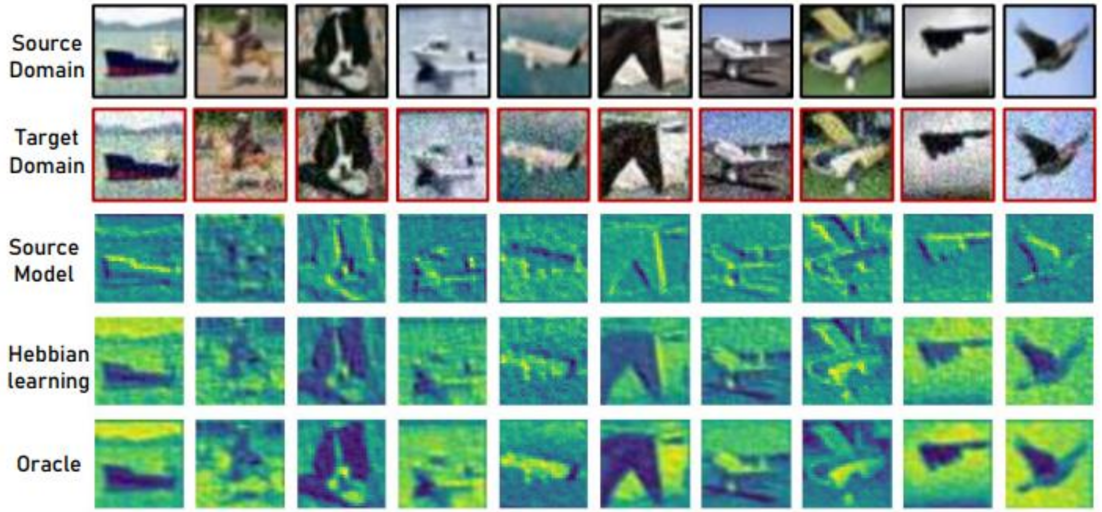
- Fully test-time adaptation, only needs access to live streams of test samples, which can dynamically adapt the source model on the fly during the testing process.

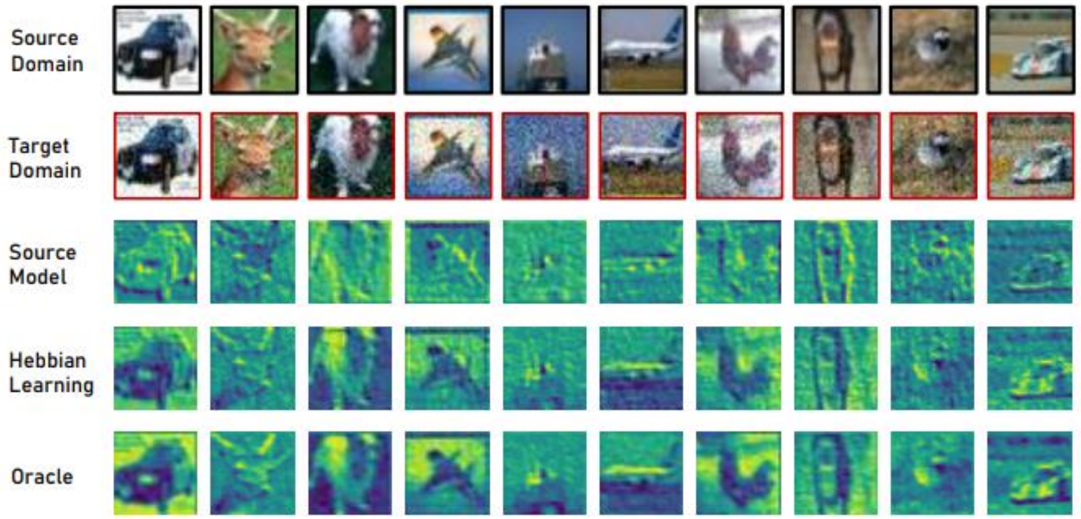


Main Idea

➢ In neural network models, the early layers of the network tend to respond to corners, edges, or colors. In contrast, deeper layers respond to more class-specific features. In the corruption test-time adaptation scenario, the class-specific features are always the same because the testing datasets are the corruption of the training domain. However, the early layers of models can be failed due to corruption.

➢ We first incorporate a soft decision rule into the feed-forward Hebbian learning to improve its competitive learning. Second, we learn a neuro-modulator to capture feedback from external responses, which controls which type of feature is consolidated and further processed to minimize the predictive error. During inference, the source model is adapted by the proposed NHL rule for each mini-batch of testing samples.



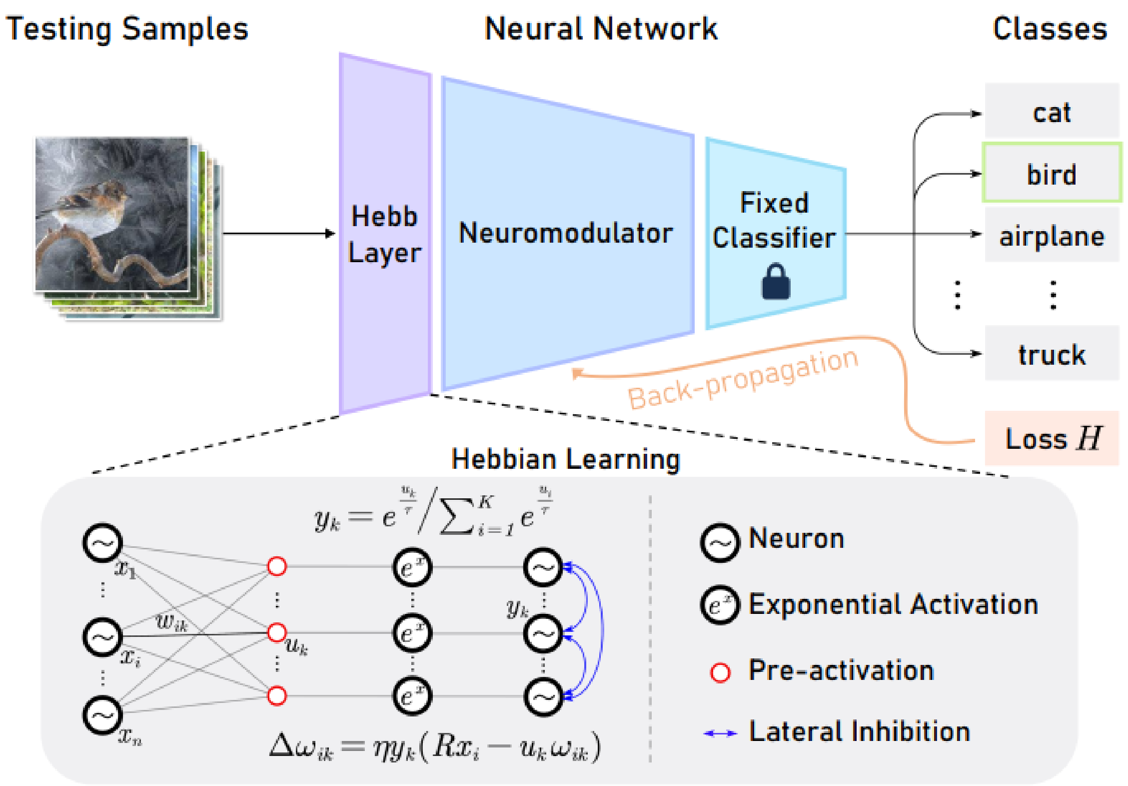


Method

- Our proposed neuro-modulated Hebbian learning consists of two major components: the feed-forward soft Hebbian learning layer and the neuro-modulator.

- The soft Hebbian learning layer aims to learn useful early layer representations without supervision based on local synaptic plasticity and soft competitive learning rules. It is able to generate early representations which are as good as those learned by end-to-end supervised training with labeled samples and back-propagation.

- During our experiments, we find that this soft Hebbian learning layer can significantly improve the performance of the network model in the target domain.



Method

- To approximate the true distribution $p_t(y|X_t)$ by a posterior approximation $q_2(y) := q_{\theta,t}(y|X_t)$, one can consider the similarity between these two distributions measured by the following Kullback-Leibler (KL) divergence

$$\mathrm{KL}[q_2(y)||p_t(y|X_t)] = \int q_2(y) \log \frac{q_2(y)}{p_t(y|X_t)} dy$$

- this minimization of KL-divergence can be converted to minimization of the *free-energy* F defined as:

$$F = \int q_2(y) \log \frac{q_2(y)}{p_t(X_t, y)} dy$$

Method

- The problem of minimizing the KL-divergence for $q_2(y)$ and its true posterior $p_t(y|X_t)$ can be formulated based on the free-energy principle:

$$\mathrm{KL}[q_2(y)||p_t(y|X_t)] = F + \log P_t(X_t)$$

where $P_t(X_t) := \int q_2(y)\, p_t(X_t)\, dy = p_t(X_t)$ is the normalization term. Note that this term does not depend on $q_2(y)$. Therefore, minimizing the KL-divergence is reduced to minimizing F. To this end, given a batch B of data in the target domain, we rewrite (use $p_t(y,B) = p_t(B|y)\, p_t(y)$) and decompose the free-energy F_B for current batch into the following two items:

$$F_B = \mathrm{KL}[q_2(y|B)||p_t(y)] - \int_y q_2(y|B) \log p_t(B|y) dy$$

- The first term is already minimized through soft Hebbian learning, while minimizing the second term requires the likelihood distribution $p_t(B|y)$. Since $p_t(B|y) = p_t(y|B)\, p_t(B)/p_t(y)$ and $q_2(y|B)$ is considered as an approximation of $p_t(y|B)$, we minimize the entropy of y given the current batch B in a discrete way as:

$$\arg\min_{\boldsymbol{w}} \; H(y|B) = \arg\max_{\boldsymbol{w}} \sum q_2(y|B) \log q_2(y|B)$$

Experimental Results

- Top-1 Classification Error (%) for each corruption in **CIFAR-10C** at the highest severity (Level 5). Backbones: ResNet-26 (top), WRN-28-10 (middle), and WRN-40-2 (bottom).

Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	67.7	63.1	69.9	55.3	56.6	42.2	50.1	31.6	46.3	39.1	17.1	74.6	34.2	57.9	31.7	49.2
TTT [64]	45.6	41.8	50.0	21.8	46.1	23.0	23.9	29.9	30.0	25.1	**12.2**	23.9	**22.6**	47.2	**27.2**	31.4
NORM [60]	44.6	43.7	49.1	29.4	45.2	26.2	26.9	25.8	27.9	23.8	18.3	34.3	29.3	37.0	32.5	32.9
TENT [66]	39.4	38.8	47.9	19.9	45.0	23.2	20.6	28.1	32.1	24.5	16.1	26.7	32.4	30.6	35.5	30.7
DUA [41]	34.9	32.6	42.2	18.7	**40.2**	24.0	18.4	**23.9**	**24.0**	20.9	12.3	27.1	27.2	26.2	28.7	26.8
Ours	**33.2**	**30.6**	**38.2**	**17.7**	41.2	**20.8**	**17.4**	24.0	27.2	**20.4**	13.5	**21.1**	28.4	**23.7**	28.9	**25.8**
Source	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	26.6	58.5	30.3	43.5
NORM [60]	28.1	26.1	36.3	12.8	35.3	14.2	12.1	17.3	17.4	15.3	8.4	12.6	23.8	19.7	27.3	20.4
TENT [66]	24.8	23.5	33.0	12.0	31.8	13.7	**10.8**	15.9	16.2	13.7	7.9	12.1	22.0	17.3	24.2	18.6
DUA [41]	27.4	24.6	35.3	13.1	34.9	14.6	11.6	16.8	17.5	13.1	**7.6**	14.1	22.7	19.3	26.2	19.9
Ours	**23.6**	**21.4**	**30.9**	**11.0**	**31.1**	**13.0**	10.9	**14.2**	**15.5**	**13.0**	8.0	**10.3**	**21.8**	**16.7**	**22.4**	**17.6**
Source	28.8	22.9	26.2	9.5	20.6	10.6	9.3	14.2	15.3	17.5	7.6	20.9	14.7	41.3	14.7	18.3
NORM [60]	18.7	16.4	22.3	9.1	22.1	10.5	9.7	13.0	13.2	15.4	7.8	12.0	16.4	15.1	17.6	14.6
TENT [66]	15.7	13.2	18.8	7.9	18.1	9.0	8.0	10.4	10.8	12.4	6.7	10.0	14.0	11.4	14.8	12.1
DUA [41]	15.4	13.4	17.3	8.0	18.0	9.1	**7.7**	10.8	10.8	12.1	6.6	10.9	13.6	13.0	14.3	12.1
Ours	**13.4**	**12.3**	**15.0**	**7.5**	**16.0**	**8.7**	**7.7**	**9.1**	**9.6**	**10.1**	**6.4**	**8.2**	**13.3**	**9.3**	**13.3**	**10.7**

Experimental Results

➢ Top-1 Classification Error (%) for each corruption in **CIFAR-100C** at the highest severity (Level 5).

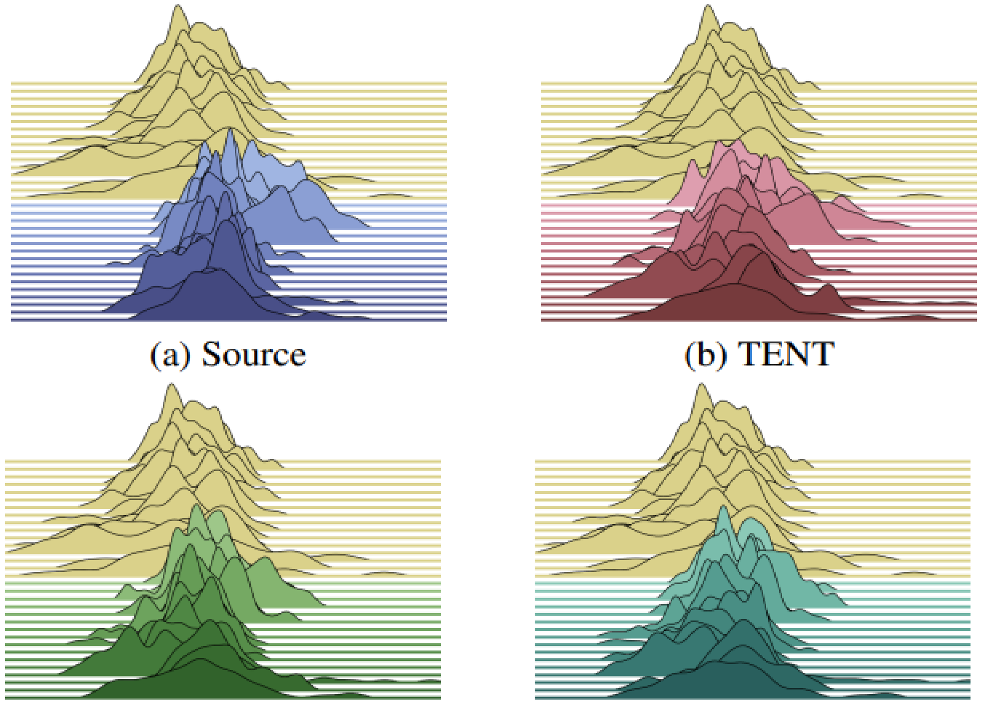
Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	65.7	60.1	59.1	32.0	51.0	33.6	32.4	41.4	45.2	51.4	31.6	55.5	40.3	59.7	42.4	46.7
NORM [60]	44.7	44.2	47.4	32.4	46.4	32.9	33.0	39.0	38.4	45.3	30.2	36.6	40.6	37.2	44.2	39.5
TENT [66]	40.3	39.9	41.8	29.8	42.3	31.0	30.0	34.5	35.2	39.5	28.0	33.9	38.4	33.4	41.4	36.0
DUA [41]	42.2	40.9	41.0	30.5	44.8	32.2	29.9	38.9	37.2	43.6	29.5	39.2	39.0	35.3	41.2	37.6
Ours	**38.4**	**37.1**	**36.2**	**28.4**	**41.0**	**29.3**	**29.7**	**32.2**	**33.1**	**36.1**	**26.4**	**30.9**	**36.2**	**30.8**	**38.3**	**33.6**

➢ Top-1 Classification Error (%) for each corruption in **ImageNet-C** at the highest severity (Level 5).

Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	98.9	97.6	99.2	93.3	89.0	90.2	82.3	87.9	92.0	99.5	75.9	99.5	65.3	60.3	54.0	85.7
TTT [64]	75.5	76.8	81.9	89.6	82.8	79.1	71.3	83.6	81.0	98.3	59.0	99.0	54.7	53.2	**49.6**	75.7
NORM [60]	60.2	60.7	59.8	76.6	68.7	67.4	64.2	64.6	66.2	74.7	57.0	88.8	55.8	53.0	52.3	64.7
TENT [66]	59.4	59.6	58.7	**72.5**	66.1	64.9	62.1	62.2	64.9	68.6	55.2	97.9	54.5	52.1	51.7	62.7
DUA [41]	71.9	72.6	72.4	90.2	80.8	83.1	74.7	76.4	77.9	87.3	62.6	99.3	60.8	58.4	52.6	74.7
Ours	**56.7**	**56.7**	**56.6**	73.3	**65.7**	**61.0**	**62.0**	**58.6**	**63.3**	**63.9**	**53.1**	**77.5**	**54.0**	**52.0**	51.5	**60.4**

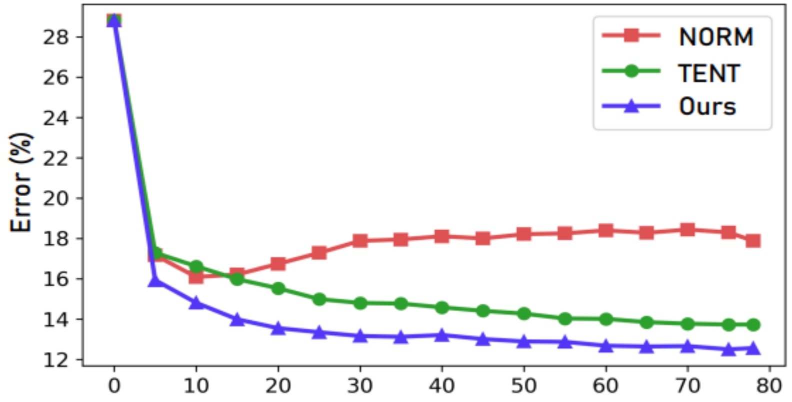
Experimental Results

- Density plots of adapted features distribution on CIFAR-10-C (Gaussian).



(a) Source (b) TENT (c) Ours (d) Oracle

- The mean error on CIFAR-10C (Gaussian) in different adaptation stages.



- Top-1 Classification Error (%) for test-time adaptation on digit recognition.

Methods	MNIST	MNIST-M	USPS	Avg.
NORM* [60]	39.6	52.1	41.4	44.4
TENT* [66]	45.8	56.2	48.3	50.1
Ours	**31.2**	**47.9**	**32.6**	**37.2**

Conclusion

- We identify that the major challenge in fully test-time adaptation lies in effective unsupervised learning of early layer representations, and explore neurobiology-inspired soft Hebbian learning for effective early layer representation learning and fully test-time adaptation.

- We develop a new neuro-modulated Hebbian learning method which combines unsupervised feed-forward Hebbian learning of early layer representation with a learned neuro-modulator to capture feedback from external responses.

- We analyze the optimal property of the proposed NHL algorithm based on free-energy principles.

- We evaluate our proposed NHL method on benchmark datasets for fully test-time adaptation, demonstrating its significant performance improvement over existing methods.