Supplementary Material: Energy-based Self-Training and Normalization for Unsupervised Domain Adaptation

Samitha Herath[1†] Basura Fernando[2] Ehsan Abbasnejad[3] Munawar Hayat[1] Shahram Khadivi[4]
Mehrtash Harandi[1] Hamid Rezatofighi[1] Gholamreza Haffari[1]
[1] Monash University, Australia [2] A*STAR, Singapore [3] The University of Adelaide, Australia [4] eBay Inc.
[†] samitha.herath1@monash.edu

1. Dataset Details

We conduct our experiments on DomainNet [6], OfficeHome [11] and VISDA2017 [7] datasets. For all our experiments we follow the protocol explained in [9].

DomainNet [6]. DomainNet is a large scale UDA benchmark for image classification task. The original DomainNet contains 0.6 million images, 6 domains and 345 classes. However, due to labeling noise, most recent work [8, 9] has adopted a cleaned version containing 40 commonly seen classes from 4 domains. We use this cleaned version of the DomainNet dataset for our experiments as in [8]. **VISDA2017** [7].VISDA2017 is a large scale dataset with the domain transfer synthetic$\rightarrow$ real. The dataset contains 12 classes and over $200K$ images. **OfficeHome** [11]. The OfficeHome dataset comprises 12 domain shifts, with each shift containing 65 classes observed in both home and office settings from 4 domains.

2. Scalability of SCAL+SCON to CNNs

While our proposed self-training method is entirely centered around ViT backbone, the SCAL+SCON modules are applicable to the CNN architecture (*i.e.*, ResNet50 [3]). In Table 1, we compare the performance when energy alignment and our normalization modules are used for distribution alignment. For this comparison, we use results for PADA [1] and DANN [2] reported in SENTRY [8]. Note, as in the case of SENTRY we compute the per-class-mean-accuracy as the evaluation metric. It can be seen that our method outperforms the considered adversarial methods.

3. Summary of Training Hyper-parameters

Optimizer parameters. We use AdamW [5] optimizer with a fixed learning rate of 2×10^{-4} with a mini-batch size of 512. For all OfficeHome [11] and DomainNet [6] we report results after 300 epochs of training. For VISDA2017 experiments, we report results after 20 epochs.

SEEBS+ parameters. We keep the loss weights, $\alpha_u = \alpha_{ea} = \alpha_n = 0.1$ constant for all experiments (*see* Overall training loss provided in the main text). We find the normalizer module parameter $\lambda = 0.01$ (*see* equation (15) of the main text) works best for DomainNet and OfficeHome experiments. For all other experiments we use $\lambda = 0.1$. In Fig. 1 we provide a

Method	Re2Cl	Cl2Pa	Pa2Re	Sk2Re
Source	65.8	60.6	84.5	77.1
PADA [1]	65.9	53.1	79.8	76.5
DANN [2]	63.4	65.7	86.9	85.7
SCON+SCAL	**78.0**	**73.8**	**89.7**	**87.6**

Table 1. SCAL/SCON modules for CNNs. Here, we compare the effectiveness of our proposed SCAL/SCON modules against other distribution alignment methods in CNN architecture.

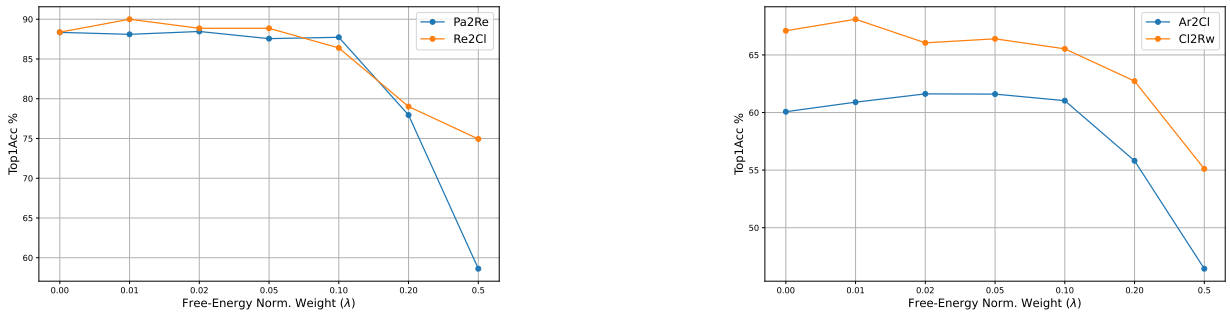

Figure 1. Sensitivity of the proposed **SEEBS+** method to the free-energy normalization weight λ *see* equation (15) of the main text. Here, we report the sensitivity analysis on two domain sets per each DomainNet **(Left)** and OfficeHome **(Right)** datasets.

Dataset	SOTA (i.e., PACMAC [9])	Ours	Relative Imp.%
OfficeHome	66.3	68.5	+3.3
DomainNet	81.5	83.3	+2.2
VISDA2017	77.1	79.5	+3.1

Table 2. Overall improvement from state-of-the-art UDA methods.

Ar2Cl	**Ar2Pr**	**Ar2Rw**	**Cl2Ar**	**Cl2Pr**	**Cl2Rw**	**Pr2Ar**	**Pr2Cl**	**Pr2Rw**	**Rw2Ar**	**Rw2Cl**	**Rw2Pr**
0.09	1.11	0.09	0.35	0.16	0.23	0.36	0.09	0.46	0.17	0.14	0.46
Re2Cl	**Re2Pa**	**Re2Sk**	**Cl2Re**	**Cl2Pa**	**Cl2Sk**	**Pa2Re**	**Pa2Cl**	**Pa2Sk**	**Sk2Re**	**Sk2Cl**	**SkPa**
0.49	0.31	0.72	0.77	0.27	0.31	0.41	0.48	0.23	0.34	0.41	0.40

Table 3. Standard deviation of **top-1 accuracy** for **SEEBS+** on three different runs with different random seeds.

sensitivity analysis on the free-energy normalization weight, λ. Here, we report the sensitivity analysis on two domain sets from DomainNet **(Left)** and OfficeHome **(Right)** datasets.

4. Overall Improvement

In Table 2, we report the relative improvement from our method compared to state-of-the-art UDA methods. As such, we compare with PACMAC [9]. We observe that our proposed method **SEEBS+** achieves a significant 2.2% - 3.3% relative improvement in comparison to PACMAC.

In Table 3, we report the standard deviation of the top-1 accuracy for the proposed **SEEBS+** for OfficeHome [11] (top row) and DomainNet [6] (bottom row), based on three separate runs with different random seeds. In almost all cases, the standard deviation is less than 1.0, with the exception of the **Ar2Pr** set in OfficeHome. However, it is worth noting that we outperform state-of-the-art methods in the **Ar2Pr** domain set by a margin of 2.5 in top-1 accuracy. Therefore, we claim that SEEBS+ is a method with consistent improvements over state-of-the-art UDA methods.

5. Quality of Instance Selections

In Fig. 2 we show the number of instances used for self-training as a proportion of the batch-size for each iteration. It can be seen that SEEBS+ outperforms PACMAC (*see* Table 2 of the main text) while using a significantly lower number of training instances for self-training. We attribute this observation to the better quality instance selections from SEEBS+.

To quantitatively evaluate the quality of instance selections, we compute the precision of our instance selections. More specifically, we compare them to ground truth instance selections where pseudo-labels are correct.

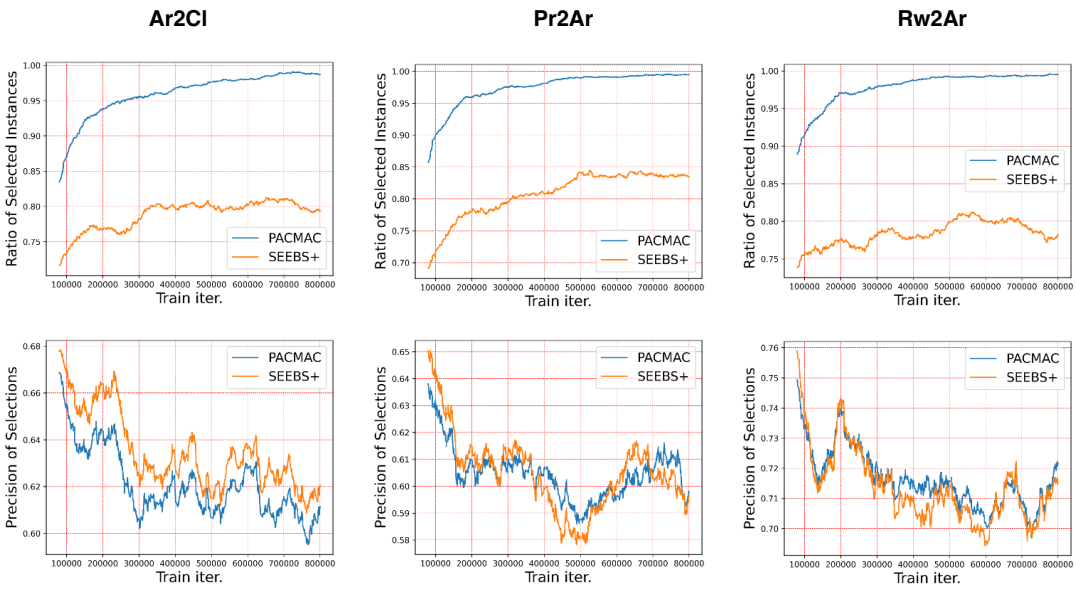

Figure 2. Here we compare the proportion and the quality of instance selections for PACMAC and SEEBS+. SEEBS+ outperform PACMAC [9] even with a significantly less number of instance selections (*see* Table 2 of the main text). We attribute this to the better quality of instance selections as shown by the precision plots.

Our analysis shows that, for a significant portion of the training, our proposed method, SEEBS+, exhibits better precision in instance selection than PACMAC. This is particularly evident in the early stages of training. This is important since a noisy self-training signal in the initial stages may cause the model to converge to an unfavorable local optimum. Previous research has emphasized the importance of steady self-training on the target domain during the early iterations of UDA. For instance, Dirt-t [10] proposes a constrained optimization technique for better self-supervision during early iterations. In our case, we improve the self-training by making reliable instance selections for the self-training objective.

6. Expanded Ablation Tables

Here, we present the expanded versions of the ablation experiments originally discussed in the main paper. In Table 4 we report the full domain set results demonstrating the significance of using the **Joint.** distribution-based instance sampling method.

In Table 5 we present an extended ablation experiment on the instance selection approach. This experiment serves to validate our instance selection approach explained in equation (4) of the main text.

In Table 6 we report the expanded set of experiments showing the significance of the SCON module when used for free-energy alignment (*i.e.*, SCAL + SCON).

In Table 7 we provide a comparison of the case when a naive normalization (*i.e.*, BatchNorm [4]) is used instead of the proposed SCON module. As per the expanded ablation tables, our choices proposed in this paper can be seen to perform better in a majority of domain sets.

	Re2Cl	Re2Pa	Re2Sk	Cl2Re	Cl2Pa	Cl2Sk	Pa2Re	Pa2Cl	Pa2Sk	Sk2Re	Sk2Cl	Sk2Pa	AVG
Cond.	87.8	82.7	77.7	78.7	73.6	76.1	88.2	83.5	77.8	83.6	82.6	78.6	**80.9**
Marg.	86.5	82.1	77.2	85.0	73.3	73.9	88.2	81.0	76.3	84.3	82.4	78.9	**80.7**
Joint.	87.1	**82.9**	**78.3**	**85.5**	**75.5**	**76.6**	87.8	82.7	**78.1**	82.5	**82.7**	78.0	**81.5**
	Ar2Cl	**Ar2Pa**	**Ar2Rw**	**Cl2Ar**	**Cl2Pr**	**Cl2Rw**	**Pr2Ar**	**Pr2Cl**	**Pr2Rw**	**Rw2Ar**	**Rw2Cl**	**Rw2Pr**	AVG
Cond.	60.4	64.7	74.6	60.5	69.1	66.7	56.7	58.5	74.2	70.6	63.8	81.3	**66.8**
Marg.	58.3	65.0	73.9	58.9	68.0	66.7	56.4	57.4	72.9	70.1	63.9	81.7	**66.1**
Joint.	**61.3**	**65.2**	74.0	60.4	**69.6**	**67.3**	**57.2**	**59.1**	73.9	**70.9**	**64.7**	**82.0**	**67.1**

Table 4. Expanded version of our analysis on self-training instance selection methods following conditional, marginal, and joint distributions.

	Re2Cl	Re2Pa	Re2Sk	Cl2Re	Cl2Pa	Cl2Sk	Pa2Re	Pa2Cl	Pa2Sk	Sk2Re	Sk2Cl	Sk2Pa	AVG
Sel-high	86.8	81.6	77.6	86.2	73.9	75.8	87.5	81.9	77.9	83.8	82.1	77.7	**81.1**
Sel-low	86.5	82.1	77.2	85.0	73.3	73.9	88.2	81.0	76.3	84.3	82.4	78.9	**80.7**
Joint-sel-high	86.3	82.3	77.7	77.4	72.0	75.2	87.9	80.0	78.3	84.2	82.1	77.5	**80.1**
Joint-sel-low	**87.1**	**82.9**	**78.3**	85.5	**75.5**	**76.6**	87.8	**82.7**	78.1	82.5	**82.7**	78.0	**81.5**
	Ar2Cl	**Ar2Pr**	**Ar2Rw**	**Cl2Ar**	**Cl2Pr**	**Cl2Rw**	**Pr2Ar**	**Pr2Cl**	**Pr2Rw**	**Rw2Ar**	**Rw2Cl**	**Rw2Pr**	AVG
Sel-high	59.8	66.3	73.4	59.3	68.1	66.1	56.4	58.4	72.9	70.0	64.2	81.8	**66.4**
Sel-low	58.3	65.0	74.1	59.3	68.0	66.7	56.4	57.4	72.9	70.1	63.9	81.7	**66.1**
Joint-sel-high	59.3	66.6	73.9	58.8	68.7	66.4	56.4	58.7	73.7	70.3	64.2	81.8	**66.6**
Joint-sel-low	**61.3**	65.2	74.0	**60.4**	**69.6**	**67.3**	**57.2**	**59.1**	**73.9**	**70.9**	**64.7**	**82.0**	**67.1**

Table 5. Expanded version of our analysis on the impact of using instances that meet the proposed free-energy based instance selection condition.

	Re2Cl	Re2Pa	Re2Sk	Cl2Re	Cl2Pa	Cl2Sk	Pa2Re	Pa2Cl	Pa2Sk	Sk2Re	Sk2Cl	Sk2Pa	AVG
Source	71.0	77.6	62.9	73.7	61.5	63.3	82.4	63.1	66.1	76.6	71.9	69.6	**70.1**
OnlySCAL	82.7	79.1	74.1	79.9	69.3	72.2	83.2	78.3	75.5	80.0	78.5	73.4	**77.2**
OnlySCON	65.3	46.3	42.7	44.0	30.7	41.1	52.2	51.6	43.6	42.4	58.1	38.4	**46.4**
SCAL+SCON	**88.6**	**81.7**	**80.5**	**86.0**	**79.1**	**76.8**	**86.9**	**80.8**	**79.0**	**84.1**	**82.7**	**79.0**	**82.1**
	Ar2Cl	**Ar2Pr**	**Ar2Rw**	**Cl2Ar**	**Cl2Pr**	**Cl2Rw**	**Pr2Ar**	**Pr2Cl**	**Pr2Rw**	**Rw2Ar**	**Rw2Cl**	**Rw2Pr**	AVG
Source	46.7	57.6	71.0	51.1	60.0	62.6	51.4	46.9	70.5	66.3	52.2	77.2	**59.4**
OnlySCAL	55.3	60.4	73.2	57.4	65.9	64.2	52.5	56.7	66.3	67.9	58.9	79.9	**63.2**
OnlySCON	51.3	55.6	69.8	39.5	44.1	40.1	44.3	36.9	50.7	64.9	48.1	65.4	**50.9**
SCAL+SCON	**60.5**	**67.7**	**74.9**	**62.1**	**68.8**	**68.0**	**60.9**	**60.2**	**75.4**	**71.5**	**66.9**	**81.9**	**68.2**

Table 6. Expanded version of the evaluation of the effects of the SCAL and SCON. Our results indicate that incorporating the normalization process, SCON, results in significant improvements.

	Re2Cl	Re2Pa	Re2Sk	Cl2Re	Cl2Pa	Cl2Sk	Pa2Re	Pa2Cl	Pa2Sk	Sk2Re	Sk2Cl	Sk2Pa	AVG
naiveSEEBS+	86.6	81.8	78.8	85.9	72.9	77.0	87.5	81.9	78.4	86.1	83.5	76.0	**81.4**
SEEBS+	**90.0**	**83.8**	**80.2**	**87.2**	**79.3**	**78.3**	**88.1**	**83.9**	**79.8**	84.6	**84.5**	**80.6**	**83.3**
	Ar2Cl	**Ar2Pr**	**Ar2Rw**	**Cl2Ar**	**Cl2Pr**	**Cl2Rw**	**Pr2Ar**	**Pr2Cl**	**Pr2Rw**	**Rw2Ar**	**Rw2Cl**	**Rw2Pr**	AVG
naiveSEEBS+	61.0	66.0	74.6	62.4	71.6	69.0	58.6	59.4	74.8	70.8	66.1	81.7	**68.0**
SEEBS+	60.9	**68.1**	**75.3**	62.1	68.7	68.1	**60.6**	**60.0**	**75.7**	**72.2**	**68.2**	**82.0**	**68.5**

Table 7. Expanded version of the reported comparisons on the effect of applying a BatchNorm instead of SCON.

References

[1] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150, 2018. 1

[2] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 1180–1189, 2015. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 448–456, 2015. 3

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[6] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019. 1, 2

[7] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 1

[8] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 8558–8567, 2021. 1

[9] Viraj Prabhu, Sriram Yenamandra, Aaditya Singh, and Judy Hoffman. Adapting self-supervised vision transformers by probing attention-conditioned masking consistency. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 23271–23283, 2022. 1, 2, 3

[10] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *Proc. Int. Conference on Learning Representations (ICLR)*, 2018. 3

[11] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5018–5027, 2017. 1, 2