Appendix: Uncertainty-guided Source-free Domain Adaptation

This appendix is organised as follows: Sec. A lists the notation used throughout the main text. Sec. B provides further details about Laplace approximations for approximate posterior inference in Bayesian neural networks. Sec. C provides the algorithm of U-SFAN. Sec. D includes additional summary statistics on the data sets used for the empirical evaluation and lists additional results on the Office31 and visda-c data set for the closed-set DA task.

A Notation

The following notation is used throughout the paper:

Notation	Description
$\mathcal{D}^{[\mathrm{S}]} = \{(\mathbf{x}_i^{[\mathrm{S}]}, \mathbf{y}_i^{[\mathrm{S}]})\}_{i=1}^{n^{[\mathrm{S}]}}$	Source data set
$\mathcal{D}^{[\mathrm{T}]} = \{\mathbf{x}_i^{[\mathrm{T}]}\}_{i=1}^{n^{[\mathrm{T}]}}$	Target data set
$\mathbf{x}^{[\mathrm{S}]} \in \mathcal{X}^{[\mathrm{S}]}$	Source inputs
$\mathbf{y}^{[\mathrm{S}]} \in \mathcal{Y}^{[\mathrm{S}]}$	Source class labels
$\mathbf{x}^{[\mathrm{T}]} \in \mathcal{X}^{[\mathrm{T}]}$	Target inputs
$\mathrm{L}^{[\mathrm{S}]}, \mathrm{L}^{[\mathrm{T}]}$	Label sets
f, f'	Model functions (source and target)
g	Feature extractor
h	Hypothesis function
β, θ	Parameterization of f and g
$\mathbf{z} = g(\mathbf{x})$	Latent feature of observation $\mathbf{x}$
K	Number of classes
$\phi_k(\cdot)$	Softmax function
$\mathbf{H}$	Hessian matrix

B Laplace Approximation

In Bayesian neural networks, we aim to incorporate uncertainty about the model and the model predictions. The standard approach places prior distributions ($p(\theta)$) onto the network parameters, which induces a probability distribution over the model predictions. By conditioning the prior (in the weight-space) onto observed source data ($\mathcal{D}^{[\mathrm{S}]}$), we obtain the posterior distribution over the network parameters $p(\theta \mid \mathcal{D}^{[\mathrm{S}]})$, allowing us to perform predictions by computing the posterior predictive distribution (see Eq. (4) in the main).

Let $\Psi(\theta)$ denote the unnormalised posterior distribution, *i.e.*,

$$\Psi(\theta) = p(\theta)\, p(\mathcal{D}^{[\mathrm{S}]} \mid \theta)\,, \tag{A1}$$

then the posterior distribution may be written as

$$p(\theta \,|\, \mathcal{D}^{[\mathtt{S}]}) = \frac{1}{Z_{\Psi}} \Psi(\theta)\,, \tag{A2}$$

where Z_{Ψ} denotes the normalisation constant. However, computing the posterior distribution and, subsequently, the posterior predictive distribution is intractable in general. We will, therefore, resort to a Laplace approximation to the posterior distribution.

Let θ_{MAP} denote the maximum or a mode of the posterior distribution in Eq. (A2). Then the second-order Taylor expansion of $\log \Psi(\theta)$ around θ_{MAP} is given as:

$$\log \Psi(\theta) \approx \log \Psi(\theta_{\mathrm{MAP}}) - \frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\,, \tag{A3}$$

where $\mathbf{H} = -\nabla^2_{\theta} \log \Psi(\theta) \,|\, _{\theta=\theta_{\mathrm{MAP}}}$ is the negative Hessian of the log joint ($\log \Psi(\theta)$) evaluated at θ_{MAP}. Substituting the value of $\log \Psi(\theta)$ in Eq. (A2) gives us:

$$\begin{aligned} p(\theta \,|\, \mathcal{D}) &= \frac{\Psi(\theta)}{\int \Psi(\theta)\, \mathrm{d}\theta} \\ &\approx \frac{\Psi(\theta_{\mathrm{MAP}}) \exp\left(-\frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\right)}{\Psi(\theta_{\mathrm{MAP}}) \int \exp\left(-\frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\right)\, \mathrm{d}\theta} \\ &= \frac{\exp\left(-\frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\right)}{\int \exp\left(-\frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\right)\, \mathrm{d}\theta}\,. \end{aligned} \tag{A4}$$

The posterior can now be calculated in closed-form, and is given by:

$$\begin{aligned} p(\theta \,|\, \mathcal{D}) &\approx \sqrt{\frac{\det \mathbf{H}}{2\pi}} \exp\left(-\frac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\, \mathbf{H}\, (\theta - \theta_{\mathrm{MAP}})\right) \\ &= \mathrm{N}(\theta \,|\, \mu_{\mathrm{MAP}}, \Sigma_{\mathrm{MAP}}), \end{aligned} \tag{A5}$$

where $\mu_{\mathrm{MAP}} = \theta_{\mathrm{MAP}}$ and $\Sigma_{\mathrm{MAP}} = \mathbf{H}^{-1}$.

The posterior predictive distribution of an unseen datum $\mathbf{x}^{[\mathtt{T}]}$ can now be approximated through Monte Carlo integration, *i.e.*,

$$p(\mathbf{x}^{[\mathtt{T}]} \,|\, \mathcal{D}^{[\mathtt{S}]}) \approx \frac{1}{M} \sum_{j=1}^{M} p(\mathbf{x}^{[\mathtt{T}]} \,|\, \theta_j)\,, \tag{A6}$$

where $\theta_j \sim \mathrm{N}(\theta \,|\, \mu_{\mathrm{MAP}}, \Sigma_{\mathrm{MAP}})$.

C Algorithm

We report the pseudo-code for our U-SFAN in Algo. 1.

Algorithm 1: Uncertainty-guided Source-free DA

Input : A probabilistic source model $f = h \circ g$ with parameters $\{\beta^{[\mathrm{S}]}_{\mathrm{MAP}}, \theta^{[\mathrm{S}]}_{\mathrm{MAP}}, \mathbf{H}^{-1}\}$, target data set $\mathcal{D}^{[\mathrm{T}]}$ containing $n^{[\mathrm{T}]}$ samples, mini-batch size b, temperature τ, and M MC steps.

Output: Target-specific feature extractor parameters $\beta^{[\mathrm{T}]}$.

repeat
- $\mathbf{X} \leftarrow \mathrm{sampleMiniBatch}(\mathcal{D}^{[\mathrm{T}]}, b)$
- $\mathbf{Z} \leftarrow g_{\beta^{[\mathrm{T}]}}(\mathbf{X})$
- $\hat{\mathbf{Y}} \leftarrow b \times K$ matrix of zeros
- ▷ Estimate predictive mean
- **for** $j = 1, \ldots, M$ **do**
  - $\theta_j \sim \mathrm{N}(\theta_j \mid \theta^{[\mathrm{S}]}_{\mathrm{MAP}}, \mathbf{H}^{-1})$
  - $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} + \mathrm{softmax}(h_{\theta_i}(\mathbf{Z})/\tau)$
- **end**
- $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}}/M$
- ▷ Compute model uncertainties
- **for** $i = 1, \ldots, b$ **do**
  - $w_i \leftarrow \exp(-H(\hat{\mathbf{y}}_i))$
- **end**
- Compute uncertainty-guided entropy ▷ Eq. (7)
- Compute divergence term ▷ Eq. (3)
- Compute U-SFAN loss
- Update parameters $\beta^{[\mathrm{T}]}$

until converged

D Data Set Details and Experiments

We have summarized the statistics of the SFDA benchmark data sets used for the comparison against the state-of-the-art in Table A1. To demonstrate the challenging aspect of having a strong domain-shift between the source and the target, we used the data set DOMAIN-NET. Moreover, the high number of semantic categories (345 classes) in DOMAIN-NET poses a challenge for the existing IM-based SFDA methods because of the lack of representative samples from every class in a given mini-batch.

Hyperparameter Selection. We re-use the hyperparameters from the baseline of [6], *e.g.*, the standard optimization technique for training such as SGD with an initial learning rate of 10^{-2} and 10^{-3} for ResNet-50 and ResNet-101, respectively. The learning rate is decayed by power decay [2]. We used the a batch size of 64 and we set $\alpha = 0.1$ and $\gamma = 0.5$. Exclusive to our method, we set the prior precision in LA equal to the weight decay, *i.e.* $5 \cdot 10^{-4}$, and set the temperature $\tau = 0.4$ for all our experiments.

Additionally, we have reported the results of the experiments on OFFICE31 in Table A2. Similar to the results obtained on the other data sets reported in the main paper, U-SFAN outperforms SHOT-IM on OFFICE31. It must be noted that for data sets like OFFICE31, the performance is already saturated, and the performance improvements of U-SFAN over SHOT-IM are minor. More-

Table A1. Data set summary for source-free domain adaptation

Data set	#domains	#classes	#images
Office31	3	31	4,652
Office-Home	4	65	15,500
Visda-C	2	12	$\sim$ 200K
Domain-Net	6	345	$\sim$ 0.6M

over, the data set shift is mild in most adaptation directions, evident from the saturated numbers. Thus, as discussed in the main paper, U-SFAN does not yield remarkable improvement when the domain-shift is milder, and is most effective when much of the target data resides outside the source manifold. Nevertheless, when our method is combined with nearest centroid pseudo-labelling (like in SHOT), U-SFAN+ further improve the performance. Through these extensive experiments on several SFDA benchmarks, we presented the advantages of our proposed method for the task of SFDA.

Table A2. Comparison of the classification accuracy on the Office31 for the closed-set SFDA using ResNet-50. Results on the small-scale office31 are known to be saturated. The visual appearance between the domains do not vary much, thus making the domain shift *milder*. The improvement of U-SFAN upon SHOT is moderate, but competitive w.r.t. A^{2}Net[11], which requires complex training objectives

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
ResNet-50	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN [3]	79.7	82.0	68.2	96.9	67.4	99.1	82.2
DAN [7]	78.6	80.5	63.6	97.1	62.8	99.6	80.4
SAFN [12]	90.7	90.1	73.0	98.6	70.2	99.8	87.1
CDAN [8]	92.9	94.1	71.0	98.6	69.3	100.	87.7
SHOT-IM [6]	90.6	91.2	72.5	98.3	71.4	99.9	87.3
U-SFAN (Ours)	91.8	92.3	75.8	97.7	74.4	99.8	88.6
A^{2}Net[11]	94.5	94.0	76.7	99.2	76.1	100.0	90.1
SHOT [6]	94.0	90.1	74.7	98.4	74.3	99.9	88.6
U-SFAN+ (Ours)	94.2	92.8	74.6	98.0	74.4	99.0	88.8

Due to lack of space in the main paper, in Table A3 we report the class-wise accuracy on the visda-c data set, whose average accuracy has been reported in the Table 4 (a) of the main paper. While our U-SFAN is competitive with SHOT-IM and SHOT, it underperforms with respect to A^{2}Net[11]. Nevertheless, U-SFAN does not optimize a multitude of loss functions, making it more intuitive than the A^{2}Net.

Table A3. Comparison of the classification accuracy on the Visda-C for the closed-set DA, pertaining to the *Synthetic* → *Real* direction, using ResNet-101. † indicates the numbers of [6] that are obtained using the official code from the authors. Note that several SFDA methods perform equally well for VISDA-C, hinting at saturating performance

METHOD	PLANE	BCYCL	BUS	CAR	HORSE	KNIFE	MCYCL	PERSON	PLANT	SKTBRD	TRAIN	TRUCK	AVG.
ResNet-101	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN [3]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
ADR [10]	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [8]	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [1]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN [12]	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [4]	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
DANCE [9]	-	-	-	-	-	-	-	-	-	-	-	-	70.2
SHOT-IM† [6]	94.2	87.6	78.6	48.6	92.1	92.9	76.4	76.2	89.4	86.6	88.8	52.7	80.3
U-SFAN (Ours)	95.1	87.0	76.8	50.1	92.9	94.3	79.0	78.0	88.4	87.5	87.7	57.3	81.2
3C-GAN [5]	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
A^2Net[11]	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
SHOT† [6]	94.9	87.1	76.9	55.0	94.2	95.4	80.8	80.0	89.5	88.7	85.6	60.5	82.4
U-SFAN + (Ours)	94.9	87.4	78.0	56.4	93.8	95.1	80.5	79.9	90.1	90.1	85.3	60.4	82.7

References

1. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 1081–1090 (2019)
2. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 1180–1189 (2015)
3. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(59), 1–35 (2016)
4. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10285–10295 (2019)
5. Li, R., Jiao, Q., Cao, W., Wong, H.S., Wu, S.: Model adaptation: Unsupervised domain adaptation without source data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9641–9650 (2020)
6. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 6028–6039 (2020)
7. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: Proceedings of the International conference on machine learning (ICML). pp. 97–105 (2015)
8. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1647–1657 (2018)

9. Saito, K., Kim, D., Sclaroff, S., Saenko, K.: Universal domain adaptation through self supervision. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
10. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Adversarial dropout regularization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
11. Xia, H., Zhao, H., Ding, Z.: Adaptive adversarial network for source-free domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 9010–9019 (2021)
12. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1426–1435 (2019)
13. Yang, S., Wang, Y., van de Weijer, J., Herranz, L., Jui, S.: Generalized source-free domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 8978–8987 (2021)