Learning CNN on ViT: A Hybrid Model to Explicitly Class-specific Boundaries for Domain Adaptation
(Supplementary Material)

Ba Hung Ngo[1,*], Nhat-Tuong Do-Tran[2,*], Tuan-Ngoc Nguyen[3], Hae-Gon Jeon[4], Tae Jong Choi[1,†]
[1]Graduate School of Data Science, Chonnam National University, South Korea
[2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[3]Digital Transformation Center, FPT Telecom, VietNam, [4]AI Graduate School, GIST, South Korea
ngohung@chonnam.ac.kr tuongdotn.cs11@nycu.edu.tw tuannn55@fpt.com haegonj@gist.ac.kr ctj17@jnu.ac.kr

In this Supplementary Materials, we provide expanded details on our analyses and additional experimental results. This section encompasses four key appendices: Appendix 1 delves into the notations used throughout our study. Appendix 2 describes the algorithm we adopted. Appendix 3 shows the sensitivity of two fixed thresholds, specifically τ_{vit} and τ_{cnn}, providing an in-depth analysis of their impact. Appendix 4 presents additional experiment results.

1. Notations

We present the notations commonly employed in our method, as outlined in Tab. 1.

2. Algorithm

In summary, the whole algorithm to train the proposed ECB is shown in Algorithm 1.

3. Sensitivity of threshold τ_{vit} and τ_{cnn}

In the semi-supervised domain adaptation (SSDA) scenario, we assess the performance sensitivity of our method on the CNN branch by varying the threshold values τ_{vit} and τ_{cnn} in the $rel \rightarrow clp$ scenario under 3-shot setting on *DomainNet* [13]. Specifically, we select threshold values of $\{0.6, 0.7, 0.8, 0.85, 0.9, \text{and } 0.95\}$ for our analysis as shown in Fig. 1. In total, 36 experiments are conducted to gauge the impact of these thresholds on our approach. Notably, the optimal performance of 87.4% is attained when $\tau_{vit} = 0.6$ and $\tau_{cnn} = 0.9$. This suggests that ViT's predictions tend to generate pseudo labels more confidently than those of CNN during the training phase. Thus, a lower ViT threshold ($\tau_{vit} = 0.6$) not only boosts the number of pseudo labels but also produces reliable ones to improve the guidance of the CNN branch more effectively. Conversely, the CNN branch shows less certainty with unlabeled target samples, requiring a higher threshold ($\tau_{cnn} = 0.9$). Furthermore, in the case $\tau_{vit} = 0.85$ and $\tau_{cnn} = 0.95$, we observe considerable performance effectiveness. A stricter threshold for the CNN branch is essential to avoid introducing noise pseudo labels into the ViT branch, which in turn significantly improves the ViT branch's performance. Nevertheless, opting for a high threshold for both two branches can lead to overlooking significant information in the unlabeled target domain, rendering it less effective in addressing data bias. As a result, we have chosen the thresholds of $\tau_{vit} = 0.6$ and $\tau_{cnn} = 0.9$ for all our experiments.

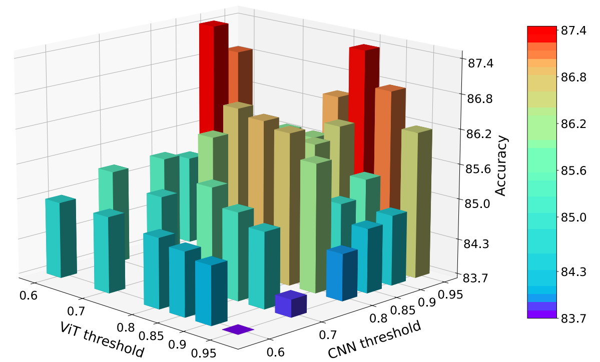

Figure 1. We evaluated our method's performance on the CNN branch by adjusting τ_{vit} and τ_{cnn} to the values $\{0.6, 0.7, 0.8, 0.85, 0.9 \text{ and } 0.95\}$. All experiments were conducted in the SSDA scenario under 3-shot setting of $rel \rightarrow clp$ task.

4. Additional Experiment Results

4.1. Experiments Setup

Dataset Details. *DomainNet* is one of the largest and most diverse datasets for domain adaptation. It contains $596,010$

*Co-first author. †Corresponding author.

Notations	Descriptions
$\mathcal{D}_{\mathcal{S}}$	The set of source samples.
$x_i^{\mathcal{S}}$	The i-th sample in the source domain.
$y_i^{\mathcal{S}}$	The label of the i-th sample in the source domain.
$\mathcal{N}_{\mathcal{S}}$	The number of source samples.
$\mathcal{D}_{\mathcal{T}_l}$	The set of labeled target samples.
$x_i^{\mathcal{T}_l}$	The i-th sample in the labeled target domain.
$y_i^{\mathcal{T}_l}$	The label of the i-th sample in the labeled target domain.
$\mathcal{N}_{\mathcal{T}_l}$	The number of labeled target samples.
$\mathcal{D}_{\mathcal{T}_u}$	The set of unlabeled target samples.
$x_i^{\mathcal{T}_u}$	The i-th unlabeled target sample in the target domain.
$y_i^{\mathcal{T}_u}$	The label of the i-th sample in the unlabeled target domain.
$\mathcal{N}_{\mathcal{T}_u}$	The number of unlabeled target samples.
$\mathcal{D}_l$	The set of labeled samples.
x_i^l	The i-th labeled sample.
y_i^l	The label of the i-th labeled sample.
$\mathcal{N}_l$	The number of labeled samples.
$Aug_w(\cdot)$	The weak augmentation.
$Aug_{str}(\cdot)$	The strong augmentation.
$x_{i,w}^{\mathcal{T}_u}$	The weakly augmented i-th unlabeled target sample.
$x_{i,str}^{\mathcal{T}_u}$	The strongly augmented i-th unlabeled target sample.
$E_1(\cdot;\boldsymbol{\theta}_{E_1})$	The ViT encoder.
$E_2(\cdot;\boldsymbol{\theta}_{E_2})$	The CNN encoder.
$F_1(\cdot;\boldsymbol{\theta}_{F_1})$	The classifier of ViT branch.
$F_2(\cdot;\boldsymbol{\theta}_{F_2})$	The classifier of CNN branch.
$p_1^l(x_i^l)$	The ViT branch's probability on the labeled sample x_i^l.
$p_2^l(x_i^l)$	The CNN branch's probability on the labeled sample x_i^l.
$p_1^{find}(x_i^{\mathcal{T}_u})$	The probability output of F_1 with ViT encoder E_1 on the unlabeled target sample, $x_i^{\mathcal{T}_u}$.
$p_2^{find}(x_i^{\mathcal{T}_u})$	The probability output of F_2 with ViT encoder E_1 on the unlabeled target sample, $x_i^{\mathcal{T}_u}$.
$p_1^{conq}(x_i^{\mathcal{T}_u})$	The probability output of F_1 with CNN encoder E_2 on the unlabeled target sample, $x_i^{\mathcal{T}_u}$.
$p_2^{conq}(x_i^{\mathcal{T}_u})$	The probability output of F_2 with CNN encoder E_2 on the unlabeled target sample, $x_i^{\mathcal{T}_u}$.
τ_{vit}	The fixed threshold of $vit \rightarrow cnn$.
τ_{cnn}	The fixed threshold of $cnn \rightarrow vit$.
$\hat{q}_i^v$	The pseudo label generated by the ViT branch on the weakly unlabeled target sample $x_{i,w}^{\mathcal{T}_u}$.
$p^c(x_{i,str}^{\mathcal{T}_u})$	The CNN branch's probability on the strongly unlabeled target sample $x_{i,str}^{\mathcal{T}_u}$.
$\hat{q}_i^c$	The pseudo label generated by the CNN branch on the weakly unlabeled target sample $x_{i,str}^{\mathcal{T}_u}$.
$p^v(x_{i,str}^{\mathcal{T}_u})$	The ViT branch's probability on the strongly unlabeled target sample $x_{i,str}^{\mathcal{T}_u}$.

Table 1. The notations commonly employed in our method.

images distributed across 345 categories and 6 domains: Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real (rel), and Sketch (skt). In the context of unsupervised domain adaptation (UDA), we encounter significant labeling noise within its full version. This is particularly evident in some classes on certain domains with many mislabeled outliers, as demonstrated in COAL [20] and SENTRY [14]. Rather than using the full set, we opt for a subset from *DomainNet* featuring 40 frequently observed classes across 4 domains: rel, clp, pnt, and skt, encompassing all 12 possible domain shifts. In the context of SSDA, a subset of the *DomainNet* dataset has been selected, focusing specifically on 126 categories out of the original 345. The reduced number of categories in this subset still encompasses a wide range of objects and themes, ensuring that the dataset remains complex and challenging for SSDA research. *Office-Home* [15] is a diverse dataset designed for domain adaptation and transfer learning research, contain-

Algorithm 1 The ECB algorithm

Data setting:
- The labeled source data $\mathcal{D}_{\mathcal{S}} = \{(x_i^{\mathcal{S}}, y_i^{\mathcal{S}})\}_{i=1}^{\mathcal{N}_{\mathcal{S}}}$.
- The labeled target data $\mathcal{D}_{\mathcal{T}_l} = \{(x_i^{\mathcal{T}_l}, y_i^{\mathcal{T}_l})\}_{i=1}^{\mathcal{N}_{\mathcal{T}_l}}$. *Notably, $\mathcal{D}_{\mathcal{T}_l}$ is empty in UDA scenario.*
- The unlabeled target data $\mathcal{D}_{\mathcal{T}_u} = \{(x_i^{\mathcal{T}_u}, y_i^{\mathcal{T}_u})\}_{i=1}^{\mathcal{N}_{\mathcal{T}_u}}$.

Note: The labeled data $\mathcal{D}_l = \mathcal{D}_{\mathcal{S}} \cup \mathcal{D}_{\mathcal{T}_l}$.

Architectures:
The **ViT branch**: a ViT encoder $E_1(\cdot;\boldsymbol{\theta}_{E_1})$ and a classifier $F_1(\cdot;\boldsymbol{\theta}_{F_1})$.
The **CNN branch**: a CNN encoder $E_2(\cdot;\boldsymbol{\theta}_{E_2})$ and a classifier $F_2(\cdot;\boldsymbol{\theta}_{F_2})$.

Hyperparameters: Fixed thresholds τ_{vit} and τ_{cnn}, the number of training interations T, learning rates for ViT and CNN, η_{vit} and η_{cnn}.

Traning strategy:
for $t \leftarrow 1$ to T **do**
  \# Supervised Training
    $\boldsymbol{\theta}_{E_1}, \boldsymbol{\theta}_{F_1} \leftarrow \eta_{vit} \nabla \mathcal{L}_{vit}^{sup}$;
    $\boldsymbol{\theta}_{E_2}, \boldsymbol{\theta}_{F_2} \leftarrow \eta_{cnn} \nabla \mathcal{L}_{cnn}^{sup}$;
  \# Finding to Conquering
  ▷ *Finding Stage*
    $\boldsymbol{\theta}_{F_1}, \boldsymbol{\theta}_{F_2} \leftarrow \eta_{vit} \nabla \mathcal{L}_{find}$;
  ▷ *Conquering Stage*
    $\boldsymbol{\theta}_{E_2} \leftarrow \eta_{cnn} \nabla \mathcal{L}_{conq}$;
  \# Co-training
  ▷ *The ViT branch teaches the CNN branch*
    $\boldsymbol{\theta}_{E_2}, \boldsymbol{\theta}_{F_2} \leftarrow \eta_{cnn} \nabla \mathcal{L}_{vit \rightarrow cnn}^{unl}$;
  ▷ *The CNN branch teaches the ViT branch*
    $\boldsymbol{\theta}_{E_1}, \boldsymbol{\theta}_{F_1} \leftarrow \eta_{vit} \nabla \mathcal{L}_{cnn \rightarrow vit}^{unl}$;
end for

Inference: $\hat{y}_i^{\mathcal{T}_u} = \texttt{argmax}\left(F_2(E_2(x_i^{\mathcal{T}_u}))\right)$.

Method	$rel{\to}clp$	$rel{\to}pnt$	$rel{\to}skt$	$clp{\to}rel$	$clp{\to}pnt$	$clp{\to}skt$	$pnt{\to}rel$	$pnt{\to}clp$	$pnt{\to}skt$	$skt{\to}rel$	$skt{\to}clp$	$skt{\to}pnt$	Mean
MCD [16]	62.0	69.3	56.3	79.8	56.6	53.7	83.4	58.3	61.0	81.7	56.3	66.8	65.4
JAN [11]	65.6	73.6	67.6	85.0	65.0	67.2	87.1	67.9	66.1	84.5	72.8	67.5	72.5
DANN [3]	63.4	73.6	72.6	86.5	65.7	70.6	86.9	73.2	70.2	85.7	75.2	70.0	74.5
COAL [20]	73.9	75.4	70.5	89.6	70.0	71.3	89.8	68.0	70.5	88.0	73.2	70.5	75.9
InstaPBM [8]	80.1	75.9	70.8	89.7	70.2	72.8	89.6	74.4	72.2	87.0	79.7	71.8	77.8
SENTRY [14]	83.9	76.7	74.4	90.6	76.0	79.5	90.3	82.9	75.6	90.4	82.4	74.0	81.4
RHWD [18]	**84.8**	76.9	75.2	91.8	75.6	81.2	**91.9**	**84.6**	76.1	**91.3**	83.2	74.6	82.0
GSDE [22]	82.9	79.2	**80.8**	**91.9**	78.2	80.0	90.9	84.1	79.2	90.3	**83.4**	76.1	83.1
ECB(CNN)	84.7	**83.8**	79.7	91.6	**84.0**	**82.5**	91.0	83.2	**79.2**	86.1	82.9	**81.6**	**84.2**

Table 2. **Accuracy (%) on DomainNet** of UDA methods. **ECB (CNN)** represents the performance of our CNN branch when applied to ResNet-50. To facilitate easy identification, the best and second-best accuracy results are highlighted in **bold** and underline, respectively.

Method	$A{\to}C$	$A{\to}P$	$A{\to}R$	$C{\to}A$	$C{\to}P$	$C{\to}R$	$P{\to}A$	$P{\to}C$	$P{\to}$R	$R{\to}A$	$R{\to}C$	$R{\to}P$	Mean
							1-shot						
ENT [4]	52.9	75.0	76.7	63.2	73.6	73.2	63.0	51.9	79.9	70.4	53.6	81.9	67.9
MME [17]	59.6	75.5	77.8	65.7	74.5	74.8	64.7	57.4	79.2	71.2	61.9	82.8	70.4
DECOTA [23]	42.1	68.5	72.6	60.3	70.4	70.7	60.0	48.8	76.9	71.3	56.0	79.4	64.8
CDAC [9]	61.2	75.9	78.5	64.5	75.1	75.3	64.6	59.3	80.0	72.7	61.9	83.1	71.0
CDAC+SLA [24]	63.0	78.0	79.2	66.9	77.6	77.0	67.3	61.8	80.5	72.7	66.1	84.6	72.9
ProML [6]	64.5	79.7	81.7	69.1	80.5	79.0	69.3	61.4	81.9	73.7	67.5	86.1	74.6
ECB (CNN)	**72.9**	**88.3**	**89.6**	**84.8**	**91.3**	**89.5**	**82.9**	**71.2**	**89.9**	**85.5**	**75.4**	**92.0**	**84.4**
							3-shot						
ENT [4]	61.3	79.5	79.1	64.7	79.1	76.4	63.9	60.5	79.9	70.2	62.6	85.7	71.9
MME [17]	63.6	79.0	79.7	67.2	79.3	76.6	65.5	64.6	80.1	71.3	64.6	85.5	73.1
DECOTA [23]	64.0	81.8	80.5	68.0	83.2	79.0	69.9	68.0	82.1	74.0	70.4	87.7	75.7
CDAC [9]	65.9	80.3	80.6	67.4	81.4	80.2	67.5	67.0	81.9	72.2	67.8	85.6	74.8
CDAC+SLA [24]	67.3	82.6	81.4	69.2	82.1	80.1	70.1	69.3	82.5	73.9	70.1	87.1	76.3
ProML [6]	67.8	83.9	82.2	72.1	84.1	82.3	72.5	68.9	83.8	75.8	71.0	88.6	77.8
ECB (CNN)	**78.7**	**90.2**	**91.3**	**85.2**	**90.4**	**91.0**	**83.9**	**76.8**	**91.2**	**85.6**	**77.6**	**92.8**	**86.2**

Table 3. **Accuracy (%) on Office-Home** of SSDA methods using a ResNet-34 serving as a backbone across different domain shifts.

ing around $15,500$ images from 65 categories of everyday objects. It includes 4 significantly different domains: Art (A), Clipart (C), Product (P), and Real World (R). This variety in domains provides a challenging testbed for algorithms aiming to generalize across different visual domains. *Office-31* [21] is an earlier standard dataset for domain adaptation, which includes $4,110$ images across 31 categories collected from an office environment. It consists of three distinct domains: Amazon (A), with $2,817$ images from amazon.com product listings; Webcam (W), consisting of 795 images taken with a webcam; and DSLR (D), which includes 498 images captured with a digital SLR camera. Each domain presents unique challenges regarding image quality, lighting, and backgrounds.

Implementation Details. In this section, we delve deeper into the specifics of our implementation. Our hybrid model utilizes the ViT/B-16 [2] for the ViT encoder E_1. The ResNet [5] and AlexNet [7] for the CNN encoder E_2. These all are initialed pre-training on the ImageNet-1K [1]. Specifically, in the context of unsupervised domain adaptation (UDA), we have chosen ResNet-50 as our primary network for E_2, aligning with methodologies in prior studies [3, 14, 16, 18, 20]. Following the evaluation protocol of established SSDA methods [4, 9, 10, 17], we employ ResNet-34 to evaluate on both the *DomainNet* and *Office-Home* dataset, while AlexNet is chosen for *Office-31* evaluations. We are following ViT encoder E_1 and CNN encoder E_2 by two different classifiers, F_1 and F_2, each consisting of two fully-connected layers followed by the softmax function. Our ECB approach uses stochastic gradient descent as the optimizer for two branches, maintaining a momentum of 0.9 and a weight decay of 0.0005. Acknowledging the distinct architectures of the ViT and CNN branches, we initially set their learning rates at $1e-4$ and $1e-3$, respectively. Following [17], we employ the learning rate scheduler with the gamma and power parameters set to $1e-4$ and 0.75, respectively. We set the same mini-batch to 32 for all labeled and unlabeled samples. Due to ViT's outstanding properties, $vit \to cnn$ needs to be provided with more information for the CNN branch, leading to the confidence threshold for pseudo-label selection at $\tau_{vit} = 0.6$. To prevent the CNN branch from introducing noise for ViT, we set a higher threshold $\tau_{cnn} = 0.9$ to get reliable pseudo labels. The warmup phase for both branches on $\mathcal{D}_l$ undergoes a fine-tuning process across $100,000$ iterations. Subsequently, we train $50,000$ iterations for our approach.

Method	$W \to A$		$D \to A$		Mean	
	1_{shot}	3_{shot}	1_{shot}	3_{shot}	1_{shot}	3_{shot}
ENT [4]	50.7	64.0	50.0	66.2	50.4	65.1
MME [17]	57.2	67.3	55.8	67.8	56.5	67.6
STar [19]	59.8	69.1	56.8	69.0	58.3	69.1
MVCL [12]	56.7	69.0	59.3	69.1	58.0	69.1
CDAC [9]	63.4	70.1	62.8	70.0	63.1	70.0
G-ABC [10]	<u>67.9</u>	<u>71.0</u>	<u>65.7</u>	<u>73.1</u>	<u>66.8</u>	<u>72.0</u>
ECB (CNN)	**77.9**	**85.2**	**76.3**	**84.0**	**77.1**	**84.6**

Table 4. **Accuracy (%) on Office-31** of SSDA methods in both 1-shot and 3-shot settings. AlexNet is used as the feature extractor for the CNN branch.

4.2. Comparison Results

Results on *DomainNet* under UDA setting. We conduct a series of 12 experiments on the subset of *DomainNet*. As detailed in Tab. 2, the ECB (CNN) outperforms the SOTA method GSDE [22] by an increased margin of +1.1% in average accuracy, with an overall accuracy of 84.2%. Additionally, our method significantly surpasses others in several specific domain transitions. Notably $rel \to pnt$, $clp \to pnt$ and $skt \to pnt$ improve accuracy by +4.6%, +5.8%, and +5.5%, compared to the second-best. However, we face obstacles in the $skt \to rel$ transition, where our method's accuracy is -5.2% lower than the RHWD [18] method.

Results on *Office-Home* under SSDA setting. The performance of our method on the *Office-Home* dataset, under both 1-shot and 3-shot settings, is showcased in Tab. 3. The results clearly demonstrate that our classification outcomes exceed prior methods in all domain adaptation scenarios presented. Notably, the ECB method improves an average classification accuracy that surpasses the nearest-competitor ProML [6] by a notable +9.8% in the 1-shot setting. Furthermore, we continue to impress with an average accuracy increase of +8.4% under the 3-shot setting.

Results on *Office-31* under SSDA setting. We use AlexNet as a backbone followed previous SSDA methods [9, 10, 12, 17]. As demonstrated in Tab. 4, our method consistently outperforms all other domain adaptation scenarios regarding classification results on the target set. Remarkably, our proposed method achieves an average classification accuracy of 84.6% under the 3-shot setting. This result surpasses the nearest method G-ABC [10] by +12.6%. Furthermore, our method maintains a competitive edge with a +10.3% higher performance even in the 1-shot setting. Reveals that our approach is not significantly affected by the CNN encoder architecture.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 248–255, 2009. 3

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Thomas Unterthiner Xiaohua Zhai, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 3

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 3

[4] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Adv. Neural Inform. Process. Syst.*, 17, 2004. 3, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 770–778, 2016. 3

[6] Xinyang Huang, Chuang Zhu, and Wenkai Chen. Semi-supervised domain adaptation via prototype-based multi-level learning. *Proc. IJCAI*, 2023. 3, 4

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, page 1097–1105, 2012. 3

[8] Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*, 2020. 3

[9] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2505–2514, 2021. 3, 4

[10] Jichang Li, Guanbin Li, and Yizhou Yu. Adaptive betweenness clustering for semi-supervised domain adaptation. *IEEE Trans. Image Process.*, 2023. 3, 4

[11] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217. PMLR, 2017. 3

[12] Ba Hung Ngo, Ju Hyun Kim, Yeon Jeong Chae, and Sung In Cho. Multi-view collaborative learning for semi-supervised domain adaptation. *IEEE Access*, volume 9:166488–166501, 2021. 4

[13] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Int. Conf. Comput. Vis.*, pages 1406–1415, 2019. 1

[14] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 8558–8567, 2021. 2, 3

[15] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Deep hashing network for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[16] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[17] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Int. Conf. Comput. Vis.*, pages 8050–8058, 2019. 3, 4

[18] Lingyu Si, Hongwei Dong, Wenwen Qiang, Changwen Zheng, Junzhi Yu, and Fuchun Sun. Regularized hypothesis-induced wasserstein divergence for unsupervised domain adaptation. *Knowledge-Based Systems*, page 111162, 2023. 3, 4

[19] Anurag Singh, Naren Doraiswamy, Sawa Takamuku, Megh Bhalerao, Titir Dutta, Soma Biswas, Aditya Chepuri, Balasubramanian Vengatesan, and Naotake Natori. Improving semi-supervised domain adaptation using effective target selection and semantics. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2709–2718, 2021. 4

[20] Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: an empirical odyssey. In *Eur. Conf. Comput. Vis. Works.*, pages 585–602. Springer, 2020. 2, 3

[21] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Adapting visual category models to new domains. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 3

[22] Thomas Westfechtel, Hao-Wei Yeh, Dexuan Zhang, and Tatsuya Harada. Gradual source domain expansion for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1946–1955, 2024. 3, 4

[23] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q. Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 8906–8916, 2021. 3

[24] Yu-Chu Yu and Hsuan-Tien Lin. Semi-supervised domain adaptation with source label adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3