Simplifying and Stabilizing Model Selection in Unsupervised Domain Adaptation

Dapeng Hu[1] **Mi Luo**[3] **Jian Liang**[4 *] **Chuan-Sheng Foo**[2,1 *]
[1]Centre for Frontier AI Research, A*STAR, Singapore
[2]Institute for Infocomm Research, A*STAR, Singapore
[3]National University of Singapore
[4]CRIPAC & MAIS, Institute of Automation, Chinese Academy of Sciences
lhxxhb15@gmail.com, romyluo7@gmail.com,
liangjian92@gmail.com, foo_chuan_sheng@i2r.a-star.edu.sg

Abstract

Ensuring reliable model selection is crucial for unleashing the full potential of advanced unsupervised domain adaptation (UDA) methods to improve model performance in unlabeled target domains. However, existing model selection methods in UDA often struggle to maintain reliable selections across diverse UDA methods and scenarios, suffering from highly risky worst-case selections. This limitation significantly hinders their practicality and reliability for researchers and practitioners in the community. In this paper, we introduce EnsV, a novel ensemble-based approach that makes pivotal strides in reliable model selection by avoiding the selection of the worst model. EnsV is built on an off-the-shelf ensemble that is theoretically guaranteed to outperform the worst candidate model, ensuring high reliability. Notably, EnsV relies solely on predictions of unlabeled target data without making any assumptions about domain distribution shifts, offering high simplicity and versatility for various practical UDA problems. In our experiments, we compare EnsV to 8 competitive model selection approaches. Our evaluation involves 12 UDA methods across 5 diverse UDA benchmarks and 5 popular UDA scenarios. The results consistently demonstrate that EnsV stands out as a highly simple, versatile, and reliable approach for practical model selection in UDA scenarios. Code is available at https://github.com/LHXXHB/EnsV.

1 Introduction

Deep learning has achieved incredible advancements in various tasks through supervised learning with large labeled datasets [1]. However, obtaining labels can be expensive, and deep models often struggle to generalize to unlabeled data sampled from unseen distributions [2]. Domain adaptation [3] tackles this challenge by transferring knowledge from a labeled source domain to a target domain with limited labels but a similar task. Unsupervised domain adaptation [4] (UDA), particularly, has garnered significant attention due to its practical assumption that the target domain is entirely unlabeled, witnessing the development of many effective methods [5–8] and practical settings [9–12].

However, the successful application of UDA methods across diverse tasks relies heavily on selecting appropriate hyperparameters. Sub-optimal hyperparameters can cause state-of-the-art UDA methods to underperform compared to the source model without adaptation [13, 14], emphasizing the significance of model selection, also called hyperparameter selection or validation, in UDA. In a typical model selection scenario, we are presented with a set of m candidate models with

*Corresponding author.

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

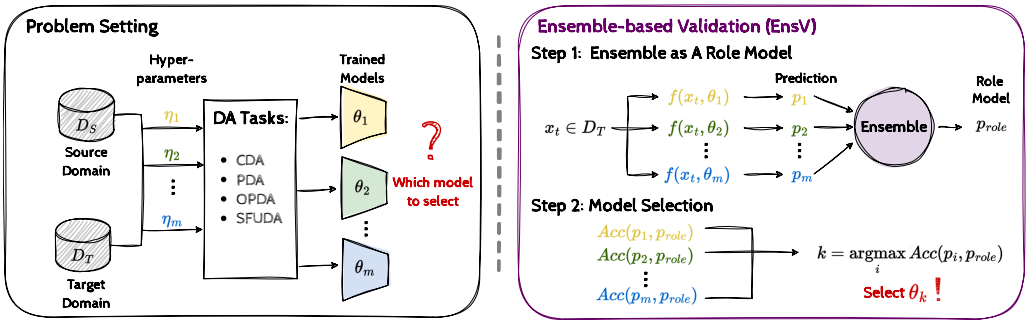

Figure 1: Overview of our model selection approach EnsV for unsupervised domain adaptation.

the weights $\{\theta_i\}_{i=1}^m$. These models are trained using a given UDA method with a corresponding set of hyperparameters $\{\eta_i\}_{i=1}^m$. The goal is to identify the candidate model that exhibits the best performance on the unlabeled target domain and subsequently adopt the associated hyperparameter value for η. This model selection problem remains challenging and under-explored in UDA due to cross-domain distribution shifts and the absence of labeled target data. Existing approaches can be categorized into two types. The first type involves leveraging labeled source data for target-domain model selection [9, 15–17]. The second type designs unsupervised metrics based on priors of the learned target-domain structure and utilizes them for model selection [18, 13, 14, 19]. Despite their specific designs, all these methods encounter challenges in avoiding the selection of poor or even the worst models across various UDA methods and settings. This renders the adaptation ineffective or even harmful, thereby constraining their adoption by researchers and practitioners in the community [14]. For instance, in Table 1, we compare the worst-case selection statistics for all these model selection methods in standard closed-set UDA and partial-set UDA settings, two settings extensively studied in prior works [16, 13]. The comparison reveals that all the methods exhibit occasional or even frequent worst-case model selection situations.

In this paper, we resolve this predicament by introducing EnsV, a novel ensemble-based validation approach. Our method emerges from a meticulous examination of the model selection problem, revealing that the problem setting inherently provides an off-the-shelf ensemble of candidate models. Surprisingly, many existing model selection studies overlook this "free lunch", treating each candidate model independently. Through a straightforward theoretical analysis of the ensemble, we observe that it strictly surpasses the worst candidate model, grounded in a very weak and reasonable assumption. EnsV takes an additional step, utilizing the ensemble as a role model for directly assessing candidate models during the model selection process. This strategy ensures the secure avoidance of selecting the worst candidate model, thereby enhancing the reliability of model selection.

Table 1: Statistics for worst-case selections are provided across 110 closed-set UDA tasks (potentially an additional 21 tasks on DomainNet [20]) and 24 partial-set UDA tasks for all the considered model selection methods. The statistics are presented as the count of worst-case selections divided by the total count of tasks. **Bold** font indicates the best worst-case avoidance.

Method	closed-set UDA	partial-set UDA
SourceRisk [9]	16 / 110	2 / 24
IWCV [15]	15 / 110	3 / 24
DEV [16]	9 / 110	1 / 24
RV [17]	2 / 110	1 / 24
Entropy [18]	15 / 131	7 / 24
InfoMax [14]	9 / 131	12 / 24
SND [13]	33 / 131	3 / 24
Corr-C [19]	80 / 131	4 / 24
EnsV (Ours)	**0 / 131**	**0 / 24**

2 Methodology

We consider a C-way image classification task to introduce the concept of unsupervised domain adaptation (UDA). In UDA, we typically have a labeled source domain $\mathcal{D}_\mathrm{s} = \{(x_\mathrm{s}^i, y_\mathrm{s}^i)\}_{i=1}^{n_\mathrm{s}}$ comprising n_s annotated source images x_s and their corresponding labels y_s. Additionally, there is an unlabeled target domain, $\mathcal{D}_\mathrm{t} = \{x_\mathrm{t}^i\}_{i=1}^{n_\mathrm{t}}$, containing only n_t unlabeled target images x_t. Despite the tasks being similar, there exist data distribution shifts between the two domains. The primary objective of UDA is to accurately predict the unavailable target labels, $\{y_\mathrm{t}^i\}_{i=1}^{n_\mathrm{t}}$, by leveraging a discriminative mapping $f(x, \theta)$, which is learned using data from two domains. Here, $\theta \in \mathbb{R}^d$ represents the weights of the trained UDA model. When presented with an input image x, the model generates a probability prediction vector, $p = f(x, \theta)$, where $p \in \mathbb{R}^C$ and $\sum_{i=1}^C p^i = 1$.

For model selection in UDA, we aim to determine the optimal hyperparameter η from a set of m candidate values $\{\eta_i\}_{i=1}^m$. The hyperparameter η can represent the learning rate, loss coefficients, architectural settings, training iterations, and more. By training UDA models using the m different values of η, we obtain corresponding models with weights denoted as $\{\theta_i\}_{i=1}^m$. In UDA, the objective of model selection is to pinpoint the model θ_k that demonstrates the best performance on the unlabeled target domain. Subsequently, we select the corresponding hyperparameter η_k as the optimal choice for potential adaptation with unlabeled target samples from the exact target domain. We illustrate the problem setting in Figure 1. Without loss of generality, in this paper, we assume m is greater than 1, and candidate models have different weights θ, resulting in different discriminative mappings of $f(x, \theta)$. For clarity, we treat both θ and the model interchangeably in our presentation. This also applies to model selection, hyperparameter selection, and validation.

2.1 Ensemble: The Overlooked "Free Lunch" in Model Selection

We first adopt a novel perspective in analyzing the challenge of model selection in UDA via the lens of the ensemble. In this paper, unless otherwise specified, the ensemble refers to prediction-based ensembling, i.e., $\frac{1}{m}\sum_{i=1}^m f(x, \theta_i)$ for a sample x. Typically, two concerns arise with the ensemble: one pertains to the efficiency issue caused by training multiple models, and the other relates to the lack of diversity among candidate models. In model selection, we observe that the problem setting itself inherently offers a range of off-the-shelf candidate models, naturally addressing the efficiency issue. Furthermore, all candidate models are trained using a UDA method with varying hyperparameters, yielding diverse yet effective discriminative abilities. This naturally eases the diversity concern. As a surprising consequence, the ensemble appears to be a "free lunch" in the context of model selection in UDA, a point that has been previously overlooked by researchers. To gain a deeper insight into the effectiveness of the ensemble, we present a theoretical analysis grounded in the proposition below.

Proposition 1 *Given negative log-likelihood (NLL) as the loss function, defined as $l(p, y) = -\log p^y$, and considering a random target sample x with label y, the following inequality can be established between the loss of the ensemble $\frac{1}{m}\sum_{i=1}^m f(x, \theta_i)$, the averaged loss of all candidate models $\{\theta_i\}_{i=1}^m$, and the loss of the worst model θ_{worst}:*

$$l(\frac{1}{m}\sum_{i=1}^m f(x, \theta_i), y) < \frac{1}{m}\sum_{i=1}^m l(f(x, \theta_i), y) < l(f(x, \theta_{\text{worst}}), y).$$

Kindly refer to Appendix A for the proof. This proposition theoretically guarantees that the ensemble always outperforms the worst candidate model. In contrast, as demonstrated in Table 1, existing model selection methods cannot guarantee to avoid selecting the worst candidate model.

2.2 Ensemble as a Role Model for Model Selection

When tackling model selection in UDA, recent trends have favored target-domain specific methods [13, 14, 18, 19]. These methods typically utilize unlabeled data to indirectly gauge specific properties of target predictions output by each candidate model, often enjoying high simplicity and effectiveness. In contrast, we initially consider a straightforward upper-bound model selection solution. This involves selecting models based on their accuracy, measured against the unattainable target ground truth $\{y_{\text{t}}^i\}_{i=1}^{n_{\text{t}}}$. The ideal solution implies that if we can obtain a reliable approximation of the true target labels, we can directly use it for accurate model selection. To achieve this, we employ the previously mentioned off-the-shelf ensemble as a reliable role model and select the model that generates predictions closest to this role model among all candidates. These two direct steps constitute an elegantly simple model selection approach known as ensemble-based validation (EnsV). We present a comprehensive illustration of EnsV in Figure 1.

Step 1: Ensemble as a role model. To begin with, for each unlabeled target sample x, we consider the ensemble $\frac{1}{m}\sum_{i=1}^m f(x, \theta_i)$ as a reliable estimation of its ground truth. This enables us to obtain reliable predictions for all target data, denoted as $\{\frac{1}{m}\sum_{i=1}^m f(x_j, \theta_i)\}_{j=1}^{n_{\text{t}}}$. These ensembles serve as our role model, providing guidance for accurate model selection in the subsequent step.

Step 2: Model selection. In this step, we utilize the role model to assess all candidate models and select the one with the highest similarity. For simplicity, EnsV involves a direct measurement of accuracy between the role model $\{\frac{1}{m}\sum_{i=1}^m f(x_j, \theta_i)\}_{j=1}^{n_{\text{t}}}$ and the predictions made by each candidate model, such as $\{f(x_j, \theta_i)\}_{j=1}^{n_{\text{t}}}$ for the model with weights θ_i. We then select the model θ_k with the highest accuracy and determine the optimal value η_k for the hyperparameter η.

Table 2: Closed-set UDA (CDA) accuracy (%) on *DomainNet-126*. **bold**: Best value.

Method	CDAN [6]					BNM [8]					ATDOC [26]				
	→ C	→ P	→ R	→ S	avg	→ C	→ P	→ R	→ S	avg	→ C	→ P	→ R	→ S	avg
Entropy [18]	**67.09**	**65.80**	74.42	**59.34**	**66.66**	63.36	64.28	74.31	48.69	62.66	63.75	61.85	79.60	52.17	64.34
InfoMax [14]	**67.09**	**65.80**	74.42	**59.34**	**66.66**	67.05	64.28	74.31	55.67	65.33	63.75	61.85	79.60	52.17	64.34
SND [13]	**67.09**	64.68	74.42	**59.34**	66.38	56.56	54.50	74.31	42.37	56.93	63.75	61.85	79.60	47.00	63.05
Corr-C [19]	57.35	62.88	74.42	54.63	62.32	59.75	63.41	**77.62**	42.37	60.79	59.98	62.27	74.42	53.69	62.59
EnsV	65.88	65.27	**74.44**	57.42	65.75	**67.86**	**66.06**	**77.62**	**57.69**	**67.31**	**70.30**	**68.44**	**80.01**	**61.73**	**70.12**
Worst	57.35	60.76	73.44	51.41	60.74	55.79	54.50	74.31	42.37	56.74	59.98	61.85	74.42	47.00	60.81
Best	67.09	65.80	74.44	59.34	66.66	67.86	66.50	78.68	58.49	67.88	70.30	68.44	80.38	62.23	70.34

3 Experiments

Setup. We use diverse and widely-used UDA benchmarks: *Office-31* [21], *Office-Home* [22], *VisDA* [23], *DomainNet-126* [20], and GTAV [24]-to-Cityscapes [25]. As for baselines, we assess all the model selection methods listed in Table 4 (Appendix). Kindly refer to Appendix B for the introduction of these model selection methods and Appendix C for detailed computations. With these validation methods, we perform model selection for various UDA methods across different UDA settings. For CDA, we consider ATDOC [26], BNM [8], CDAN [6], MCC [27], MDD [28], and SAFN [7]. For partial-set UDA, we consider PADA [10] and SAFN [7]. For OPDA, we consider DANCE [11]. For SFUDA, we consider SHOT [12] and DINE [26]. For segmentation, we consider AdaptSeg [29] and AdvEnt [30]. Detailed hyperparameter settings are provided in Appendix D.

Table 3: OPDA H-score [31] (%) on *Office-Home*. SFUDA accuracy (%) on *Office-31* and *VisDA*.

Method	DANCE [11]					SHOT [12]				*DINE [26]*
	→Ar	→Cl	→Pr	→Re	avg	→A	→D	→W	avg	T→V
Entropy [18]	32.00	39.48	27.52	38.08	34.27	71.67	90.76	88.68	83.70	71.99
InfoMax [14]	32.00	39.48	27.52	38.01	34.25	71.67	90.76	88.68	83.70	71.99
SND [13]	15.05	4.33	23.75	16.79	14.98	71.67	90.76	88.68	83.70	**74.43**
Corr-C [19]	29.60	4.33	23.75	16.79	18.62	71.58	90.76	90.19	84.18	71.99
EnsV	**77.01**	**51.36**	**78.81**	**68.65**	**68.96**	**74.85**	**94.78**	**91.82**	**87.15**	**74.43**
Worst	15.05	4.33	15.17	16.79	12.84	71.56	90.76	88.68	83.67	71.99
Best	77.01	66.29	78.81	69.81	72.98	75.06	94.78	93.33	87.72	76.17

Results. For closed-set UDA (CDA), we compare all target-specific validation methods on the large-scale benchmark *DomainNet-126* (Table 2). EnsV consistently keeps the leading performance, while other approaches exhibit high instability. In OPDA with label shift, we choose a typical method DANCE for validation on *Office-Home* (Table 3). Prior model selection works have not explored this challenging setting, resulting in poor selections. In contrast, our EnsV achieves selections close to the best. For source-free UDA (SFUDA), we choose SHOT and DINE (Table 3). EnsV consistently maintains near-best selections, while other target-specific approaches occasionally make near-worst selections. Kindly refer to Appendix E for the full results and analysis of our EnsV method.

4 Discussions

Limitations. While EnsV consistently avoids worst-case selections, it encounters challenges related to poor model selection performance in two specific scenarios: (*i*) The task of selecting the sole optimal candidate from a pool where the majority are extremely poor, and (*ii*) Deciding between a single poor model and a high-performing model.

Conclusions. Following a thorough empirical comparison of existing UDA model selection methods, several key conclusions emerge: *(i)* The significance of model selection in impacting UDA methods' deployment performance becomes evident. Relying on fixed hyperparameters or limited analyses falls short. We stress the importance of increased attention and transparent reporting of validation methods, aligning with recommendations in [16, 13, 14]. *(ii)* Existing model selection methods prove unreliable in diverse UDA methodologies and real-world settings like open-set and source-free UDA. These methods struggle to maintain effectiveness, presenting a substantial risk to the successful application of UDA in various scenarios. *(iii)* Our EnsV distinguishes itself with exceptional performance across various UDA scenarios, including open-partial-set UDA and source-free UDA, consistently avoiding worst-case selections. As a post-hoc method, EnsV leverages an off-the-shelf ensemble of pre-existing candidate models, eliminating the need for extra memory and time. We believe EnsV can serve as a simple, versatile, and highly reliable model selection approach in UDA studies.

References

[1] Russakovsky, O., J. Deng, H. Su, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[2] Hendrycks, D., K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[3] Pan, S. J., Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[4] Pan, S. J., I. W. Tsang, J. T. Kwok, et al. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010.

[5] Saito, K., K. Watanabe, Y. Ushiku, et al. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[6] Long, M., Z. Cao, J. Wang, et al. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*. 2018.

[7] Xu, R., G. Li, J. Yang, et al. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *IEEE International Conference on Computer Vision*. 2019.

[8] Cui, S., S. Wang, J. Zhuo, et al. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[9] Ganin, Y., V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*. 2015.

[10] Cao, Z., L. Ma, M. Long, et al. Partial adversarial domain adaptation. In *European Conference on Computer Vision*. 2018.

[11] Saito, K., D. Kim, S. Sclaroff, et al. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing Systems*. 2020.

[12] Liang, J., D. Hu, J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*. 2020.

[13] Saito, K., D. Kim, P. Teterwak, et al. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *IEEE International Conference on Computer Vision*. 2021.

[14] Musgrave, K., S. Belongie, S.-N. Lim. Benchmarking validation methods for unsupervised domain adaptation. *arXiv preprint arXiv:2208.07360*, 2022.

[15] Sugiyama, M., M. Krauledat, K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 2007.

[16] You, K., X. Wang, M. Long, et al. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*. 2019.

[17] Ganin, Y., E. Ustinova, H. Ajakan, et al. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.

[18] Morerio, P., J. Cavazza, V. Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *arXiv preprint arXiv:1711.10288*, 2017.

[19] Tu, W., W. Deng, T. Gedeon, et al. Assessing model out-of-distribution generalization with softmax prediction probability baselines and a correlation method, 2023.

[20] Peng, X., Q. Bai, X. Xia, et al. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*. 2019.

[21] Saenko, K., B. Kulis, M. Fritz, et al. Adapting visual category models to new domains. In *European Conference on Computer Vision*. 2010.

[22] Venkateswara, H., J. Eusebio, S. Chakraborty, et al. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[23] Peng, X., B. Usman, N. Kaushik, et al. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[24] Richter, S. R., V. Vineet, S. Roth, et al. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*. 2016.

[25] Cordts, M., M. Omran, S. Ramos, et al. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[26] Liang, J., D. Hu, J. Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2021.

[27] Jin, Y., X. Wang, M. Long, et al. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*. 2020.

[28] Zhang, Y., T. Liu, M. Long, et al. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*. 2019.

[29] Tsai, Y.-H., W.-C. Hung, S. Schulter, et al. Learning to adapt structured output space for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[30] Vu, T.-H., H. Jain, M. Bucher, et al. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[31] Bucci, S., M. R. Loghmani, T. Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *European Conference on Computer Vision*. 2020.

[32] Panareda Busto, P., J. Gall. Open set domain adaptation. In *IEEE International Conference on Computer Vision*. 2017.

[33] Li, R., Q. Jiao, W. Cao, et al. Model adaptation: Unsupervised domain adaptation without source data. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[34] Liang, J., D. Hu, J. Feng, et al. Dine: Domain adaptation from single and multiple black-box predictors. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2022.

[35] Gong, B., Y. Shi, F. Sha, et al. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2012.

[36] Fernando, B., A. Habrard, M. Sebban, et al. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision*. 2013.

[37] Long, M., Y. Cao, J. Wang, et al. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*. 2015.

[38] Sun, B., K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision, Workshop*. 2016.

[39] Yang, Y., S. Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095. 2020.

[40] Hoffman, J., E. Tzeng, T. Park, et al. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*. 2018.

[41] Tzeng, E., J. Hoffman, K. Saenko, et al. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[42] Shu, R., H. H. Bui, H. Narui, et al. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[43] Bridle, J., A. Heading, D. MacKay. Unsupervised classifiers, mutual information and'phantom targets. In *Advances in Neural Information Processing Systems*. 1991.

[44] Perrone, M. P., L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In *How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper*. World Scientific, 1995.

[45] Opitz, D., R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 1999.

[46] Bauer, E., R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 1999.

[47] Dietterich, T. G. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop*. 2000.

[48] Lakshminarayanan, B., A. Pritzel, C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*. 2017.

[49] Ovadia, Y., E. Fertig, J. Ren, et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*. 2019.

[50] Lee, S., S. Purushwalkam, M. Cogswell, et al. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

[51] Wen, Y., D. Tran, J. Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

[52] Dusenberry, M., G. Jerfel, Y. Wen, et al. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning*. 2020.

[53] Huang, G., Y. Li, G. Pleiss, et al. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[54] Garipov, T., P. Izmailov, D. Podoprikhin, et al. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*. 2018.

[55] Benton, G., W. Maddox, S. Lotfi, et al. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*. 2021.

[56] Izmailov, P., D. Podoprikhin, T. Garipov, et al. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[57] Wortsman, M., G. Ilharco, S. Y. Gadre, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*. 2022.

[58] Matena, M. S., C. A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*. 2022.

[59] Rame, A., J. Zhang, L. Bottou, et al. Pre-train, fine-tune, interpolate: a three-stage strategy for domain generalization. In *Advances in Neural Information Processing Systems, Workshop*. 2022.

[60] Ramé, A., K. Ahuja, J. Zhang, et al. Recycling diverse models for out-of-distribution generalization. *arXiv preprint arXiv:2212.10445*, 2022.

[61] Freund, Y., R. E. Schapire, et al. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*. 1996.

[62] Fort, S., H. Hu, B. Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[63] Wenzel, F., J. Snoek, D. Tran, et al. Hyperparameter ensembles for robustness and uncertainty quantification. In *Advances in Neural Information Processing Systems*. 2020.

[64] Zaidi, S., A. Zela, T. Elsken, et al. Neural ensemble search for uncertainty estimation and dataset shift. In *Advances in Neural Information Processing Systems*. 2021.

[65] Gontijo-Lopes, R., Y. Dauphin, E. D. Cubuk. No one representation to rule them all: Overlapping features of training methods. *arXiv preprint arXiv:2110.12899*, 2021.

[66] Cortes, C., M. Mohri, M. Riley, et al. Sample selection bias correction theory. In *Algorithmic Learning Theory*. 2008.

[67] Zhong, E., W. Fan, Q. Yang, et al. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2010.

[68] Grandvalet, Y., Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*. 2004.

[69] Chapelle, O., A. Zien. Semi-supervised classification by low density separation. In *International Workshop on Artificial Intelligence and Statistics*. 2005.

[70] He, K., X. Zhang, S. Ren, et al. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[71] Fu, B., Z. Cao, M. Long, et al. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*. 2020.

[72] Dosovitskiy, A., L. Beyer, A. Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. 2021.

A Proof of Proposition 1

Given the use of negative log-likelihood (NLL) as the loss function, defined as $l(p, y) = -\log p^y$. We first prove the first inequality using Jensen's inequality, which states that for a real-valued, convex function φ with its domain as a subset of $\mathbb{R}$ and numbers $t_1, \ldots, t_n$ in its domain, the inequality $\varphi\left(\frac{1}{n}\sum_{i=1}^{n} t_i\right) \leq \frac{1}{n}\sum_{i=1}^{n}\varphi(t_i)$ holds. Given that $-\log$ is a convex function, and assuming $m > 1$ with candidate models having different weights θ, resulting in distinct discriminative mappings of $f(x, \theta)$, we can strictly obtain $l(\frac{1}{m}\sum_{i=1}^{m} f(x, \theta_i), y) < \frac{1}{m}\sum_{i=1}^{m} l(f(x, \theta_i), y)$ without the equal situation. Next, we leverage the property of inequalities to prove the second inequality. Here, θ_{worst} denotes the worst candidate model, i.e., the one with the largest loss. For any other candidate model θ_i, we have $l(f(x, \theta_i), y) < l(f(x, \theta_{\text{worst}}), y)$. This ensures that $\frac{1}{m}\sum_{i=1}^{m} l(f(x, \theta_i), y) < \frac{1}{m}\sum_{i=1}^{m} l(f(x, \theta_{\text{worst}}), y)$, or more explicitly, $\frac{1}{m}\sum_{i=1}^{m} l(f(x, \theta_i), y) < l(f(x, \theta_{\text{worst}}), y)$. Substituting the NLL loss with any strongly convex loss function would still uphold the proposition. This proposition theoretically guarantees that the ensemble strictly outperforms the worst candidate model.

B Related Work

Table 4: Comparing mainstream methods for model selection in unsupervised domain adaptation.

Method	covariate shift	label shift	w/o source data	w/o hyperparameters	w/o extra training	worst-case avoidance
SourceRisk [9]	✗	✗	✗	✗	✓	✗
IWCV [15]	✓	✗	✗	✗	✗	✗
DEV [16]	✓	✗	✗	✗	✗	✗
RV [17]	✓	✗	✗	✗	✗	✗
Entropy [18]	✓	✗	✓	✓	✓	✗
InfoMax [14]	✓	✗	✓	✓	✓	✗
SND [13]	✓	✓	✓	✗	✓	✗
Corr-C [19]	✓	✗	✓	✓	✓	✗
EnsV (Ours)	✓	✓	✓	✓	✓	✓

Unsupervised domain adaptation (UDA) is initially studied in a closed-set setting (CDA) where only covariate shift [15] is considered as the domain shift, and the two domains share the same label set. Recent research has explored many real-world UDA scenarios by incorporating label shift, where the two domains have distinct label sets. This includes partial-set UDA (PDA) [10], where several source classes are missing in the target domain, open-set UDA (ODA) [32], where the target domain contains samples from unknown classes, and open-partial-set UDA (OPDA) [11], where there are only some overlaps in the label sets across domains. More recently, source-free UDA settings (SFUDA) [33, 12] have been explored, where only the source model instead of source data is available for target adaptation, potentially addressing privacy concerns in the source domain. Subsequently, in the context of black-box domain adaptation [34], the privacy of the source domain is fully safeguarded. Specifically, the research community has made significant efforts to develop effective UDA methods in image classification [9, 6] and semantic segmentation [29, 30], which can be seen through two distinct research directions. The first direction focuses on aligning the distributions across domains by minimizing specific discrepancy measures [35–39] or using adversarial learning to maximize domain confusion [9]. Especially, adversarial learning has become a popular approach and has been explored at different levels for domain alignment, including image-level [40], manifold-level [9, 41, 6], and prediction-level [5, 29, 30, 28]. The second direction focuses on target-oriented learning, aiming to learn a good structure for the target domain. This includes self-training approaches [42, 12, 26] and target-specific regularizations [7, 8, 27]. To thoroughly assess the efficacy of model selection baselines, we opt for a diverse set of UDA methods across various UDA scenarios in our model selection experiments and then utilize these baselines to choose the appropriate hyperparameters for different UDA methods.

Model selection in UDA is significant in the practical deployment of UDA methods but remains relatively under-explored. Efforts to address this challenge can be broadly categorized into two lines. Early approaches to model selection in UDA focused on estimating the target domain risk through labeled source data. SourceRisk [9] utilized a hold-out labeled source validation set to guide model selection based on source risk. To mitigate the impact of domain shift on source estimation, [15]

introduced Importance-Weighted Cross-Validation (IWCV), which re-weights source risk using a source-target density ratio estimated in the input space. Building upon this, [16] improved IWCV by introducing Deep Embedded Validation (DEV), which estimates the density ratio in the feature space and offers lower variance. [17] proposed a novel Reverse Validation approach (RV) that leveraged reversed source risk for selection. However, source-based validation methods often necessitate additional model training to handle domain shifts, rendering them cumbersome and less reliable. In contrast, recent model selection methods have shifted their focus exclusively to unlabeled target data, employing specifically designed metrics for model selection. For instance, [18] introduced the mean Shannon's Entropy of target predictions as a model selection metric, promoting confident predictions. [14] proposed the use of Input-Output Mutual Information Maximization (InfoMax)[43] as a metric, augmented with class-balance regularization over Entropy. [13] introduced Soft Neighborhood Density (SND), a novel metric focusing on neighborhood consistency. [19] presented Corr-C, a class correlation-based metric that evaluates both class diversity and prediction certainty simultaneously. Our EnsV approach aligns with the latter line of research. EnsV approaches the model selection problem from a novel perspective, leveraging the power of the off-the-shelf ensemble. Importantly, it operates without making any assumptions about cross-domain distribution shifts or the learned target-domain structure, making it suitable for a variety of UDA scenarios. A comprehensive comparison, as presented in Table 4, underscores that EnsV stands out as a simple and versatile approach.

Ensemble methods, which harness the collective power of a pool of models through prediction averaging, have been extensively studied in the machine learning community for enhancing model performance [44–47] and improving model calibration [48, 49]. In the era of deep learning, the efficiency of ensembling has garnered significant attention due to the high training cost of deep models. Efficient solutions have been proposed, such as using partially shared parameters [50–52] and leveraging intermediate snapshots [53–55]. Recently, weight averaging has gained attention as an efficient alternative to prediction averaging during inference [56–60]. In addition, diversity is considered crucial for effective ensembles. Various approaches have been explored to achieve diverse checkpoints, including bootstrapping [61], random initializations [62], tuning hyperparameters [63, 64, 57], and combining multiple strategies [65]. Different from existing ensemble applications, our work innovatively and elegantly applies ensemble to help address the open problem of unsupervised model selection in domain adaptation.

C Model Selection Baselines

Let $\{p_{\mathrm{t}}^i\}_{i=1}^{n_{\mathrm{t}}}$ represent the target probability output, and let $P \in \mathbb{R}^{n_{\mathrm{t}} \times C}$ denote the prediction matrix. We introduce the practical computation involved in the existing model selection approaches.

Source risk. The Source risk approach (SourceRisk) [9] utilizes a hold-out source validation set to select the model θ_k with the best performance on this set as the final choice. However, this method is limited in its ability to handle significant domain shifts between domains and introduces additional hyperparameters during the splitting of the validation set.

Importance-weighted source risk. Directly taking source risk as target risk is unreliable due to domain distribution shifts between domains. To address this challenge, [15] propose Importance-Weighted Cross Validation (IWCV), which re-weights the source risk using a source-target density ratio estimated in the input space. [16] further enhance IWCV by introducing Deep Embedded Validation (DEV), which estimates the density ratio in the feature space using a domain discriminator and controls the variance. Both IWCV and DEV rely on the importance weighting technique [66], which assumes that the target distribution is included in the source distribution [15], making the weighting unreliable in scenarios with severe covariate shift and label shift. In addition, both IWCV and DEV involve hyperparameters in extra model training for the density ratio estimation.

Reversed source risk. Building upon the concept of reverse cross-validation [67], [17] propose a novel Reverse Validation approach (RV). This method first conducts source-to-target adaptation to obtain a UDA model, which enables the acquisition of pseudo labels for the target unlabeled data. Subsequently, Reverse Validation performs a reversed adaptation from the pseudo-labeled target to the source and utilizes the source risk in this reversed adaptation task for validation. Reverse Validation heavily relies on the symmetry between domains and is unable to handle label shift. Additionally, this approach involves hyperparameters for dataset splitting.

Entropy. [18] propose using the mean Shannon's Entropy of target predictions as a validation metric, which encourages confident predictions. The motivation behind this is that the decision boundary should avoid crossing high-density regions in the target structure [68, 69]. Lower Entropy scores indicate better model performance for this metric.

$$\text{Entropy} = -\frac{1}{n_\text{t}} \sum_{i=1}^{n_\text{t}} \sum_{j=1}^{C} P_{ij} \log P_{ij}, \qquad \text{InfoMax} = -\sum_{j=1}^{C} \bar{p} \log \bar{p} + \frac{1}{n_\text{t}} \sum_{i=1}^{n_\text{t}} \sum_{j=1}^{C} P_{ij} \log P_{ij}$$

Information maximization. The Entropy score only considers sample-wise certainty, which can be misleading when confident predictions are biased towards a small fraction of classes [13]. To address this challenge, [14] utilize input-output mutual information maximization (InfoMax) [43] as a validation metric. In contrast to Entropy, InfoMax includes an additional class-balance regularization by encouraging the averaged prediction $\bar{p} = \frac{1}{n_\text{t}} \sum_{i=1}^{n_\text{t}} P_{ij}, \quad \bar{p} \in \mathbb{R}^C$ to have a large entropy. Higher InfoMax scores indicate better model performance according to this metric.

Neighborhood consistency. [13] introduce Soft Neighborhood Density (SND), a novel metric that focuses on neighborhood consistency. SND leverages softmax predictions as features and constructs a sample-sample similarity matrix. This matrix is transformed into a probabilistic distribution using the softmax function: $S = \text{softmax}(PP^T/\tau), \quad S \in \mathbb{R}^{n_\text{t} \times n_\text{t}}$. Here, τ is a small temperature parameter that sharpens the distribution, enabling the differentiation between nearby and distant samples. SND promotes high neighborhood consistency by encouraging samples to have similar predictions to other points in their neighborhood, resulting in larger SND scores.

$$\text{SND} = -\frac{1}{n_\text{t}} \sum_{i=1}^{n_\text{t}} \sum_{j=1}^{n_\text{t}} S_{ij} \log S_{ij}, \qquad \text{Corr-C} = \frac{\text{sum}(\text{diag}(P^T P))}{\|P^T P\|_\text{F}}$$

Class correlation. [19] introduce Corr-C, a class correlation-based metric that evaluates class diversity and prediction certainty simultaneously. Corr-C calculates the cosine similarity between the class correlation matrix and an identity matrix. Lower Corr-C scores are indicative of better model performance based on this metric.

D Hyperparameter Configurations

In our experiments, we adopt the setting of previous studies [16, 13] by tuning a single hyperparameter for various UDA methods. The comprehensive hyperparameter settings can be found in Table 5. For MCC [27] and MDD [28], we also explore different bottleneck dimensions: $256, 512, 1024, 2048$. Additionally, in semantic segmentation tasks, we consider the training iteration following SND [13].

E Full Experiments

E.1 Setup

Datasets. Our experiments encompass diverse and widely-used image classification benchmarks: (*i*) *Office-31*[21] with 31 classes and 3 domains (Amazon (A), DSLR (D), and Webcam (W)); (*ii*) *Office-Home*[22] with 65 classes and 4 domains (Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Re)); (*iii*) *VisDA*[23] with 12 classes and 2 domains (training (T) and validation (V)); and (*iv*) *DomainNet-126*[20, 5] with 126 classes and 4 domains (Real (R), Clipart (C), Painting (P), and Sketch (S)). Additionally, we conduct experiments in synthetic-to-real semantic segmentation, specifically targeting the transfer from GTAV[24] to Cityscapes[25].

UDA methods. In our experiments, we assess all the model selection methods listed in Table 4. With these validation methods, we perform model selection for various UDA methods across different UDA settings. For CDA, we consider ATDOC [26], BNM [8], CDAN [6], MCC [27], MDD [28], and SAFN [7]. For PDA, we consider PADA [10] and SAFN [7]. For OPDA, we consider DANCE [11]. For SFUDA, we consider the white-box method SHOT [12] and the black-box method DINE [34]. For domain adaptive semantic segmentation, we consider AdaptSeg [29] and AdvEnt [30]. During

Table 5: Overview of the UDA methods validated and their associated hyperparameters

UDA method	UDA Type	Hyperparameter	Search Space	Default Value
ATDOC [26]	closed-set self-training	loss coefficient λ	$\{0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0\}$	0.2
BNM [8]	closed-set output regularization	loss coefficient λ	$\{0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0\}$	1.0
CDAN [6]	closed-set feature alignment	loss coefficient λ	$\{0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$	1.0
MCC [27]	closed-set output regularization	temperature T	$\{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$	2.5
MDD [28]	closed-set output alignment	margin factor γ	$\{0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0\}$	4.0
SAFN [7]	closed/partial-set feature regularization	loss coefficient λ	$\{0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2\}$	0.05
PADA [10]	partial-set feature alignment	loss coefficient λ	$\{0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$	1.0
DANCE [11]	open-partial-set self-supervision	loss coefficient η	$\{0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0\}$	0.05
SHOT [12]	source-free hypothesis transfer	loss coefficient β	$\{0.03, 0.05, 0.1, 0.3, 0.5, 1.0, 3.0\}$	0.3
DINE [26]	black-box knowledge distillation	loss coefficient β	$\{0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$	1.0
AdaptSeg [29]	closed-set output alignment	loss coefficient λ	$\{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03\}$	0.0002
AdvEnt [30]	closed-set output alignment	loss coefficient λ	$\{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03\}$	0.001

Table 6: CDA accuracy (%) on *Office-Home* (*Home*). **bold**: Best value.

Method	ATDOC [26]					BNM [8]					CDAN [6]				
	→Ar	→Cl	→Pr	→Re	avg	→Ar	→Cl	→Pr	→Re	avg	→Ar	→Cl	→Pr	→Re	avg
SourceRisk [9]	66.63	52.54	78.57	76.61	68.59	62.44	50.74	77.53	74.76	66.37	55.00	42.65	69.50	68.81	58.99
IWCV [15]	67.97	54.03	78.31	79.26	69.89	66.56	48.16	74.09	73.28	65.52	61.31	41.24	67.17	71.93	60.41
DEV [16]	67.39	54.23	77.78	79.39	69.70	65.76	56.39	73.92	77.59	68.41	67.23	57.04	68.76	76.91	67.49
RV [17]	68.68	56.13	78.93	79.64	70.85	68.25	56.75	**78.08**	78.67	70.44	67.66	56.74	76.01	77.68	69.52
Entropy [18]	63.67	55.83	76.54	78.36	68.60	66.28	54.49	74.15	77.64	68.14	67.66	**57.56**	76.37	77.45	69.76
InfoMax [14]	63.67	55.63	77.61	78.36	68.82	66.28	54.49	74.15	77.64	68.14	67.66	**57.56**	76.37	77.45	69.76
SND [13]	63.67	55.63	76.54	77.54	68.34	66.28	54.49	74.15	77.64	68.14	**67.94**	**57.56**	76.96	77.68	70.04
Corr-C [19]	63.51	50.39	73.89	73.88	65.42	58.10	45.37	68.97	70.59	60.76	53.84	41.21	64.96	67.65	56.91
EnsV	**68.70**	**58.05**	**79.81**	**80.41**	**71.74**	**68.61**	**57.38**	**78.08**	**79.54**	**70.90**	67.88	**57.56**	**77.39**	**78.19**	**70.25**
Worst	62.89	50.39	73.89	73.88	65.26	58.10	45.37	68.96	70.59	60.75	53.80	41.21	64.78	67.65	56.86
Best	68.97	58.35	80.27	80.58	72.04	68.93	57.51	78.43	79.57	71.11	68.19	57.90	77.44	78.19	70.43

Method	MCC [27]					MDD [28]					SAFN [7]					*Home*
	→Ar	→Cl	→Pr	→Re	avg	→Ar	→Cl	→Pr	→Re	avg	→Ar	→Cl	→Pr	→Re	avg	AVG
SourceRisk [9]	66.57	56.53	79.55	80.90	70.89	62.53	54.43	75.27	75.55	66.94	63.54	51.34	73.66	74.54	65.77	66.26
IWCV [15]	68.69	58.93	80.37	80.08	72.02	64.20	56.50	73.78	74.28	67.19	64.31	**52.36**	72.31	74.29	65.82	66.81
DEV [16]	68.81	58.07	78.54	80.10	71.38	64.42	56.94	76.85	75.94	68.54	63.15	50.47	71.20	74.54	64.84	68.39
RV [17]	**70.40**	58.80	**80.63**	80.39	72.56	**66.57**	55.75	76.60	76.90	68.96	64.31	50.13	73.77	74.93	65.78	69.68
Entropy [18]	69.29	59.33	**80.63**	**80.96**	72.55	66.54	57.63	77.27	**77.45**	69.72	59.85	46.41	72.51	73.18	62.99	68.63
InfoMax [14]	66.58	58.48	79.12	80.81	71.25	66.54	57.74	77.27	**77.45**	**69.75**	64.56	49.71	73.77	73.18	65.31	68.84
SND [13]	69.05	55.61	79.72	79.10	70.87	51.34	38.01	**77.61**	68.46	58.86	57.90	46.41	67.04	68.18	59.88	66.02
Corr-C [19]	69.05	55.61	79.72	79.10	70.87	47.79	31.69	63.40	60.63	50.88	62.66	46.41	68.83	68.18	61.52	61.06
EnsV	69.92	**59.50**	80.30	80.86	**72.65**	66.46	**57.81**	**77.61**	76.51	69.60	**65.91**	52.18	**74.51**	**75.57**	**67.04**	**70.36**
Worst	62.72	54.63	76.19	78.19	67.93	47.79	31.69	63.40	60.63	50.88	57.90	46.41	67.04	68.18	59.88	60.26
Best	70.68	59.95	80.93	81.02	73.14	66.75	58.36	77.61	77.45	70.04	66.59	53.14	74.90	75.57	67.55	70.72

selection, we explore 7 candidate values for each hyperparameter. Specifically, we select the loss coefficient for ATDOC, BNM, CDAN, PADA, SAFN, DANCE, SHOT, DINE, AdaptSeg, and AdvEnt, while the margin is selected for MDD and the temperature for MCC. Additionally, we perform two complex two-hyperparameter validation tasks. For classification, we select the bottleneck dimension among 4 options in MCC and MDD, whereas for segmentation, we select the training iteration among 8 options in AdaptSeg and AdvEnt.

Implementation details. We train UDA models using the Transfer Learning Library[2] on a single RTX TITAN 16GB GPU with a batch size of 32 and a total number of iterations of 5000. Unless specified, checkpoints are saved at the last iteration.We adopt ResNet-101 [70] for *VisDA* and segmentation tasks, ResNet-34 [70] for *DomainNet*, and ResNet-50 [70] for other benchmarks. We

[2] https://github.com/thuml/Transfer-Learning-Library

Table 7: CDA accuracy (%) on *Office-31* (*Office*) and *VisDA*.

Method	ATDOC [26]					BNM [8]					CDAN [6]					MCC [27]					MDD [28]					SAFN [7]					*Office*	*VisDA*
	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	AVG	AVG
SourceRisk [9]	72.56	88.96	**87.80**	83.11	67.79	72.92	**90.36**	**89.43**	84.24	70.51	63.90	91.16	**89.06**	81.37	64.50	73.11	90.96	91.07	85.05	80.46	75.72	91.06	86.23	84.34	72.25	69.20	83.73	87.17	80.03	70.71	83.02	71.04
IWCV [15]	72.56	86.14	86.54	81.75	67.79	72.92	85.54	**89.43**	82.63	**76.94**	63.90	69.08	58.74	63.91	64.50	73.11	91.16	88.55	84.27	81.48	75.49	91.16	89.18	85.28	72.25	69.32	86.55	80.38	78.75	66.33	79.43	71.55
DEV	72.56	86.14	86.54	81.75	70.34	72.92	85.54	**89.43**	82.63	**76.94**	63.90	91.16	88.30	81.12	64.50	72.70	89.16	93.08	84.98	81.48	75.65	91.16	89.18	85.33	72.25	68.21	86.55	80.38	78.38	66.33	82.36	71.97
RV [17]	**74.93**	89.96	87.23	84.04	**77.37**	70.71	88.55	**89.43**	82.90	74.58	**73.27**	91.16	88.30	84.24	76.02	**73.97**	89.06	93.08	85.37	**82.22**	74.46	**92.57**	86.79	84.61	77.23	68.69	90.83	87.17	82.23	66.33	83.90	75.62
Entropy [18]	73.29	86.14	**87.80**	82.41	62.85	72.67	85.54	83.14	80.45	58.36	71.62	91.16	**89.06**	83.95	**80.46**	73.93	90.56	**93.46**	85.98	**82.22**	76.31	**92.57**	**90.82**	86.57	**78.95**	68.23	**91.57**	85.66	81.82	70.20	83.53	72.17
InfoMax [14]	73.29	86.14	**87.80**	82.41	76.49	70.52	85.54	83.14	79.73	58.36	71.62	91.16	88.30	83.69	**80.46**	73.93	89.16	88.55	83.88	81.48	**76.50**	**92.57**	**90.82**	**86.63**	**78.95**	68.23	**91.57**	**87.42**	82.41	70.20	83.13	74.32
SND [13]	73.29	**92.37**	**87.80**	84.49	**77.37**	74.44	85.54	83.14	81.04	69.65	71.55	92.37	88.55	84.16	**80.46**	73.93	**91.97**	**93.46**	**86.45**	69.35	**76.50**	92.17	**90.82**	86.50	**78.95**	68.23	89.96	85.66	81.28	58.15	83.99	72.32
Corr-C [19]	71.05	90.96	84.40	82.14	67.79	67.16	84.34	78.99	76.83	70.51	58.29	67.67	59.62	61.86	64.50	73.93	91.37	**93.46**	86.25	69.35	74.25	91.57	85.66	83.83	72.25	68.39	86.75	80.38	78.51	62.52	78.24	67.82
EnsV	74.83	90.96	**87.80**	**84.53**	73.36	**74.87**	**90.36**	**89.43**	**84.89**	74.58	73.20	**92.77**	88.55	**84.84**	79.05	73.75	90.56	91.45	85.25	**82.22**	75.92	**92.57**	**90.82**	86.44	77.23	**69.67**	90.96	87.17	**82.60**	**73.96**	**84.76**	**76.73**
Worst	71.05	86.14	84.40	80.53	62.85	67.16	84.34	78.99	76.83	23.08	58.29	67.67	57.11	61.02	64.50	70.56	86.75	87.17	81.49	69.35	73.06	87.35	85.66	82.02	72.25	67.27	83.73	80.38	77.13	58.15	76.50	58.36
Best	75.31	92.37	87.80	85.16	77.37	75.52	90.36	89.43	85.10	76.94	73.38	92.77	89.06	85.07	80.46	74.42	91.97	93.46	86.62	82.23	76.52	92.57	92.20	87.10	78.95	70.06	91.57	87.42	83.02	75.30	85.34	78.54

Table 8: PDA accuracy (%) on *Office-Home*.

Method	PADA [10]					SAFN [7]				
	→ Ar	→ Cl	→ Pr	→ Re	avg	→ Ar	→ Cl	→ Pr	→ Re	avg
SourceRisk [9]	57.21	41.90	64.48	71.89	58.87	66.82	54.71	74.41	76.48	68.11
IWCV [15]	59.65	50.51	66.84	72.96	62.49	69.36	53.91	71.78	76.38	67.86
DEV [16]	66.88	49.29	72.40	70.46	64.76	69.36	54.94	73.95	76.06	68.58
RV [17]	57.79	40.87	63.87	70.83	58.34	68.98	52.74	72.83	77.14	67.92
Entropy [18]	60.08	46.51	53.16	62.47	55.56	**71.75**	55.62	76.36	76.59	70.08
InfoMax [14]	60.08	51.40	60.20	66.67	59.59	63.67	51.74	69.64	73.62	64.67
SND [13]	67.80	50.71	59.46	67.13	61.27	**71.75**	51.74	76.36	78.36	69.55
Corr-C [19]	61.34	45.65	54.90	62.25	56.04	71.23	55.70	**76.94**	**79.13**	**70.75**
EnsV	**68.54**	**55.60**	**69.86**	**78.23**	**68.06**	70.98	**56.12**	75.67	78.48	70.31
Worst	56.29	39.76	50.49	59.31	51.46	62.48	49.91	68.50	73.62	63.63
Best	69.33	55.86	74.55	79.59	69.83	73.37	58.09	77.35	79.33	72.03

repeat trials with three random seeds and report the mean for results. Source-based validation methods allocate 80% of the source data for training and the remaining 20% for validation.

E.2 Comprehensive Comparison Results

Consistent with prior model selection studies [16, 13, 14], we extensively compare our EnsV with 8 other methods in standard UDA settings, including CDA and PDA. Averaged results are presented for UDA tasks sharing the same target domain. 'Worst' refers to the selection with the lowest target-domain performance, while 'Best' indicates the opposite.

CDA : We provide model selection results for 6 typical closed-set UDA methods on *Office-Home*, *Office-31*, and *VisDA* in Tables 6 and 7. EnsV method consistently outperforms other validation methods in terms of the average selection accuracy on each benchmark and furthermore, consistently achieves near-best results. Among existing methods, we find the reverse validation (RV) approach is consistently the best among the three benchmarks. However, RV requires extra model re-training, making it impractical when compared to the efficient target-specific validation methods.

PDA : For partial-set UDA with label shift of missing classes, we conduct hyperparameter selections for two different UDA methods on *Office-Home* (Table 8). Notably, existing methods, except for DEV and SND, suffer from frequent low-accuracy selections. In contrast, EnsV consistently achieves high-accuracy selections and, on average, outperforms both DEV and SND.

E.3 Comparison with Target-Specific Baselines

Recent advancements in UDA model selection [13, 14] indicate that validation using only unlabeled target data achieves superior performance compared to source-based methods, with increased simplicity. Eliminating the reliance on source data facilitates easy application in various real-world UDA scenarios, extending beyond conventional closed-set settings. We particularly compare EnsV

with other target-specific validation methods on the large-scale benchmark DomainNet and in two practical UDA settings: OPDA and SFUDA.

CDA : We compare all target-specific validation methods on the large-scale benchmark *DomainNet-126* (Table 2). EnsV consistently keeps the leading validation performance, while other approaches exhibit high instability.

OPDA : In open-partial-set UDA with label shift of unknown classes, we choose a typical method DANCE for validation on *Office-Home* (Table 3) and measure the H-score [31, 71]. Previous validation works have not studied this challenging setting [13], and all of them suffer from poor model selections. In contrast, EnsV consistently achieves high-accuracy selections, close to the best.

SFUDA : In source-free UDA (SFUDA), where source-based validation methods are not applicable, we select SHOT for the white-box setting on *Office-31* and DINE for the black-box setting on *VisDA* (Table 3). EnsV consistently maintains near-best selections, while other target-based approaches frequently make worst-case selections.

Table 9: Two-hyperparameters validation accuracy (%) on *Office-Home*.

Method	MDD [28]					MCC [27]					*Home*
	Ar → Cl	Cl → Pr	Pr → Re	Re → Ar	avg	Ar → Cl	Cl → Pr	Pr → Re	Re → Ar	avg	AVG
SourceRisk [9]	55.99	73.15	78.77	69.39	69.33	57.91	76.84	81.13	72.89	72.19	70.76
IWCV [15]	37.89	72.92	80.42	58.43	62.42	46.09	77.74	80.68	74.45	69.74	66.08
DEV [16]	52.60	72.11	53.36	67.70	61.44	59.47	76.84	81.94	74.08	73.08	67.26
RV [17]	**57.59**	72.25	80.83	70.79	70.37	59.13	76.84	82.03	71.98	72.50	71.44
Entropy [18]	57.21	**73.19**	80.06	**72.31**	70.69	59.75	77.77	82.37	74.33	73.56	72.13
InfoMax [14]	**57.59**	72.92	80.06	**72.31**	**70.72**	59.70	**78.73**	**82.58**	70.33	72.84	71.78
SND [13]	38.10	56.45	70.03	65.10	57.42	53.49	74.97	77.25	74.12	69.96	63.69
Corr-C [19]	30.17	44.74	57.15	50.76	45.71	44.90	56.75	74.32	67.61	60.90	53.31
EnsV	56.91	72.74	**80.93**	71.16	70.44	**60.39**	78.71	82.28	**74.91**	**74.07**	**72.26**
Worst	30.17	39.81	53.36	50.76	43.53	43.02	56.75	73.47	67.24	60.12	51.83
Best	57.59	73.35	80.93	72.52	71.10	61.10	78.94	83.04	75.36	74.61	72.86

E.4 Further Analysis

Validation with two-hyperparameters. We conduct practical two-hyperparameters model selection experiments on classification tasks (Table 9) and segmentation tasks (Table 11). Most validation studies focus on classification, with limited attention [13] to segmentation. We find EnsV achieves near-optimal selections on both tasks, outperforming other generic methods like Entropy and SND.

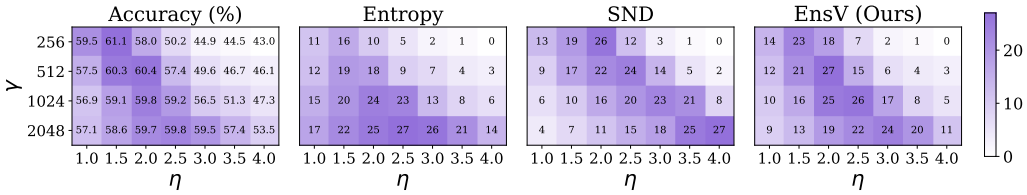

Figure 2: Qualitative comparisons of two-hyperparameters validation for MCC on Ar → Cl.

Qualitative comparison. We perform a qualitative comparison between two state-of-the-art target-based model selection methods, Entropy and SND, and our EnsV. In Figure 2, we present the rankings of the 28 candidate checkpoints in ascending order based on the respective selection metric of each approach. On the left side, we show the rankings according to the real target accuracy and denote the accuracy for each candidate model. Our EnsV demonstrates a high level of consistency with the real target accuracy, while the other methods exhibit significant deviations. This highlights the superior reliability of our EnsV over other methods.

Robustness to architectures. Architecture plays a significant role in the ensemble. In our experiments, we assess the effectiveness of EnsV using various ResNet backbone variants and observe consistent success across different scales. For further study, we conduct validation experiments using the ViT-B [72] architecture on the R→S task with BNM. The validation results, presented in Table 10, demonstrate that EnsV achieves the best selection. However, all other target-based methods except SND make the worst selection.

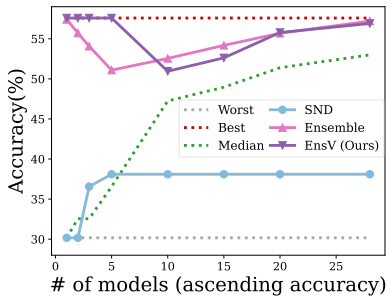

Figure 3: MDD on Ar→Cl.

Table 10: ViT results.

Method	BNM [8]
Entropy [18]	28.21
InfoMax [14]	28.21
SND [13]	52.42
Corr-C [19]	28.21
EnsV	**55.16**
Worst	28.21
Best	55.16

Table 11: Segmentation mIoU (%).

Method	AdaptSegt	AdvEnt
SourceRisk [9]	39.52	39.08
Entropy [18]	39.47	38.41
SND [13]	**40.69**	40.02
EnsV	**40.69**	**40.67**
Worst	35.32	34.22
Best	42.20	41.78

Performance of role models. The effectiveness of our ensemble-based validation method, EnsV, relies on the performance of the role model. We evaluate the target performance of role models for various UDA methods in 4 UDA settings on *Office-Home* and present the results in Table 12. Through a comparison of ensemble performance with model selection performance in our empirical experiments, we demonstrate that the ensemble consistently exhibits high performance. The success of EnsV can be attributed to the robust role model provided by the ensemble. For a comprehensive study, we further present the results of the weight-based ensemble [57], denoted as 'W-Avg,' and the EnsV variant based on this ensemble, denoted as 'EnsV-W.' While the weight-based ensemble also shows competitiveness, it requires all candidate models to share the same architecture and lacks a theoretical guarantee. Thus, we recommend the simple and generic prediction-based ensemble.

Table 12: Accuracy (%) of the ensemble on *Office-Home*.

Method	CDA						PDA		OPDA	SFUDA
	ATDOC [26]	BNM [8]	CDAN [6]	MCC [27]	MDD [28]	SAFN [7]	PADA [10]	SAFN [7]	DANCEDANCE [11]	SHOT [12]
W-Avg	72.04	70.48	69.30	72.77	69.39	66.65	67.46	70.11	64.97	71.82
Ensemble	**72.13**	70.86	**70.32**	**72.82**	**69.80**	67.12	**68.23**	70.71	**69.31**	**71.94**
EnsV-W	71.72	70.74	69.81	72.70	69.23	**67.38**	68.21	**71.91**	66.85	71.74
EnsV	71.74	**70.90**	70.25	72.65	69.60	67.04	68.06	70.71	68.96	71.88
Worst	65.26	60.75	56.86	67.93	50.88	59.88	51.46	63.63	12.84	67.21
Best	72.04	71.11	70.43	73.14	70.04	67.55	69.83	72.03	72.98	72.05

Robustness to bad candidates. The robustness of the ensemble to bad checkpoints is critical for its effectiveness. We conduct two-hyperparameter validation experiments using MDD on Ar→Cl to assess this. In the worst-case scenario where we have only one good checkpoint and several bad checkpoints, the ensemble results may be heavily influenced by the bad checkpoints, leading to poor selections. To analyze this, we rank the 28 candidate checkpoints based on their true target accuracy. Starting with the best and worst checkpoints, we gradually introduce more bad checkpoints into the ensemble. By observing the ensembling and validation performance in Figure 3, we study the impact of bad checkpoints. Despite the presence of bad checkpoints, both the prediction-average Ensemble and our EnsV consistently prioritize selections above the median, demonstrating their resilience. In contrast, the state-of-the-art method SND falls short in surpassing the median selection.

E.5 Task-Level Results

In our evaluation, we conduct hyperparameter selection for both classification and segmentation tasks. For open-set experiments, we utilize the H-score (%) [71, 31] metric, which combines the accuracy of known classes and unknown samples. For semantic segmentation tasks, we employ the mean intersection-over-union (mIoU) (%) [29, 30] metric. For all other classification tasks, we measure the accuracy (%). For clarity, we consolidate the results of UDA tasks with the same target domain. For example, in the case of the *Office-Home* dataset, UDA tasks including 'Cl→Ar', 'Pr→Ar', and 'Re→Ar' share the common target domain 'Ar.' As a result, we have averaged the results of these three UDA tasks and reported the averaged value in the tables within our main text under the row labeled '→ Ar'. Furthermore, it's important to distinguish between the 'avg' row, which signifies the average results within each UDA method's rows to the left of the 'avg' row, and the 'AVG' row, which represents the averaged results across all 'avg' rows associated with different UDA methods. Consequently, the 'AVG' row can be considered more reliable and representative for drawing conclusions. Please refer to Table 13 to Table 27 for the specific task-level validation results.

Table 13: Accuracy (%) of a closed-set UDA method ATDOC [26] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	51.43	**77.31**	78.17	**66.87**	74.36	75.60	61.85	48.06	76.06	71.16	58.14	**84.05**	68.59
IWCV [15]	55.88	76.57	78.88	66.25	74.50	78.33	65.60	48.06	80.58	**72.06**	58.14	83.87	69.89
DEV [16]	51.43	76.55	78.88	66.25	74.36	77.67	64.77	51.29	81.62	71.16	**59.98**	82.43	69.70
RV [17]	56.38	76.12	80.01	66.25	76.80	78.33	67.82	55.62	80.58	71.98	56.40	83.87	70.85
Entropy [18]	55.88	74.14	78.88	59.25	74.52	77.67	64.19	54.39	78.54	67.57	57.23	80.96	68.60
InfoMax [14]	55.88	74.14	78.88	59.25	77.74	77.67	64.19	54.39	78.54	67.57	56.61	80.96	68.82
SND [13]	55.88	74.14	78.88	59.25	74.52	75.21	64.19	54.39	78.54	67.57	56.61	80.96	68.34
Corr-C [19]	51.41	72.00	76.04	59.37	69.36	69.54	61.85	48.04	76.06	69.30	51.71	80.31	65.42
EnsV-W	**57.85**	76.57	**81.04**	66.25	**79.48**	**78.52**	**67.94**	55.62	**82.17**	71.9	59.24	84.03	71.72
EnsV	**57.85**	76.57	80.54	66.25	78.82	**78.52**	**67.94**	**57.07**	**82.17**	71.9	59.24	84.03	**71.74**
Worst	51.41	72.00	76.04	59.25	69.36	69.54	61.85	48.04	76.06	67.57	51.71	80.31	65.26
Best	58.01	77.31	81.04	66.91	79.48	78.52	67.94	57.07	82.17	72.06	59.98	84.03	72.04

Table 14: Accuracy (%) of a closed-set UDA method BNM [8] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	56.93	**77.00**	77.74	57.64	**73.33**	69.36	56.45	42.38	77.19	73.22	52.90	82.26	66.37
IWCV [15]	46.46	**77.00**	79.30	63.86	61.34	62.54	63.95	42.38	78.01	71.86	55.65	83.92	65.52
DEV [16]	57.75	71.62	79.30	57.64	67.90	75.46	**66.21**	**54.04**	78.01	**73.42**	57.37	82.25	68.41
RV [17]	**58.67**	**77.00**	79.30	65.68	**73.33**	75.46	65.64	52.05	**81.25**	**73.42**	**59.54**	83.92	70.44
Entropy [18]	53.40	67.04	78.04	63.41	71.44	73.93	63.58	52.69	80.95	71.86	57.37	**83.96**	68.14
InfoMax [14]	53.40	67.04	78.04	63.41	71.44	73.93	63.58	52.69	80.95	71.86	57.37	**83.96**	68.14
SND [13]	53.40	67.04	78.04	63.41	71.44	73.93	63.58	52.69	80.95	71.86	57.37	**83.96**	68.14
Corr-C [19]	46.46	67.06	74.82	49.73	61.34	62.54	56.45	42.38	74.41	68.11	47.26	78.51	60.76
EnsV-W	**58.67**	**77.00**	**80.61**	**66.21**	**73.33**	**76.75**	**66.21**	53.93	**81.25**	**73.42**	57.59	83.92	70.74
EnsV	**58.67**	**77.00**	**80.61**	**66.21**	**73.33**	**76.75**	**66.21**	53.93	**81.25**	**73.42**	**59.54**	83.92	**70.90**
Worst	46.46	67.04	74.82	49.73	61.34	62.54	56.45	42.38	74.41	68.11	47.26	78.51	60.75
Best	58.67	77.00	80.61	67.16	74.16	76.75	66.21	54.04	81.36	73.42	59.82	84.12	71.11

Table 15: Accuracy (%) of a closed-set UDA method CDAN [6] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	43.41	62.51	75.51	43.96	61.59	57.70	53.75	37.53	73.22	67.28	47.01	84.39	58.99
IWCV [15]	43.18	62.51	77.81	44.71	54.61	56.14	65.14	37.53	**81.85**	74.08	43.02	84.39	60.41
DEV [16]	57.16	71.75	77.81	62.46	55.64	71.08	65.14	56.54	**81.85**	74.08	57.43	78.89	67.49
RV [17]	57.16	71.75	77.78	63.62	72.92	73.40	65.14	54.50	**81.85**	**74.21**	**58.56**	83.37	69.52
Entropy [18]	**57.55**	72.43	77.74	63.62	72.92	73.40	**65.27**	**56.66**	81.20	74.08	58.47	83.76	69.76
InfoMax [14]	**57.55**	72.43	77.74	63.62	72.92	73.40	**65.27**	**56.66**	81.20	74.08	58.47	83.76	69.76
SND [13]	**57.55**	72.43	77.78	**64.61**	**73.73**	73.40	65.14	**56.66**	**81.85**	74.08	58.47	**84.73**	70.04
Corr-C [19]	43.14	63.05	73.61	43.96	54.58	56.12	51.75	37.50	73.22	65.80	43.00	77.25	56.91
EnsV-W	57.18	73.30	77.78	63.37	73.89	73.38	65.14	55.44	81.36	73.88	**58.56**	84.39	69.81
EnsV	**57.55**	**73.71**	**78.33**	**64.61**	**73.73**	**74.39**	65.14	56.56	**81.85**	73.88	**58.56**	**84.73**	**70.25**
Worst	43.14	62.51	73.61	43.96	54.58	56.12	51.63	37.50	73.22	65.80	43.00	77.25	56.86
Best	57.55	73.71	78.33	64.61	73.89	74.39	65.76	56.66	81.85	74.21	59.50	84.73	70.43

Table 16: Accuracy (%) of a closed-set UDA method MCC [27] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	57.23	78.19	**81.75**	60.65	76.50	**78.79**	64.15	53.15	82.17	**74.91**	59.20	83.96	70.89
IWCV [15]	**60.02**	78.15	81.34	68.73	**78.51**	77.85	64.15	**57.85**	81.04	73.18	58.92	84.46	72.02
DEV [16]	57.16	78.15	81.34	**69.10**	73.01	76.80	64.15	**57.85**	82.17	73.18	59.20	84.46	71.38
RV [17]	59.34	**78.53**	80.70	**69.10**	77.83	78.22	**67.20**	**57.85**	82.24	**74.91**	59.20	**85.54**	72.56
Entropy [18]	59.31	**78.53**	81.59	66.87	77.83	**78.79**	**67.20**	**57.85**	**82.51**	73.79	60.82	**85.54**	72.55
InfoMax [14]	**60.02**	74.66	**81.75**	64.98	78.24	78.49	64.15	54.52	82.19	70.62	60.89	84.46	71.25
SND [13]	53.56	77.43	79.46	67.28	76.48	76.80	65.06	54.34	81.04	74.82	58.92	85.24	70.87
Corr-C [19]	53.56	77.43	79.46	67.28	76.48	76.80	65.06	54.34	81.04	74.82	58.92	85.24	70.87
EnsV-W	59.31	77.86	81.59	**69.10**	**78.51**	**78.79**	66.87	**57.85**	82.19	73.79	**61.35**	85.22	**72.70**
EnsV	59.31	77.86	81.59	**69.10**	77.83	**78.79**	66.87	**57.85**	82.19	73.79	**61.35**	85.22	72.65
Worst	53.56	73.44	79.25	60.65	73.01	75.76	59.74	53.15	79.55	67.78	57.18	82.11	67.93
Best	60.02	78.53	81.75	69.22	78.51	78.79	67.90	58.49	82.51	74.91	61.35	85.74	73.14

Table 17: Accuracy (%) of a closed-set UDA method MDD [28] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	54.85	73.35	77.05	58.76	69.95	72.23	60.03	51.02	77.36	68.81	57.42	82.50	66.94
IWCV [15]	56.40	69.52	76.59	58.76	67.40	69.43	61.89	56.43	76.82	71.94	56.68	**84.43**	67.19
DEV [16]	57.71	**75.42**	77.05	58.76	**72.99**	70.51	**63.95**	56.43	80.26	70.54	56.68	82.14	68.54
RV [17]	**58.05**	**75.42**	76.59	63.54	69.95	**73.74**	**63.95**	51.02	**80.38**	**72.23**	58.17	**84.43**	68.96
Entropy [18]	57.73	74.54	**78.22**	**64.07**	**72.99**	**73.74**	**63.95**	55.85	**80.38**	71.61	59.31	84.28	69.72
InfoMax [14]	**58.05**	74.54	**78.22**	**64.07**	**72.99**	**73.74**	**63.95**	55.85	**80.38**	71.61	59.31	84.28	**69.75**
SND [13]	**58.05**	**75.42**	77.05	44.99	**72.99**	48.06	37.08	21.60	80.26	71.94	34.39	**84.43**	58.86
Corr-C [19]	39.08	59.74	69.61	44.99	54.58	48.06	37.08	21.60	64.22	61.31	34.39	75.87	50.88
EnsV-W	54.89	**75.42**	78.01	61.89	**72.99**	72.23	63.08	56.43	79.66	**72.23**	**60.02**	83.96	69.23
EnsV	56.40	**75.42**	77.05	**64.07**	**72.99**	72.23	63.08	**57.02**	80.26	**72.23**	**60.02**	**84.43**	69.60
Worst	39.08	59.74	69.61	44.99	54.58	48.06	37.08	21.60	64.22	61.31	34.39	75.87	50.88
Best	58.05	75.42	78.22	64.07	72.99	73.74	63.95	57.02	80.38	72.23	60.02	84.43	70.04

Table 18: Accuracy (%) of a closed-set UDA method SAFN [7] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	50.78	69.72	76.06	59.66	70.29	69.86	60.90	46.07	**77.71**	70.05	**57.16**	80.96	65.77
IWCV [15]	50.24	69.72	77.28	62.63	67.24	69.86	58.84	49.69	75.72	**71.45**	**57.16**	79.97	65.82
DEV [16]	51.07	69.72	76.64	59.66	67.24	71.26	58.84	49.69	75.72	70.95	50.65	76.64	64.84
RV [17]	51.07	71.41	76.64	62.63	68.44	70.44	58.84	44.49	**77.71**	**71.45**	54.82	**81.46**	65.78
Entropy [18]	45.93	69.72	75.49	55.29	67.22	68.35	54.26	43.30	75.69	70.00	49.99	80.60	62.99
InfoMax [14]	50.47	69.72	75.49	62.46	**70.98**	68.35	61.23	43.30	75.69	70.00	55.37	80.60	65.31
SND [13]	45.93	64.36	70.60	55.29	60.13	62.50	54.26	43.30	71.43	64.15	49.99	76.64	59.88
Corr-C [19]	45.93	69.72	70.60	55.29	60.13	62.50	61.23	43.30	71.43	**71.45**	49.99	76.64	61.52
EnsV-W	**51.73**	72.07	76.64	**64.65**	**70.98**	71.26	**63.66**	**50.52**	77.48	70.99	**57.16**	**81.46**	**67.38**
EnsV	51.07	**72.27**	**77.30**	63.58	70.29	**71.70**	62.71	49.69	**77.71**	**71.45**	55.78	80.96	67.04
Worst	45.93	64.36	70.60	55.29	60.13	62.50	54.26	43.30	71.43	64.15	49.99	76.64	59.88
Best	51.73	72.27	77.30	64.65	70.98	71.70	63.66	50.52	77.71	71.45	57.16	81.46	67.55

Table 19: Accuracy (%) of closed-set UDA methods on *Office-31*.

Method	ATDOC [26]					BNM [8]					CDAN [6]				
	A → D	A → W	D → A	W → A	AVG	A → D	A → W	D → A	W → A	AVG	A → D	A → W	D → A	W → A	AVG
SourceRisk [9]	88.96	**87.80**	73.65	71.46	80.47	**90.36**	**89.43**	73.13	72.70	81.41	91.16	**89.06**	66.33	61.46	77.00
IWCV [15]	86.14	86.54	73.65	71.46	79.45	85.54	**89.43**	73.13	72.70	80.20	69.08	58.74	66.33	61.46	63.90
DEV [16]	86.14	86.54	73.65	71.46	79.45	85.54	**89.43**	73.13	72.70	80.20	91.16	88.30	66.33	61.46	76.81
RV [17]	89.96	87.23	74.28	**75.58**	81.76	88.55	**89.43**	74.90	66.52	79.85	91.16	88.30	**76.18**	**70.36**	81.50
Entropy [18]	86.14	**87.80**	73.87	72.70	80.13	85.54	83.14	71.07	74.26	78.50	91.16	**89.06**	72.88	**70.36**	80.87
InfoMax [14]	86.14	**87.80**	73.87	72.70	80.13	85.54	83.14	71.07	69.97	77.43	91.16	88.30	72.88	**70.36**	80.68
SND [13]	**92.37**	**87.80**	73.87	72.70	81.69	85.54	83.14	74.62	74.26	79.39	92.37	88.55	72.88	70.22	81.01
Corr-C [19]	90.96	84.40	71.88	70.22	79.37	84.34	78.99	67.80	66.52	74.41	67.67	59.62	58.15	58.43	60.97
EnsV-W	**92.37**	**87.80**	**74.65**	75.01	**82.46**	88.55	**89.43**	**75.43**	**75.29**	82.18	**92.77**	88.55	**76.18**	70.22	**81.93**
EnsV	90.96	**87.80**	**74.65**	75.01	82.11	**90.36**	**89.43**	**75.43**	74.30	**82.38**	**92.77**	88.55	**76.18**	70.22	**81.93**
Worst	86.14	84.40	71.88	70.22	78.16	84.34	78.99	67.80	66.52	74.41	67.67	57.11	58.15	58.43	60.34
Best	92.37	87.80	75.04	75.58	82.70	90.36	89.43	75.75	75.29	82.71	92.77	89.06	76.18	70.57	82.15

Table 20: Accuracy (%) of closed-set UDA methods on *Office-31*.

Method	MCC [27]					MDD [28]					SAFN [7]				
	A → D	A → W	D → A	W → A	AVG	A → D	A → W	D → A	W → A	AVG	A → D	A → W	D → A	W → A	AVG
SourceRisk [9]	90.96	91.07	73.33	72.89	82.06	91.06	86.23	76.68	74.76	82.18	83.73	87.17	68.96	69.44	77.33
IWCV [15]	91.16	88.55	73.33	72.89	81.48	91.16	89.18	76.68	74.30	82.83	86.55	80.38	68.96	**69.68**	76.39
DEV [16]	89.16	93.08	73.33	72.06	81.91	91.16	89.18	76.68	74.62	82.91	86.55	80.38	68.96	67.45	75.84
RV [17]	89.06	93.08	74.42	73.52	82.52	**92.57**	86.79	73.94	**74.97**	82.07	90.83	87.17	68.76	68.62	78.85
Entropy [18]	90.56	**93.46**	**74.83**	73.02	82.97	**92.57**	**90.82**	**78.03**	74.58	84.00	**91.57**	85.66	67.20	69.26	78.42
InfoMax [14]	89.16	88.55	74.16	**73.70**	81.39	**92.57**	**90.82**	**78.03**	**74.97**	**84.10**	**91.57**	**87.42**	67.20	69.26	78.86
SND [13]	**91.97**	**93.46**	**74.83**	73.02	**83.32**	92.17	**90.82**	**78.03**	**74.97**	84.00	89.96	85.66	67.20	69.26	78.02
Corr-C [19]	91.37	**93.46**	**74.83**	73.02	83.17	91.57	85.66	73.91	74.58	81.43	86.75	80.38	67.09	**69.68**	75.98
EnsV-W	90.56	91.07	74.16	**73.70**	82.37	**92.57**	**90.82**	77.53	74.30	83.80	**91.57**	87.17	**70.22**	69.12	**79.52**
EnsV	90.56	91.45	73.80	**73.70**	82.38	**92.57**	**90.82**	77.53	74.30	83.80	90.96	87.17	**70.22**	69.12	79.37
Worst	86.75	87.17	71.18	69.93	78.76	87.35	85.66	73.91	72.20	79.78	83.73	80.38	67.09	67.45	74.66
Best	91.97	93.46	74.83	74.01	83.57	92.57	92.20	78.03	75.01	84.45	91.57	87.42	70.43	69.68	79.78

Table 21: Accuracy (%) of a closed-set UDA method CDAN [6] on *DomainNet-126*.

Method	C → S	P → C	P → R	R → C	R → P	R → S	S → P	AVG
Entropy [18]	**58.04**	**64.78**	74.42	**69.39**	**68.65**	**60.63**	**62.94**	**65.55**
InfoMax [14]	**58.04**	**64.78**	74.42	**69.39**	**68.65**	**60.63**	**62.94**	**65.55**
SND [13]	**58.04**	**64.78**	74.42	**69.39**	**68.65**	**60.63**	60.70	65.23
Corr-C [19]	**58.04**	57.73	74.42	56.98	65.07	51.23	60.70	60.60
EnsV-W	55.15	60.98	73.86	60.99	65.07	55.50	60.27	61.69
EnsV	56.73	64.67	**74.44**	67.08	67.97	58.12	62.57	64.51
Worst	51.59	57.73	73.44	56.98	63.06	51.23	58.46	58.93
Best	58.04	64.78	74.44	69.39	68.65	60.63	62.94	65.55

Table 22: Accuracy (%) of a closed-set UDA method BNM [8] on *DomainNet-126*.

Method	C → S	P → C	P → R	R → C	R → P	R → S	S → P	AVG
Entropy [18]	56.42	61.57	74.31	65.15	65.15	40.95	63.42	61.00
InfoMax [14]	56.42	68.95	74.31	65.15	65.15	54.93	63.42	64.05
SND [13]	43.78	61.57	74.31	51.55	54.40	40.95	54.59	54.45
Corr-C [19]	43.78	60.03	**77.62**	59.47	67.19	40.95	59.64	58.38
EnsV-W	**58.48**	68.42	**77.62**	66.05	**67.79**	**57.65**	**64.34**	65.76
EnsV	57.73	**69.63**	**77.62**	**66.10**	**67.79**	**57.65**	**64.34**	**65.84**
Worst	43.78	60.03	74.31	51.55	54.40	40.95	54.59	54.23
Best	58.48	69.63	78.68	66.10	67.79	58.50	65.20	66.34

Table 23: Accuracy (%) of a closed-set UDA method ATDOC [26] on *DomainNet-126*.

Method	C → S	P → C	P → R	R → C	R → P	R → S	S → P	AVG
Entropy [18]	46.43	65.98	79.60	61.52	64.24	57.92	59.46	62.16
InfoMax [14]	46.43	65.98	79.60	61.52	64.24	57.92	59.46	62.16
SND [13]	46.43	65.98	79.60	61.52	64.24	47.58	59.46	60.69
Corr-C [19]	54.71	60.63	74.42	59.33	64.58	52.66	59.95	60.90
EnsV-W	**63.12**	69.57	78.33	67.93	69.32	60.85	66.33	67.92
EnsV	62.11	**71.14**	**80.01**	**69.45**	**69.79**	**61.35**	**67.10**	**68.71**
Worst	46.43	60.63	74.42	59.33	64.24	47.58	59.46	58.87
Best	63.12	71.14	80.38	69.45	69.79	61.35	67.10	68.90

Table 24: Accuracy (%) of a partial-set UDA method PADA [10] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	45.03	68.85	81.89	43.25	46.83	57.26	57.12	36.42	76.53	71.26	44.24	77.76	58.87
IWCV [15]	**55.58**	65.10	84.54	51.42	**61.29**	53.01	57.02	35.16	81.34	70.52	**60.78**	74.12	62.49
DEV [16]	54.81	**78.15**	78.02	**58.13**	**61.29**	50.14	67.86	35.16	83.21	74.66	57.91	77.76	64.76
RV [17]	43.22	65.10	81.89	42.70	48.74	52.79	57.21	35.16	77.80	73.46	44.24	77.76	58.34
Entropy [18]	40.12	40.11	55.94	52.43	37.25	50.14	57.30	47.22	81.34	70.52	52.18	82.13	55.56
InfoMax [14]	54.81	69.24	78.02	52.43	37.25	50.14	57.30	47.22	71.84	70.52	52.18	74.12	59.59
SND [13]	40.12	40.11	55.94	**58.13**	56.13	64.11	70.62	**51.22**	81.34	74.66	**60.78**	82.13	61.27
Corr-C [19]	40.12	40.11	55.94	54.18	46.89	53.01	58.59	38.93	77.80	71.26	57.91	77.70	56.04
EnsV-W	**55.58**	77.25	86.14	**58.13**	60.17	**67.86**	**73.00**	37.97	**84.04**	76.77	57.91	83.75	**68.21**
EnsV	54.81	69.24	**86.53**	**58.13**	56.13	64.11	70.62	**51.22**	**84.04**	**76.86**	**60.78**	**84.20**	68.06
Worst	40.12	40.11	55.94	41.41	37.25	50.14	56.93	34.87	71.84	70.52	44.30	74.12	51.46
Best	55.58	78.15	86.53	58.13	61.29	68.19	73.00	51.22	84.04	76.86	60.78	84.20	69.83

Table 25: Accuracy (%) of a partial-set UDA method SAFN [7] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
SourceRisk [9]	**59.40**	77.14	81.34	63.97	67.00	71.29	65.60	46.21	76.81	70.89	58.51	79.10	68.11
IWCV [15]	52.24	74.45	**82.16**	70.98	62.41	70.18	63.45	53.49	76.81	73.65	56.00	78.49	67.86
DEV [16]	55.22	74.45	80.07	70.98	67.00	71.29	63.45	51.70	76.81	73.65	57.91	80.39	68.58
RV [17]	53.67	71.60	81.34	67.58	67.00	73.27	65.70	48.54	76.81	73.65	56.00	79.89	67.92
Entropy [18]	58.93	74.90	80.73	70.98	**74.12**	69.80	70.16	50.09	79.24	74.10	57.85	80.06	70.08
InfoMax [14]	51.82	67.62	76.97	64.65	65.77	69.80	59.69	50.09	74.10	66.67	53.31	75.52	64.67
SND [13]	51.82	74.90	80.73	70.98	**74.12**	**75.10**	70.16	50.09	79.24	74.10	53.31	80.06	69.55
Corr-C [19]	**59.40**	**77.20**	**82.16**	67.58	72.89	**75.10**	70.16	**55.70**	80.12	**75.94**	52.00	**80.73**	70.75
EnsV-W	**59.40**	**77.20**	**82.16**	**71.72**	72.89	74.82	**72.45**	**55.70**	**80.73**	**75.94**	**59.16**	**80.73**	**71.91**
EnsV	55.22	76.30	81.28	67.58	70.31	74.05	70.16	54.63	80.12	75.21	58.51	80.39	70.31
Worst	51.52	67.62	76.97	61.07	62.35	69.80	59.69	46.21	74.10	66.67	52.00	75.52	63.63
Best	59.40	77.20	82.16	71.72	74.12	75.10	72.45	55.70	80.73	75.94	59.16	80.73	72.03

Table 26: H-score [71, 31] (%) of an open-partial-set UDA method DANCE [11] on *Office-Home*.

Method	Ar → Cl	Ar → Pr	Ar → Re	Cl → Ar	Cl → Pr	Cl → Re	Pr → Ar	Pr → Cl	Pr → Re	Re → Ar	Re → Cl	Re → Pr	AVG
Entropy [18]	38.29	26.08	36.51	32.92	17.10	32.19	37.69	46.40	45.53	25.39	33.75	39.37	34.27
InfoMax [14]	38.29	26.08	36.51	32.92	17.10	32.19	37.69	46.40	45.33	25.39	33.75	39.37	34.25
SND [13]	1.00	0.00	12.73	0.00	42.84	1.95	19.77	11.99	35.69	25.39	0.00	28.40	14.98
Corr-C [19]	1.00	0.00	12.73	0.00	42.84	1.95	19.77	11.99	35.69	69.02	0.00	28.40	18.62
EnsV-W	**67.00**	75.15	**66.57**	67.87	67.35	59.05	66.41	**62.59**	**69.40**	59.86	**67.54**	73.40	66.85
EnsV	38.40	**76.96**	**66.57**	**71.76**	**75.17**	**69.99**	**77.42**	48.15	**69.40**	**81.84**	**67.54**	**84.31**	**68.96**
Worst	1.00	0.00	12.73	0.00	17.10	1.95	19.77	11.99	35.69	25.39	0.00	28.40	12.84
Best	67.00	76.96	66.57	71.76	75.17	69.99	77.42	64.32	72.87	81.84	67.54	84.31	72.98

Table 27: Accuracy (%) of a source-free UDA method SHOT [12] on *Office-31*.

Method	A → D	A → W	D → A	W → A	AVG
Entropy [18]	90.76	88.68	71.21	72.13	80.69
InfoMax [14]	90.76	88.68	71.21	72.13	80.69
SND [13]	90.76	88.68	71.21	72.13	80.69
Corr-C [19]	90.76	90.19	71.21	71.96	81.03
EnsV-W	**94.78**	**91.82**	**75.15**	**74.55**	**84.08**
EnsV	**94.78**	**91.82**	**75.15**	**74.55**	**84.08**
Worst	90.76	88.68	71.21	71.92	80.64
Best	94.78	93.33	75.58	74.55	84.56