Towards Reliable Model Selection for Unsupervised Domain Adaptation: An Empirical Study and A Certified Baseline

Anonymous Author(s)
Affiliation
Address
email

Abstract

Selecting appropriate hyperparameters is crucial for unlocking the full potential of advanced unsupervised domain adaptation (UDA) methods in unlabeled target domains. Although this challenge remains under-explored, it has recently garnered increasing attention with the proposals of various model selection methods. Reliable model selection should maintain performance across diverse UDA methods and scenarios, especially avoiding highly risky worst-case selections—selecting the model or hyperparameter with the worst performance in the pool. Are existing model selection methods reliable and versatile enough for different UDA tasks? In this paper, we provide a comprehensive empirical study involving 8 existing model selection approaches to answer this question. Our evaluation spans 12 UDA methods across 5 diverse UDA benchmarks and 5 popular UDA scenarios. Surprisingly, we find that none of these approaches can effectively avoid the worst-case selection. In contrast, a simple but overlooked ensemble-based selection approach, which we call EnsV, is both theoretically and empirically certified to avoid the worst-case selection, ensuring high reliability. Additionally, EnsV is versatile for various practical but challenging UDA scenarios, including validation of open-partial-set UDA and source-free UDA. Finally, we call for more attention to the reliability of model selection in UDA: avoiding the worst-case is as significant as achieving peak selection performance and should not be overlooked when developing new model selection methods. Code is available in the supplementary materials.

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 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, successful applications of UDA methods across diverse tasks rely heavily on selecting appropriate hyperparameters. Sub-optimal hyperparameters can cause state-of-the-art UDA methods to underperform compared to the source-trained model without target-domain adaptation [19, 18]. This

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

Table 1: Statistics for worst-case selections by various model selection methods are provided across 110 closed-set UDA tasks (potentially an additional 21 tasks on DomainNet [13]), 24 partial-set UDA tasks, and 17 source-free UDA tasks (only for applicable methods). These statistics represent the count of worst-case selections divided by the total count of tasks, with **bold** font indicating the best worst-case avoidance. 'n.a.' indicates that certain methods are not applicable without source data.

Method	Closed-set UDA	Partial-set UDA	Source-free UDA
SourceRisk [9]	16 / 110	2 / 24	n.a.
IWCV [14]	15 / 110	3 / 24	n.a.
DEV [15]	9 / 110	1 / 24	n.a.
RV [16]	2 / 110	1 / 24	n.a.
Entropy [17]	15 / 131	7 / 24	16 / 17
InfoMax [18]	9 / 131	12 / 24	16 / 17
SND [19]	33 / 131	3 / 24	11 / 17
Corr-C [20]	80 / 131	4 / 24	3 / 17
EnsV (Ours)	**0 / 131**	**0 / 24**	**0 / 17**

phenomenon emphasizes the significance of model selection, also called hyperparameter selection or validation, in UDA. Taking the typical one-hyperparameter validation task of a given UDA method as an example, we need to determine the optimal value of a hyperparameter η among a set of m different candidate values $\{\eta_i\}_{i=1}^m$. By applying these different η_i with the same UDA method, we can obtain a set of m different models with the parameter weights $\{\theta_i\}_{i=1}^m$. 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, 14–16]. The second type designs unsupervised metrics based on priors of the learned target-domain structure and utilizes the metrics for model selection [17, 19, 18, 20]. It is natural to ask: Are these approaches reliable in model selection tasks, i.e., can they maintain good performance for various practical UDA tasks?

To answer this question, we conduct an extensive empirical study to assess the performance of all selection methods across various practical UDA settings, including closed-set UDA [21], partial-set UDA [10], open-partial-set UDA [11], and source-free UDA [12, 22]. Notably, the model selection problem of open-partial-set UDA has not been investigated before. Surprisingly, we find that 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 [18]. For instance, Table 1 compares the worst-case selection statistics of all these model selection methods across various practical UDA settings. These settings include standard closed-set UDA and partial-set UDA, which have been extensively studied in prior works [15, 19], and source-free UDA, where the model selection problem has not been widely investigated. The comparison reveals that all the methods occasionally or even frequently suffer from worst-case model selection situations, indicating high unreliability.

In contrast, we note that a simple ensemble-based validation baseline, dubbed EnsV, can effectively avoid the worst-case selection. Through a straightforward theoretical analysis of the ensemble, we observe that it is guaranteed to surpass the worst candidate model's performance. Our introduced EnsV takes a further simple 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. Moreover, EnsV only uses target-domain predictions inferred by all candidate models. This eliminates the need for specific domain shift assumptions and access to source data, while also requiring no additional effort, such as time and memory, as all models are provided within the given problem context. This simplicity and versatility make EnsV suitable for various practical UDA scenarios, including the

Table 2: Comparisons of unsupervised model selection approaches used for UDA.

Method	covariate shift	label shift	w/o source data	w/o extra hyperparameter	w/o extra training	worst-case avoidance
SourceRisk [9]	✗	✗	✗	✗	✓	✗
IWCV [14]	✓	✗	✗	✗	✗	✗
DEV [15]	✓	✗	✗	✗	✗	✗
RV [16]	✓	✗	✗	✗	✗	✗
Entropy [17]	✓	✗	✓	✓	✓	✗
InfoMax [18]	✓	✗	✓	✓	✓	✗
SND [19]	✓	✓	✓	✗	✓	✗
Corr-C [20]	✓	✗	✓	✓	✓	✗
EnsV (Ours)	✓	✓	✓	✓	✓	✓

unexplored challenges of validation for UDA with unknown open classes [19]. Despite EnsV not being certified for peak-performance selection, we hope that, as the first to focus on the practical aspect of worst-case avoidance in model selection, our empirical study and simple baseline can inspire future efforts in developing more reliable model selection methods.

2 Related Work

Unsupervised domain adaptation (UDA) is initially studied in a closed-set setting (CDA) where only covariate shift [14] 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) [23], 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) [24, 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 [22], 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 [25, 26], which can be seen through two distinct research directions. The first direction focuses on aligning the distributions across domains by minimizing specific discrepancy measures [27, 28, 21, 29, 30] 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 [31], manifold-level [9, 32, 6], and prediction-level [5, 25, 26, 33]. The second direction focuses on target-oriented learning, aiming to learn a good structure for the target domain. This includes self-training approaches [34, 12, 35] and target-specific regularizations [7, 8, 36]. 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, [14] 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, [15] improved IWCV by introducing Deep Embedded Validation (DEV), which estimates the density ratio in the feature space and offers lower variance. [16] 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, [17] introduced the mean Shannon's Entropy of target predictions as a model selection metric, promoting confident predictions. [18] proposed the use of Input-Output Mutual Information Maximization (InfoMax)[37] as a metric, augmented with class-balance regularization over Entropy. [19] introduced Soft Neighborhood Density (SND), a novel metric focusing on neighborhood consistency. [20] presented Corr-C, a class correlation-based metric that evaluates both class diversity and prediction certainty simultaneously. Our EnsV baseline aligns with the latter line of research. 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 2, 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 [38–41] and improving model calibration [42, 43]. 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 [44–46] and leveraging intermediate snapshots [47–49]. Recently, weight averaging has gained attention as an efficient alternative to prediction averaging during inference [50–54]. In addition, diversity is considered crucial for effective ensembles. Various approaches have been explored to achieve diverse checkpoints, including bootstrapping [55], random initializations [56], tuning hyperparameters [57, 58, 51], and combining multiple strategies [59]. Different from mainstream ensemble applications, our work innovatively and elegantly applies ensemble to help address the open problem of unsupervised model selection in various domain adaptation scenarios. In addition, [60] leverages ensembles for hyperparameter selection in CDA but directly uses prediction-based ensembling as the output, unlike our EnsV, which includes a selection step.

3 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 parameter 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 $\sum_{i=1}^{C} p^i = 1$.

Model selection in UDA is essentially equivalent to the hyperparameter selection challenge. Here, we aim to determine the optimal value for the hyperparameter η from a set of m candidate values $\{\eta_i\}_{i=1}^m$. The hyperparameter η can encompass various aspects, including 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 the presentation. This also applies to model selection, hyperparameter selection, and validation.

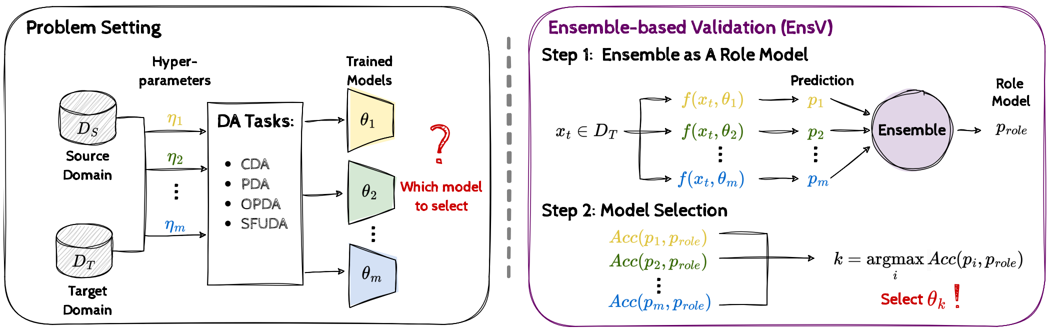

Figure 1: **Left**: Depiction of the unsupervised model selection problem in domain adaptation scenarios, where the objective is to identify the optimal model for the unlabeled target domain. **Right**: Overview of our approach, EnsV, for model selection, which relies solely on predictions of target data by all candidate models.

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

Model selection in UDA is challenging due to the absence of labeled target data for directly evaluating candidate models. Existing selection approaches typically address this challenge from two perspectives: leveraging labeled source data [15] or designing unsupervised metrics based on specific assumed priors [19]. Surprisingly, we've observed that all existing model selection methods treat each candidate model independently, overlooking the collective potential offered by the off-the-shelf ensemble created by these candidates. In this paper, unless otherwise specified, the ensemble refers to prediction-based ensembling, which involves averaging probability predictions across all models to obtain the averaged prediction, i.e., $\frac{1}{m}\sum_{i=1}^{m} f(x, \theta_i)$ for a sample x.

In contrast, we first investigate the potential of the ensemble within the model selection problem. When contemplating the use of the ensemble, two primary concerns often arise, one concerning low efficiency due to training multiple models and the other related to the potential lack of diversity among candidate models. Upon closer inspection of model selection, we observe that the problem setting inherently offers a range of pre-existing candidate models, effectively addressing the efficiency concern without requiring extra model training. Furthermore, all candidate models are trained using a UDA method with varying hyperparameter values, resulting in diverse yet effective discriminative abilities. This naturally mitigates the diversity concern. *Interestingly, the ensemble emerges as a "free lunch" in UDA model selection, a previously overlooked insight.* To delve deeper 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 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 models* $\{\theta_i\}_{i=1}^{m}$, *and the loss of the worst one* 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 the Appendix for the proof. This proposition theoretically guarantees that the ensemble strictly outperforms the worst candidate model.

3.2 Ensemble-based Validation (EnsV): Ensemble as a Role Model for Model Selection

Intuitively, 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. 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 unavailable ground truth. This enables us to obtain reliable predictions for all target data, denoted as $\{\frac{1}{m}\sum_{i=1}^{m} f(x_j, \theta_i)\}_{j=1}^{n_t}$. These ensembles can be viewed as the output of a reliable role model, aiding in accurate model selection in the subsequent step. We then utilize the role model to assess all candidate models and select the one with the highest similarity. For simplicity, EnsV involves direct measurement of accuracy between the role model output $\{\frac{1}{m}\sum_{i=1}^{m} f(x_j, \theta_i)\}_{j=1}^{n_t}$ and the predictions made by each candidate model, such as $\{f(x_j, \theta_i)\}_{j=1}^{n_t}$ for the model with weights θ_i. We select the model θ_k with the highest accuracy and determine the optimal value η_k for the hyperparameter η. Figure 1 provides a vivid illustration of our approach, EnsV. Guided by a reliable role model, EnsV can safely avoid selecting the worst candidate model, a distinct advantage over all existing model selection approaches.

4 Experiments

4.1 Setup

Datasets Our experiments encompass diverse and widely-used image classification benchmarks: (*i*) *Office-31*[61] with 31 classes and 3 domains (Amazon (A), DSLR (D), and Webcam (W)); (*ii*) *Office-Home*[62] with 65 classes and 4 domains (Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Re)); (*iii*) *VisDA*[63] with 12 classes and 2 domains (training (T) and validation (V)); and (*iv*) *DomainNet-126*[13, 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*[64] to *Cityscapes*[65].

UDA methods In our experiments, we assess all the model selection approaches listed in Table 2. Kindly refer to the Appendix for detailed introductions of them. With these approaches, we perform model selection for various UDA methods across different UDA settings. For CDA of image classification, we consider ATDOC [35], BNM [8], CDAN [6], MCC [36], MDD [33], 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 [22]. For domain adaptive semantic segmentation, we consider AdaptSeg [25] and AdvEnt [26]. Following previous model selection studies [15, 19], we primarily focus on one-hyperparameter validation and present the comprehensive hyperparameter settings for all UDA methods in the Appendix. For each hyperparameter, we generally explore 7 candidate values. Additionally, we perform two types of challenging two-hyperparameter validation tasks. For classification tasks, we select the bottleneck dimension as the second hyperparameter from 4 options: $256, 512, 1024, 2048$ in MCC and MDD. For segmentation tasks, following SND [19], we select the training iteration as the second hyperparameter from 8 options, ranging from 16,000 to 30,000 iterations at intervals of 2,000 iterations, in AdaptSeg and AdvEnt.

Implementation details For all UDA methods, we train UDA models using the Transfer Learning Library[1] or the official GitHub code 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 [66] for *VisDA* and segmentation tasks, ResNet-34 [66] for *DomainNet*, and ResNet-50 [66] for other benchmarks. We assess the selection performance of all model selection methods on our trained models for fair comparisons. As a result, comparing our reported values with those from the original papers [15, 19] would be inappropriate. We 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.

4.2 Comprehensive Comparison of All Model Selection Methods

Following prior studies [15, 19, 18], 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. For example, results of 'Cl→Ar', 'Pr→Ar', and 'Re→Ar' on *Office-Home* are averaged and reported under the column labeled '→ Ar'. In addition, the column

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

Table 3: Validation accuracy (%) of CDA on *Office-Home* (*Home*). **bold**: Best value.

Method	ATDOC [35]					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 [14]	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 [15]	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 [16]	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 [17]	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 [18]	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 [19]	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 [20]	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 [36]					MDD [33]					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 [14]	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 [15]	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 [16]	**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 [17]	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 [18]	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 [19]	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 [20]	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

Table 4: Validation accuracy (%) of CDA on *Office-31* (*Office*) and *VisDA*.

Method	ATDOC [35]					BNM [8]					CDAN [6]				
	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V
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
IWCV [14]	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
DEV [15]	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
RV [16]	**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
Entropy [17]	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**
InfoMax [18]	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**
SND [19]	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**
Corr-C [20]	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
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
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
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

Method	MCC [36]					MDD [33]					SAFN [7]					*Office*	*VisDA*
	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	→A	→D	→W	avg	T→V	AVG	AVG
SourceRisk [9]	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 [14]	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 [15]	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 [16]	**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 [17]	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 [18]	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 [19]	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 [20]	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	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	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	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

'avg' signifies the averaged results for each UDA method while the 'AVG' row represents the averaged results across different UDA methods. 'Worst' refers to the worst candidate model with the lowest target-domain performance, while 'Best' indicates the best candidate model with the highest performance. Kindly refer to the Appendix for full results.

CDA We provide model selection results for 6 typical closed-set UDA methods on *Office-Home*, *Office-31*, and *VisDA* in Tables 3 and 4. EnsV consistently outperforms other validation methods in terms of the average selection accuracy on each benchmark and consistently achieves near-best

Table 5: Validation accuracy (%) of PDA on *Office-Home*.

Method	SAFN [7]					PADA [10]					*Home*
	→ Ar	→ Cl	→ Pr	→ Re	avg	→ Ar	→ Cl	→ Pr	→ Re	avg	AVG
SourceRisk [9]	66.82	54.71	74.41	76.48	68.11	57.21	41.90	64.48	71.89	58.87	63.49
IWCV [14]	69.36	53.91	71.78	76.38	67.86	59.65	50.51	66.84	72.96	62.49	65.18
DEV [15]	69.36	54.94	73.95	76.06	68.58	66.88	49.29	72.40	70.46	64.76	66.67
RV [16]	68.98	52.74	72.83	77.14	67.92	57.79	40.87	63.87	70.83	58.34	63.13
Entropy [17]	**71.75**	55.62	76.36	76.59	70.08	60.08	46.51	53.16	62.47	55.56	62.82
InfoMax [18]	63.67	51.74	69.64	73.62	64.67	60.08	51.40	60.20	66.67	59.59	62.13
SND [19]	**71.75**	51.74	76.36	78.36	69.55	67.80	50.71	59.46	67.13	61.27	65.41
Corr-C [20]	71.23	55.70	**76.94**	**79.13**	**70.75**	61.34	45.65	54.90	62.25	56.04	63.40
EnsV	70.98	**56.12**	75.67	78.48	70.31	**68.54**	**55.60**	**69.86**	**78.23**	**68.06**	**69.19**
Worst	62.48	49.91	68.50	73.62	63.63	56.29	39.76	50.49	59.31	51.46	57.55
Best	73.37	58.09	77.35	79.33	72.03	69.33	55.86	74.55	79.59	69.83	70.93

model selection 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 model selection methods.

PDA For partial-set UDA with label shift of missing source-domain classes, we conduct hyper-parameter selections for two different UDA methods on *Office-Home* (Table 5). 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.

4.3 Comparison of Target-specific Model Selection Methods

Table 6: Validation accuracy (%) of CDA on *DomainNet-126 (DNet)*.

Method	CDAN [6]					BNM [8]					ATDOC [35]					*DNet*
	→ C	→ P	→ R	→ S	avg	→ C	→ P	→ R	→ S	avg	→ C	→ P	→ R	→ S	avg	AVG
Entropy [17]	**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	64.55
InfoMax [18]	**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	65.44
SND [19]	**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	62.12
Corr-C [20]	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	61.90
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**	**67.73**
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	59.43
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	68.29

Recent advancements in UDA model selection [19, 18] indicate that validation using only unlabeled target data can achieve 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 extra practical UDA settings: OPDA and SFUDA.

CDA We compare all target-specific validation methods on the large-scale benchmark *DomainNet-126* (Table 6). EnsV consistently keeps the leading validation performance, while other approaches exhibit high variance.

Table 7: H-score [67, 68] (%) of an OPDA 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 [17]	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 [18]	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 [19]	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 [20]	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	**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

OPDA In open-partial-set UDA with label shift of unknown classes, we choose a representative method DANCE for validation on *Office-Home* (Table 7) and measure the H-score [68, 67]. Previous validation works have not studied this challenging setting [19], and all of them encounter issues with poor model selections. In contrast, EnsV consistently achieves high-accuracy selections.

Table 8: Validation accuracy (%) of SFUDA on *Office-Home*, *Office-31*, and *VisDA*.

Method	SHOT [12]					SHOT [12]				DINE [22]
	→Ar	→Cl	→Pr	→Re	avg	→A	→D	→W	avg	T→V
Entropy [17]	63.38	50.45	77.35	77.65	67.21	71.67	90.76	88.68	83.70	71.99
InfoMax [18]	63.38	50.45	77.35	77.65	67.21	71.67	90.76	88.68	83.70	71.99
SND [19]	64.58	54.17	78.23	77.65	68.66	71.67	90.76	88.68	83.70	**74.43**
Corr-C [20]	69.13	56.32	79.29	79.14	70.97	71.58	90.76	90.19	84.18	71.99
EnsV	**69.58**	**56.78**	**80.40**	**80.76**	**71.88**	**74.85**	**94.78**	**91.82**	**87.15**	**74.43**
Worst	63.38	50.45	77.35	77.65	67.21	71.56	90.76	88.68	83.67	71.99
Best	69.83	57.08	80.55	80.76	72.05	75.06	94.78	93.33	87.72	76.17

SFUDA In source-free UDA, where source-based model selection methods are not applicable due to no access to source data, we select SHOT for the white-box setting on *Office-31* and DINE for the black-box setting on *VisDA* (Table 8). EnsV consistently maintains near-best selections, while other target-based approaches frequently make worst-case selections.

Table 9: CDA accuracy (%) on *Office-Home* when two hyperparameters are validated.

Method	MDD [33]					MCC [36]					
	Ar → Cl	Cl → Pr	Pr → Re	Re → Ar	avg	Ar → Cl	Cl → Pr	Pr → Re	Re → Ar	avg	AVG
SourceRisk	55.99	73.15	78.77	69.39	69.33	57.91	76.84	81.13	72.89	72.19	70.76
IWCV [14]	37.89	72.92	80.42	58.43	62.42	46.09	77.74	80.68	74.45	69.74	66.08
DEV [15]	52.60	72.11	53.36	67.70	61.44	59.47	76.84	81.94	74.08	73.08	67.26
RV [16]	**57.59**	72.25	80.83	70.79	70.37	59.13	76.84	82.03	71.98	72.50	71.44
Entropy [17]	57.21	**73.19**	80.06	**72.31**	70.69	59.75	77.77	82.37	74.33	73.56	72.13
InfoMax [18]	**57.59**	72.92	80.06	**72.31**	**70.72**	59.70	**78.73**	**82.58**	70.33	72.84	71.78
SND [19]	38.10	56.45	70.03	65.10	57.42	53.49	74.97	77.25	74.12	69.96	63.69
Corr-C [20]	30.17	44.74	57.15	50.76	45.71	44.90	56.75	74.32	67.61	60.90	53.31
EnsV-P	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

4.4 Further Comparisons

Validation with two hyperparameters We conduct two-hyperparameters model selection experiments with a large pool of model candidates, i.e., 28 models for image classification (Table 9) and 48 models for image segmentation (Table 10). EnsV consistently achieves near-optimal selections in both scenarios, surpassing other versatile validation methods such as Entropy and SND.

Table 10: Segmentation mIoU (%) of AdaptSeg and AdvEnt on *GTAV* → *Cityscapes* when two hyperparameters are validated.

Method	AdaptSeg [25]	AdvEnt [26]
SourceRisk [9]	39.52	39.08
Entropy [17]	39.47	38.41
SND [19]	**40.69**	40.02
EnsV	**40.69**	**40.67**
Worst	35.32	34.22
Best	42.20	41.78

Table 11: CDA accuracy (%) of BNM with ViT as the backbone.

Method	BNM [8]
Entropy [17]	28.21
InfoMax [18]	28.21
SND [19]	52.42
Corr-C [20]	28.21
EnsV	**55.16**
Worst	28.21
Best	55.16

Robustness to architectures In our experiments, we evaluate the robustness of EnsV across various ResNet backbone variants, observing consistent success across different scales. We also conduct validation experiments using the ViT-B architecture [69] on the R→S task with BNM. The validation results, presented in Table 11, demonstrate that EnsV achieves the best selection. However, all other target-based methods except SND make the worst selection.

5 Conclusion

Following a thorough empirical comparison of existing UDA model selection approaches, several key conclusions emerge: *i)* The significance of model selection in influencing the deployment performance of UDA methods becomes evident. Relying on fixed hyperparameters or limited analyses is inadequate. We emphasize the importance of increased attention and transparent reporting of validation methods, consistent with recommendations in [15, 19, 18]. *ii)* Among existing validation methods, we recommend the reverse validation (RV) approach, which, despite being overlooked in previous studies [15, 19, 18], proves to be the most reliable method for widely studied closed-set UDA scenarios when source data is available. However, it requires additional model re-training, making it less lightweight compared to target-based validation methods. Moreover, all existing model selection methods demonstrate unreliability across diverse UDA methodologies and real-world settings such as open-set and source-free UDA. These methods struggle to maintain effectiveness, posing a significant risk to the successful application of UDA in various scenarios. *iii)* Regarding our proposed baseline, EnsV, we believe it is a simple and versatile model selection method that is certified to avoid worst-case selections. While it may not always achieve peak performance, especially when the ensemble result is suboptimal, EnsV offers valuable insights for future explorations in reliable model selection methods.

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

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

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

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

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

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

[19] 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.

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

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

[30] 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.

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

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

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

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

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

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

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

[38] 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.

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

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

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

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

[43] 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.

[44] 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.

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

[46] 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.

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

[48] 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.

[49] 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.

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

[51] 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.

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

[53] 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.

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

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

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

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

[58] 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.

[59] 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.

[60] Dinu, M.-C., M. Holzleitner, M. Beck, et al. Addressing parameter choice issues in unsupervised domain adaptation by aggregation. In *International Conference on Learning Representations*. 2023.

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

[62] 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.

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

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

[65] 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.

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

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

[68] 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.

[69] 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.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]