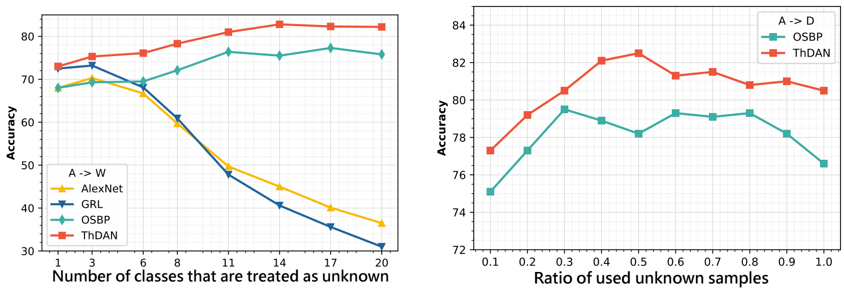
Reply to Reviewer#2

I. In the main idea, it is not clear the sampling proportion between the data from known and unknown classes. How does this proportion affect the subsequent training?

Response:

Thanks for your comments. Without label information in the target domain, the sampling proportion between the data from known and unknown classes is not given in practice. During training, the proposed ThDAN is set to randomly sample data from the known and unknown classes. Nevertheless, ThDAN can be trained under different sampling proportions. To evaluate how the sampling proportion between the data from known and unknown classes could affect the performance, results of different number of unknown training samples are reported in Section *5.3.4*.



(a) Accury *w.r.t.* number of classes that are treated as the unknown

(b) Accury *w.r.t.* ratio of used unknown samples

Fig.R2. 1: (**a**): The accuracy when we change the ratio of unknown samples in the adaptation task $A \rightarrow D$. (**b**): The accuracy when we change the number of classes that are treated as the unknown in the adaptation task $A \rightarrow W$.

In Fig.R2.1a, we use the first 10 classes (31 classes in total) as the known classes. By increasing the number of classes that are treated as the unknown, the sampling proportion for data of the unknown class increases. It is worth noting that the methods for close set domain adaptation will incur severe performance degradations. This is because these methods cannot alleviate the negative transfer brought by the unknown class. On the contrary, the methods proposed for open set domain adaptation will gain performance improvement by correctly rejecting samples from the unknown class. However, threating too many classes as the unknown will deteriorate the knowledge transfer for the classification task. Therefore the accuracy tends to decay at the end.

In Fig.R2.1b, we use the same 10 classes as the known classes and the rest as the unknown class. To change the sampling proportion between the data from the known and unknown classes, we vary the ratio of used unknown samples. Specifically, **ratio 0.1** of used unknown samples means that for every class that is treated as the unknown, only one-tenth samples are used to train ThDAN. So lowing the ratio of used unknown samples will decrease the sampling proportion for data of the unknown class. In this case, the models can hardly identify samples from the unknown class due to the lack of training data. On the other hand, when this ratio is high, the negative effect of the unknown class will substantially increase. This leads to performance degeneration of categorizing the known classes.

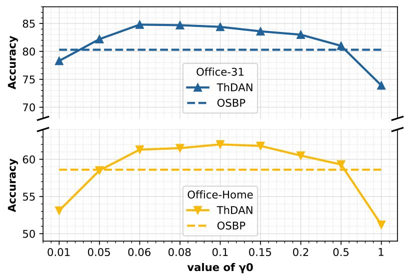
The experiments above show that the sampling proportion between the data from known and unknown classes can substantially affect model performance for both the known and unknown classes. Sampling too much data from the unknown class will degrade the performance of categorizing known classes. On the other hand, sampling too little data from the unknown class will affect the performance of rejecting "unknown".

II. From their reported results in Fig.7c, the adjust threshold of γ_0 is not sensitive to the accuracy of training, it is a bit strange. Authors are encouraged to carefully check their claims by using more detailed reasons or more diverse data sets.

Response:

Thanks for your suggestions. We have checked that the results reported in Fig.7c are correct. In the revised manuscript, we give more detailed analysis to illustrate that the adjust threshold of γ_0 is not sensitive to the accuracy of training.

Fig.7c of the original manuscript is presented as follows,



(a) Accury *w.r.t.* value of γ_0

Fig.R2. 2: The prediction accuracy of our method when we change the value of γ_0 on dataset Office-31 and Office-Home.

Fig.R2.2a illustrates the average accuracy of **6 tasks on Office-31** and **12 tasks on Office-Home** when γ_0 changes. Compared with the baseline model OSBP, the proposed ThDAN can deliver considerable results on both Office-31 and Office-Home datasets in a wide range of γ_0.

To select transferable target samples for domain adversarial training, a transferability threshold is adaptively computed during training,

$$\beta(X_s, \gamma) = \mathbb{E}_{x \in X_s} w(x) - \gamma. \tag{R2.1}$$

The threshold β is calculated by averaging the transferability score of a batch of source samples and the predefined γ. In the training phase, we dynamically increase γ from 0 to γ_0 so that more samples of known classes can be selected for domain adversarial training. The changing function of γ is as follows,

$$\gamma(n) = \begin{cases} 0 & , n \in N_1 \\ \gamma_0 \times \sigma(n) & , n \in N_2 \end{cases}. \tag{R2.2}$$

Here σ is a *monotonically increasing function* with an upper bound of 1.

It is worth noting that in the process of training, the learning rate is gradually reduced for better convergence. As a result, the transferable samples that are selected in the early stage would "dominate" the model training. These samples engage in training sooner, so the gradient they produce can update the model with a larger learning rate. As γ approaches γ_0, the learning rate is relatively low, and the previously selected samples are likely to be selected again. Therefore the newcomers have little impact on the model performance. This directly reflects the insensitivity of value of γ_0 to classification accuracy.

The above explanation has been added in the revised manuscript. Please refer to Section *5.3.6* in the revised manuscript for details.

1. **The abstract needs to be redesigned. Open set domain adaptation (OSDA) should be a scenario. Authors claim to solve the OSDA problem. What is the problem?? This is unclear. It seems the author should firstly explain the OSDA setting and then present which issue they want to address. The first sentence of the abstract is a bit far away from their focus. More seriously, the introduction don not connect the abstract. For example, the beginning of the abstract explains the unsupervised domain adaptation, but the beginning of the introduction suddenly use deep learning to start this paper without unsupervised domain adaptation. Authors are encouraged to carefully reorganize their work, where the introduction must connect the abstract tightly. One sentence in the abstract should connect one logic in the introduction section.**

Response:

Thanks for your comments, the suggestions are of great help to improve our work. In the revised manuscript, we redesigned the **abstract** and **introduction** for tighter logical connections.

The **abstract** has being redesigned as follows,

> *In recent years, many unsupervised domain adaptation (UDA) methods have been proposed to tackle the domain shift problem. Most existing UDA methods are derived for Close Set Domain Adaptation (CSDA) in which source and target domains are assumed to share the same label space. However, target domain may contain unknown class different from the known ones in the source domain in practice, i.e., Open Set Domain Adaptation (OSDA). Due to the presence of unknown class, aligning the whole distribution of the source and target domain for OSDA as in the previous methods will lead to negative transfer. Existing methods developed for OSDA attempt to assign smaller weights to target samples of unknown class. Despite promising performance achieved by existing methods, the samples of the unknown class are still used for distribution alignment, which make the model suffer from the risk of negative transfer. Instead of reweighting, this paper presents a novel method namely Thresholded Domain Adversarial Network (ThDAN), which progressively selects transferable target samples for distribution alignment. Based on the fact that samples from the known classes must be more transferable than target samples of the unknown one, we derive a criterion to quantify the transferability by constructing classifiers to categorize known classes and to discriminate unknown class. In ThDAN, an adaptive threshold is calculated by averaging transferability scores of source domain samples to select target samples for training. The threshold is tweaked progressively during the training process so that more and more target samples from the known classes can be correctly selected for adversarial training. Extensive experiments show that the proposed method outperforms state-of-the-art domain adaptation and open set recognition approaches on benchmarks.*

According to the redesigned abstract, we have modified the **introduction** to make it connect to the abstract tightly. The following Table 1 shows the logical connection between each paragraph of the introduction and each sentence of the abstract.

Table 1: The logical connection between paragraphs of the introduction and sentences of the abstract.

Paragraph of Introduction	**Sentence of Abstract**
# 1	*In recent years, many unsupervised domain adaptation (UDA) methods have been proposed to tackle the domain shift problem.*
# 2	*Most existing UDA methods are derived for Close Set Domain Adaptation (CSDA) in which source and target domains are assumed to share the same label space. However, target domain may contain unknown class different from the known ones in the source domain in practice, i.e., Open Set Domain Adaptation (OSDA).*
# 3	*Existing methods developed for OSDA attempt to assign smaller weights to target samples of unknown class. Despite promising performance achieved by existing methods, the samples of the unknown class are still used for training, which make the model suffer from the risk of negative transfer.*
# 4	*Instead of reweighting, this paper presents a novel method namely Thresholded Domain Adversarial Network (ThDAN), which progressively selects transferable target samples for distribution alignment.* ......

2. What is “transferable”? How to evaluate it? Any definition to support this term?

Response:

Unfortunately, there is no formal definition or mathematical formula for “transferable” in domain adaptation. Generally, the word “transferable” is used to describe features. Here we quote from [1]:

> *the transferable features are the features that generalize well to novel tasks for domain adaption*

While the later work [2] uses “transferable” to describe samples that contribute to the transfer task of domain adaptation.

In the scenario of *open set domain adaptation*, we consider the target samples from the known classes are transferable samples since they can boost the classification performance of the model. On the contrary, the target samples from the unknown class are considered as untransferable samples, this is because aligning distribution with them will incur negative transfer.

The proposed method evaluates transferability from two perspectives:

Target samples from the known classes are able to confuse G_d. Here G_d is the domain discriminator that gives the probability of being target domain. For a target sample, if $G_d(z)$ approaches to 1, then the sample has a high probability of coming from the unknown class. That is because the unknown class is only included in the target domain and can be almost perfectly discriminated from the source samples. On the other hand, if $G_d(z)$ approaches to 0, then the sample is more likely from the known classes that shared by domains. So the transferable target samples are able to confuse G_d to label them as the source samples. Therefore we can define the transferability $w_d(x)$ as inversely related to $G_d(z)$,

$$w_d(x) = 1 - G_d(z). \tag{R2.3}$$

Target samples from the known classes can be categorized by $G_{c,known}$. Here $G_{c,known}$ is the classifier that gives the probability distribution of known classes. Due to the overlapping in the marginal distributions across domains, target samples from the known classes could be categorized correctly by the classifier $G_{c,known}$ that trained on the source samples. This leads to low entropy (high transferability) for the samples from the known classes. For the samples from the unknown class, because they cannot be classified into one of the K known classes, the uncertainty of the prediction measured by the entropy is large. Therefore the transferability $w_c(x)$ can be defined as inversely related to the *normalized* classification entropy H,

$$w_c(x) = 1 - H(G_{c,\,known}(z)). \tag{R2.4}$$

Since Eq.(R2.3) and Eq.(R2.4) measure the transferability from two independent perspectives, we can unify the transferability criterions as,

$$w(x) = 1 - G_d(z) \cdot H(G_{c,\,known}(z)). \tag{R2.5}$$

The experiments in Section *4* show that the transferability criterion of Eq.(R2.5) works well for the proposed model to select target samples from the known classes for domain adversarial training.

3. In Fig.1, it is unclear why (d) must outperform the other methods.

Response:

Thanks for your comments. As a matter of fact, Fig.1 is used to illustrate the general idea of the proposed method. Fig.1 of the original manuscript is presented as follows,

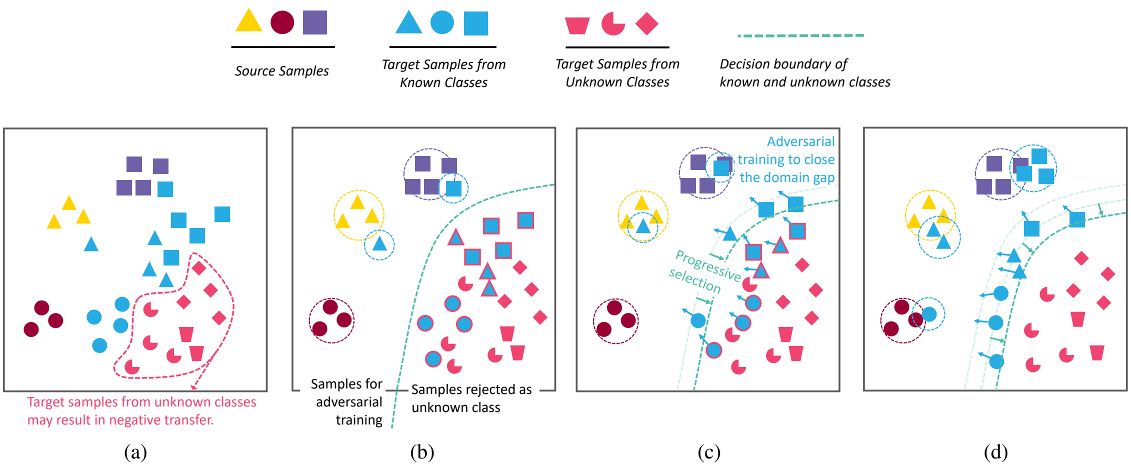


Fig.R2. 3

To make a clearer presentation for Fig.R2.3, we modify the caption in the revised manuscript as follows,

> *The general idea of the proposed Thresholded Domain Adversarial Network (**ThDAN**). (**a**): Training samples for open set domain adaptation. (**b**): ThDAN builds a decision boundary of the transferability threshold to separate transferable and untransferable samples. (**c**): To bridge the domain gap, the transferable target samples are selected for domain adversarial training. Then ThDAN tweaks the transferability threshold to collect more target samples of known classes to enhance knowledge transfer. (**d**): More transferable target samples are selected for domain adversarial training. All the unselected target samples are rejected as "unknown"*

4. What is "domain-invariant features" ?

Response:

In domain-invariant feature space, the source and target domains have the same (or similar) marginal distributions, and the posterior distributions of the labels are the same across domains too [3]. Hence, a classifier trained on the labeled source domain is likely to perform well on the target domain.

Specifically, the domain-invariant features can be obtained by *domain adversarial training* [4, 5, 6]. The domain adversarial training is a minmax game: a domain discriminator is trained to separate the feature representation of the source domain from the one of the target domain, at the same time, feature generators are trained to deceive the domain discriminator. In this work, we adopt the training scheme proposed in [6] to enable the domain discriminator to identify samples of unknown class for open set domain adaptation. Formally, the training procedure can be written as,

$$\begin{aligned}\min_{G_f^t}\max_{G_d} \mathcal{L}(G_d, G_f^s, G_f^t) = &\ \mathbb{E}_{x\sim p_t(x)}\left[\log\left(G_d\left(G_f^t(x)\right)\right)\right] \\ &+ \mathbb{E}_{x\sim p_s(x)}\left[\log\left(1 - G_d\left(G_f^s(x)\right)\right)\right].\end{aligned} \tag{R2.6}$$

G_f^s and G_f^t are the feature extractors for source and target samples, which share weights as in [6]. G_d is a binary domain discriminator with all the source samples labelled as 0 and all the target samples labelled as 1. By optimizing Eq.(R2.6) with fixed G_d, G_f^t is confined to generate features that can not be discriminated by G_d (*i.e.* domain-invariant features). Therefore the classifier G_c that trained on the source domain can perform well for target samples.

5. In the related work, CSDA does not have references.

Response:

Thanks for your suggestion. CSDA is referred to Close Set Domain Adaptation and we have add references [7, 8, 9] for the methods for CSDA in the revised manuscript.

6. In section 2.1, ”Another way to solve domain adaptation” is not suitable. Domain adaptation is a setting in detailed ML tasks or a probability distribution issue. Authors can obtain more information from S Ben-David’s paper.

Response:

Thanks for your comment. We agree that the phrase “Another way to solve domain adaptation” is not precise. This phrase is revised as “Another approach for domain adaptation”. Similar unprecise wordings have been modified in the revised manuscript. Also, we have carefully read Ben-David’s paper [7] to get a better understanding of domain adaptation and has cited this paper in the revised manuscript.

7. Section 2.2 missed some new work from open-set domain adaptation.

Response:

Thanks for your suggestions. We add some new work [10, 11] of open set domain adaptation in Section 2.2, they are respectively from ICML2019 and CVPR2019.

Furthermore, we implement *Factorized Representations For Open Set Domain Adaptation* ***(FRFOSDA)*** model proposed in [10] in experiments on **Office-Hone** and **Office31** for comparisons. Please refer to Section 4 for more details.

8. In Section 3, authors missed the definition on C_t. Is it the label space of target domain?

Response:

Yes, C_t is the label space of target domain. In the revised manuscript, the definition on C_t has been added in section *3.1 open set domain adaptation* as follows,

> C_s and C_t denote the label space of the source and target domain, respectively. In the setting of open set domain adaptation, the label space of target domain contains the label space of source domain, *i.e.*, $C_s \subset C_t$. We refer to classes from C_s as the known classes and classes from $C_t \backslash C_s$ as the unknown class.

9. In Section 3.1, what is “untransferable ones”?

Response:

This question is related to Question 2. In the setting of open set domain adaptation, the untransferable ones refer to the samples of unknown class in the target domain. The proposed method selects untransferable samples by evaluating transferability as follows,

$$X_t^u = \{x | w(x) < \beta, x \in X_t\}. \qquad \text{(R2.7)}$$

Here w is transferability calculator of Eq.(R2.5). To select untransferable samples from a batch of target data X_t, a transferability threshold β is computed by Eq.(R2.1). Then samples from X_t whose transferability score $w(x)$ less than the threshold β will be selected as untransferable samples, denoted as X_t^u.

10. In Eq.(2), more strong reasons need to be presented to explain the G_f and G_d. Why the use such settings to define them? Only based other's work?

Response:

The reason why we use the setting "G_f and G_d can be considered as the generative network and discriminate network of GAN respectively" to define them in Eq.(2) is that the domain adversarial training between G_f and G_d is similar to the training procedure of the original *Generative Adversarial Nets (GAN)* [12].

In GAN, the *discriminate network* D and the *generative network* G are trained with the following objective,

$$\begin{aligned}\min_G \max_D V(D,G) = & \ \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ & + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))].\end{aligned} \tag{R2.8}$$

In the learning framework of GAN, D is trained to separate the feature representation of the true images and fake images, and G is trained to generate fake images that can confuse D. While in the domain adversarial training of Eq.(R2.6), G_d is trained to separate the feature representation of the source domain from the target domain, and G_f (G_f^t) is trained to generated target features that are able to confuse G_d. The training procedure for D and G is very similar to the domain adversarial training for G_d and G_f. Therefore we can define G_f and G_d by the setting of GAN.

To better explain why Eq.(2) gives the optimal G_d, we have removed this phrase and added more formal proof in the revised manuscript. Please refer to the reply of the next question for details.

11. In the following, why does G_d converge to the optimal?

Response:

Eq.(2) in the revised manuscript is as follows,

$$G_d^*(z) = \frac{p_t(z)}{p_s(z) + p_t(z)}. \tag{R2.9}$$

Here z is the deep feature representation of a sample, *i.e.,* $z = G_f(x)$. We give the proof that Eq.(R2.9) is the optimal G_d for Eq.(R2.6) as follows,

Proof. For any G_f^s and G_f^t, we train G_d to maximize Eq.(R2.6):

$$\begin{aligned}\max_{G_d} \mathcal{L}(G_d, G_f^s, G_f^t) &= \int_x p_t(x) \log\left(G_d\left(G_f^t(x)\right)\right) + p_s(x)\log\left(1 - G_d\left(G_f^s(x)\right)\right) dx \\ &= \int_z p_t(z)\log\left(G_d(z)\right) + p_s(z)\log\left(1 - G_d(z)\right) dz.\end{aligned} \tag{R2.10}$$

We take the partial differential of the objective Eq.(R2.10) with respect to G_d, and apply the Leibnizs rule to exchange the order of differentiation and integration to achieve optimal G_d in [0, 1] at Eq.(R2.9). □

The proof has been added in Section 3.2 of the revised manuscript.

12. In Eq.(2), how to define p_s and p_t?

Response:

Eq.(2) in the revised manuscript is as follows,

$$G_d^*(z) = \frac{p_t(z)}{p_s(z) + p_t(z)}. \tag{R2.11}$$

Here z is the deep feature representation of a sample, *i.e.,* $z = G_f(x)$. As stated in *Section 3.1 open set domain adaptation* of the revised manuscript, $p_s(z)$ and $p_t(z)$ are probability distributions of feature representations in source and target domains respectively, as defined in [13].

13. In Eq.(9), what is X_t^k?

Response:

Thanks for your comments. X_t^k and X_t^k respectively denote the transferable and untransferable samples split from a batch of target samples X_t. Eq.(9) of the original manuscript indicates the process,

$$\begin{aligned} X_t^k &= \{x|w(x) \geq \beta, x \in X_t\}, \\ X_t^u &= \{x|w(x) < \beta, x \in X_t\}. \end{aligned} \tag{R2.12}$$

Here w is the transferability calculator defined in Eq.(R2.5). A transferability threshold β is computed by Eq.(R2.1) to split a batch of target training data X_t into two parts, *i.e.,* X_t^k and X_t^u. Specifically, X_t^k denotes the transferable target samples, which will be selected for domain adversarial training. While X_t^u denotes the untransferable target samples, which the model learns to identify as "unknown".

We add Section *4.1* in the revised manuscript to clarify the proposed methods and equations, in which X_t^k and X_t^u are explained.

14. A same controversial issue appears again: Sample selection is the key step for this model to address open set domain adaptation.

Response:

Thanks for pointing out the mistakes. This sentence has been modified as "The sample selection algorithm will substantially affect the performance of ThDAN for open set domain adaptation"

Similar mistakes have bee corrected in the revised manuscript.

15. In the experiments, why do you use different setting to begin the experiments in Section 4.2.1? Authors are encouraged to explain the inherent reasons.

Response:

Thanks for your suggestions. To evaluate the efficacy of the threshold tweaking techniques proposed for ThDAN, ablation studies on Office-31, OfficeHome and VisDA datasets are reported in Section *4.2.1* and *4.2.2*. The ablation studies involve 3 kinds of ThDAN variants: **(1)** ***ThDAN-m-dy*** is the original setting that calculates the transferability threshold based on mini-batch samples and a fixed γ. (**2**) ***ThDAN-dy*** is the variant that applies exponential moving average to update the transferability threshold. (**3**) ***ThDAN*** further takes advantage of dynamic γ to tweak the transferability threshold.

In order to better understand how the proposed method performs under ablation settings, we add Section *5.3.1 Ablation Study* in the revised manuscript. Please refer to the revised manuscript for details.

16. A same issue appear again in the settings of Section 4.2.2.

Response:

Thanks for your reminding. The ablation analysis in the original Section 4.2.2 has been reorganized in Section *5.3.1 Ablation Study* of the revised manuscript.

17. The results analysis in Section 4.2.3 is unclear. More detailed reasons need to be present to support your conclusion.

Response:

The results of Section 4.2.3 are presented as follows,

Table 2: Accuracy (%) of each method with 6 shared class on VisDA.

Method	class-wise accuracy on VisDA								
	Bcycle	Bus	Car	Mcycle	Train	Truck	Unkown	*Avg.All*	*Avg.Known*
OSVM [ECCV, 2014]	4.8	45.0	44.2	43.5	59.0	10.5	57.4	*37.8*	*34.5*
MMD [NIPS, 2007]	0.2	30.9	49.1	54.8	56.1	8.1	61.3	*37.2*	*33.2*
GRL [JMLR, 2016]	9.1	50.5	53.9	79.8	69.0	8.1	42.5	*44.7*	*45.1*
OSBP [ECCV, 2018]	48.0	67.4	39.2	80.2	69.4	**24.9**	80.3	*58.5*	*54.8*
ThDAN-dy	52.6	69.5	58.8	83.1	72.3	12.3	84.5	*61.8*	*58.1*
ThDAN	**55.2**	**70.8**	**61.4**	**85.6**	**74.8**	10.0	**88.8**	***63.8***	***59.6***

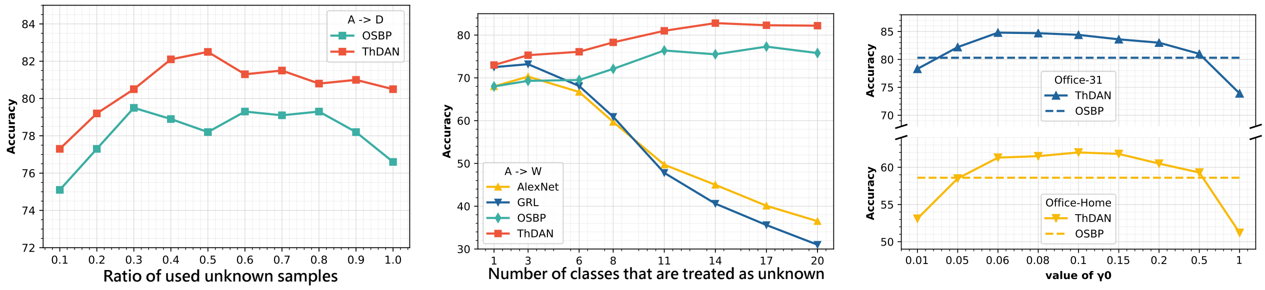
In the experiments on VisDA, the *training* split is used as the source domain and the *validation* one as the target domain. We choose 6 categories, i.e., bicycle, bus, car, motorcycle, train and truck as the known classes, and the other 6 categories as the unknown class. The classification results are shown in Table 2, where *Avg.All* and *Avg.Known* indicate the accuracy averaged over all classes and the known classes, respectively. For the classification for known classes, the proposed ThDAN can exceed other methods in almost every class and in the average, which means our method can effetely transfer knowledge for the adaptation task. This is because ThDAN avoids the negative transfer by aligning distributions with target samples of high transferability. Also, ThDAN improves the discrimination accuracy for the unknown class by a big margin. This further verifies the efficiency of the sample selection algorithm, since ThDAN is learnt to reject the unselected samples as "unknown".

The analysis mentioned above has been added to the corresponding section in the revised manuscript.

18. Please carefully check the claim in Fig.7.

Response:

Thanks for your reminding. Fig.7 of the original manuscript is included as follows,



(a) Accury *w.r.t.* ratio of used unknown samples (b) Accury *w.r.t.* number of classes that are treated as the unknown (c) Accury *w.r.t.* value of γ_0

Fig.R2. 4

In the revised manuscript, the claims of Fig.R2.4 are modified for clearer representation. The rewritten claims have been stated in the previous replies:

- In the reply of **Question I**, we modify the claims of Fig.R2.4a and Fig.R2.4b by adding more analysis to explain how the sampling proportion between the data from known and unknown classes could affect the performance.
- In the reply of **Question II**, we modify the claims of Fig.R2.4c by providing a more detailed analysis on why the proposed method is insensitive to the setting of γ_0.

19. Some equations missed "," or "." at their ends.

Response:

Thanks for your reminding. The equations in the revised manuscript have been corrected.

References

[1] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, arXiv preprint arXiv:1502.02791 (2015).
[2] X. Wang, L. Li, W. Ye, M. Long, J. Wang, Transferable attention for domain adaptation, in: Thirty-Third AAAI Conference on Artificial Intelligence, 2019.
[3] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, arXiv preprint arXiv:1412.3474 (2014).
[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, The Journal of Machine Learning Research 17 (1) (2016) 2096–2030.
[5] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.
[6] K. Saito, S. Yamamoto, Y. Ushiku, T. Harada, Open set domain adaptation by backpropagation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 153–168.
[7] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine learning 79 (1-2) (2010) 151–175.
[8] M. Wang, W. Deng, Deep visual domain adaptation: A survey, Neurocomputing 312 (2018) 135–153.

[9] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.
[10] M. Baktashmotlagh, M. Faraki, T. Drummond, M. Salzmann, Learning factorized representations for open-set domain adaptation, in: International Conference on Learning Representations, 2019.
[11] H. Liu, Z. Cao, M. Long, J. Wang, Q. Yang, Separate to adapt: Open set domain adaptation via progressive separation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2927–2936.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[13] J. Zhang, Z. Ding, W. Li, P. Ogunbona, Importance weighted adversarial nets for partial domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8156–8164.