MULTI-DOMAIN ACTIVE LEARNING: A COMPARATIVE STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-domain learning (MDL) refers to learning a set of models simultaneously, with each one specialized to perform a task in a certain domain. Generally, high labeling effort is required in MDL, as data need to be labeled by human experts for every domain. Active learning (AL), which reduces labeling effort by only using the most informative data, can be utilized to address the above issue. The resultant paradigm is termed multi-domain active learning (MDAL). However, currently little research has been done in MDAL, not to mention any off-the-shelf solution. To fill this gap, we present a comprehensive comparative study of 20 different MDAL algorithms, which are established by combining five representative MDL models under different information-sharing schemes and four well-used AL strategies belonging to different categories. We evaluate the algorithms on five datasets, involving textual and visual classification tasks. We find that the models which capture both domain-dependent and domain-specific information are more likely to perform well in the whole AL loops. Besides, the simplest informative-based uncertainty strategy surprisingly performs well on most datasets. As our off-the-shelf recommendation, the combination of Multinomial Adversarial Networks (MAN) with the best vs second best (BvSB) uncertainty strategy shows its superiority in most cases, and this combination is also robust across datasets and domains.

1 INTRODUCTION

Building classifiers on the data which are collected from different domains is common. These domains usually refer to different datasets which have different distributions. For example, in a sentiment classification task for product reviews, the goal is to predict the polarity of reviews from different categories of products. These categories are considered as domains. The conventional approach is to independently build models on each domain to guarantee the performance, or jointly build one single model on the data from all the domains. However, independent training requires sufficient labeled data on each domain and neglects the correlation among domains, and the joint training eliminates the unique information from each domain. When the number of domains is large, such limitation could be more serious. Under this circumstance, multi-domain learning (**MDL**)(Dredze & Crammer, 2008) attracts much attention. MDL is proposed to capture both the domain-invariant information and the domain-specific information to overcome the mentioned limitations.

However, MDL still requires a well-annotated training set which contains the instances from all the domains. In the real life, obtaining such a training set usually costs a lot. Active learning **AL** (Settles, 2010) is developed to reduce the annotating cost for decades by selecting the most informative instances. So it is natural to use AL to reduce the annotating cost in MDL. However, most of the current researches on AL are designed for single domain learning setting, and it is unclear how to efficiently apply AL on MDL setting.

In this paper, utilizing AL on MDL setting is referred to as **MDAL**. Different from single-domain active learning, if the active strategy only selects the instances which most benefit to one specific domain, the model would fail to use the information shared across domains and waste the budget. The goal of MDAL is to actively select the most helpful instances from all the domains and train models under the MDL setting with as few labeled instances as possible.

Despite the practical importance of MDAL, there only exists few works in the literature. In particular, these MDAL works are based on the specific models for specific applications, e.g. Support Vector Machines (SVMs) for sentiment classification (Li et al., 2012), Rating-Matrix Generative Model (RMGM) for collaborate filtering (Zhang et al., 2016). The AL strategies designed by these works cannot be used on other models, such as the more recent, state-of-the-art Deep Neural Networks (DNNs). So there is a research gap for MDAL. In this paper, our goal is to provide an off-the-shelf solution for MDAL. Thus, a comparative study is necessary.

The major contributions of this work could be summarized as follows:

- We provided a formal definition of MDAL. Besides, the relative works are thoroughly reviewed in the way how they share information among domains in the model level.
- As far as we know, this is the first comparative work for MDAL setting. According to the experiment results, we provided an off-the-shelf solution for MDAL. As our recommendation, MAN model with the BvSB uncertainty strategy shows its superiority in most cases, and it is robust over different datasets and domains.

2 Problem Description

In this section, we will give a formal definition of the MDAL setting.

Definition 1 (Multi–domain learning) *Given K different data sources (domains) $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$, a set of data pools $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$ which contains both labeled and unlabeled data could be collected from each source in advance. The labeled data from each pool constitute a labeled data set $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K\}$. MDL is to find a set of hypothesis $H = \{h_1, h_2, \ldots, h_K\}$ for K domains from the hypothesis space $\mathcal{H}$ by utilizing the common knowledge of different domains, which could be expressed as follows:*

$$\min_{H} \mathrm{Loss}_{\mathrm{sup}}(H; \mathcal{L}) + \Omega(H; \mathcal{P}) \tag{1}$$

$\mathrm{Loss}_{\mathrm{sup}}(H; \mathcal{L})$ represents the supervised loss on the labeled set $\mathcal{L}$, $\Omega(H; \mathcal{P})$ represents a loss on the set of data pools $\mathcal{P}$ for capturing the common knowledge.

Definition 2 (Multi–domain active learning) *When the collected set of data pools $\mathcal{P}$ is unlabeled in the MDL setting, an oracle (e.g. human experts) could be available for annotation. MDAL is to annotate a set of labeled data $\mathcal{L}_A = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K\}$ from $\mathcal{P}$ according to a selection strategy A. A set of hypothesis $H = \{h_1, h_2, \ldots, h_K\}$ would be found with the labeled dataset $\mathcal{L}_A$ and the set of data pools $\mathcal{P}$. On the one hand, MDAL keeps the sharing of knowledge in MDL setting to improve the performance, on the other hand, it tries to maintain the labeled set $\mathcal{L}_A$ as small as possible to reduce the labeling cost.*

3 Related Works

There are two primary parts in MDAL. One is the model which shares information, and the other is the AL query strategy. In this section, both concepts are reviewed. In the literature, there are too many technologies that begin with "multi-", such as multi-task Learning, multi-label Learning, etc. To avoid confusions, the relations between MDAL and many other research fields are discovered in Appendix A.

The models which could utilize the information shared among domains are reviewed in section 3.1. It is worth noting that these models are used either in domain adaptation or MDL. The works of active learning in the settings where multiple domains exist are reviewed in section 3.2.

3.1 Models for Multiple Domains

In our taxonomy, the models are classified by how they share information among different domains. We summarize the model-level intuitions into the following 3 categories.

Sharing domain-invariant representations only The intuition is that if the discrepancy among domains are small enough, the knowledge is easier to be transferred. Thus, many works try to unify the domain distributions, then directly make predictions under the learned representations.

Matching the marginal distribution is a well-used method to reduce the difference among different domains. Plenty of works achieve this goal by designing a discrepancy loss, usually minimizing the Maximum Mean Discrepancy (MMD). Transfer Component Analysis (TCA) (Pan et al., 2011) is a classic method which applies this idea with conventional kernel based models. Deep Adaptation Networks (DAN) (Long et al., 2015), Residual Transfer Networks (RTN) (Long et al., 2016) and Joint Adaptation Networks (JAN) (Long et al., 2017) are neural network based models adopting this idea for domain adaptation setting.

Many other works utilized adversarial training to match the marginal distribution. The intuition is that if a discriminator cannot tell which domain the instances are from, the representation extraction would be effective. Domain-Adversarial Neural Network (DANN) (Ganin et al., 2016), Adversarial Discriminative Domain Adaptation (ADDA) (Tzeng et al., 2017), and Conditional Adversarial Domain Adaptation (CADA) (Long et al., 2018), add discriminators into their models for domain adaptation. Feng et al. (2019) included discriminators to learn representations in MDL.

Some other works propose to match conditional distributions. The intuition is that the instances in the same class from different domains are expected to be mapped nearby. Joint Distribution Adaptation (JDA) (Long et al., 2013), Deep Supervised Domain Adaptation (Deep SDA) (Motiian et al., 2017), Multi-Adversarial Domain Adaptation (MADA) (Pei et al., 2018), and MSTN (Xie et al., 2018) match the distributions either by the existing true labels or the created pseudo-labels on the target domain. Saito et al. (2018) tried to align the distribution of a target domain by matching the decision boundaries.

Sharing domain-invariant representations with domain-specific information The intuition is that only using the domain-invariant representation might be insufficient to contain all the original information. Under this circumstance, domain-specific information also could be introduced to guide the predictions with the domain-invariant representations.

Many works concatenated the domain-invariant representation and domain-specific representation together to make prediction. III (2007) proposed Frustratingly Easy Domain Adaptation (FEDA) for multi-domain adaptation and first applied this idea into MDL. Domain Separation Networks (DSN) (Bousmalis et al., 2016) and multinomial adversarial networks (MAN) Chen & Cardie (2018) utilized this concatenation into neural networks. Liu et al. (2019) further added the orthogonal regularization between private features across domains.

Another type of models applies domain-specific classifiers on the shared representations. Nam & Han (2016) proposed MDNet for the multi-domain visual tracking problem. Saito et al. (2017) proposed an asymmetric tri-training for unsupervised domain adaptation and the classifiers are trained on the pseudo-labels on the target domain. Xiao et al. (2016) proposed a domain guided dropout for the person re-identification problem on different domains.

Yuan et al. (2018) proposed a Domain Attention Model (DAM) for multi-domain sentiment analysis to capture the specific information from different domains. When there are more than one feature extractors (many channels), the representations (Li et al., 2019) or channels (Xiao et al., 2020) could be weighted differently to make predictions on different domains.

Sharing model parameters without share representations Few other works share parameters without explicitly sharing representations. This type of works normally mainly concerns how to learn a single network that can compactly represent all the domains with minimal number of domain-specific parameters. Rebuffi et al. (2017) and Rebuffi et al. (2018) designed networks with residual adapters to minimize the domain-specific parameters. Li & Vasconcelos (2019) used a domain-specific covariance normalization (CovNorm) layer instead. Rosenfeld & Tsotsos (2020) proposed a method called Deep Adaptation Modules (DAM) that constrains newly learned filters to be linear combinations of existing ones.

3.2 ACTIVE LEARNING FOR MULTIPLE DOMAINS

In this section, the works of AL with the presence of multiple domains are reviewed. We follow the taxonomy used in the previous section. We note that these researches in the literature didn't cover all types of the models mentioned in the previous section.

When the models share domain-invariant representations only, several AL strategies select the instances which would minimize the discrepancy to learn the domain-invariant representations. Chattopadhyay et al. (2013) tried to select instances from the target domain by reducing MMD between two different domains. Huang & Chen (2016) adopted this idea to another problem setting, where the selection is from the source domain. Deng et al. (2018) trained a multi-kernel SVM classifier with the source and target data, then they used the margin criteria to select instances. Su et al. (2020) proposed Active Adversarial Domain Adaptation (AADA), where a discriminator was trained to weight the uncertainty of the target unlabeled examples.

When the domain-specific representations are included, the predictions are made on the concatenated features for each domain. Li et al. (Li et al., 2012) proposed to use multiple SVM models on the concatenated features for each domain in a MDAL setting. Their strategy selects the instances which could most reduce the version space of the SVM models.

4 COMPARISON DESIGN

Currently, very few works have been done in MDAL. Under this circumstance, as the most convenient approach, combining the models for MDL setting and conventional AL strategies might be applicable. If so, how good and how robust would these combinations be? To discover the performance of these combinations, a comparative work should be done.

The goal of making this comparison is to find out an off-the-shelf solution for the MDAL setting. More specifically, we are curious:

1. What is the best model-strategy combination and how good it could be?
2. How robust the solution could be for different datasets?
3. How robust the solution could be on different domains?

In this section, we describe the details of our comparison for MDAL. The selected models, strategies, datasets and the implementation details are introduced.

Datasets 5 datasets are selected from the MDL and domain adaptations settings. These datasets at least contain two domains, and all the tasks are classification tasks. The details of train/test partition for each dataset could be found in the appendix B.

- **Synthetic Double and Triple Inter-Twin Moons Toy Datasets**: The double inter-twin moons toy dataset is first used in (Ganin et al., 2016). In our comparison, a variance dataset with triple inter-tween moons is also included.
- **Digits**: This digit dataset is used in (Ganin et al., 2016), which contains two domains (sub-datasets): 'MNIST' (LeCun et al., 1998) and 'MNIST-M'. MNIST-M sub-dataset is generalized in (Ganin et al., 2016) by blending digits from the MNIST set over patches randomly extracted from color photos from BSDS500 (Arbelaez et al., 2010).
- **Amazon** (as mSDA representations): In the amazon dataset, there are four domains: 'books', 'dvd', 'electronics' and 'kitchen'. The original sentence data are processed by marginalized denoising autoencoders (mSDA) (Chen et al., 2012).
- **Office-31**: This Office-31 dataset (Gong et al., 2012) contains 31 categories from three domains 'Amazon', 'Webcam' and 'DSLR'. Briefly, the DeCaf representations (Donahue et al., 2014) are used.
- **ImageCLEF-DA** [1]: ImageCLEF-DA is a well-balanced dataset in the ImageCLEF domain adaptation challenge in 2014.

[1] https://www.imageclef.org/2014/adaptation

Models Several representative models from the previous review are selected in our comparison. The neural network is used as our base model due to its strong ability for representation extraction. 3 models are selected: **DANN** (Ganin et al., 2016), **MDNet** (Nam & Han, 2016) and **MAN** (Chen & Cardie, 2018). Besides, 2 baseline models which do not handle the domain information are also included: **SDL-separate** and **SDL-joint**. For SDL-separate, multiple networks are independently trained on the corresponding domains. For SDL-joint, the differences among domains are ignored, and a single neural network was trained on the data mixed from all the domains. The sketches of models and the specific model structures for each dataset could be found in Appendix C.

Strategies Most of the strategies from the review are not appropriate to be combined with the neural network models in our comparison. They either heavily depends on the models other than neural network, or could only select instances from single target domain. So we'd rather compare the conventional single domain AL strategies, which directly evaluate instances from the models outputs. Here, 4 well-used strategies which could be easily implemented on neural networks are selected as follows:

- **Random**: Randomly select instances from each domain.
- **Uncertainty** ((Joshi et al., 2009)): Best vs Second Best (BvSB) is an uncertainty measurement which selects instances with the greatest difference in inference scores between the most likely class and the second most likely class.
- **EGL** (Settles & Craven, 2008; Zhang et al., 2017): Expected Gradient Length (EGL) strategy is designed for the models which could be optimized by gradients. The instance which would lead the longest expected gradient length to the last fully connected layer would be selected.
- **BADGE** (Ash et al., 2020): Batch Active learning by Diverse Gradient Embeddings (BADGE) calculates the gradients of the last fully connected layer lead by the pseudo-labels of the unlabeled instances. A k-means++ initialization is applied on all the calculated gradients to ensure the diversity of the batches.

Neural Network Training Adam optimizer and Cross Entropy Loss are used to train our classifiers. L2 regularization is used. The learning rate, weight decay, the batch size and the maximum number of epochs are set to be the same for each model on the same dataset to ensure the fairness of comparison. For DANN and MAN, there is a trade-off parameter for balancing the loss from classifiers and discriminators. Besides, the early stopping technique is used during the training process. The details are listed in Table 2 in Appendix D.

Active Learning Setting We still consider this MDAL setting as a pool-based scenario. The learning process would repeat until the budgets are consumed. The model would be totally re-trained in each iteration. The details are listed in Table 3 in Appendix D.

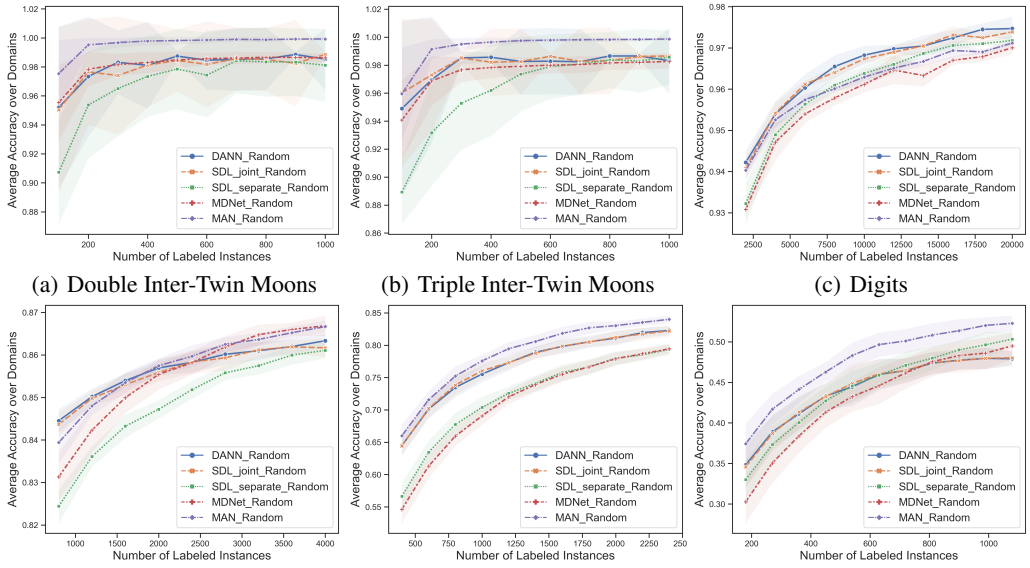
5 Comparison Results and Discussions

As we introduced in section 4, there are three research questions. To answer these questions, this comparison contains four parts. At first, the selected models are compared in section 5.1 to get a first impression. Then, in section 5.2, the strategies on each model are compared to answer the first research question. In section 5.3, we try to discover the robustness of the combinations over datasets. Finally, robustness over domains are discussed in section 5.4.

The average prediction accuracy on the test sets would be recorded for each model-strategy combination. As an AL process, the performances would be presented as learning curves. Usually, in a set of learning curves, the x-axis represents the cost which has been consumed (the number of labeled instances), and the y-axis represents the accuracy under the current cost, and each curve represents a model-strategy combination.

5.1 Comparisons over Models

In this section, the models are compared to get a first impression without using AL strategies. Specifically, the random selection was applied to check the performance in the whole learning process. The results on each dataset are plotted in Fig. 1.



(a) Double Inter-Twin Moons (b) Triple Inter-Twin Moons (c) Digits

(d) Amazon (e) Office-31 (f) ImageCLEF-DA

Figure 1: The results of passive models on different datasets

To analyze the results, the whole learning process was divided into an earlier stage and a latter stage. The earlier stage is the period at the beginning of the AL process, where the labeled instances are relatively few. The latter stage is the period where all the budget is almost consumed, and the labeled instances are relatively adequate. To present the results more clearly, for each dataset, the performance at the first AL iteration and the last iteration are recorded in Table 4 in Appendix E.1.

For DANN and SDL-joint, all parts of models are trained on instances from all the domains. They usually perform well at the earlier stage. On digits and amazon datasets, they achieve the top initial performance as shown in Fig. 1(c,d). On the rest four datasets, they get above-average performance at the earlier stage. At the latter stage, these two models perform worse than the other models in the most datasets except on digits dataset. This joint training relieves the labeled data scarce situation at the beginning. But, it fails to learn the domain-specific information, and the performance are damaged. The learning curves of DANN and SDL-joint are relatively flatter, which means the improvements they get from an addition of labeled instances are less than the other models. Besides, on all the datasets, the performances of DANN and SDL-joint are very similar. It reflects that solely adding a discriminator may not bring improvements in MDL setting.

SDL-separate and MDNet both have domain-specific classifiers. They usually do not perform good at the earlier stage. However, their learning curves are relatively steeper than the others, which means their performances increase rapidly with the increasing number of labeled instances. So at the latter stage, these two models might reach or outperform other models. On amazon dataset, MDNet performs good at the end of the learning process as shown in Fig. 1(d). It is worth to note that, when the labeled data are adequate, training these models are equivalent to train different models on different domain. In this case, all the information from each domain could be well captured from adequate labeled data. But when there are only few labeled instances, these models would not perform well.

MAN model performs well at the earlier stage. At the latter stage, MAN could also keep the superiority over other models in most cases. On digit dataset, MAN only get medium performance, but it still maintains a small margin from the performance of the best model according to Table 4. According to the good performance of MAN, we believe that the shared feature extractor ensures a good initial performance and the domain-specific extractor captures the domain-specific information at the latter stage of learning. In short, MAN has shown its superiority over other models in most cases. Compared to other models, it is the best model considering the whole learning process.

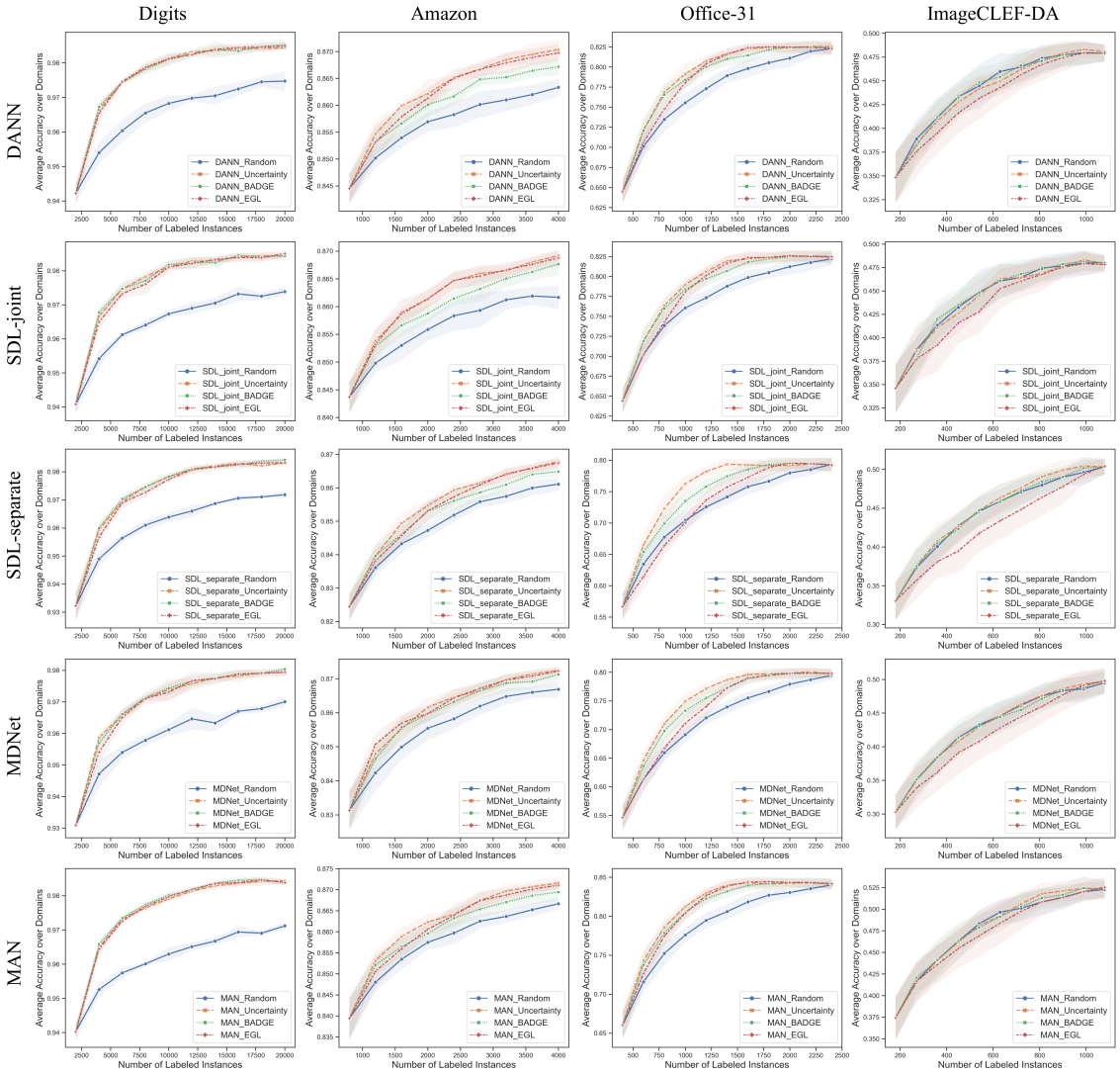


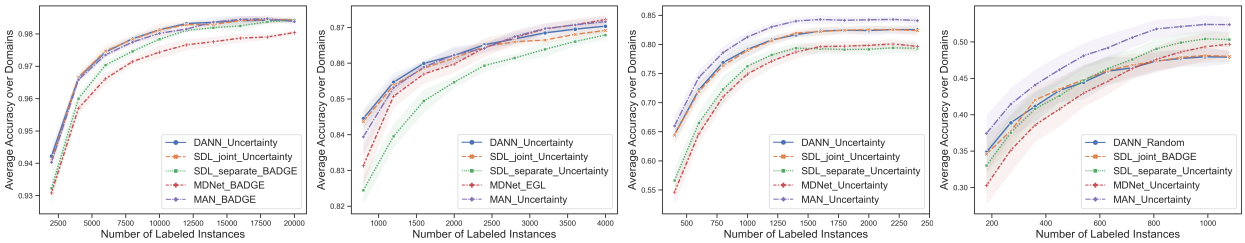
Figure 2: The results of AL strategies on each model: the performance on digits, amazon, office-31 and imageCLEF-DA datasets are plotted in each column correspondingly.

5.2 Comparisons over Strategies

In this section, all models and strategies are combined and compared on each dataset. We plan to answer the first research question: which would be the best off-the-shelf model-strategy combination. The performances on four datasets (except inter-twin moons toy datasets) are plotted in Fig. 2. Each sub-graph represents the results of all the strategies with one model on one dataset. The results from the same dataset are presented in each column, and the results from the same model are presented in each row.

From the result in Fig. 2, the most important finding is that the uncertainty selection (BvSB) already performs very well in most cases. In the worst case, it still maintains a very small margin from the best performed strategies.

To reveal the best off-the-shelf model-strategy combination, the results of the best strategies on each model for each dataset are plotted together in Fig. 3. MAN with uncertainty performs clearly better than the others on office-31 and imageCLEF-DA datasets. On amazon dataset, DANN and MAN with uncertainty both obtain top performance. DANN performs better at the earlier stage, while



(a) Digits (b) Amazon (c) Office-31 (d) ImageCLEF-DA

Figure 3: The results of the best strategies from each model

MAN performs better at the latter stage. On digit dataset, DANN and SDL-joint with uncertainty strategy perform best, and MAN with BADGE still obtains a good performance. According to Fig. 2, the performance difference between MAN with uncertainty and MAN with BADGE is small, which means the performance of MAN with uncertainty can't be bad at this dataset.

In this comparison, except the combination of MAN and uncertainty strategy, other combinations would perform bad on at least one or two datasets. In conclusion, for the first research question, the MAN model with uncertainty (BvSB) strategy would be our off-the-shelf recommendation due to its good performance.

5.3 Robustness over Datasets

Beside the absolute performance of the combinations, as one of our research question, their robustness over datasets are also concerned. Specifically, we are curious about whether the model-strategy combinations would obtain consistent performance on different datasets. If one combination is the best on one dataset but the worst on another in the comparison, it won't be a good solution.

At first, the strategy level robustness is discussed on different models. From the results in Fig. 2, the relative orders of curves are similar at each sub-graph in the same column. It means that on the same dataset, the strategies have similar level of superiority over models. In other words, if one strategy performs well on one model, it is more likely to consistently perform well on the others. The selected uncertainty strategies are robust over models.

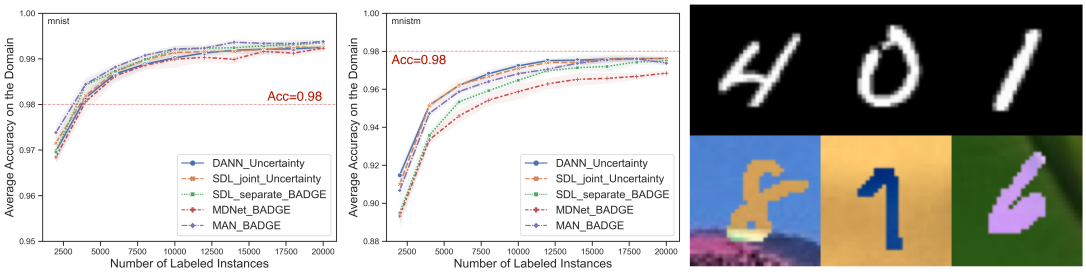
On different datasets, the superiority of strategies varies. The uncertainty strategy consistently performs well on all the datasets, but the performances of the other strategies diverse a lot. BADGE performs better than EGL on imageCLEF-DA datasets, while it performs worse or similar to EGL on the rest datasets. Besides, as an active selection strategy, EGL doesn't consistently outperform passive selection on all the datasets. In Fig. 2, EGL even significantly performs worse than random selection on imageCLEF-DA dataset. The other strategies only perform equivalently or slightly better than random selection as well. So from the results, only the uncertainty (BvSB) strategy is relatively robust over different datasets and consistently achieve good performance.

In such condition, the combinations with the strategies other than uncertainty could easily perform inconsistently on different datasets. The combination-level robustness comes from both the models and the strategies. In section 5.1, the superiority of MAN over other models on different datasets has been shown. In this case, the combination of MAN with uncertainty strategy shows its robustness over datasets in general as revealed in Fig. 3.

5.4 Robustness over Domains

In this section, we plan to answer the third research question, how robust the combinations could be on different domains? In MDL, although the overall performance is concerned, the performances on each domain are also important.

For the best strategy selected for each model in the section 5.2, the domain level performances are also plotted. On digit dataset, the results for each domain are plotted in Fig. 4. The classification task constructed for MNIST-M is supposed to be harder due to the more complex backgrounds as



(a) MNIST (b) MNIST-M (c) Digits (Ganin et al., 2016)

Figure 4: The results on two domains from digits dataset

shown in Fig. 4(c). The accuracy on MNIST-M is relatively lower than the accuracy on MNIST as we expected. On domain MNIST-M, all the combinations obtain accuracy evaluations lower than 98% when the budgets are consumed. But on domain MNIST, the accuracy of 98% has been exceeded with less than 5000 labeled instances. Besides, the model-strategy combinations perform more concentrated on MNIST than on MNIST-M. These facts might reflect the difficulty of the task on domain MNIST-M. The performances of different domains on the rest datasets are plotted in Appendix E.2. From the results, MAN with uncertainty strategy consistently performs well on each domain.

In conclusion, from our result, the most of the model-strategy pairs doesn't consistently outperform the other combinations on all the domains. But our off-the-shelf recommendation MAN with uncertainty relatively performs well on different domains (in the worst case, it could still obtain a medium performance on the worst performed domain). In general, MAN with uncertainty selection shows its robustness over different domains.

6 Conclusions

In this work, a comparative study was made for MDAL setting. At first, we provided the formal definition of MDAL setting. Then, we thoroughly reviewed the literatures and categorized the models and the corresponding AL strategies. From the review, there isn't many works directly related to MDAL setting. Under this circumstance, our goal is to provide an off-the-shelf solution.

The combinations of 5 models from MDL setting with 4 AL strategies are compared. First, in the comparison of models, we found the superiority of MAN. Then, in the comparison of AL strategies, we found that the simplest uncertainty strategy outperforms the others. Finally, in the perspective of model-strategy combination, combining MAN with uncertainty strategy would consistently lead a good performance in general. The robustness and stability of this combination is also better than the other combinations over datasets and domains. In conclusion, MAN with uncertainty strategy would be our recommended solution for MDAL setting.

References

Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 2020.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. pp. 343–351, December 2016.

Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Joint transfer and batch-mode active learning. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 253–261, Atlanta, GA, USA, June 2013.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. June 2012.

Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pp. 1226–1240, New Orleans, Louisiana, USA, June 2018.

Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: a deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 647–655, Beijing, China, June 2014.

Mark Dredze and Koby Crammer. Online methods for multi-domain learning and adaptation. In *2008 Conference on Empirical Methods in Natural Language Processing*, pp. 689–697, Honolulu, Hawaii, USA, October 2008.

Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning from multi-domain data. In *IEEE/CVF International Conference on Computer Vision*, pp. 3244–3254, Seoul, Korea (South), October 2019.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17:59:1–59:35, 2016.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, Providence, RI, USA, June 2012.

Sheng-Jun Huang and Songcan Chen. Transfer learning with active queries from source domain. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1592–1598, New York, NY, USA, July 2016.

Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007.

Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2372–2379, Miami, Florida, USA, June 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lianghao Li, Xiaoming Jin, Sinno Jialin Pan, and Jian-Tao Sun. Multi-domain active learning for text classification. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1086–1094, Beijing, China, August 2012.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Semi-supervised stochastic multi-domain learning using variational inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pp. 1923–1934, Florence, Italy, July 2019.

Yunsheng Li and Nuno Vasconcelos. Efficient multi-domain learning by covariance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5424–5433, Long Beach, CA, USA, June 2019.

Yajing Liu, Xinmei Tian, Ya Li, Zhiwei Xiong, and Feng Wu. Compact feature learning for multi-domain image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7193–7201, Long Beach, CA, USA, June 2019.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision*, pp. 2200–2207, Sydney, Australia, December 2013.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 97–105, Lille, France, July 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. pp. 136–144, December 2016.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2208–2217, Sydney, NSW, Australia, August 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*, pp. 1647–1657, Montréal, Canada, December 2018.

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision*, pp. 5716–5726, Venice, Italy, October 2017.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4293–4302, Las Vegas, NV, USA, June 2016.

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 3934–3941, New Orleans, Louisiana, USA, February 2018.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 506–516, Long Beach, CA, USA, December 2017.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8119–8127, Salt Lake City, UT, USA, June 2018.

Amir Rosenfeld and John K. Tsotsos. Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3):651–663, 2020.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. pp. 2988–2997, August 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, Salt Lake City, UT, USA, June 2018.

Burr Settles. Active learning literature survey. 2010.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, Honolulu, Hawaii, USA, October 2008. ACL.

Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 728–737, Snowmass Village, CO, USA, March 2020.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2962–2971, Honolulu, HI, USA, July 2017.

Jin Xiao, Shuhang Gu, and Lei Zhang. Multi-domain learning for accurate and few-shot color constancy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3255–3264, Seattle, WA, USA, June 2020.

Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1249–1258, Las Vegas, NV, USA, June 2016.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5419–5428, Stockholmsmässan, Stockholm, Sweden, July 2018.

Zhigang Yuan, Sixing Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. Domain attention model for multi-domain sentiment classification. *Knowledge-Based Systems*, 155:1–10, 2018.

Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3386–3392, San Francisco, California, February 2017.

Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. Multi-domain active learning for recommendation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2358–2364, Phoenix, Arizona, USA, February 2016.

A RELATION TO OTHER FIELDS

As shown in Fig 5, we specified our target field in the red dotted box.

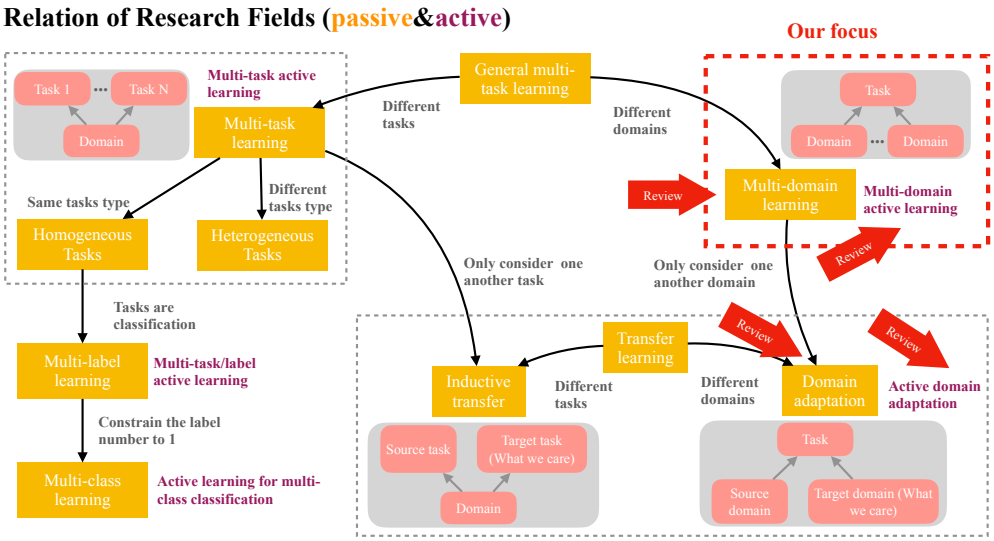
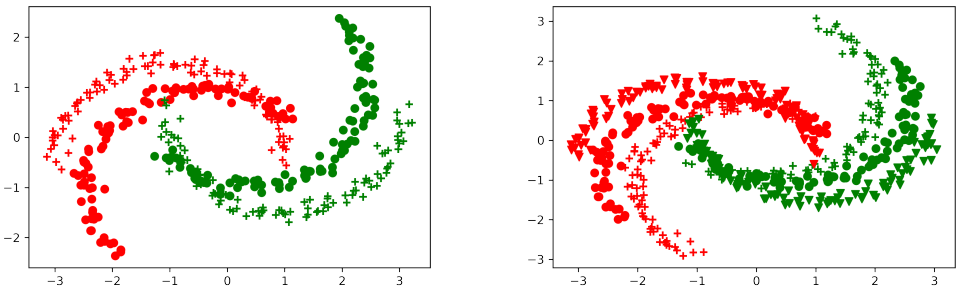


Figure 5: The relationships between different terminologies. Our focus is the MDAL in the red dot box. The fields pointed by the red arrows are reviewed.

B DATASETS

The details of each dataset are recorded in this section. The number of instances and the train/test partition are included.

- **Synthetic Double and Triple Inter-Twin Moons Toy Datasets**: The datasets are created by the make_moon() function in scikit-learn library (Pedregosa et al., 2011) in Python. As 2-D datasets, they could be easily visualized as shown in Fig. 6(a) and Fig. 6(b). This is a binary classification task. On double inter-twin moons dataset, we set the number of instances to 1600/400/400 for training/validation/test set on each domain. The second pair of moons is created by rotating the original distribution a small angle of $2/9\pi$. On triple twin moons dataset, we set the number of instances to 1000/200/200 for training/validation/test set on each domain. The second and the third pairs of moons are created by rotating the original distribution $1/9\pi$ and $2/9\pi$ correspondingly.
- **Digits**: This is a 10 class classification task. There are several examples of instances in Fig. 4(c). For each domain, there are 50000 training samples, 10000 validation samples and 10000 test samples. Each instance is a digit picture with dimension (28,28,3).
- **Amazon** (mSDA representations): After the construction of features by using mSDA (Chen et al., 2012), each instance would have 30000 features. For each domain, there are 2000 training samples, 1000 validation samples. There are 3465/2586/4681/4945 test samples on domain 'books', 'dvd', 'electronics' and 'kitchen' correspondingly.
- **Office-31**: Under the DeCaf representations (Donahue et al., 2014), the images are encoded as vectors with length 4096. There are 2817/498/795 samples on domain 'amazon', 'dslr' and 'webcam' correspondingly. The ratio 6:2:2 is used to split the training/validation/task sets.
- **ImageCLEF-DA** [2]: This dataset contains 12 categories from 3 domains. Each image is encoded as a vector with length 1024. There are 600 samples on each domain correspondingly. The ratio 6:2:2 is used to split the training/validation/task set.



(a) Double (b) Triple

Figure 6: A visualization of inter-twin moons datasets

[2] https://www.imageclef.org/2014/adaptation

C MODEL STRUCTURES

5 models were compared in this work. For the structures of models, we note that we didn't precisely use the structures introduced in the original paper due to the difference of the dimensions of inputs. In the comparison, the macro-structures of the models maintain the same, i.e. the models will contain the same number of feature extractors, classifiers and discriminators compared to the models in the original paper. The constitution of different models was sketched in Fig. 7. For each dataset, the 5 compared models share the same micro-structure, i.e. the structure of feature extractors, classifiers and discriminators of different models are set to be the same. The particular structures of modules were presented in Table 1. For the datasets which is used in the corresponding paper, the model structures are remained. The DANN models used on double inter-twin moons and digits datasets are same to the original DANN work (Ganin et al., 2016). For other datasets, we use shallow networks with one hidden layer.

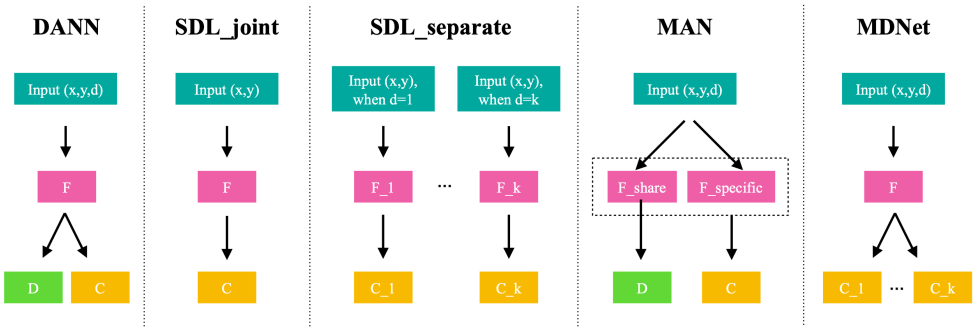


Figure 7: Sketches of different models: F represents feature extractors. D represents domain discriminators. C represents classifiers. x, y and d represent the input features, labels of instances and the domain ID of the instances correspondingly. Arrows represent the forward propagation.

Table 1: Structures of different modules

Module Name / Dataset Name	Feature Extractor	Classifier	Discriminator
Double Inter-Twin Moons	Linear(2,15) Sigmoid layer	Linear(15,2) Softmax layer	Linear(15,2) Softmax layer
Triple Inter-Twin Moons	Linear(2,15) Sigmoid layer	Linear(15,2) Softmax layer	Linear(15,3) Softmax layer
Digits	Conv2d(3, 32, kernel_size=5) BatchNorm2d(32) nn.MaxPool2d(2) ReLU Conv2d(32, 48, kernel_size=5) BatchNorm2d(48) Dropout2d() MaxPool2d(2) ReLU	Linear(48 * 4 * 4, 100) BatchNorm1d(100) ReLU Dropout Linear(100, 100) BatchNorm1d(100) ReLU Linear(100, 10) Softmax Layer	Linear(48 * 4 * 4, 100) BatchNorm1d(100) ReLU Linear(100, 2) Softmax Layer
Amazon	Linear(30000,50) Sigmoid layer	Linear(50,2) Softmax layer	Linear(50,4) Softmax layer
Office-31	Linear(4096,50) Sigmoid layer	Linear(50,31) Softmax layer	Linear(50,3) Softmax layer
ImageCLEF-DA	Linear(1024,50) Sigmoid layer	Linear(50,12) Softmax layer	Linear(50,3) Softmax layer

D HYPER-PARAMETERS

The hyper-parameters for training the neural networks are recorded in Table 2.

Table 2: Hyper-parameters for the network training

Dataset Name	Learning Rate	Trade-Off in Models	Batch Size	Epochs	Patience for Early Stopping
Inter-twin Moons (Double)	0.003	0.1	32	300	50
Inter-twin Moons (Triple)	0.003	0.1	32	300	50
Digit	0.0001	0.1	1024	300	50
Amazon	0.0001	0.05	256	300	10
Office-31	0.0001	0.1	128	300	50
ImageCLEF-DA	0.0001	0.1	32	300	50

The hyper-parameters for the AL process are recorded in Table 3. At the beginning, a part of instances are randomly selected to train the initial model (recorded as initial labeled size), which is referred to as a warm start process. Then, a fixed number of instances are iteratively selected (recorded as AL batch size). When the total budget has been consumed, the AL process terminates. The learning process are repeated multiple times (recorded as repeat time) to get the average results.

Table 3: Hyper-parameters for AL process

Dataset Name	Total Budget	Initial Labeled Size	AL Batch Size	Repeat Times
Inter-twin Moons (Double)	500	100	40	50
Inter-twin Moons (Triple)	500	100	40	50
Digit	20000	2000	2000	5
Amazon	4000	800	400	10
Office-31	2400	400	200	20
ImageCLEF-DA	1080	180	90	30

E RESULTS OF COMPARISONS

E.1 RESULTS OF COMPARISONS OVER MODELS

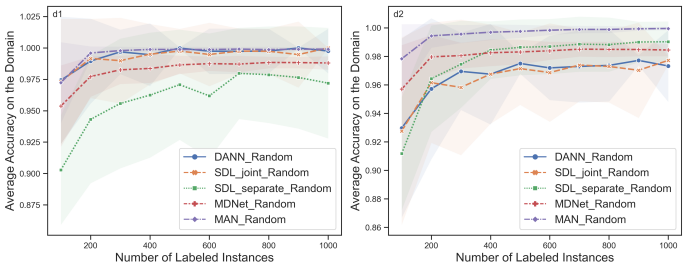
To present the results of comparison over models more clearly, the performances are also recorded in Table 4. For each dataset, the performance at the first selection iteration and the last selection iteration are recorded. These two stages reflect the performances of models when the number of labeled instances is small and large.

Table 4: The classification accuracy (%) of models on each dataset at the first and last AL iterations

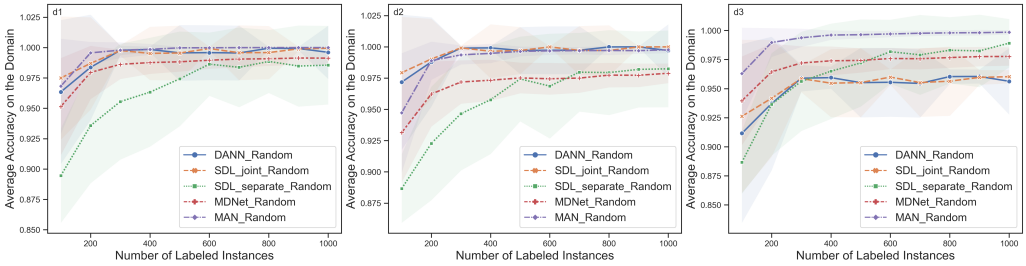
Model \ Datasets	Double Moons	Triple Moons	Digit	Amazon	Office-31	ImageCLEF
Labeled Number	100	100	2000	800	400	180
DANN	95.20	94.90	**94.22**	**84.45**	64.48	34.83
MAN	**97.54**	95.95	94.03	83.94	**66.01**	**37.44**
MDNet	95.54	94.08	93.09	83.13	54.62	30.28
SDL-joint	95.04	**96.02**	94.08	84.37	64.42	34.59
SDL-separate	90.73	88.93	93.23	82.44	56.64	33.02
Labeled Number	1000	1000	2000	4000	2400	1080
DANN	98.54	98.32	**97.47**	86.33	82.28	47.91
MAN	**99.93**	**99.87**	97.12	86.67	**84.01**	**52.28**
MDNet	98.62	98.26	97.00	**86.69**	79.45	49.50
SDL-joint	98.86	98.66	97.39	86.17	82.21	48.02
SDL-separate	98.11	98.57	97.18	86.11	79.35	50.33

E.2 RESULTS OF COMPARISONS OVER DOMAINS

The performances of different models on each domain of double and triple inter-twin moons are shown in Fig. 8. The domain performances of the best model-strategy pairs on amazon, office-31 and imageCLEF-DA dataset are shown in Fig. 9 and Fig. 10.

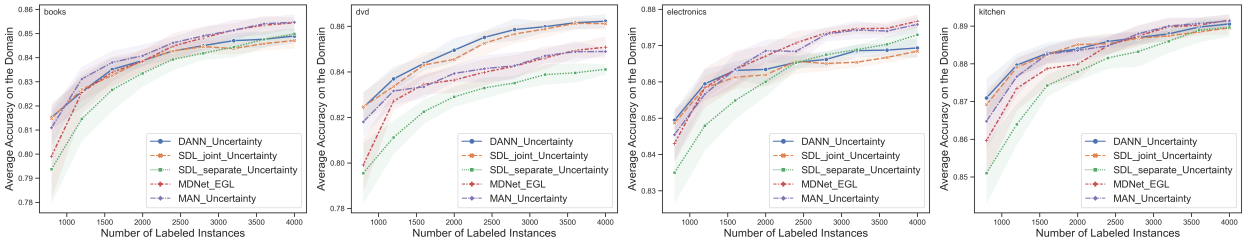


(a) Double inter-twin moons dataset



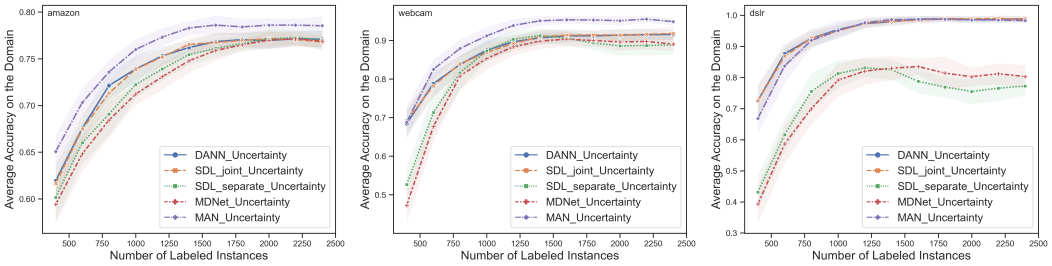
(b) Triple inter-twin moons dataset

Figure 8: Learning curves for different domains from inter-twin moons datasets

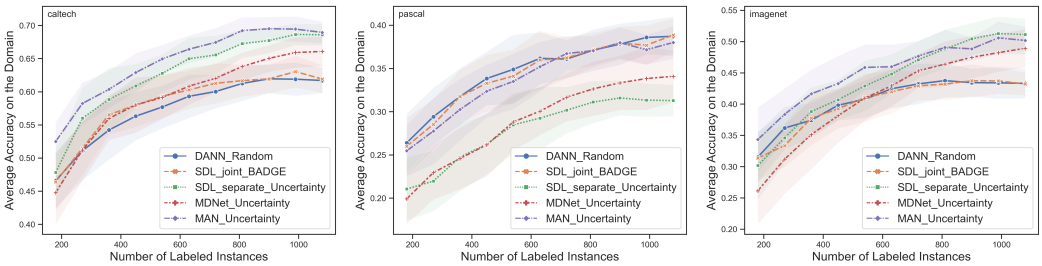


(a) Books (b) DVD (c) Electronics (d) Kitchen

Figure 9: Learning curves for 4 different domains from amazon dataset



(a) Office-31 dataset



(b) ImageCLEF dataset

Figure 10: Learning curves for different domains from office-31 and imageCLEF-DA datasets