ORIENT: Submodular Mutual Information Measures for Data Subset Selection under Distribution Shift

Athresh Karanam*
University of Texas at Dallas
athresh.karanam@utdallas.edu

Krishnateja Killamsetty*
University of Texas at Dallas
krishnateja.killamsetty@utdallas.edu

Harsha Kokel*
University of Texas at Dallas
hkokel@utdallas.edu

Rishabh K Iyer
University of Texas at Dallas
rishabh.iyer@utdallas.edu

Abstract

Real-world machine-learning applications require robust models that generalize well to distribution shift settings, which is typical in real-world situations. Domain adaptation techniques aim to address this issue of distribution shift by minimizing the disparities between domains to ensure that the model trained on the source domain performs well on the target domain. Nevertheless, the existing domain adaptation methods are computationally very expensive. In this work, we aim to improve the efficiency of existing supervised domain adaptation (SDA) methods by using a subset of source data that is similar to target data for faster model training. Specifically, we propose ORIENT, a subset selection framework that uses the submodular mutual information (SMI) functions to select a source data subset similar to the target data for faster training. Additionally, we demonstrate how existing robust subset selection strategies, such as GLISTER, GRADMATCH, and CRAIG, when used with a held-out query set, fit within our proposed framework and demonstrate the connections with them. Finally, we empirically demonstrate that SDA approaches like d-SNE, CCSA, and standard Cross-entropy training, when employed together with ORIENT, achieve a) faster training and b) better performance on the target data.

1 Introduction

The recent success of deep learning frameworks in applications such as image classification [10], speech recognition [21], and object detection [14] stems primarily from the availability of large amounts of labeled data. However, due to enormous labeling costs and the need for specialists in specific domains such as medical imaging, it is not always possible to obtain vast amounts of labeled data in all situations. On the contrary, training deep models on limited amounts of data can lead to poor performance due to overfitting [2]. Consequently, where obtaining large amounts of labeled data is difficult for the target domain, closely related domain (source domain) is used to train the model. However, this may result in a deep model with suboptimal performance on target domain as a result of the distribution shift [1, 6, 5, 50], i.e., change of data distribution from source domains to target domains. A change in the data distribution often renders the features learned by the model on the source domain irrelevant for the target domain.

To address the problem of distribution shift, many domain adaptation [54, 42] and domain generalization [41] techniques have been proposed in recent years. Domain adaptation methods assume that some target domain information is available (usually target data), whereas domain generalization methods do not. Domain adaptation (DA) methods can be categorized into unsupervised [16, 13, 33, 52]

*equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

(using unlabeled target data), semi-supervised [17, 59, 44, 48] (using labeled and unlabeled target data), and supervised [40, 39, 58, 38, 20] (using labeled target data) methods. UDA methods that do not require any labeled target data assume access to large volumes of unlabeled target data. When providing the same amount of target data, SDA methods are more effective than UDA methods [40]. As it is not difficult to obtain a few samples (as less as 2 examples per class) of labeled target data in practice, SDA methods are an attractive approach in scenarios with limited target data. Hence, in this work we are explicitly focused on the SDA setting.

Recently, various effective SDA methods are proposed [40, 58, 38]. However, they are usually compute intensive. For example, using one of the state-of-the-art SDA method, d-SNE [58], to train a ResNet50 model [19] on the Office-Home dataset [53] with 3022 samples of the a source data and 338 samples a target data for 300 epochs takes > 18 hours using a `GTX 1080 Ti` GPU. To put that in perspective, training a ResNet50 model on the larger CIFAR100 dataset with 45000 training samples for 300 epochs on the same machine for standard cross-entropy loss takes only 14 hours. The increase in training time not only increases energy consumption and CO2 emissions but also restricts the usefulness of SDA methods in resource-constrained environments. We address this problem by seeking an answer to the following question: ***Can we improve the efficiency of SDA methods by training on subsets of source data to ensure faster adaptation and reduction in training time?***

To this end, we propose ORIENT, a subset selection framework that utilizes Submodular Mutual Information (SMI) measures [22, 18] to select subsets of source data that are similar to target data for training. Our key insight is that by training the model on data points that are similar to target data, we can speed up the training. The ORIENT framework, illustrated in Figure 1, complements existing SDA methods and can be effectively combined with any of them, enabling us to achieve a reduction in training times, energy costs, and CO2 emissions.

The key contributions of our paper include: (1) ORIENT, an SMI-based subset selection framework for efficient and effective supervised domain adaptation. (2) In Section 3, we illustrate that ORIENT unifies three existing subset selection methods: CRAIG [36], GLISTER [26], and GRAD-MATCH [25]. Specifically, these methods fit into ORIENT's framework when used with a held-out validation set from the target domain. (3) We empirically show the effectiveness of ORIENT when used with existing SDA methods like CCSA [40], d-SNE [58], and Standard Cross-entropy training on two real-world datasets: Office-31 [43] and Office-Home [53]. Our experiments also demonstrate that model learn better class discrimination features with ORIENT, resulting in better classification performance.

1.1 Related work

SDA When the labeled training data is scarce for a domain τ (or a task), a powerful way to boost performance is by pretraining a model on a related domain and fine-tuning it in the target domain [15, 60, 9]. I SDA techniques have been investigated for various distribution shift settings where the target domain ($\tau^t = \{X^t, Y^t\}$) has different marginal or conditional distributions than the source domain ($\tau^s = \{X^s, Y^t\}$). In this work, we consider *covariate shift* [1], wherein the marginal distibution is different $\big($i.e. $P(X^s) \neq P(X^t)\big)$, but the conditional distribution remains the same $\big($i.e. $P(Y^s|X^s) = P(Y^t|X^t) = P(Y|X)\big)$. A notable body of work for covariate shift SDA has focused on identifying a latent feature space that is domain-invariant [40, 39, 58, 38, 20]. This line of work train a network consisting of a feature extractor and a classifier end-to-end. The feature extractor learns a non-linear transformation of the samples from different domains to a shared latent space, and the classifier learns to assign class labels. Different researchers have proposed different optimization objectives and loss functions to encourage domain confusion and class separability.

Tzeng et al. [51] designed a domain-confusion loss to optimize for domain invariance and match the distribution over classes in the source domain to the soft label in the target domain. Motiian et al. [40] propose a classification and contrastive semantic alignment loss (CCSA). CCSA uses contrastive loss with Siamese Network to encourage the same class samples from different distributions to be nearby in the latent space (semantic alignment loss), and the different class samples from different distributions to be far apart (separation loss). Few-shot adversarial domain adaptation (FADA) [39] proposes to learn similar latent space with adversarial training. Domain-adaptation using stochastic neighborhood embedding (d-SNE) [58] proposes to maximizes the smallest distance between the samples of different classes and minimizes the largest distance between the samples of the same class while being domain invariant. Another approach, Second- or Higher-order Transfer of Knowledge

(So-HoT) [30], aims to align the with-in class scatters and maintain separation of between-class scatters. Finally, Morsing et al. [38] propose domain adaptation using graph embedding (DAGE) and show that CCSA and d-SNE can also be expressed as graph embedding methods [20].

Subset Selection methods Submodular function [34, 12] is one of the effective approach of data subset selection which is used in variety of applications [7]. For example, it has been employed for efficient training in speech recognition [56, 55], machine translation [29], computer vision [23], supervised classification [36, 26, 25], semi-supervised classification [27], active learning methods [57, 47, 3, 26], and hyper-parameter optimization [28]. Another widely used method for selecting subsets is coreset construction. The coresets [11] are weighted subsets of the data that approximate the semantic characteristics of the entire dataset (e.g., loss, marginal probability, etc.). A number of recent coreset selection-based methods [37, 26, 25, 27] have demonstrated the promise to efficiently and robustly train deep models. Coresets are used for other deep learning applications like continual learning, active learning and data summarization [8, 49].

Subset selection plays a significant role in robust learning under realistic scenarios, such as imbalances or rare classes, out-of-distribution(OOD) data, or redundancy. Axelrod et al. [4] propose a simple cross-entropy based data selection methods for effective domain adaptation in statistical machine learning. Kothawade et al. [31] employed SMI measures for active learning in realistic scenarios tackling imbalance, OOD, and redundancy. PRISM [32] employed SMI measures for targeted data summarization. GLISTER [26] and GRADMATCH [25] shows that using a clean held-out validation set with SMI mitigates the label noise and class imbalance in the training set. Mirzasoleiman et al. [37] showed the data samples with clean labels cluster together, whereas the ones with noisy labels spread out in the gradient space, thus making k-medoid clustering an effective method for reducing noise in labeled data. Furthermore, Killamsetty et al. [27] tackles OOD and imbalance in the unlabeled data in semi-supervised learning setting through data subset selection. We show how these previous subset selection strategies including a modified version of CRAIG [36] were also using instantiations of SMI measures in Section 3.

2 Preliminaries

In this section, we introduce submodular functions, SMIs and SDA loss functions.

Submodular Functions: Let D be a set of n data points $D = \{1, \cdots, n\}$ and $f : 2^D \rightarrow \mathbb{R}$ be a set function returning real-value for any subset of set D. A function f is *submodular* [12] if $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$, $\forall x \in D$, $\forall A \subseteq B \subseteq D$ and $x \notin B$. Monotone Submodular functions, when maximized for a cardinality constraint using a simple greedy algorithm, admit a constant factor of $1 - \frac{1}{e}$. This can be done in near-linear time using a stochastic-greedy algorithm.

Submodular Mutual Information (SMI): The SMI between two sets $A, B \subseteq D$, *instantiated* with a submodular function f, is defined as $I_f(A; B) = f(A) + f(B) - f(A \cup B)$ [22]. In this work, we aim to select data points $A \subset D$ that maximize the SMI between A and B. Intuitively, this allows selection of data points that are *similar* to B while being diverse.

SDA Loss Functions: For supervised domain adaptation, specialized loss functions are introduced as described in the previous section. Here we highlight two state-of-the-art domain adaptation loss functions, CCSA and d-SNE, which are used in our experiments.

In CCSA, the classifier is modeled as a composition of two functions–$h \circ g$. Here, $g : X \rightarrow Z$ is a feature extractor that transforms the input from the feature space X to an embedding space Z, and $h : Z \rightarrow Y$ is a predictor function. The CCSA loss for supervised domain adaptation is defined as,

$$\mathcal{L}_{CCSA}(h \circ g) = \mathcal{L}_{\text{CE}}(h \circ g) + \mathcal{L}_{SA}(g) + \mathcal{L}_{S}(g),$$

where $\mathcal{L}_{\text{CE}}(h \circ g)$ is the *cross-entropy loss* for multi-class classification, $\mathcal{L}_{SA}$ is a *semantic alignment loss* encouraging the samples from different domains but same label to map nearby in the embedding space, and $\mathcal{L}_S$ is a *separation loss* encouraging the samples from different domains and different labels to map far away in the embedding space.

The d-SNE loss function for SDA is defined as

$$\mathcal{L}_{d\text{-SNE}}(h \circ g) = \tilde{\mathcal{L}}(g) + \alpha \mathcal{L}^s_{\text{CE}}(h \circ g) + \beta \mathcal{L}^t_{\text{CE}}(h \circ g),$$

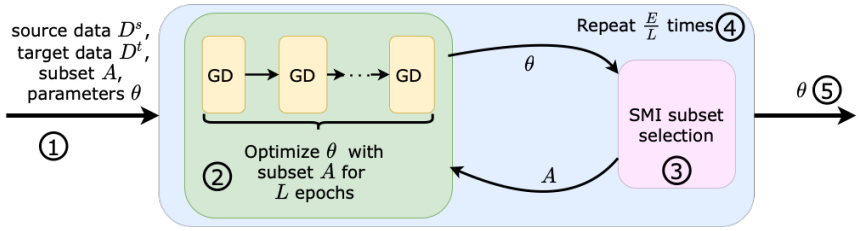


Figure 1: Illustration of ORIENT framework. **(1)** Given the target data D^t and source data D^s, a subset $A \subseteq D^s$ and model parameters θ are randomly initialized. **(2)** Model parameters θ are optimized for L epochs on the subset A with any gradient descent (GD) based method. **(3)** Subset A is updated using the SMI measure and current model parameters θ after every L^{th} epoch. **(4)** Model is trained for E epochs, with subset selection after every L interval. **(5)** The final model parameters are returned after E epochs.

Name	$f(A)$	$I_f(A; D^t)$
FLMI	$\sum_{i \in D^t} \max_{j \in A} S_{ij}$	$\sum_{i \in D^t} \max_{j \in A} S_{ij} + \eta \sum_{i \in A} \max_{j \in D^t} S_{ij}$
LOGDETMI	$\log\det(S_A)$	$\log\det(S_A) - \log\det(S_A - \eta^2 S_{A,D^t} S_{D^t}^{-1} S_{A,D^t}^T)$
GCMI	$\sum_{i \in A, j \in D^s} S_{ij} - \lambda \sum_{i,j \in A} S_{ij}$	$2\lambda \sum_{i \in A, j \in D^t} S_{ij}$
COM	Appendix Eq. 3	$\eta \sum_{i \in A} \psi(\sum_{j \in D^t} S_{ij}) + \sum_{j \in D^t} \psi(\sum_{i \in A} S_{ij})$

Table 1: Instantiations of submodular mutual information functions [32].

where $\mathcal{L}^s_{\text{CE}}$ and $\mathcal{L}^t_{\text{CE}}$ are the cross-entropy loss on the source and the target domain examples, respectively, and $\tilde{\mathcal{L}}$ minimizes the largest distance between the samples from different domains with same label and maximizes the smallest distance between the samples from different domain with different label in the embedding space. SDA loss functions are further elaborated in Section A.4.

3 ORIENT

For our SDA setting, we assume a small amount of labeled data (as less as 2 examples per class) is available from the target domain τ^t and sufficient labeled data is available from the source domain τ^s. We denote the source dataset as $D^s = (x_i, y_i)_{i=1}^{|D^S|}$ and the target dataset as $D^t = (x_i, y_i)_{i=1}^{|D^t|}$. The source dataset is used for training the model $\mathcal{M}$ using an SDA loss function $\mathcal{L}$. We will introduce other necessary notations in the remainder of the paper, as necessary. A complete list of notations is presented in Appendix (A.1).

First, we extend the SMI measure to incorporate data subsets from different domains. For a source domain subset $A \subseteq D^s$ and the target domain data set D^t, we denote SMI measure as $I_f(A; D^t)$. Essentially, assuming that both the domains form a set $\mathcal{D}$, and $A \subseteq D^s \subset \mathcal{D}$ and $D^t \subset \mathcal{D}$. With S denoting the similarity matrix defined over a subset A and D^t, Table 1 presents the mathematical expression of the SMI functions for different instantiations of submodular functions (f) [32]. We defer the readers to Kothawade et al. [32] for more details. Note that the FLMI, GCMI and COM only need the pairwise similarities of the data points in A and D^t. Consequently, the similarity kernel is of size $|A| \times |D^t|$, making it very efficient to optimize.

For efficient SDA, we want to select a subset A of the source domain τ^s such that training a model on A results in a classifier that is proficient on the target domain τ^t. To this effect, we use the gradients $\boldsymbol{\chi}$ of the current model to represent the data points and use it to compute the similarity matrix. Specifically, we define the pairwise similarity between two data points as the cosine similarity

between the gradients,

$$S_{ij} = \frac{\chi_i \, \chi_i}{|\chi_i| \, |\chi_j|} \tag{1}$$

$$\text{where, } \chi_i = \nabla_\theta \mathcal{L}(x_i, y_i) \text{ and } \chi_j = \nabla_\theta \mathcal{L}(x_j, y_j)$$

Given a set size b, we select a *diverse* subset A of the source data D^s that is *similar* to the target data D^t by maximizing the following SMI measure,

$$\underset{A \subseteq D^s, |A| \leq b}{\arg\max} \; I_f(A; D^t) \tag{2}$$

$I_f(A; D^t)$ is an instance of cardinality constrained monotone submodular maximization for all SMI functions except for LogDetMI. Therefore, we can achieve a $1 - \frac{1}{e}$ approximation using the lazy greedy algorithm [35]. Even though LogDetMI is not submodular, previous works [32, 31] have reported good empirical performance using the lazy greedy algorithm for maximization of the LogDetMI function. Following the footsteps of Kothawade et al. [32, 31], we also use the lazy greedy algorithm to maximize the LogDetMI function. However, as the number of parameters in the deep learning model can be extremely high, our gradients χ can be very high dimensional. Following the success of targeted subset selection approaches, GLISTER [26], SIMILAR [31], CRAIG [37] and BADGE [3], we circumvent this problem by using last-layer gradient approximations to represent the data point in the similarity matrix S. Further, we select a new subset every L epochs as the model is trained, and the subsets chosen are adapted accordingly.

Algorithm: We present the complete training procedure of ORIENT in Algorithm 1. We first randomly initialize the training subset A and the model parameters θ in **line 1** and **2**, respectively. For each epoch, we update the model parameters by optimizing a gradient-descent (GD) based loss $\mathcal{L}$ on the current training subset A in **line 4**. After a fixed interval of L epochs, we update the subset A (**line 5–11**). To do this, we compute the gradients $\boldsymbol{\chi}^s$ and $\boldsymbol{\chi}^t$ for all the datapoints in D^s and D^t, in **line 6** and **7**,respectively. Next, we use the gradients to compute the similarity matrix S in **line 8**, as described in Equation 1. In **line 9** we instantiate the SMI function I_f with S and update the subset A in **line 10** using Equation 2. Our implementation is made available online[2].

Algorithm 1 ORIENT

Input: source domain data D^s, query data D^t, batch size b, total epochs E, subset selection interval L, SMI function I_f.

Output: Final model parameters θ

1: $A \leftarrow \text{RandomSubset}(D^s)$ ▷ *Randomly initialize a subset*
2: θ ▷ *initial model parameters*
3: **for** epoch e in E **do** ▷ *for each epoch*
4: $\quad \theta \leftarrow \text{mini-batch-gd}(\theta, \mathcal{L}(A))$ ▷ *mini-batch gradient descent*
5: $\quad$ **if** $e \mod L == 0$ **then** ▷ *perform subset selection every L epochs*
6: $\quad\quad \boldsymbol{\chi}^s \leftarrow \nabla_\theta \mathcal{L}(D^s)$ ▷ *compute gradients for source data*
7: $\quad\quad \boldsymbol{\chi}^t \leftarrow \nabla_\theta \mathcal{L}(D^t)$ ▷ *compute gradients for query data*
8: $\quad\quad S \leftarrow \text{SIMILARITY}(\boldsymbol{\chi}^s, \boldsymbol{\chi}^t)$ ▷ *compute the similarity matrix*
9: $\quad\quad$ Instantiate I_f with S
10: $\quad\quad A \leftarrow \underset{A \subseteq D^s, |A| \leq b}{\arg\max} \; I_f(A; D^t)$ ▷ *update subset A*

Figure 1 illustrates the ORIENT workflow. The general framework of ORIENT can incorporate any gradient decent (GD) based SDA technique. It can train any model with loss function $\mathcal{L}$ by using a subset of the source data, selected every L epochs. In our experiments, we demonstrate the effectiveness of our method in conjunction with standard cross-entropy loss as well as two state-of-the-art domain adaptation losses, CCSA [40] and d-SNE [58].

Connections to Previous Work

ORIENT generalizes three approaches, GLISTER [26], GRADMATCH [25], and a slightly adapted version of CRAIG [36], which were initially proposed for efficient and robust subset selection.

[2]Anonymized link: https://anonymous.4open.science/r/6845_Orient

Specifically, these subset selection approaches maximize a submodular function corresponding to a different instantiations of SMI functions for subset selection. Following theorems summarize the connection of SMI functions with these subset selection approaches.

Theorem 1 *When the outer level loss of the discrete bi-level optimization problem of* GLISTER *is hinge loss, logistic loss, and perceptron loss, then the optimization problem becomes an instance of Concave over Modular (COM) SMI function.*

Theorem 2 *When the optimization problem of* GRADMATCH *is used to match gradients of a held-out validation set, it becomes an instance of the summation of GCMI and a diversity function.*

Theorem 3 *When the optimization problem of* CRAIG *is adapted to match gradients of a held-out validation set, it becomes an instance of the FLMI function with* $\eta = 0$.

We present the proofs of these theorems in Appendix (A.3). In conclusion, SMI measures have been used in previous subset selection strategies for robust learning to deal with the class imbalance and noisy labels in training data sets and achieved comparable performance to current state-of-the-art robust supervised learning approaches.

4 Experimental Evaluation

Our experiments explicitly aim at answering the following questions,

Q1: Does the proposed ORIENT approach substantially reduce the training time while maintaining comparable performance to training on complete source dataset?

Q2: Can the proposed ORIENT approach augment the existing domain adaptation approaches?

We evaluate our proposed approach on two domain adaptation datasets: Office-31 [43] and Office-Home [53]. Office-31 dataset consists of 4110 images of 31 object categories from three different domains: Amazon (A), DSLR (D), and Webcam (W). For our experiments we used all the images from the source domain in the training set (D^s) and two examples of each category in the query set (D^t). Office-Home dataset consists of 10812 images of 65 object categories from four different domains: Art (A), Clipart (C), Product (P), and Real-world (R). Here, the target data (D^t) consists of about 20% of the images from the target domain. We use $L = 20$, that is, we sample the subset after every 20 epochs and use $b = 0.3\%$, that is, we sample 30% of the source domain for training.

To answer the first question, we use ORIENT to train a ResNet50 architecture using the stochastic gradient descent (SGD) algorithm with the momentum of 0.9 and the weight decay ratio of $5e4$. We compare our approach against three baselines: **Full**—training on the source data and the target data, i.e. $D^s \cup D^t$; **Random**—a random subset of the source dataset and the target set, i.e. $R \cup D^t, R = \text{RandomSubset}(D^s)$; and **CRAIG**—a coreset of the source dataset, i.e $\text{coreset}(D^t)$, without any information of the target dataset. We use standard categorical cross-entropy loss function (for multi-class classification) and five different instantiations of the submodular mutual information function: Facility Location Mutual Information (**ORIENT (FLMI)**), Graph Cut Mutual Information (**ORIENT (GCMI)**), Log Determinant Mutual Information (**ORIENT (LDMI)**), GLISTER (**ORIENT (G)**), and GradMatch (**ORIENT (GM)**).

		A → D	A → W	D → A	D → W	W → A	W → D
CCSA	Full	0.78	0.72	0.55	**0.93**	0.55	**0.97**
	Random	0.76	0.72	0.54	0.81	0.54	0.92
	ORIENT	0.77	**0.76**	0.55	0.89	0.55	**0.96**
d-SNE	Full	0.77	0.69	0.53	**0.93**	0.54	**0.98**
	Random	0.76	0.68	0.53	0.86	0.53	0.94
	ORIENT	0.78	0.71	**0.55**	0.90	0.56	**0.97**

Table 2: Test accuracy for office-31 with SDA methods

Figure 2 presents the scatter plot of the speed up against the prediction accuracy of Office-31 dataset using cross-entropy loss.. In all our experiments, we use 0.3 fraction of the source data as the subset

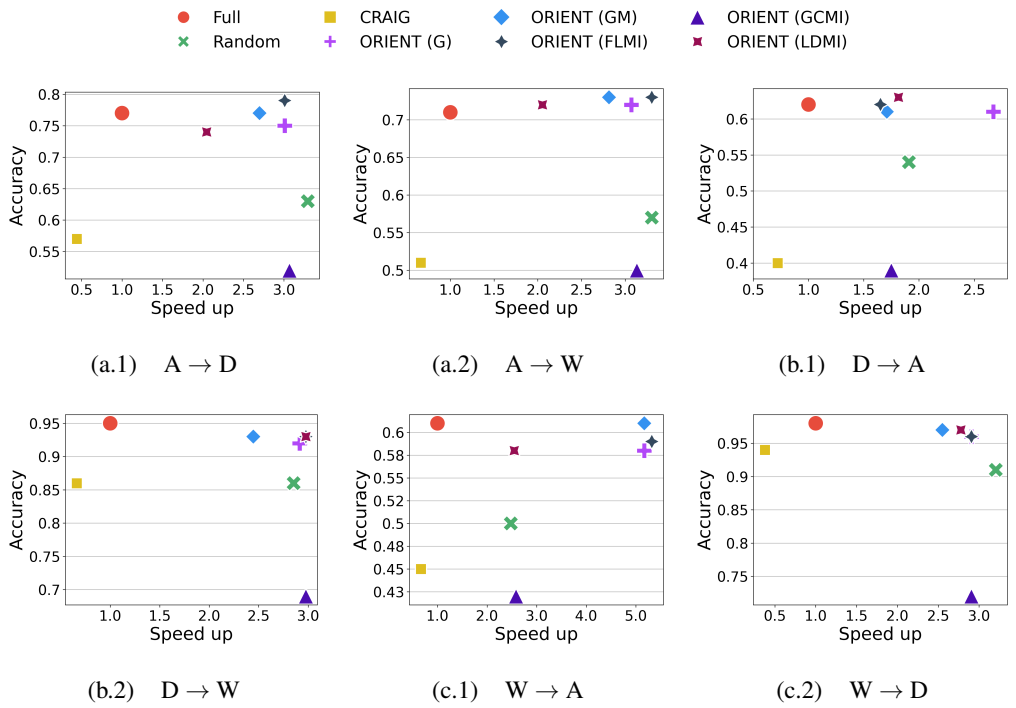


Figure 2: Speed up vs prediction accuracy on three domains of Office31 dataset: Amazon (A), DSLR (D), and Webcam (W). X-axis represents the speed up by the model, i.e. the ratio of the time taken to train on complete source dataset (Full) to the time taken by the model. Y-axis represents the prediction accuracy of the model on the target domain.

		R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
CCSA	Full	0.74	**0.55**	0.62	0.46	0.56	0.67	0.70	**0.64**	0.48	0.57	**0.49**	0.41
	Random	0.71	0.47	0.62	0.45	0.54	0.66	0.69	0.57	0.48	0.5	0.47	0.45
	ORIENT	**0.78**	**0.54**	**0.65**	**0.5**	**0.61**	**0.71**	0.71	**0.65**	**0.51**	**0.59**	**0.54**	**0.47**
d-SNE	Full	0.77	**0.53**	**0.62**	**0.50**	**0.60**	0.71	**0.72**	**0.63**	0.49	**0.52**	0.44	0.40
	Random	0.75	0.50	0.60	0.45	0.57	0.69	0.68	0.59	0.49	0.46	0.43	0.40
	ORIENT	0.77	**0.52**	**0.63**	**0.50**	**0.60**	0.71	**0.71**	0.61	0.51	**0.52**	0.44	0.42

Table 3: Test accuracy for Office-Home with SDA methods

and see $> 2.5\times$ speed up. Figure 3 presents the convergence curves of the Office-Home dataset using cross-entropy loss. It is evident from these charts that our proposed approach ORIENT can achieve better or comparable performance to the Full training in significantly less amount of time, indicating better efficiency. This analysis helps us answer our Q1. ORIENT reduces the training time substantially, the speed up achieved is reciprocal to the fraction of the subset used, without trading the performance. Rather, in some combinations, we observe that ORIENT outperforms Full.

Figure 4 presents the bar plots of the speed up when ORIENT(FLMI) is used in conjunction with d-SNE loss function on Office-31 and Office-Home datasets. Here, we see a consistent $3\times$ speed up as compared to full model training with d-SNE loss. These bar plots demonstrate that our proposed approach ORIENT can augment existing domain adaptation approaches and substantially reduce the training times. Additionally, Tables 2 and 3 present the test accuracy while using d-SNE and CCSA loss in conjunction with ORIENT(FLMI) on Office-31 and Office-Home datasets, respectively. It's evident that augmenting existing domain adaptation approaches with ORIENT achieves better or comparable performance to the Full training. These two observations help us answer our Q2 in the affirmative. Precise values of prediction accuracy and training time for all the experiments are provided in Appendix (A.5). Further, in Appendix Appendix (A.7) and Appendix (A.8), we analyze the effect of different subset sizes and subset sampling frequencies.

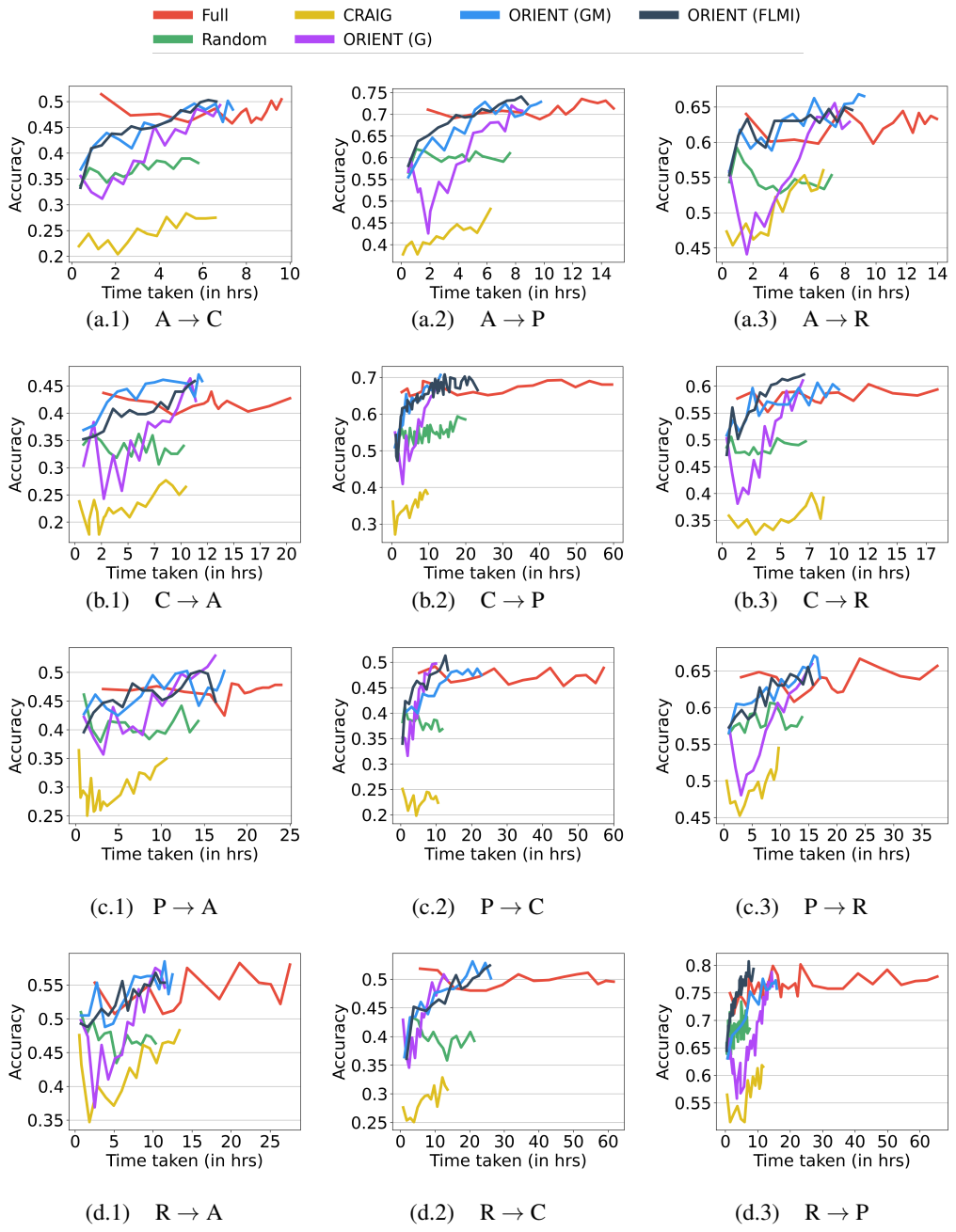
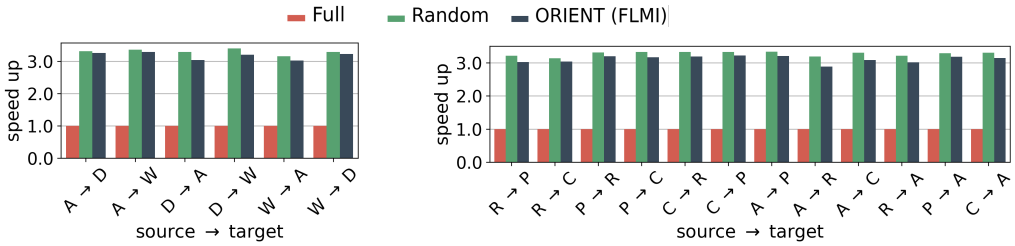


Figure 3: Convergence curves on four domains of Office-Home dataset: Art (A), Clipart (C), Product (P), and Real World (R). X-axis presents the training time in hours and Y-axis presents the prediction accuracy on the target domain.

Figure 5 presents the GradCam [46] class-activation maps of trained models on the Office-Home dataset ($P \rightarrow R$ setting) using d-SNE loss for both Full and ORIENT (FLMI). These activation maps show that the model trained with the ORIENT framework learned better class discrimination features than the model trained with Full. This might explain why the ORIENT framework performs better sometimes than Full.

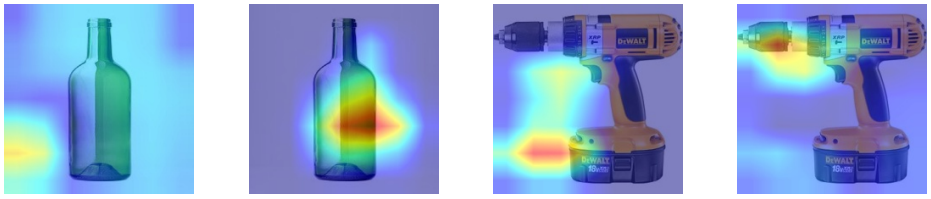
Comparison of different SMI functions: Domain adaptation results on Office31 (Figure 2) and OfficeHome (Figure 3) datasets show that ORIENT(GCMI) performs suboptimally compared to ORIENT using other SMI functions in terms of target domain accuracy. Although ORIENT(LDMI)



(a) Office-31 dataset

(b) Office-Home dataset

Figure 4: Speed up achieved by combining d-SNE with ORIENT



(a) d-SNE + Full

(b) d-SNE + ORIENT (FLMI)

(c) d-SNE + Full

(d) d-SNE + ORIENT (FLMI)

Figure 5: GradCam [46] activation maps of the models learned using d-SNE + Full and d-SNE + ORIENT (FLMI) on the Office-Home dataset with "Product" as the source domain and "Real World" as the target domain. As evidenced by class activation maps, the ORIENT framework enabled the model to learn more effective class discriminative features than Full data training.

achieves reasonable target domain accuracy, it is computationally expensive and does not achieve the best performance-speedup trade-off. In comparison to the ORIENT using remaining SMI functions, i.e., ORIENT(GM), ORIENT(G), and ORIENT(FLMI), ORIENT(FLMI) consistently achieves the best performance versus speed-up trade-off. We further present synthetic experiments to provide intuitions on how different SMI functions selects data subsets in Appendix A.6

To summarize, our experiments on two adaptation datasets suggest that our proposed approach ORIENT can substantially reduce the training time while maintaining comparable performance to training on complete source data. Additionally, our experiments using d-SNE and CCSA loss functions suggest that ORIENT can be used in conjunction with existing SDA methods to achieve significant speed ups in training time, while maintaining or improving the prediction accuracy. In particular, two instantiations of our proposed approach, ORIENT(FLMI) and ORIENT(GM) consistently achieve comparable or better performance compared to Full training while being $\sim 3\times$ faster to train.

5 Conclusion, Limitations, and Broader Impact

We introduce ORIENT, a subset selection framework based on SMI functions for supervised domain adaptation. The submodularity of SMI functions allows us to use scalable greedy algorithms to select the data subsets efficiently. In addition, we demonstrate how ORIENT is a unified framework that integrates previous approaches based on subset selection for robust learning. Empirically, we show that ORIENT is very effective for SDA. Specifically, it achieves $\sim 3\times$ speed up over existing SDA approaches like d-SNE and CCSA while achieving comparable or better performance. Our findings confirm that ORIENT has a significant social impact by making existing SDA algorithms significantly faster and more energy-efficient, reducing CO2 emissions and energy consumption incurred during training, thus contributing to a Green AI [45]. The main limitation of our work is that although ORIENT significantly reduces the training time, it requires more memory to store the similarity kernel required for subset selection. This makes running ORIENT harder on devices with low memory.

References

[1] Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000. ISSN 0378-3758.

[2] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 2017.

[3] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020.

[4] A. Axelrod, X. He, and J. Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `https://aclanthology.org/D11-1033`.

[5] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, pages 137–144. MIT Press, 2006.

[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

[7] J. A. Bilmes. Submodularity in machine learning and artificial intelligence. *CoRR*, abs/2202.00132, 2022.

[8] Z. Borsos, M. Mutný, M. Tagliasacchi, and A. Krause. Data summarization via bilevel optimization. *CoRR*, abs/2109.12534, 2021.

[9] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *ECCV Workshops (3)*, volume 9915 of *Lecture Notes in Computer Science*, pages 435–442, 2016.

[10] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE Computer Society, 2012.

[11] D. Feldman. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 23–44. Springer, 2020.

[12] S. Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

[13] Y. Ganin and V. S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org, 2015.

[14] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann. Generalisation in humans and deep neural networks. In *NeurIPS*, pages 7549–7561, 2018.

[15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587. IEEE Computer Society, 2014.

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE Computer Society, 2012.

[17] Y. Guo and M. Xiao. Cross language text classification via subspace co-regularized multi-view learning. In *ICML*. icml.cc / Omnipress, 2012.

[18] A. Gupta and R. Levin. The online submodular cover problem. In *SODA*, pages 1525–1537. SIAM, 2020.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[20] L. Hedegaard, O. A. Sheikh-Omar, and A. Iosifidis. Supervised domain adaptation: A graph embedding perspective and a rectified experimental protocol. *IEEE Trans. Image Process.*, 30: 8619–8631, 2021.

[21] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Comput. Speech Lang.*, 24(1):45–66, 2010.

[22] R. K. Iyer, N. Khargonkar, J. A. Bilmes, and H. Asnani. Generalized submodular information measures: Theoretical properties, examples, optimization algorithms, and applications. *IEEE Trans. Inf. Theory*, 68(2):752–781, 2022.

[23] V. Kaushal, R. K. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *WACV*, pages 1289–1299. IEEE, 2019.

[24] K. Killamsetty, D. Bhat, G. Ramakrishnan, and R. Iyer. CORDS: COResets and Data Subset selection for Efficient Learning, March 2021. URL `https://github.com/decile-team/cords`.

[25] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, A. De, and R. K. Iyer. GRAD-MATCH: gradient matching based data subset selection for efficient deep model training. In *ICML*, volume 139, pages 5464–5474. PMLR, 2021.

[26] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. K. Iyer. GLISTER: generalization based data subset selection for efficient and robust learning. In *AAAI*, pages 8110–8118. AAAI Press, 2021.

[27] K. Killamsetty, X. Zhao, F. Chen, and R. K. Iyer. RETRIEVE: coreset selection for efficient and robust semi-supervised learning. In *NeurIPS*, pages 14488–14501, 2021.

[28] K. Killamsetty, G. S. Abhishek, Aakriti, A. V. Evfimievski, L. Popa, G. Ramakrishnan, and R. Iyer. AUTOMATA: Gradient based data subset selection for compute-efficient hyper-parameter tuning, 2022. URL `https://arxiv.org/abs/2203.08212`.

[29] K. Kirchhoff and J. A. Bilmes. Submodularity for data selection in machine translation. In *EMNLP*, pages 131–141. ACL, 2014.

[30] P. Koniusz, Y. Tas, and F. Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *CVPR*, pages 7139–7148. IEEE Computer Society, 2017.

[31] S. Kothawade, N. Beck, K. Killamsetty, and R. K. Iyer. SIMILAR: submodular information measures based active learning in realistic scenarios. In *NeurIPS*, pages 18685–18697, 2021.

[32] S. Kothawade, V. Kaushal, G. Ramakrishnan, J. A. Bilmes, and R. K. Iyer. PRISM: a rich class of parameterized submodular information measures for guided subset selection. *AAAI*, 36, 2022.

[33] M. Liu and O. Tuzel. Coupled generative adversarial networks. In *NeurIPS*, pages 469–477, 2016.

[34] L. Lovász. Submodular functions and convexity. In *ISMP*, pages 235–257. Springer, 1982.

[35] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pages 234–243. Springer, 1978.

[36] B. Mirzasoleiman, J. A. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In *ICML*, volume 119, pages 6950–6960. PMLR, 2020.

[37] B. Mirzasoleiman, K. Cao, and J. Leskovec. Coresets for robust training of neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

[38] L. H. Morsing, O. A. Sheikh-Omar, and A. Iosifidis. Supervised domain adaptation using graph embedding. In *ICPR*, pages 7841–7847. IEEE, 2020.

[39] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *NIPS*, pages 6670–6680, 2017.

[40] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5716–5726. IEEE Computer Society, 2017.

[41] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *ICML (1)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 10–18. JMLR.org, 2013.

[42] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Process. Mag.*, 32(3):53–69, 2015.

[43] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV (4)*, volume 6314 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 2010.

[44] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019.

[45] R. Schwartz, J. Dodge, N. Smith, and O. Etzioni. Green ai. *Communications of the ACM*, 63:54 – 63, 2020.

[46] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.

[47] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*. OpenReview.net, 2018.

[48] A. Singh. CLDA: contrastive learning for semi-supervised domain adaptation. In *NeurIPS*, pages 5089–5101, 2021.

[49] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy. GCR: Gradient coreset based replay buffer selection for continual learning, 2021.

[50] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347.

[51] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076. IEEE Computer Society, 2015.

[52] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971. IEEE Computer Society, 2017.

[53] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5385–5394. IEEE Computer Society, 2017.

[54] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

[55] K. Wei, Y. Liu, K. Kirchhoff, C. D. Bartels, and J. A. Bilmes. Submodular subset selection for large-scale speech training data. In *ICASSP*, pages 3311–3315. IEEE, 2014.

[56] K. Wei, Y. Liu, K. Kirchhoff, and J. A. Bilmes. Unsupervised submodular subset selection for speech data. In *ICASSP*, pages 4107–4111. IEEE, 2014.

[57] K. Wei, R. K. Iyer, and J. A. Bilmes. Submodularity in data subset selection and active learning. In *ICML*, volume 37, pages 1954–1963. JMLR.org, 2015.

[58] X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR*, pages 2497–2506. Computer Vision Foundation / IEEE, 2019.

[59] T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, pages 2142–2150. IEEE Computer Society, 2015.

[60] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 3 for theoretical results and Section 4 for empirical evaluation of our claims.
   (b) Did you describe the limitations of your work? [Yes] We discuss limitations of our proposed approach in Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss the potential societal impact of our approach in Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We've ensured that our paper conforms to the guidelines to the best of our knowledge.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 3 for theoretical results and A.3 for details
   (b) Did you include complete proofs of all theoretical results? [Yes] See A.3.
3. If you ran experiments...
   (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] See C.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Please refer to A.5 for precise results for all our experiments.
   (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] Please refer to Section A.5
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? [Yes] Please refer to 4 dataset and model architecture citations.
   (b) Did you mention the license of the assets? [Yes] Please refer to Section D.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Please refer to Section C for our code.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Please refer to Section D for licenses for the datasets used.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Our data does not use personally identifiable information or offensive content. Please refer to Section D and Section A.5 for details regarding the data used.
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] Our work does not use research involving human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Our work does not use research involving human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Our work does not use research involving human subjects.

A Appendix

A.1 Notations

Table 4 summarize the notations used in this paper.

	Notation	Description
DATA	τ, τ^s, τ^t	domain or task
	superscript ^{s}	source domain
	superscript ^{t}	target domain
	X	Feature space
	Y	Label space
	X	Embedding space
	$P(X), P(X^s), P(X^t)$	Marginal distributions
	$P(Y\|X)$	Conditional distribution
	D	Set of data points
	A, B	Subset of D
	D^s, D^t	Datasets
SMI	f	Sub modular function
	I_f	SMI function
	S	Similarity matrix
ORIENT	$\mathcal{L}$	Loss function
	b	Batch size
	E	Total training epochs
	L	Epoch interval
SDA	g	feature extractor
	h	predictor function
	$\circ$	Composition

Table 4: Notations

A.2 Details on Concave over Modular Mutual Information

$f_\eta(A)$ is a restricted submodular function [32] defined over sets V, V' and ψ is a concave function. Let n be the size of set A.

$$\begin{aligned} f_\eta(A) =& \eta \sum_{i \in V'} \max \left(\psi \left(\sum_{j \in A \cap V} S_{ij} \right), \psi \left(\sqrt{n} \sum_{j \in A \cap V'} S_{ij} \right) \right) \\ &+ \sum_{i \in V} \max \left(\psi \left(\sum_{j \in A \cap V'} S_{ij} \right), \psi \left(\sqrt{n} \sum_{j \in A \cap V} s_{ij} \right) \right) \end{aligned} \tag{3}$$

A.3 Proof of the Technical Results

Theorem 4 *When the outer level loss of the discrete bi-level optimization problem of* GLISTER *is hinge loss, logistic loss, and perceptron loss, then the optimization problem becomes an instance of maximization of Concave over Modular (COM) SMI function.*

PROOF Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, subset $\mathcal{S}$, subset size k, and validation set $\mathcal{V}$, the objective of GLISTER can be written as follows:

$$\begin{aligned} &\min_{\mathcal{S} \subseteq \mathcal{D}, |\mathcal{D}| = k} L_V(\theta^*, \mathcal{V}) \\ &\text{where } \theta^* = \arg\min_{\theta} L_T(\theta, \mathcal{S}) \end{aligned} \tag{4}$$

Loss on the subset denoted by $L_T(\theta, \mathcal{S})$ is the summation of the losses of individual data samples in the subset. i.e., $L_T(\theta, \mathcal{S}) = \sum_{(x,y)\in\mathcal{S}} L_T(\theta, x, y)$. Using the one-step gradient approximation of GLISTER, the above optimization problem can be written as:

$$\begin{aligned}&\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} L_V(\theta - \eta\nabla_\theta L_T(\theta, \mathcal{S}), \mathcal{V})\\ &\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} L_V(\theta - \eta \sum_{(x,y)\in\mathcal{S}} \nabla_\theta L_T(\theta, x, y), \mathcal{V})\end{aligned} \tag{5}$$

where η is the learning rate.

We can convert the above minimization problem to a maximization problem as following:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} -L_V(\theta - \eta \sum_{(x,y)\in\mathcal{S}} \nabla_\theta L_T(\theta, x, y), \mathcal{V}) \tag{6}$$

Using the Proof of Theorem-1 in GLISTER [26], the optimization problem in Equation 6 for different validation losses can be written as follows:

Case 1 When L_V is hinge loss or perceptron loss, the optimization problem of GLISTER is:

$$\begin{aligned}&\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} \min(0, C_i + \sum_{j\in\mathcal{S}} \hat{g}_{ij})\\ &\text{where } \sum_{j\in\mathcal{S}} \hat{g}_{ij} \geq 0\end{aligned} \tag{7}$$

We can write the above optimization problem as,

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} |\mathcal{V}|C_i + \sum_{i=1}^{|\mathcal{V}|} \min(-C_i, \sum_{j\in\mathcal{S}} \hat{g}_{ij}) \tag{8}$$

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} \min(-C_i, \sum_{j\in\mathcal{S}} \hat{g}_{ij}) \tag{9}$$

Note that in the above equation, $\min(C, x)$ is a concave function in x and $\sum_{j\in\mathcal{S}} \hat{g}_{ij}$ is a non-negative modular function. Hence, the above formulation is submodular and is an instance of concave over modular function.

Case 2 When L_V is logistic loss, the optimization problem of GLISTER is:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} C - \log\left(1 + C_i \exp\left(\alpha \sum_{j\in\mathcal{S}} \hat{g}_{ij}\right)\right) \tag{10}$$

Note that in the above equation, $-\log(1 + C\exp -x)$ is concave in x and $\sum_{j\in\mathcal{S}} \hat{g}_{ij}$ is modular. Hence, the above formulation of set function is submodular and is an instance of concave over modular function. In our setting, we use labeled target data D^t as the validation set. Furthermore, the above given formulations of GLISTER corresponds to instantiation of maximization of COM MI function ($\eta \sum_{i\in A} \psi(\sum_{j\in D^t} S_{ij}) + \sum_{j\in D^t} \psi(\sum_{i\in A} S_{ij})$) with $\eta = 0$.

Theorem 5 *When the optimization problem of* GRADMATCH *with equal sample weights is used to match gradients of a held-out validation set, it becomes an instance of maximization of summation of GCMI and a diversity function.*

PROOF Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, and validation set $\mathcal{V}$. Let us denote loss of i^{th} sample in the training dataset as $L_T^i(\theta)$ and the loss of j^{th} sample in the validation dataset as $L_V^j(\theta)$.

The objective of GRADMATCH with equal sample weights can be written as follows:

$$\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \left\| \frac{1}{k}\sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta_t) - \frac{1}{|\mathcal{V}|}\sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta) \right\|^2 \tag{11}$$

Without the loss of generality we assumed sample weights to be 1 in the above equation.

We can convert this to a maximization problem as follows:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} -\left\| \frac{1}{k}\sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta_t) - \frac{1}{|\mathcal{V}|}\sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta) \right\|^2 \tag{12}$$

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \frac{-1}{k^2}\left\|\sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta)\right\|^2 + \frac{-1}{|\mathcal{V}|^2}\left\|\sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta_t)\right\|^2 + 2\frac{1}{k|\mathcal{V}|}\sum_{i\in\mathcal{S},j\in\mathcal{V}} \nabla_\theta L_T^i(\theta)^T\cdot\nabla_\theta L_V^j(\theta) \tag{13}$$

In the above equation second term is not dependent on $\mathcal{S}$ and can be ignored during optimization. Following which the above optimization problem can be written as:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \frac{-1}{k^2}\left\|\sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta)\right\|^2 + 2\frac{1}{k|\mathcal{V}|}\sum_{i\in\mathcal{S},j\in\mathcal{V}} \nabla_\theta L_T^i(\theta)^T\cdot\nabla_\theta L_V^j(\theta) \tag{14}$$

Note that in our setting, labeled target data D^t is used as the validation set. In the above equation, the second term corresponds to GCMI function($2\lambda \sum_{i\in A, j\in D^t} S_{ij}$) with $\lambda = 1$.

Expanding on the first term we have,

$$\frac{-1}{k^2}\left\|\sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta)\right\|^2 = \frac{-1}{k^2}\sum_{i\in\mathcal{S}}\left\|\nabla_\theta L_T^i(\theta)\right\|^2 - \frac{2}{k^2}\sum_{i,j\in\mathcal{S}|i\neq j} \nabla_\theta L_T^i(\theta)^T\cdot\nabla_\theta L_T^j(\theta) \tag{15}$$

The function given in Equation 15 is a diversity function.

Hence, the optimization problem of GRADMATCH with equal sample weights when matched with validation set is a instance of maximization of summation of the GCMI function and diversity function.

Theorem 6 *When the optimization problem of* CRAIG *is adapted to match gradients of a held-out validation set, it becomes an instance of maximization of the FLMI function with* $\eta = 0$.

PROOF Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, and validation set $\mathcal{V}$. Let us denote loss of i^{th} sample in the training dataset as $L_T^i(\theta)$ and the loss of j^{th} sample in the validation dataset as $L_V^j(\theta)$.

The objective of CRAIG matching gradients of a held-out validation set is as follows:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \sum_{i\in\mathcal{V}} \max_{j\in\mathcal{S}} \nabla_\theta L_V^i(\theta)^T\cdot\nabla_\theta L_T^j(\theta) \tag{16}$$

Note that in our setting, labeled target data D^t is used as the validation set. The set function in the above equation corresponds to FLMI function($\sum_{i\in D^t}\max_{j\in A} S_{ij} + \eta\sum_{i\in A}\max_{j\in D^t} S_{ij}$) with $\eta = 0$.

Hence, the optimization problem of CRAIG adapted to match gradients of a held-out validation set is an instance of maximization of the FLMI function with $\eta = 0$.

A.4 SDA Loss

CCSA: In CCSA, the classifier is modeled as a composition of two functions–$h \circ g$. Here, $g : X \to Z$ is a feature extractor that transforms the input from the feature space X to an embedding space Z, and $h : Z \to Y$ is a predictor function. Let X_y^s and X_y^t denote the source and the target domain samples with label $y \in Y$, respectively. The semantic alignment loss ($\mathcal{L}_{SA}$) for CCSA is defined as,

$$\mathcal{L}_{SA}(g) = \sum_{y \in Y} d\left(P\left(g\left(X_y^s\right)\right), P\left(g\left(X_y^t\right)\right)\right),$$

where $d(.)$ indicates a distance measure between the distributions of X_y^s and X_y^t in the embedding space and $P(.)$ indicates the distribution. $\mathcal{L}_{SA}$ encourages the samples from different domains and the same label to map nearby in the embedding space. The separation loss ($\mathcal{L}_S$) is defined as

$$\mathcal{L}_S(g) = \sum_{a,b \in Y | a \neq b} k\left(P\left(g\left(X_a^s\right)\right), P\left(g\left(X_b^t\right)\right)\right),$$

where k is a similarity measure that returns a higher value when the distribution of X_a^s and X_b^t is close in the embedding space. Hence, $\mathcal{L}_S$ encourages the samples from different domains and different labels to map far away in the embedding space. Overall the CCSA loss is defined as a combination of cross-entropy loss, semantic alignment loss, and separation loss,

$$\mathcal{L}_{CCSA}(h \circ g) = \mathcal{L}_{\text{CE}}(h \circ g) + \mathcal{L}_{SA}(g) + \mathcal{L}_S(g).$$

The distance measure (d) in the semantic alignment loss and the similarity measure (k) in the separation loss are computed as average of pairwise similarities and distances between all the samples from the source and the target domain, respectively. Further assumptions on d and k lends them into a well known contrastive loss function (cf. Motiian et al. [40] for details).

d-SNE: Instead of minimizing the average of distances between all pairs of samples, Xu et al. [58], in d-SNE, proposes to minimize the largest distance of the samples from different source with same label, and maximizing the smallest distance of the samples from different source and different labels. Let D_y^s denote the subset of source data with label y, then the loss function $\tilde{L}(g)$ is defined as,

$$\tilde{L}(g) = \sum_{x_j \in D^t} \left(\sup_{x \in D_{y_j}^s} \left\{ a \mid a \in d\big(g(x), g(x_j)\big) \right\} - \inf_{x \in D^s \setminus D_{y_j}^s} \left\{ b \mid b \in d\big(g(x), g(x_j)\big) \right\} \right).$$

The complete d-SNE loss is defined as a combination of $\tilde{L}$ and cross-entropy loss on source and target domain.

$$\mathcal{L}_{d\text{-SNE}}(h \circ g) = \tilde{\mathcal{L}}(g) + \alpha \mathcal{L}_{\text{CE}}^s(h \circ g) + \beta \mathcal{L}_{\text{CE}}^t(h \circ g).$$

A.5 Additional Results

Figure 6 presents the GradCam [46] class-activation maps of trained models on the Office-Home dataset ($P \to R$ setting) using d-SNE loss for both Full and ORIENT (FLMI). These activation maps show that the model trained with ORIENT framework learn effective class discriminative features more consistently than Full.

We present test accuracy of models trained using cross-entropy loss on a combination of source and target domain data, $D^s \cup D^t$, of Office-31 and Office-Home datasets in 5 and 9, respectively. We see that ORIENT(FLMI) and ORIENT(GM) perform similar to Full and outperform Random across all experiments. Tables 6 and 10 show the training times for these settings. We see that all instantiations of ORIENT except ORIENT(L) achieve $\sim 3\times$ speed-up compared to Full.

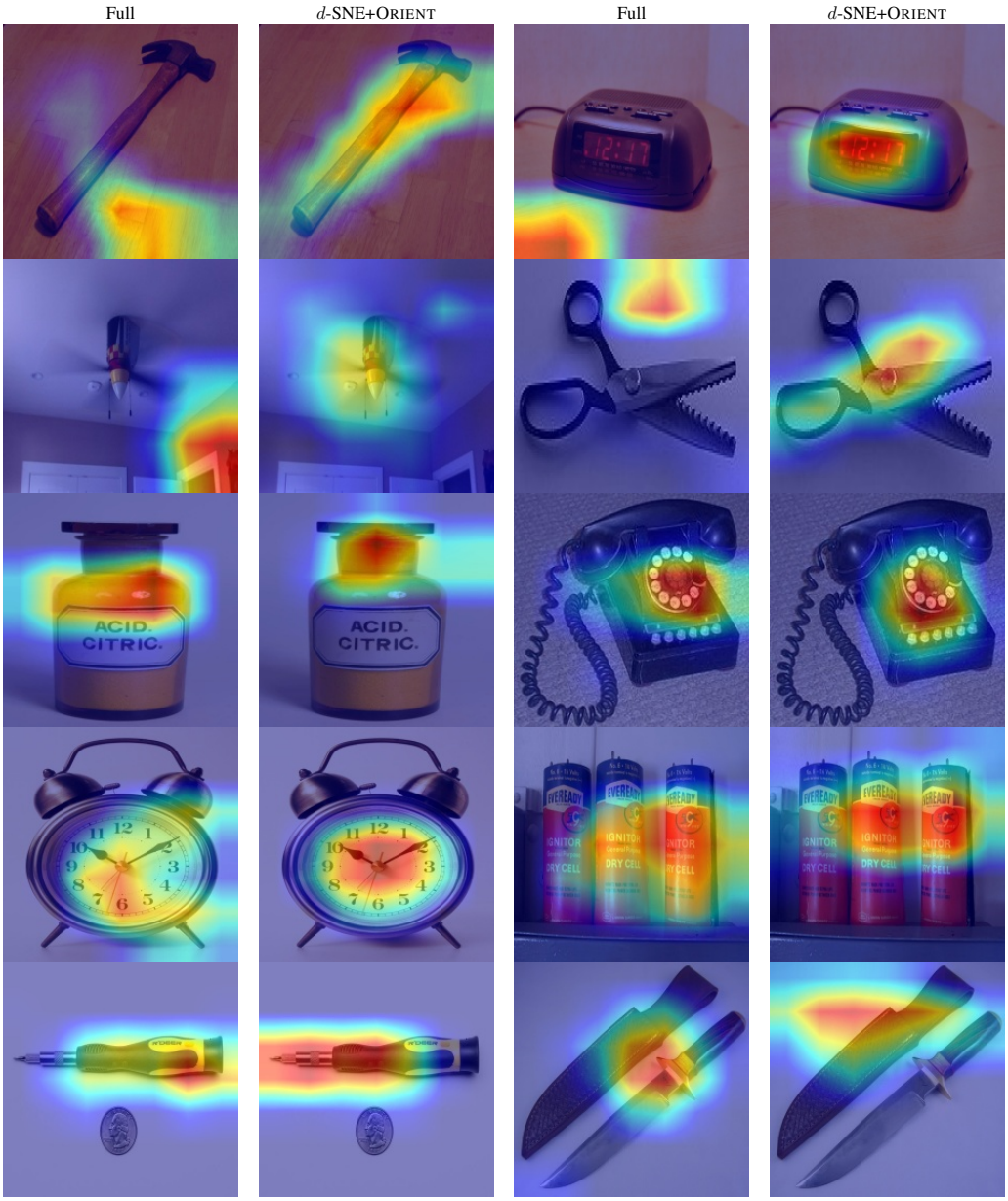


Figure 6: GradCam [46] activation maps of the models learned using d-SNE + Full and d-SNE + ORIENT (FLMI) on the Office-Home dataset with "Product" as the source domain and "Real World" as the target domain. As evidenced by class activation maps, the ORIENT framework enabled the model to learn more effective class discriminative features than Full data training consistently.

	A → D	A → W	D → A	D → W	W → A	W → D
Full	0.77 ± 0.02	0.71 ± 0.02	0.62 ± 0.01	0.95 ± 0.01	0.61 ± 0.01	0.98 ± 0.01
Random	0.63 ± 0.03	0.57 ± 0.04	0.54 ± 0.02	0.86 ± 0.02	0.5 ± 0.03	0.91 ± 0.02
ORIENT (FLMI)	0.79 ± 0.02	0.73 ± 0.01	0.62 ± 0.01	0.93 ± 0.01	0.59 ± 0	0.96 ± 0.01
ORIENT (GC)	0.52 ± 0.12	0.5 ± 0.08	0.39 ± 0.07	0.69 ± 0.11	0.42 ± 0.09	0.72 ± 0.08
ORIENT (L)	0.74 ± 0.03	0.72 ± 0.02	0.63 ± 0.01	0.93 ± 0.01	0.58 ± 0.01	0.97 ± 0.01
ORIENT (G)	0.75 ± 0.01	0.72 ± 0.01	0.61 ± 0.01	0.92 ± 0.01	0.58 ± 0.01	0.96 ± 0
ORIENT (GM)	0.77 ± 0.02	0.73 ± 0.02	0.61 ± 0.01	0.93 ± 0.01	0.61 ± 0.01	0.97 ± 0.01

Table 5: Test accuracy on Office-31 dataset.

	A → D	A → W	D → A	D → W	W → A	W → D
Full	3.13 ± 0.02	3.04 ± 0.02	1.47 ± 0.52	1.37 ± 0	1.81 ± 1.12	1.25 ± 0.01
Random	0.95 ± 0.01	0.92 ± 0	0.77 ± 0.04	0.48 ± 0.13	0.73 ± 0.01	0.39 ± 0
ORIENT (FLMI)	1.04 ± 0.09	0.82 ± 0.18	0.89 ± 0.01	0.46 ± 0.01	0.34 ± 0.01	0.43 ± 0
ORIENT (GC)	1.02 ± 0.02	0.97 ± 0	0.84 ± 0.01	0.46 ± 0	0.7 ± 0.02	0.43 ± 0.01
ORIENT (L)	1.53 ± 0	1.48 ± 0.02	0.81 ± 0.01	0.46 ± 0	0.71 ± 0.01	0.45 ± 0
ORIENT (G)	1.04 ± 0.02	0.99 ± 0	0.55 ± 0.11	0.47 ± 0	0.35 ± 0.01	0.43 ± 0
ORIENT (GM)	1.16 ± 0.03	1.08 ± 0.01	0.86 ± 0.22	0.56 ± 0.02	0.35 ± 0	0.49 ± 0.01

Table 6: Training time (in hours) for 300 epochs on Office-31 dataset

		A → D	A → W	D → A	D → W	W → A	W → D
CCSA	Full	0.78 ± 0	0.72 ± 0	0.55 ± 0	0.93 ± 0	0.55 ± 0	0.97 ± 0
	Random	0.76 ± 0.01	0.72 ± 0.01	0.54 ± 0.02	0.81 ± 0.02	0.54 ± 0.01	0.92 ± 0.02
	ORIENT (G)	0.78 ± 0.03	0.73 ± 0.02	0.55 ± 0.04	0.92 ± 0.01	0.56 ± 0	0.97 ± 0.01
	ORIENT (GM)	0.78 ± 0.01	$0.74\pm$	0.51 ± 0.04	0.88 ± 0.02	0.55 ± 0.02	0.97 ± 0.02
	ORIENT (FLMI)	0.77 ± 0	0.76 ± 0.01	0.55 ± 0	0.89 ± 0.02	0.55 ± 0.02	0.96 ± 0.01
d-SNE	Full	0.77 ± 0	0.69 ± 0	0.53 ± 0.01	$\mathbf{0.93 \pm 0}$	0.54 ± 0	0.98 ± 0
	Random	0.76 ± 0.02	0.68 ± 0.02	0.53 ± 0.02	0.86 ± 0.02	0.53 ± 0.01	0.94 ± 0.02
	ORIENT (G)	0.73 ± 0	0.69 ± 0.01	0.52 ± 0.02	$\mathbf{0.93 \pm 0.01}$	0.54 ± 0	0.97 ± 0.01
	ORIENT (GM)	0.76 ± 0.01	$0.69\pm$	0.52 ± 0.04	0.89 ± 0.02	0.54 ± 0.02	0.97 ± 0.01
	ORIENT (FLMI)	0.78 ± 0	0.71 ± 0.01	$\mathbf{0.55 \pm 0}$	0.90 ± 0.02	0.56 ± 0.02	0.97 ± 0.01

Table 7: Test accuracy on Office-31 dataset with SDA methods

		A → D	A → W	D → A	D → W	W → A	W → D
CCSA	Full	24.11 ± 0.02	27.81 ± 1.0	7.16 ± 0.1	7.84 ± 0.01	10.46 ± 0.01	10.54 ± 1.15
	Random	14.71 ± 2.5	14.64 ± 0.19	3.37 ± 0.02	2.64 ± 0.39	4.11 ± 0.08	3.25 ± 0.77
	ORIENT (G)	16.59 ± 0.42	15.53 ± 1.2	3.63 ± 0	3.45 ± 0.48	4.53 ± 0.21	4.14 ± 0.77
	ORIENT (GM)	17.11 ± 1.11	$15.6\pm$	3.14 ± 0.95	4.1 ± 0.34	5.33 ± 0.6	5.21 ± 1.29
	ORIENT (FLMI)	13.93 ± 8.85	13.24 ± 8.47	3.79 ± 0.65	3.53 ± 1.3	5.34 ± 1.27	5.21 ± 1.25
d-SNE	Full	11.57 ± 0.02	8.33 ± 0.5	2.01 ± 0.1	2.21 ± 0.1	2.18 ± 0.1	3.42 ± 0.2
	Random	3.49 ± 0.05	2.48 ± 0.01	0.61 ± 0.02	0.65 ± 0.013	0.69 ± 0.01	1.04 ± 0.02
	ORIENT (G)	3.53 ± 0.05	2.5072 ± 0.02	0.66 ± 0.05	0.69 ± 0.01	0.72 ± 0.01	1.06 ± 0.01
	ORIENT (GM)	3.56 ± 0.01	2.56 ± 0.01	0.68 ± 0.02	0.71 ± 0.03	0.73 ± 0.03	1.10 ± 0.02
	ORIENT (FLMI)	3.55 ± 0.01	2.53 ± 0.04	0.66 ± 0.01	0.69 ± 0.01	0.72 ± 0.02	1.06 ± 0.02

Table 8: Training time in hours on Office-31 with SDA methods

	R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
Full	0.79	0.5	0.62	**0.49**	0.59	0.68	0.71	0.63	**0.5**	**0.58**	0.48	0.43
Random	0.69	0.39	0.58	0.37	0.5	0.58	0.61	0.55	0.38	0.46	0.42	0.34
ORIENT (G)	0.77	0.5	**0.66**	0.48	0.61	0.67	0.71	0.63	0.49	0.55	**0.53**	0.42
ORIENT (GM)	0.76	0.5	0.64	0.47	0.59	0.71	**0.73**	**0.67**	0.48	0.57	0.5	**0.46**
ORIENT (FLMI)	0.79	**0.52**	0.63	0.48	**0.62**	**0.69**	0.72	0.65	**0.5**	0.55	0.45	**0.46**

Table 9: Test accuracy on Office-Home dataset

	R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
Full	44.28	61.58	37.72	57.2	18.47	59.6	14.86	13.96	9.61	27.55	23.97	20.32
Random	6.92	21.34	13.8	11.72	7.16	17.5	7.6	7.11	5.79	10.32	14.3	10.27
ORIENT (G)	14.41	13.65	15.57	9.95	6.93	12.61	8.46	8.29	6.8	11.49	16.28	11.42
ORIENT (GM)	15.88	26.15	17	22.78	10.02	13.18	9.77	9.23	7.38	12.48	17.34	12.02
ORIENT (FLMI)	7.78	25.84	15.73	13.33	7.02	19.97	8.85	8.45	6.62	11.47	16.33	11.31

Table 10: Training time(in hours) for 300 epochs on Office-Home dataset

		R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
CCSA	Full	0.74	**0.55**	0.62	0.46	0.56	0.67	0.70	0.64	0.48	0.57	0.49	0.41
	Random	0.71	0.47	0.62	0.45	0.54	0.66	0.69	0.57	0.48	0.5	0.47	0.45
	ORIENT(G)	**0.78**	0.54	0.63	**0.5**	0.59	0.7	**0.72**	0.62	0.5	0.57	0.49	**0.5**
	ORIENT(GM)	0.75	0.5	0.63	0.48	0.59	0.69	0.69	0.63	0.49	0.51	0.5	0.46
	ORIENT(FLMI)	**0.78**	**0.54**	**0.65**	**0.5**	**0.61**	**0.71**	0.71	**0.65**	**0.51**	**0.59**	**0.54**	0.47
d-SNE	Full	**0.77**	**0.53**	**0.62**	**0.50**	**0.60**	**0.71**	**0.72**	**0.63**	0.49	0.52	0.44	0.40
	Random	0.75	0.50	0.60	0.45	0.57	0.69	0.68	0.59	0.49	0.46	0.43	0.40
	ORIENT(G)	0.75	0.51	0.62	0.49	0.59	0.7	**0.72**	0.62	0.5	**0.54**	0.44	0.41
	ORIENT(GM)	0.76	**0.52**	0.62	**0.50**	0.59	0.70	0.71	0.61	0.49	0.51	**0.46**	**0.42**
	ORIENT(FLMI)	**0.77**	**0.52**	**0.63**	**0.50**	**0.60**	**0.71**	**0.71**	0.61	**0.51**	0.52	0.44	**0.42**

Table 11: Test accuracy on Office-Home with SDA methods

		R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
CCSA	Full	17.73	17.53	18.23	14.93	17.25	16.42	10.32	9.45	16.13	17.67	8.29	11.35
	Random	5.57	5.17	4.74	4.87	5.57	4.62	3.48	3.12	4.25	5.88	2.52	3.45
	ORIENT (G)	6.01	6.94	5.78	5.15	6.61	5.92	3.71	3.41	4.61	6.13	2.87	4.12
	ORIENT (GM)	5.73	5.69	5.15	5.39	6.12	5.34	3.78	3.85	4.66	6.27	3.17	3.95
	ORIENT (FLMI)	6.12	6.13	5.73	6.17	5.56	5.99	3.94	3.34	4.89	6.72	2.78	4.23
d-SNE	Full	16.84	16.68	17.42	9.55	17.05	9.60	6.64	9.94	6.29	18.81	11.69	11.27
	Random	5.23	5.31	5.26	2.87	5.12	2.88	1.99	3.11	1.90	5.85	3.55	3.41
	ORIENT (G)	5.48	5.61	5.42	2.98	5.32	2.96	2.11	3.48	1.95	6.21	3.63	3.53
	ORIENT (GM)	6.08	6.01	5.50	3.09	5.51	3.12	2.17	3.65	2.08	6.62	3.81	3.93
	ORIENT (FLMI)	5.56	5.48	5.44	3.01	5.34	2.98	2.07	3.44	2.04	6.23	3.67	3.58

Table 12: Training time(in hours) on Office-Home with SDA methods

We present test accuracy of models trained using SDA loss on source domain data, D^s, of Office-31 dataset in 7. Again, we see that all instantiations of ORIENT perform similar to Full while outperforming Random. Table 8 shows the training times for this setting. Again, we see that all instantiations of ORIENT achieve $\sim 2.5\times$ speed-up compared to Full.

Finally, Table 11 presents the test accuracy of models trained using SDA loss on source domain data, D^s, of Office-Home dataset. Here, we see that instantiations of ORIENT perform substantially better than Full and Random in some experiments and perform similar to Full in the rest. In particular, ORIENT(FLMI) outperforms Full in the settings of $R \rightarrow P$, $C \rightarrow R$, $C \rightarrow P$ and $P \rightarrow A$. Table 12 presents the corresponding training times. Once again, we see that instantiations of ORIENT achieve $\sim 3\times$ speed-up compared to Full.

A.6 Synthetic Experiments

To provide better intuition into how different SMI functions select data subsets, we present a comparison of the subset selected for two toy datasets in Figures 7 and 8. Figure 7a and 8a presents the synthetic dataset with 5000 samples in the source domain. The target domain consists of the 500 data points highlighted with a different color. The query set is of size 50. We present the subset selected by ORIENT with Facility Location Mutual Information (**ORIENT (FLMI)**), GradMatch (**ORIENT (GM)**), GLISTER (**ORIENT (G)**), Log Determinant Mutual Information (**ORIENT**

(LDMI)), and Graph Cut Mutual Information (**ORIENT (GCMI)**). From the results, it is evident that the FLMI function selects sample sources closer to the target domain than other SMI functions. In contrast, the GM function selects representative samples from the source domain. Our results show that the G function selects samples near the decision boundaries of the source domain data. The GCMI function selects samples from a source domain that are very similar and clustered together. In synthetic results, we found that the LDMI function tends to prioritize the selection of data samples from certain classes compared to others.

A.7 Analysis of Data Subset Size

Table 13 presents the target domain accuracies achieved using different subset sizes on the office-31 dataset using d-SNE loss, and Table 14 presents the training times taken by using different subset sizes on the office-31 dataset using d-SNE loss. As seen, on reducing the subset size from 0.3 fraction to 0.1, model experiences loss of accuracy but gains in terms of training time. Hence, there is a trade-off between training time and accuracy of the model. Higher fraction of the data would lead to better performance in accuracy but also require more training time. Where as, lower fraction would require less training time but might result in lower accuracy.

	Fraction	A $\rightarrow$ D	A $\rightarrow$ W	D $\rightarrow$ A	D $\rightarrow$ W	W $\rightarrow$ A	W $\rightarrow$ D
Full	1.0	0.77	0.69	0.53	0.93	0.54	0.98
Random	0.1	0.65	0.58	0.43	0.62	0.48	0.72
Random	0.3	0.76	0.68	0.53	0.86	0.53	0.94
ORIENT(FLMI)	0.1	0.70	0.67	0.50	0.77	0.48	0.92
ORIENT(FLMI)	0.3	0.78	0.71	0.55	0.90	0.56	0.97
ORIENT(G)	0.1	0.75	0.60	0.42	0.70	0.48	0.89
ORIENT(G)	0.3	0.76	0.66	0.50	0.87	0.52	0.96

Table 13: Comparison of test accuracy for office-31 with d-SNE loss function and different fractions of subset selection.

	Fraction	A $\rightarrow$ D	A $\rightarrow$ W	D $\rightarrow$ A	D $\rightarrow$ W	W $\rightarrow$ A	W $\rightarrow$ D
Full	1.0	11.57	8.33	2.01	2.21	2.18	3.42
Random	0.1	1.18	0.83	0.20	0.22	0.23	0.34
Random	0.3	3.49	2.48	0.61	0.65	0.69	1.04
ORIENT (FLMI)	0.1	1.25	0.89	0.25	0.26	0.26	0.38
ORIENT (FLMI)	0.3	3.55	2.53	0.66	0.69	0.72	1.06
ORIENT (G)	0.1	1.21	0.87	0.25	0.26	0.25	0.37
ORIENT (G)	0.3	3.53	2.51	0.66	0.69	0.72	1.06

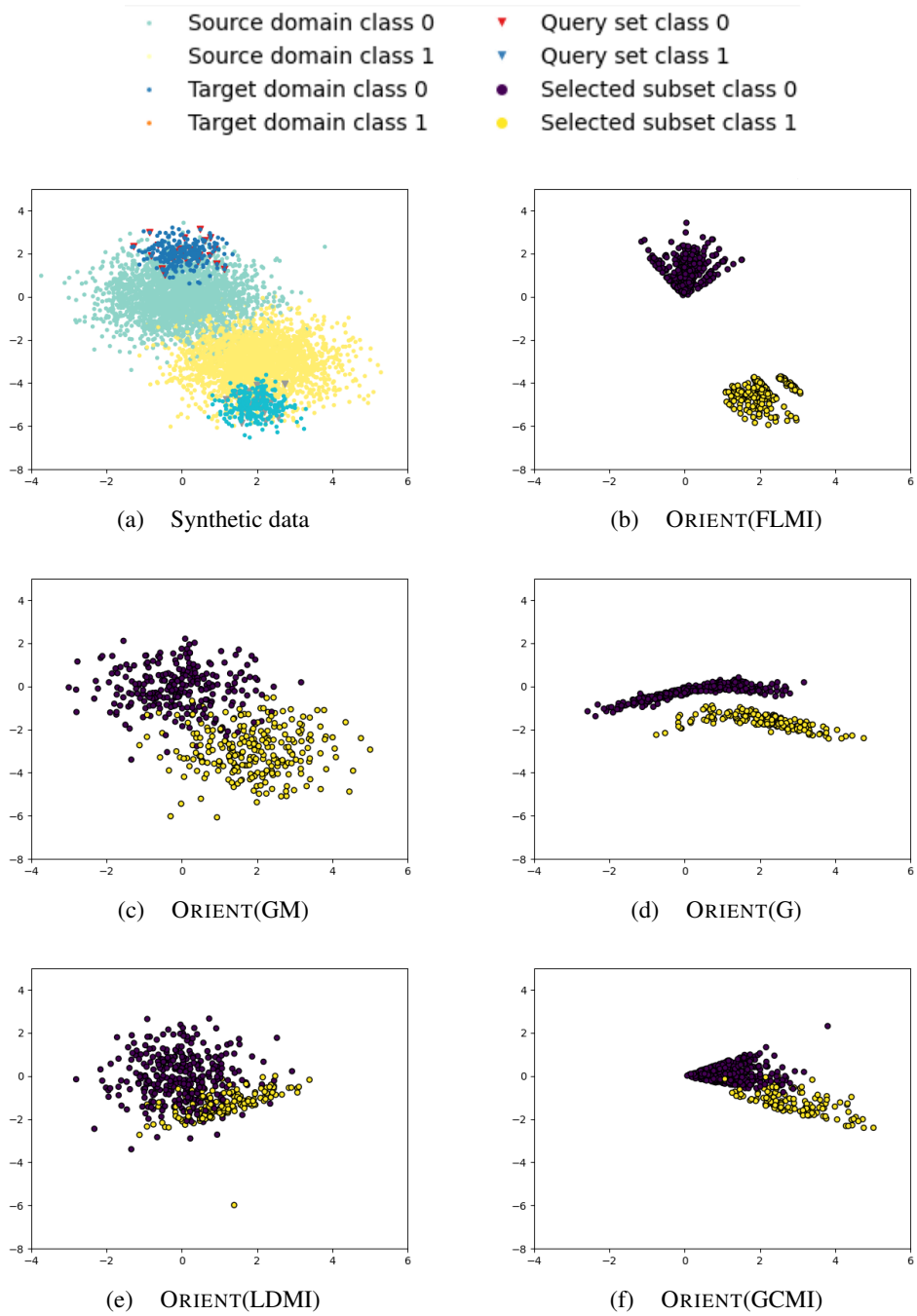
Table 14: Training time in hours on Office-31 with d-SNE loss function and different fractions of subset selection.

A.8 Analysis of L for subset selection

We present comparison of target domain accuracy achieved by ORIENT (FLMI) for different L values of 5, 10, 20, 40 on Office 31 (A->D) using d-SNE loss and 30% subset in Table 15. Note that smaller the value of L is, greater the frequency of subset selection. Results demonstrate that using $L = 5, 10$ (i.e., more frequent subset selection) results in higher training time with no improvement in accuracy. Whereas using $L = 40$(i.e., less frequent subset selection) results in lower target domain accuracy with not much significant improvement in training time. Hence, we used $L = 20$ in our experiments.

A.9 Analysis of time taken

Additionally, we present the convergence curve of training time against the validation loss in Fig. 9. As different methods use different losses, the absolute values of loss are not directly comparable. But we still like to present these plots to show that even though the Full starts with reasonable



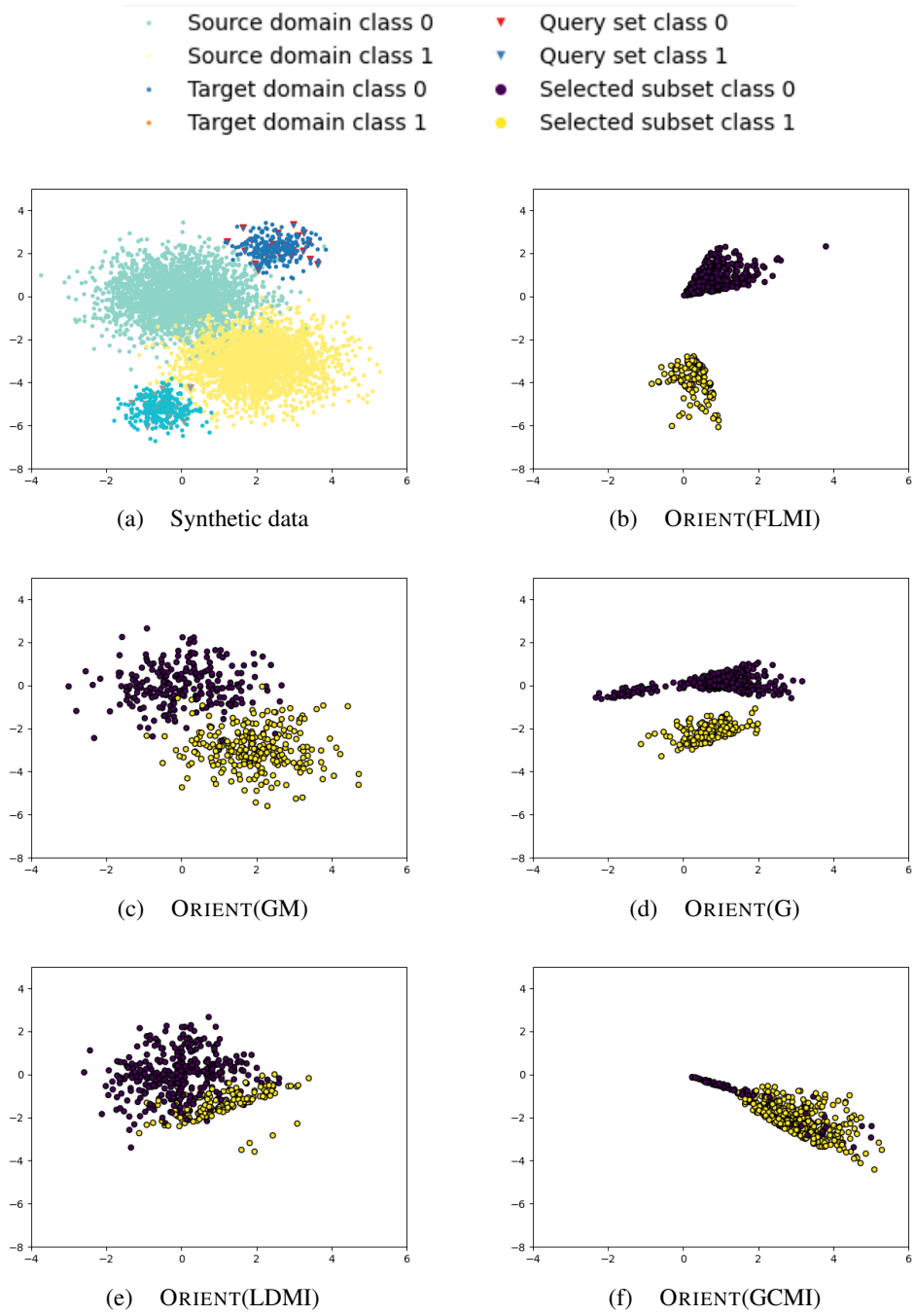
(a) Synthetic data (b) ORIENT(FLMI)

(c) ORIENT(GM) (d) ORIENT(G)

(e) ORIENT(LDMI) (f) ORIENT(GCMI)

Figure 7: Subsets selected by different instantiations of ORIENT on a synthetic dataset. (a) Synthetic data - We sample 50 examples from the target distributions for the query set. (b) ORIENT (FLMI) selects samples close to the target distribution. (c) ORIENT (GM) selects representative samples from the source domain. (d) ORIENT (G) selects samples near the decision boundaries of the source domains. (e) ORIENT(LDMI) prioritizes selection of data samples from certain classes over others (f) ORIENT(GCMI) selects samples from a source domain that are very similar and clustered together.

performance in terms of accuracy (in Fig. **??**) it does not start with lowest validation loss, and multiple training epochs are necessary before it converges. The plots show that the training time required by the ORIENT methods to converge is consistently lower than the training time required by Full to converge.



(a) Synthetic data (b) ORIENT(FLMI)

(c) ORIENT(GM) (d) ORIENT(G)

(e) ORIENT(LDMI) (f) ORIENT(GCMI)

Figure 8: Subsets selected by different instantiations of ORIENT on a synthetic dataset. (a) Synthetic data - We sample 50 examples from the target distributions for the query set. Target domain is skewed to the right for class 0 and left for class 1 as compared to the source domain. (b) ORIENT (FLMI) selects samples close to the target distribution. (c) ORIENT (GM) selects representative samples from the source domain. (d) ORIENT (G) selects samples near the decision boundaries of the source domains. (e) ORIENT(LDMI) prioritizes selection of data samples from certain classes over others (f) ORIENT(GCMI) selects samples from a source domain that are very similar and clustered together.

Tables 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, and 27 present ratio of time taken by subset selection methods with respect to Full training to achieve test accuracy in the range of $(0.3, 0.8)$ with increments of 0.05. We could see that ORIENT(FLMI) achieves a performance threshold faster

Method	Epoch Interval (L)	Target domain accuracy	Time Taken(in hrs)	
ORIENT(FLMI)	5	0.78	3.75	
ORIENT(FLMI)	10	0.78	3.61	
ORIENT(FLMI)	20	0.78	3.55	
ORIENT(FLMI)	40	0.77	3.51	

Table 15: Table showing target domain accuracy and training time taken(in hrs) achieved on office-31 (A -> D) using d-SNE loss function and 30% subset.

Table 16: Setting: R $\rightarrow$ P

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
Random	6.31	6.31	6.31	6.31	6.31	6.31	6.31	3.13	0.51	0.0	0.0
CRAIG	4.78	4.78	4.78	4.78	4.78	4.78	0.27	0.0	0.0	0.0	0.0
ORIENT (FLMI)	6.17	6.17	6.17	6.17	6.17	6.17	6.17	6.17	1.92	6.21	3.21
ORIENT (G)	2.49	2.49	2.49	2.49	2.49	2.49	2.49	2.49	0.22	2.01	0.0
ORIENT (GM)	3.93	3.93	3.93	3.93	3.93	3.93	3.93	1.29	0.43	3.53	0.0

Table 17: Setting: R $\rightarrow$ C

Accuracy threshold	0.3	0.35	0.4	0.45	0.5
Random	3.22	3.22	1.62	0.0	0.0
CRAIG	0.58	0.0	0.0	0.0	0.0
ORIENT (FLMI)	3.18	3.18	1.48	1.48	0.35
ORIENT (G)	7.58	7.58	7.58	0.72	0.45
ORIENT (GM)	4.83	4.83	2.04	0.97	0.3

Table 18: Setting: R $\rightarrow$ A

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55
Random	3.38	3.38	3.38	3.38	3.38	0.0
CRAIG	4.7	4.7	4.7	4.7	0.0	0.0
ORIENT (FLMI)	3.42	3.42	3.42	3.42	0.75	0.42
ORIENT (G)	3.4	3.4	3.4	3.4	0.31	0.26
ORIENT (GM)	3.35	3.35	3.35	3.35	3.35	0.9

Speedups achieved by different subset selection strategies w.r.t Full training to reach different accuracy thresholds for different combinations using R as source domain on the Officehome dataset in the augmented setting.

than Full in 70 out of 82 cases, ORIENT(GM) achieves a performance threshold faster than Full in 65 out of 82 cases, ORIENT(G) achieves a performance threshold faster than Full in 60 out of 82 cases, whereas CRAIG achieves a performance threshold faster than Full in 27 out of 82 cases and Random achieves a performance threshold faster than Full in 56 out of 82 cases consisting only of lower accuracy thresholds. Furthermore, Random always fails to achieve similar accuracy to Full. From this, it is evident that ORIENTachieves faster performance thresholds than FULL in most cases.

Table 28 presents the ratio of time taken by subset selection methods with respect to Full training to achieve validation loss withing 105% of the minimum validation loss. Even with this impractical stopping criterion we see average speedups of 2.26, 2.37 and 1.85 for ORIENT(FLMI), ORIENT(G) and ORIENT(GM), respectively. Similarly, speedups achieved in reaching $1.1\times$ minimum validation loss for ORIENT(FLMI), ORIENT(G) and ORIENT(GM) are 2.28, 2.38 and 1.85, respectively.

B Experiment Details

We use a common training process and hyperparameters for all our experiments. We use the following hyperparameters:

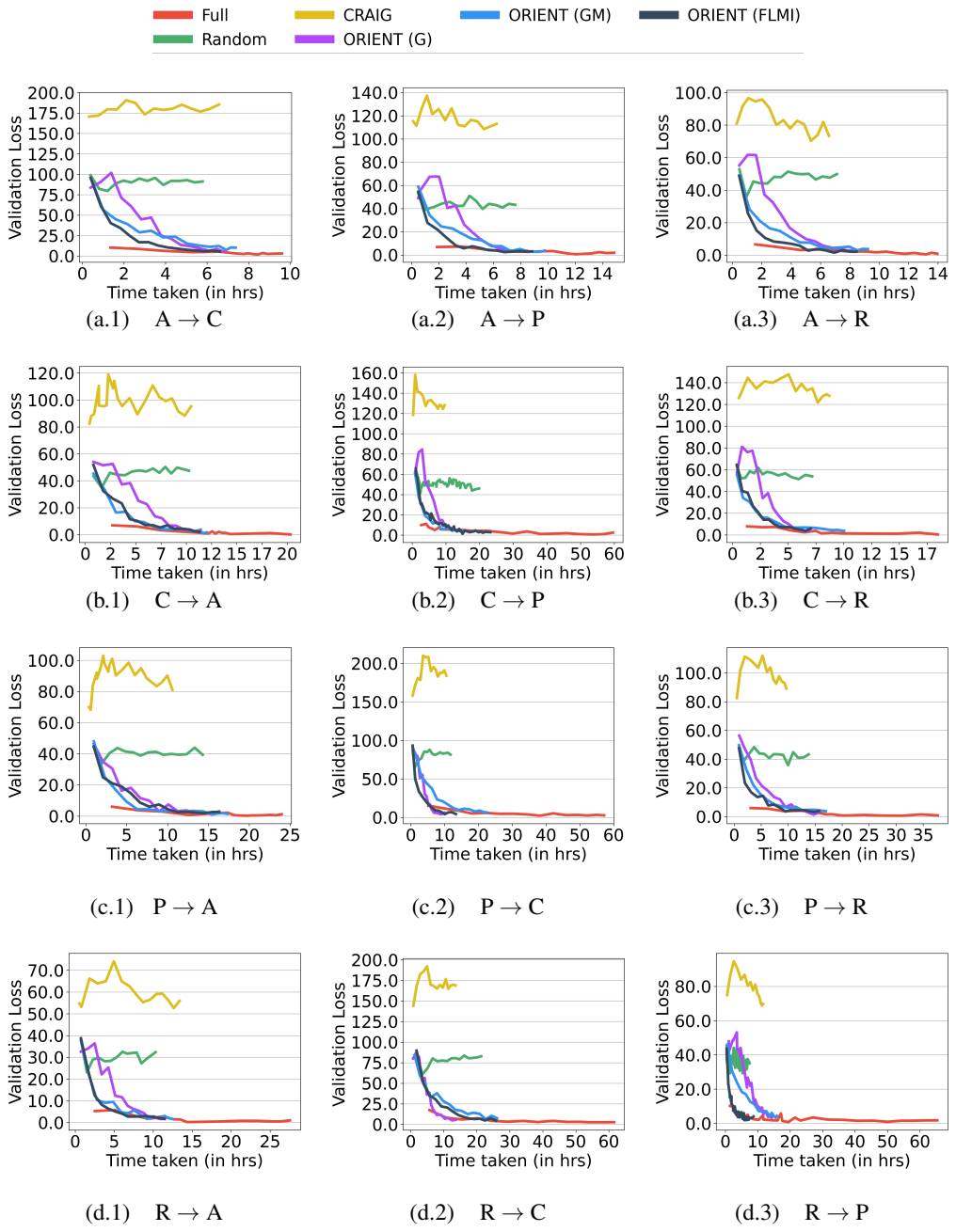


Figure 9: Convergence curves on four domains of Office-Home dataset: Art (A), Clipart (C), Product (P), and Real World (R). X-axis presents the training time in hours and Y-axis presents the Validation loss on the target domain.

- **Optimization Algorithm:** SGD with Momentum
- **Learning Rate:** 0.001 with Cosine Annealing
- **Momentum:** 0.9
- **Weight Decay:** 5e-4
- **SMI query diversity hyperparameter(η):** 1
- **Number of epochs:** 300

Table 19: Setting: P $\rightarrow$ R

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Random	3.31	3.31	3.31	3.31	3.31	3.31	0.37	0.0
CRAIG	6.24	6.24	6.24	6.24	0.39	0.0	0.0	0.0
ORIENT (FLMI)	3.38	3.38	3.38	3.38	3.38	3.38	0.48	1.62
ORIENT (G)	3.25	3.25	3.25	3.25	3.25	3.25	0.32	1.54
ORIENT (GM)	3.39	3.39	3.39	3.39	3.39	3.39	1.38	1.72

Table 20: Setting: P $\rightarrow$ C

Accuracy threshold	0.3	0.35	0.4	0.45	0.5
Random	9.46	9.46	4.47	0.0	0.0
CRAIG	0.0	0.0	0.0	0.0	0.0
ORIENT (FLMI)	9.72	4.02	4.02	1.48	Improved
ORIENT (G)	8.88	8.88	1.27	0.91	0.0
ORIENT (GM)	4.02	4.02	4.02	0.47	0.0

Table 21: Setting: P $\rightarrow$ A

Accuracy threshold	0.3	0.35	0.4	0.45	0.5
Random	3.31	3.31	3.31	3.31	0.0
CRAIG	8.32	8.32	0.0	0.0	0.0
ORIENT (FLMI)	3.34	3.34	1.54	0.74	Improved
ORIENT (G)	3.3	3.3	3.3	0.36	Improved
ORIENT (GM)	3.28	3.28	3.28	1.39	Improved

Speedups achieved by different subset selection strategies w.r.t Full training to reach different accuracy thresholds for different combinations using P as source domain on the Officehome dataset in the augmented setting.

We use a single `GTX 1080 Ti` GPU for experiments.

C Code

The anonymized code of ORIENT is available at the following link: `https://anonymous.4open.science/r/6845_Orient`

D Licenses

We release the code repository of ORIENT with MIT license, and it is available for everybody to use freely. We use the CORDS—COResets and Data Subset selection—library [24] for the implementation of ORIENT framework and SubModLib—Submodular optimization library—for the SMI optimization. The datasets, Office-31 and Office-Home, are made publicly available by the original authors. Office-Home dataset is made available under a custom license for non-commercial research and educational purposes. Office-Home dataset contains copyrighted material and is made available in accordance with Title 17 U.S.C. Section 107 of the US Copyright Law. In addition, the datasets used do not contain any personally identifiable information.

Table 22: Setting: C $\rightarrow$ R

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6
Random	3.38	3.38	3.38	3.38	1.69	0.0	0.0
CRAIG	2.26	2.26	0.17	0.0	0.0	0.0	0.0
ORIENT (FLMI)	3.33	3.33	3.33	3.33	1.49	1.49	2.99
ORIENT (G)	3.29	3.29	3.29	3.29	3.29	0.24	1.81
ORIENT (GM)	3.29	3.29	3.29	3.29	3.29	0.5	1.54

Table 23: Setting: C $\rightarrow$ P

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Random	2.22	2.22	2.22	2.22	2.22	0.44	0.0	0.0	0.0
CRAIG	9.98	9.98	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ORIENT (FLMI)	2.46	2.46	2.46	2.46	2.46	1.1	0.7	0.29	Improved
ORIENT (G)	3.09	3.09	3.09	3.09	3.09	0.37	0.3	0.24	0.0
ORIENT (GM)	3.2	3.2	3.2	3.2	3.2	1.34	0.69	0.69	Improved

Table 24: Setting: C $\rightarrow$ A

Accuracy threshold	0.3	0.35	0.4	0.45
Random	3.34	1.66	0.0	0.0
CRAIG	0.0	0.0	0.0	0.0
ORIENT (FLMI)	3.38	3.38	0.75	Improved
ORIENT (G)	3.33	1.52	0.28	Improved
ORIENT (GM)	3.34	3.34	0.89	Improved

Speedups achieved by different subset selection strategies w.r.t Full training to reach different accuracy thresholds for different combinations using C as source domain on the Officehome dataset in the augmented setting.

Table 25: Setting: A $\rightarrow$ R

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Random	3.28	3.28	3.28	3.28	3.28	1.64	0.0	0.0
CRAIG	5.26	5.26	5.26	5.26	0.45	0.29	0.0	0.0
ORIENT (FLMI)	3.34	3.34	3.34	3.34	3.34	3.34	1.48	0.0
ORIENT (G)	3.34	3.34	3.34	3.34	3.34	3.34	0.28	Improved
ORIENT (GM)	3.29	3.29	3.29	3.29	3.29	3.29	1.34	Improved

Table 26: Setting: A $\rightarrow$ P

Accuracy threshold	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Random	3.77	3.77	3.77	3.77	3.77	3.77	1.8	0.0	0.0
CRAIG	15.32	15.32	2.53	0.32	0.0	0.0	0.0	0.0	0.0
ORIENT (FLMI)	3.77	3.77	3.77	3.77	3.77	3.77	1.6	0.98	0.33
ORIENT (G)	3.76	3.76	3.76	3.76	3.76	3.76	0.37	0.37	0.24
ORIENT (GM)	3.74	3.74	3.74	3.74	3.74	3.74	1.41	0.5	0.36

Table 27: Setting: A $\rightarrow$ C

Accuracy threshold	0.3	0.35	0.4	0.45	0.5
Random	3.4	1.67	0.0	0.0	0.0
CRAIG	0.0	0.0	0.0	0.0	0.0
ORIENT (FLMI)	3.39	1.52	1.52	0.5	0.22
ORIENT (G)	3.36	3.36	0.36	0.36	0.0
ORIENT (GM)	3.32	3.32	1.35	0.41	0.19

Speedups achieved by different subset selection strategies w.r.t Full training to reach different accuracy thresholds for different combinations using A as source domain on the Officehome dataset in the augmented setting.

	C → P	C → R	R → A	P → R	R → P	A → C	C → A	R → C	P → A	A → P	A → R	P → C	Average speedup
ORIENT (FLMI)	3.58	2.81	1.25	2.1	3.1	1.28	1.8	2.41	1.36	1.77	1.9	3.8	2.26
ORIENT (G)	4.46	2.86	1.32	2.12	1.36	1.24	1.86	4.33	1.28	1.56	1.69	4.3	2.37
ORIENT (GM)	4.49	1.84	1.56	1.83	1.25	1.22	1.74	2.39	1.27	1.34	1.55	1.68	1.85

Table 28: Speed ups achieved by different variants of ORIENT when using $1.05\times$ the minimum validation loss as a stopping criterion for training.