UNIVERSAL SOURCE-FREE DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

There is a strong incentive to develop versatile learning techniques that can transfer the knowledge of class-separability from a labeled source domain to an unlabeled target domain in the presence of a domain-shift. Existing domain adaptation (DA) approaches are not equipped for practical DA scenarios as a result of their reliance on the knowledge of source-target label-set relationship (e.g. Closed-set, Open-set or Partial DA). Furthermore, almost all the prior unsupervised DA works require coexistence of source and target samples even during deployment, making them unsuitable for incremental, real-time adaptation. Devoid of such highly impractical assumptions, we propose a novel two-stage learning process. Initially, in the *procurement-stage*, the objective is to equip the model for future *source-free* deployment, assuming no prior knowledge of the upcoming *category-gap* and *domain-shift*. To achieve this, we enhance the model's ability to reject out-of-source distribution samples by leveraging the available source data, in a novel generative classifier framework. Subsequently, in the *deployment-stage*, the objective is to design a unified adaptation algorithm capable of operating across a wide range of *category-gaps*, with no access to the previously seen source samples. To achieve this, in contrast to the usage of complex adversarial training regimes, we define a simple yet effective *source-free* adaptation objective by utilizing a novel instance-level weighing mechanism, named as Source Similarity Metric (SSM). A thorough evaluation shows the practical usability of the proposed learning framework with superior DA performance even over state-of-the-art *source-dependent* approaches.

1 INTRODUCTION

Deep learning models have proven to be highly successful over a wide variety of tasks (Krizhevsky et al., 2012; Ren et al., 2015). However, a majority of these remain heavily dependent on access to a huge amount of labeled samples to achieve a reliable level of generalization. A recognition model trained on a certain distribution of labeled samples (source domain) often fails to generalize (Chen et al., 2017) when deployed in a new environment (target domain) in the presence a discrepancy in the input distribution (Shimodaira, 2000). Domain adaptation (DA) algorithms seek to minimize this discrepancy either by learning a domain invariant feature representation (Long et al., 2015; Kumar et al., 2018; Ganin et al., 2016; Tzeng et al., 2015), or by learning independent domain transformations (Long et al., 2016) to a common latent representation through adversarial distribution matching (Tzeng et al., 2017; Nath Kundu et al., 2018), in the absence of target label information.

Most of the existing approaches (Zhang et al., 2018c; Tzeng et al., 2017) assume a common label-set shared between the source and target domains (*i.e.* $\mathcal{C}_s = \mathcal{C}_t$), which is often regarded as *Closed-Set DA* (see Fig. 1). Though this assumption helps to analyze various insights of DA algorithms, such an assumption rarely holds true in real-world scenarios. Recently researchers have independently explored two broad adaptation settings by partly relaxing the above assumption. In the first kind, *Partial DA* (Zhang et al., 2018b; Cao et al., 2018a;b), the target label space is considered as a subset of the source label space (*i.e.* $\mathcal{C}_t \subset \mathcal{C}_s$). This setting is more suited for large-scale universal source datasets, which will almost always subsume the label-set of a wide range of target domains. However, the availability of such a universal source is highly questionable for a wide range of input domains and tasks. In the second kind, regarded as *Open-set DA* (Baktashmotlagh et al., 2019; Ge et al., 2017), the target label space is considered as a superset of the source label space (*i.e.* $\mathcal{C}_t \supset \mathcal{C}_s$). The major challenge in this setting is attributed to detection of target samples from the unobserved categories in a fully-unsupervised scenario. Apart from the above two extremes, certain works define a partly mixed scenario by allowing "*private*" label-set for both source and target domains (*i.e.* $\mathcal{C}_s \setminus \mathcal{C}_t \neq \emptyset$

and $\mathcal{C}_t \setminus \mathcal{C}_s \neq \emptyset$) but with extra supervision such as few-shot labeled data (Luo et al., 2017) or access to the knowledge of common categories (Panareda Busto & Gall, 2017).

Most of the prior approaches consider each scenario in isolation and propose independent solutions. Thus, they require access to the knowledge of label-set relationship (or *category-gap*) to carefully choose a DA algorithm, which would be suitable for the problem in hand. Furthermore, all the prior unsupervised DA works require coexistence of source and target samples even during deployment, hence not *source-free*. This is highly impractical, as labeled source data may not be accessible after deployment due to several reasons such as, privacy concerns, restricted access to proprietary data, accidental loss of source data or other computational limitations in real-time deployment scenarios.

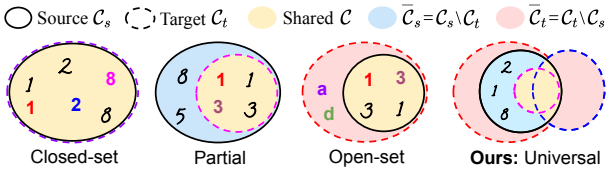

Figure 1: Various label-set relationships (*category-gap*).

Acknowledging the aforementioned shortcomings, we propose one of the most convenient DA frameworks which is ingeniously equipped to address *source-free* DA for all kinds of label-set relationships, without any prior knowledge of the associated *category-gap* (i.e. *universal-DA*). We not only focus on identifying the key complications associated with the challenging problem setting, but also devise insightful ideas to tackle such complications by adopting learning techniques much different from the available DA literature. This leads us to realize a holistic solution which achieves superior DA performance even over prior *source-dependent* approaches.

2 Related Work

We briefly review the available domain adaptation methods under the three major divisions according to the assumption on label-set relationship. **a) Closed-set DA**. The cluster of previous works under this setting focuses on minimizing the domain gap at some intermediate feature level either by minimizing well-defined statistical distance functions (Wang & Schneider, 2014; Duan et al., 2012; Zhang et al., 2013; Saenko et al., 2010) or by formalizing it as an adversarial distribution matching problem (Tzeng et al., 2017; Kang et al., 2018; Long et al., 2018; Hu et al., 2018; Hoffman et al., 2018) inspired from the Generative Adversarial Nets (Goodfellow et al., 2014). Certain prior works (Sankaranarayanan et al., 2018; Zhu et al., 2017; Hoffman et al., 2018) use GAN framework to explicitly generate target-like images translated from the source image samples, which is also regarded as pixel-level adaptation (Bousmalis et al., 2017) in contrast to other feature level adaptation works (Nath Kundu et al., 2018; Tzeng et al., 2017; Long et al., 2015; 2016). **b) Partial DA**. Focusing on *Partial DA*, Cao et al. (2018a) proposed to achieve adversarial class-level matching by utilizing multiple domain discriminators furnishing class-level and instance-level weighting for individual data samples. Zhang et al. (2018b) proposed to utilize importance weights for source samples depending on their similarity to the target domain data using an auxilliary discriminator. To effectively address the problem of *negative-transfer* (Wang et al., 2019), Cao et al. (2018b) employed a single discriminator to achieve both adversarial adaptation and class-level weighting of source samples. **c) Open-set DA**. Saito et al. (2018b) proposed a more general open-set adaptation setting without accessing the knowledge of source private labels set in contrast to the prior work (Panareda Busto & Gall, 2017). They extended the source classifier to accommodate an additional "unknown" class, which is trained adversarially against the other source classes. **Universal DA**. You et al. (2019) proposed Universal DA, which requires no prior knowledge of label-set relationship similar to the proposed setting, but considers access to both source and target samples during adaptation.

3 Proposed Approach

The problem setting for *source-free* domain adaptation is broadly divided into a two stage process.

a) Procurement stage. In this stage, we are given full access to the labeled samples of source domain, $\mathcal{D}_s = \{(x_s, y_s) : x_s \sim p,\ y_s \in \mathcal{C}_s\}$, where p is the distribution of source samples and $\mathcal{C}_s$ denotes the label-set of the source domain. Here, the objective is to equip the model for the second stage, *i.e.* the *Deployment* stage, in the presence of a discrepancy in the distribution of input target samples. To achieve this we rely on an artificially generated negative dataset, $\mathcal{D}_n = \{(x_n, y_n) : x_n \sim p_n,\ y_n \in \mathcal{C}_n\}$, where p_n is the distribution of negative source samples such that $\mathcal{C}_n \cap \mathcal{C}_s = \emptyset$.

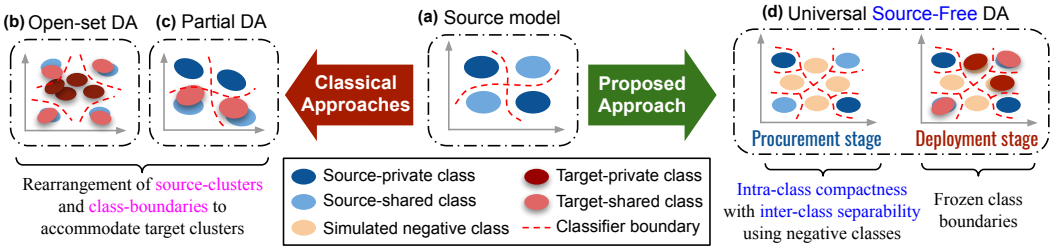

Figure 2: Latent space cluster arrangement during adaptation (see Section 3.1.1).

b) Deployment stage. After obtaining a trained model from the *Procurement* stage, the model will have its first encounter with the unlabeled target domain samples from the deployed environment. We denote the unlabeled target data by $\mathcal{D}_t = \{x_t : x_t \sim q\}$, where q is the distribution of target samples. Note that, access to the source dataset $\mathcal{D}_s$ from the previous stage is fully restricted during adaptation in the *Deployment* stage. Suppose that, $\mathcal{C}_t$ is the "*unknown*" label-set of the target domain. We define the common label space between the source and target domain as $\mathcal{C} = \mathcal{C}_s \cap \mathcal{C}_t$. The private label-set for the source and the target domains is represented as $\overline{\mathcal{C}}_s = \mathcal{C}_s \setminus \mathcal{C}_t$ and $\overline{\mathcal{C}}_t = \mathcal{C}_t \setminus \mathcal{C}_s$ respectively.

3.1 Learning in the Procurement Stage

3.1.1 Challenges. The available DA techniques heavily rely on the adversarial discriminative (Tzeng et al., 2017; Saito et al., 2018a) strategy. Thus, they require access to the source samples to reliably characterize the source domain distribution. Moreover, these approaches are not equipped to operate in a *source-free* setting. Though a generative model can be used as a memory-network (Sankaranarayanan et al., 2018; Bousmalis et al., 2017) to realize *source-free* adaptation, such a solution is not scalable for large-scale source datasets (e.g. ImageNet (Russakovsky et al., 2015)), as it introduces unnecessary extra parameters in addition to the associated training difficulties (Salimans et al., 2016). This calls for a fresh analysis of the requirements beyond the solutions found in literature.

In a general DA scenario, with access to source samples in the *Deployment* stage (specifically for *Open-set* or *Partial* DA), a widely adopted approach is to learn domain invariant features. In such approaches the placement of source category clusters is learned in the presence of unlabeled target samples which obliquely provides a supervision regarding the relationship between $\mathcal{C}_s$ and $\mathcal{C}_t$. For instance, in case of *Open-set* DA, the source clusters may have to disperse to make space for the clusters from target private $\overline{\mathcal{C}}_t$ (see Fig. 2a to 2b). Similarly, in *partial* DA, the source clusters may have to rearrange themselves to keep all the target shared clusters ($\mathcal{C} = \mathcal{C}_t$) separated from the source private $\overline{\mathcal{C}}_s$ (see Fig. 2a to 2c). However in a complete *source-free* framework, we do not have the liberty to leverage such information as source and target samples never coexist together during training. Motivated by the adversarial discriminative DA technique (Tzeng et al., 2017), we hypothesize that, inculcating the ability to reject samples that are out of the source data distribution can facilitate future *source-free* domain alignment using this discriminatory knowledge. Therefore, in the *Procurement* stage the overarching objective is two-fold.

- Firstly, we must aim to learn a certain placement of source clusters best suited for all kinds of *category-gap* scenarios acknowledging the fact that, a *source-free* scenario does not allow us to modify the placement in the presence of target samples during adaptation (see Fig. 2d).
- Secondly, the learned embedding must have the ability to reject out-of-distribution samples, which is an essential requirement for unsupervised adaptation in the presence of domain-shift.

3.1.2 Solution. In the presence of source data, we aim to restrain the model's domain and category bias which is generally inculcated as a result of the over-confident supervised learning paradigms (see Fig. 4A). To achieve this goal, we adopt two regularization strategies viz. i) regularization via generative modeling and ii) utilization of a labeled simulated negative source dataset to generalize for the latent regions not covered by the given positive source samples (see Fig. 4C).

How to configure the negative source dataset? While configuring $\mathcal{D}_n$, the following key properties have to be met. Firstly, latent clusters formed by the negative categories must lie in-between the latent clusters of positive source categories to enable a higher degree of intra-class compactness with inter-class separability (Fig. 4C). Secondly, the negative source samples must enrich the source domain

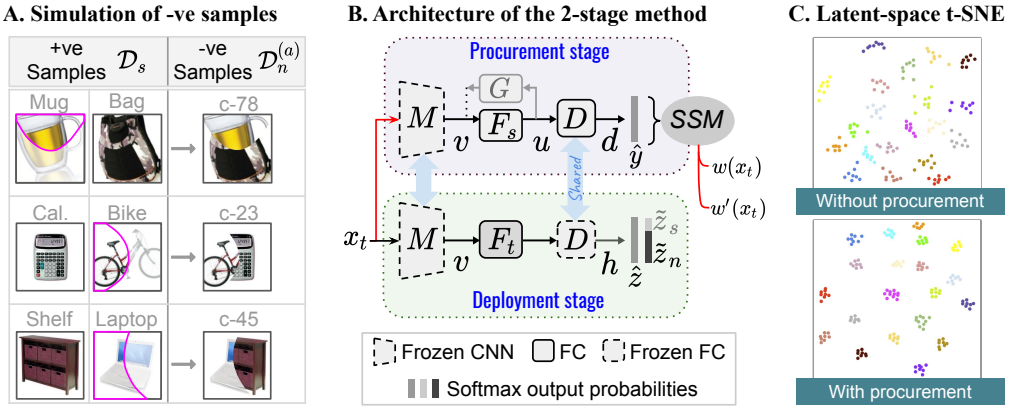

Figure 3: **A)** Simulated labeled negative samples using randomly created spline segments (in pink), **B)** Proposed architecture, **C)** *Procurement* stage yields compact source clusters on experimental data.

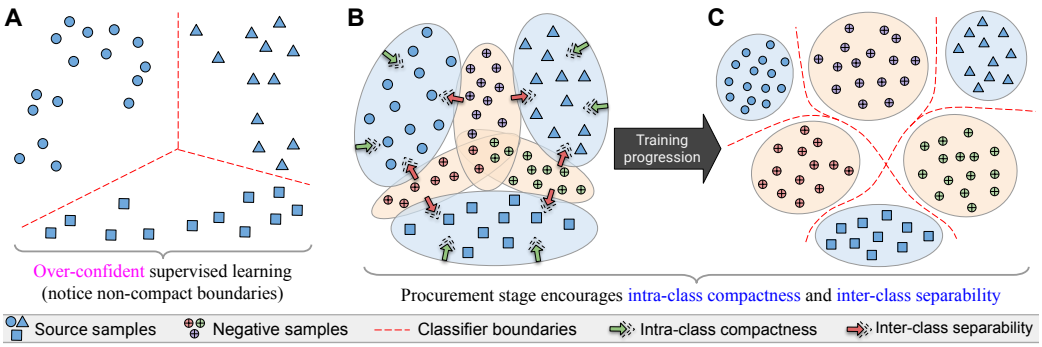

Figure 4: Achieving intra-class compactness and inter-class separability using negative dataset $\mathcal{D}_n$.

distribution without forming a new domain by themselves. This rules out the use of Mixup (Zhang et al., 2018a) or adversarial noise (Shu et al., 2018) as negative samples in this scenario. Thus, we propose the following two ways to synthesize the desired negative source dataset.

a) Image-composition as negative dataset $\mathcal{D}_n^{(a)}$. One of the key characteristics shared between the samples from source and unknown target domain is the semantics of the local part-related features specifically for image-based object recognition tasks. Relying on this assumption, we propose a systematic procedure to simulate the samples of $\mathcal{D}_n^{(a)}$ by randomly compositing local regions between a pair of images drawn from the positive source dataset $\mathcal{D}_s$ (see Fig. 3A and appendix, Algo. 2). Intuitively, composite samples x_n created on image pairs from different source categories are expected to lie in-between the two positive source clusters in the latent space, thereby introducing a combinatorial amount of new class labels *i.e.* $|\mathcal{C}_n| = {}^{|\mathcal{C}_s|}C_2$.

b) Latent-simulated negative dataset $\mathcal{D}_n^{(b)}$. As an alternative approach, in the absence of domain knowledge (e.g. non-image datasets, or for tasks beyond image-recognition such as pose estimation), we propose to sample virtual negative instances, u_n from the latent space which are away from the high confidence regions (3-sigma) of positive source clusters (Fig. 4B). For each negative sample, we assign a negative class label (one of $|\mathcal{C}_n| = {}^{|\mathcal{C}_s|}C_2$) corresponding to the pair of most confident source classes predicted by the classifier. Thus, we obtain $\mathcal{D}_n^{(b)} = \{(u_n, y_n) : u_n \sim p_n^u,\ y_n \in \mathcal{C}_n\}$ where p_n^u is the distribution of negative samples in the latent u-space (more details in appendix Algo. 3).

Training procedure. The generative source classifier is divided into three stages; i) backbone-model M, ii) feature extractor F_s, and iii) classifier D (see Fig. 3B). Output of the backbone-model is denoted as $v = M(x)$, where x is drawn from either $\mathcal{D}_s$ or $\mathcal{D}_n$. Following this, the output of F_s and D are represented as u and d respectively. D outputs a K-dimensional logit denoted as $d^{(k)}$ for $k = 1, 2...K$; $K = |\mathcal{C}_s| + |\mathcal{C}_n|$. The individual class probabilities, $\hat{y}^{(k)}$ are obtained by applying softmax over the logits *i.e.* $\hat{y}^{(k)} = exp(d^{(k)}) / \sum_{k=1}^{K} exp(d^{(k)}) = \sigma^{(k)}(D \circ F_s \circ M(x))$. Additionally, we define priors of only positive source classes as $P(u_s|c_i) = \mathcal{N}(u_s|\mu_{c_i}, \Sigma_{c_i})$ for $i = 1, 2...|\mathcal{C}_s|$ at

Algorithm 1 Training algorithm in the *Procurement* stage

1: **input:** $(x_s, y_s) \in \mathcal{D}_s$, $(x_n, y_n) \in \mathcal{D}_n$; $\theta_{F_s}, \theta_D, \theta_G$: Parameters of F_s, D and G respectively.
2: **initialization:** pretrain $\{\theta_{F_s}, \theta_D\}$ using cross-entropy loss on (x_s, y_s) followed by initialization of the sample mean μ_{c_i} and covariance Σ_{c_i} (at u-space) of $F_s \circ M(x_s)$ for x_s from class c_i; $i = 1, 2, ...|\mathcal{C}_s|$
3: **for** $iter < MaxIter$ **do**
4: $v_s = M(x_s)$; $u_s = F_s(v_s)$; $\hat{v}_s = G(u_s)$; $u_r \sim \mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$ for $i = 1, 2, ...|\mathcal{C}_s|$; $\hat{u}_r = F_s \circ G(u_r)$
5: $\hat{y}_s^{(k_s)} = \sigma^{(k_s)}(D \circ F_s \circ M(x_s))$, and $\hat{y}_n^{(k_n)} = \sigma^{(k_n)}(D \circ F_s \circ M(x_n))$ where k_s and k_n are the index of ground-truth label y_s and y_n respectively.
6: $\mathcal{L}_{CE} = -\log \hat{y}_s^{(k_s)} - \alpha \log \hat{y}_n^{(k_n)}$; $\mathcal{L}_v = |v_s - \hat{v}_s|$; $\mathcal{L}_u = |u_r - \hat{u}_r|$
7: $\mathcal{L}_p = -\log(\exp(P(u_s|c_{k_s}))/\sum_{i=1}^{|\mathcal{C}_s|}\exp(P(u_s|c_i)))$, where $P(u_s|c_i) = \mathcal{N}(u_s|\mu_{c_i}, \Sigma_{c_i})$
8: Update $\theta_{F_s}, \theta_D, \theta_G$ by minimizing $\mathcal{L}_{CE}$, $\mathcal{L}_v$, $\mathcal{L}_u$, and $\mathcal{L}_p$ alternatively using separate optimizers.
9: **if** $(iter \,\%\, UpdateIter == 0)$ **then**
10: Recompute the sample mean (μ_{c_i}) and covariance (Σ_{c_i}) of $F_s \circ M(x_s)$ for x_s from class c_i; $i = 1, 2...|\mathcal{C}_s|$ (For $\mathcal{D}_n^{(b)}$: generate fresh latent-simulated negative samples using the updated priors)

the intermediate embedding $u_s = F_s \circ M(x_s)$. Here, parameters of the normal distributions are computed during training as shown in line-10 of Algo. 1. A cross-entropy loss over these prior distributions is defined as $\mathcal{L}_p$ (line-7 in Algo. 1), to effectively enforce intra-class compactness with inter-class separability (progression from Fig. 4B to 4C). Motivated by generative variational auto-encoder (VAE) setup (Kingma & Welling, 2013), we introduce a feature decoder G, which aims to minimize the cyclic reconstruction loss selectively for the samples from positive source categories v_s and randomly drawn samples u_r from the corresponding class priors (*i.e.* $\mathcal{L}_v$ and $\mathcal{L}_u$, line-6 in Algo. 1). This along with a lower weightage α for the negative source categories (*i.e.* at the cross-entropy loss $\mathcal{L}_{CE}$, line-6 in Algo. 1) is incorporated to deliberately bias F_s towards the positive source samples, considering the level of unreliability of the generated negative dataset.

3.2 Learning in the Deployment Stage

3.2.1 Challenges. We hypothesize that, the large number of negative source categories along with the positive source classes *i.e.* $\mathcal{C}_s \cup \mathcal{C}_n$ can be interpreted as a universal source dataset, which can subsume label-set $\mathcal{C}_t$ of a wide range of target domains. Moreover, we seek to realize a unified adaptation algorithm, which can work for a wide range of *category-gaps*. However, a forceful adaptation of target samples to positive source categories will cause target private samples to be classified as an instance of the source private or the common label-set, instead of being classified as "*unknown*", i.e. one of the negative categories in $\mathcal{C}_n$.

3.2.2 Solution. In contrast to domain agnostic architectures (You et al., 2019; Cao et al., 2018a; Saito et al., 2018a), we resort to an architecture supporting domain specific features (Tzeng et al., 2017), as we must avoid disturbing the placement of source clusters obtained from the *Procurement* stage. This is an essential requirement to retain the task-dependent knowledge gathered from the source dataset. Thus, we introduce a domain specific feature extractor denoted as F_t, whose parameters are initialized from the fully trained F_s (see Fig. 3B). Further, we aim to exploit the learned generative classifier from the *Procurement* stage to complement for the purpose of separate ad-hoc networks (critic or discriminator) as utilized by the prior works (You et al., 2019; Cao et al., 2018b).

a) Source Similarity Metric (SSM). We define a weighting factor (SSM) for each target sample x_t, as $w(x_t)$. A higher value of this metric indicates x_t's similarity towards the positive source categories, specifically inclined towards the common label space $\mathcal{C}$. Similarly, a lower value of this metric indicates x_t's similarity towards the negative source categories $\mathcal{C}_n$, showing its inclination towards the private target labels $\overline{\mathcal{C}}_t$. Let, $p_{\bar{s}}$, $q_{\bar{t}}$ be the distribution of source and target samples with labels in $\overline{\mathcal{C}}_s$ and $\overline{\mathcal{C}}_t$ respectively. We define, p_c and q_c to denote the distribution of samples from source and target domains belonging to the shared label-set $\mathcal{C}$. Then, the *SSM* for the positive and negative source samples should lie on the two extremes, forming the following inequality:

$$\mathbb{E}_{x_n \sim p_n} w(x_n) \approx \mathbb{E}_{x_t \sim q_{\bar{t}}} w(x_t) \; < \mathbb{E}_{x_t \sim q_c} w(x_t) < \mathbb{E}_{x_s \sim p_c} w(x_s) \approx \mathbb{E}_{x_s \sim p_{\bar{s}}} w(x_s) \tag{1}$$

To formalize the *SSM* criterion we rely on the class probabilities defined at the output of source model only for the positive class labels, *i.e.* $\hat{y}^{(k)}$ for $k = 1, 2...|\mathcal{C}_s|$. Note that, $\hat{y}^{(k)}$ is obtained by performing softmax over $|\mathcal{C}_s| + |\mathcal{C}_n|$ categories as discussed in the *Procurement* stage. Finally, the

SSM and its complement are defined as,

$$w(x_t) = \max_{i=1,2...|\mathcal{C}_s|} \exp(\hat{y}^{(i)}), \;\; and \;\; w'(x_t) = \max_{i=1,2...|\mathcal{C}_s|} \exp(1 - \hat{y}^{(i)}) \tag{2}$$

We hypothesize that, the above definition will satisfy Eq. 1, as a result of the generative learning strategy adopted in the *Procurement* stage. In Eq. 2 the exponent is used to further amplify separation between target samples from the shared $\mathcal{C}$ and those from the private $\overline{\mathcal{C}}_t$ label-set (see Fig. 5A).

b) Source-free domain adaptation. To perform domain adaptation, the objective function aims to move the target samples with higher *SSM* value towards the clusters of positive source categories and vice-versa at the frozen source embedding, u-space (from the *Procurement* stage). To achieve this, parameters of only F_t network are allowed to be trained in the *Deployment* stage. However, the decision of weighting the loss on target samples towards the positive or negative source clusters is computed using the source feature extractor F_s *i.e.* the *SSM* in Eq. 2. We define, the deployment model as $h = D \circ F_t \circ M(x_t)$ using the target feature extractor, with softmax predictions over K categories obtained as $\hat{z}^{(k)} = \sigma(h^{(k)})$. Thus, the primary loss function for adaptation is defined as,

$$\mathcal{L}_{d1} = -w(x_t) \log(\textstyle\sum_{k=1}^{|\mathcal{C}_s|} \hat{z}^{(k)}) - w'(x_t) \log(\sum_{k=1+|\mathcal{C}_s|}^{|\mathcal{C}_s|+|\mathcal{C}_n|} \hat{z}^{(k)}) \tag{3}$$

Additionally, in the absence of label information, there would be uncertainty in the predictions $\hat{z}^{(k)}$ as a result of distributed class probabilities. This leads to a higher entropy for such samples. Entropy minimization (Grandvalet & Bengio, 2005; Long et al., 2016) is adopted in such scenarios to move the target samples close to the highly confident regions (*i.e.* positive and negative cluster centers from the *Procurement* stage) of the classifier's feature space. However, it has to be done separately for positive and negative source categories based on the *SSM* values of individual target samples to effectively distinguish the target-private set from the full target dataset. To achieve this, we define two different class probability vectors separately for the positive and negative source classes denoted as, $\tilde{z}_s^{(i)} = \exp(h^{(i)}) / \sum_{j=1}^{|\mathcal{C}_s|} \exp(h^{(j)})$ and $\tilde{z}_n^{(i)} = \exp(h^{(i+|\mathcal{C}_s|)}) / \sum_{j=1}^{|\mathcal{C}_n|} \exp(h^{(j+|\mathcal{C}_s|)})$ respectively (see Fig. 3B). Entropy of the target samples in the positive and negative regimes of the source classifier is obtained as $H_s(x_t) = -\sum_{i=1}^{|\mathcal{C}_s|} \tilde{z}_s^{(i)} \log \tilde{z}_s^{(i)}$ and $H_n(x_t) = -\sum_{i=1}^{|\mathcal{C}_n|} \tilde{z}_n^{(i)} \log \tilde{z}_n^{(i)}$ respectively. Consequently, the entropy minimization loss is formalized as,

$$\mathcal{L}_{d2} = w(x_t) H_s(x_t) + w'(x_t) H_n(x_t) \tag{4}$$

Thus, the final loss function for adapting the parameters of F_t is presented as $\mathcal{L}_d = \mathcal{L}_{d1} + \beta \mathcal{L}_{d2}$. Here β is a hyper-parameter controlling the importance of entropy minimization during adaptation.

4 Experiments

We perform a thorough evaluation of the proposed *source-free*, universal domain adaptation framework against prior state-of-the-art models across multiple datasets. We also provide a comprehensive ablation study to establish generalizability of the approach across a variety of label-set relationships and justification of the various model components.

4.1 Experimental Setup

Datasets. For all the following datasets, we resort to the experimental settings inline with the recent work by You et al. (2019) (UAN). **Office-Home** (Venkateswara et al., 2017) dataset consists of images from 4 different domains - Artistic (**Ar**), Clip-art (**Cl**), Product (**Pr**) and Real-world (**Rw**). Alphabetically, the first 10 classes are selected as $\mathcal{C}$, the next 5 classes as $\overline{\mathcal{C}}_s$, and the rest 50 as $\overline{\mathcal{C}}_t$. **VisDA2017** (Peng et al., 2018) dataset comprises of 12 categories with synthetic images as the source domain and natural images as the target domain, out of which, the first 6 are chosen as $\mathcal{C}$, the next 3 as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. **Office-31** (Saenko et al., 2010) dataset contains images from 3 distinct domains - Amazon (**A**), DSLR (**D**) and Webcam (**W**). We use the 10 classes shared by Office-31 and Caltech-256 (Gong et al., 2012) to construct the shared label-set $\mathcal{C}$ and alphabetically select the next 10 as $\overline{\mathcal{C}}_s$, with the remaining 11 classes contributing to $\overline{\mathcal{C}}_t$. To evaluate scalability, **ImageNet-Caltech** is also considered with 84 common classes inline with the setting in You et al. (2019).

Simulation of labeled negative samples. To simulate negative labeled samples for training in the *Procurement* stage, we first sample a pair of images, each from different categories of $\mathcal{C}_s$, to create

Table 1: Average per-class accuracy ($\mathcal{T}_{avg}$) for universal-DA tasks on **Office-Home** dataset (with $|C|/|\mathcal{C}_s \cup \mathcal{C}_t| = 0.15$). Scores for the prior works are directly taken from UAN (You et al., 2019).

Method	Office-Home												
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet (He et al., 2016)	59.37	76.58	87.48	69.86	71.11	81.66	73.72	56.30	86.07	78.68	59.22	78.59	73.22
IWAN (Zhang et al., 2018b)	52.55	81.40	86.51	70.58	70.99	85.29	74.88	57.33	85.07	77.48	59.65	78.91	73.39
PADA (Zhang et al., 2018b)	39.58	69.37	76.26	62.57	67.39	77.47	48.39	35.79	79.60	75.94	44.50	78.10	62.91
ATI (Busto et al., 2017)	52.90	80.37	85.91	71.08	72.41	84.39	74.28	57.84	85.61	76.06	60.17	78.42	73.29
OSBP (Saito et al., 2018b)	47.75	60.90	76.78	59.23	61.58	74.33	61.67	44.50	79.31	70.59	54.95	75.18	63.90
UAN (You et al., 2019)	63.00	82.83	87.85	**76.88**	**78.70**	85.36	78.22	58.59	86.80	**83.37**	63.17	79.43	77.02
Source-free adaptation													
Ours ***USFDA-a***	**63.35**	**83.30**	**89.35**	70.96	72.34	**86.09**	**78.53**	**60.15**	**87.35**	81.56	63.17	**88.23**	**77.03**
Ours ***USFDA-b***	62.46	82.71	88.26	71.10	70.88	85.75	78.21	59.18	86.05	82.17	**63.22**	87.68	76.47

unique negative classes in $\mathcal{C}_n$. Note that, we impose no restriction on how the hypothetical classes are created (e.g. one can composite non-animal with animal). A random mask is defined which splits the images into two complementary regions using a quadratic spline passing through a central image region (see Appendix Algo. 2). Then, the negative image is created by merging alternate mask regions as shown in Fig. 3A. For the **I**→**C** task of ImageNet-Caltech, the source domain (ImageNet), consisting of 1000 classes, results in a large number of possible negative classes (*i.e.* $|\mathcal{C}_n| = {}^{|\mathcal{C}_s|}C_2$). We address this by randomly selecting only 600 of these negative classes for ImageNet(**I**), and 200 negative classes for Caltech(**C**) in the task **C**→**I**. In a similar fashion, we generate latent-simulated negative samples only for the selected negative classes in these datasets. Consequently, we compare two models with different *Procurement* stage training - (i) ***USFDA-a***: using image-composition as negative dataset , and (ii) ***USFDA-b***: using latent-simulated negative samples as the negative dataset. We use *USFDA-a* for most of our ablation experiments unless mentioned explicitly.

4.2 Evaluation Methodology

Average accuracy on Target dataset, $\mathcal{T}_{avg}$. We resort to the evaluation protocol proposed in the VisDA2018 Open-Set Classification challenge. Accordingly, all the target private classes are grouped into a single "*unknown*" class and the metric reports the average of per-class accuracy over $|\mathcal{C}_s| + 1$ classes. In the proposed framework a target sample is marked as "*unknown*", if it is classified ($argmax_k \hat{z}^{(k)}$) into any of the negative $|\mathcal{C}_n|$ classes out of total $|\mathcal{C}_s| + |\mathcal{C}_n|$ categories. In contrast, UAN (You et al., 2019) relies on a sensitive hyperparameter, as a threshold on the sample-level weighting, to mark a target sample as "*unknown*". Also note that, our method is completely *source-free* during the *Deployment* stage, while all other methods have access to the full source-data.

Accuracy on Target-Unknown data, $\mathcal{T}_{unk}$. We evaluate the target unknown accuracy, $\mathcal{T}_{unk}$, as the proportion of actual target private samples (i.e. $\{(x_t, y_t) : y_t \in \bar{\mathcal{C}}_t\}$) being classified as "*unknown*" after adaptation. Note that, UAN (You et al., 2019) does not report $\mathcal{T}_{unk}$ which is a crucial metric to evaluate the vulnerability of the model after its deployment in the target environment. The $\mathcal{T}_{avg}$ metric fails to capture this as a result of class-imbalance in the *Open-set* scenario (Saito et al., 2018b). Hence, to realize a common evaluation ground, we train the UAN implementation provided by the authors (You et al., 2019) and denote it as UAN* in further sections of this paper. We observe that, the UAN(You et al., 2019) training algorithm is often unstable with a decreasing trend of $\mathcal{T}_{unk}$ and $\mathcal{T}_{avg}$ over increasing training iterations. We thus report the mean and standard deviation of the peak values of $\mathcal{T}_{unk}$ and $\mathcal{T}_{avg}$ achieved by UAN*, over 5 separate runs on Office-31 dataset (see Table 7).

Implementation Details. We implement our network in PyTorch and use ResNet-50 (He et al., 2016) as the backbone-model M, pre-trained on ImageNet (Russakovsky et al., 2015) inline with UAN (You et al., 2019). The complete architecture of other components with fully-connected layers is provided in the Supplementary. A sensitivity analysis of the major hyper-parameters used in the proposed framework is provided in Fig. 5B-C, and Appendix Fig. 8B. In all our ablations across the datasets, we fix the hyperparameters values as $\alpha = 0.2$ and $\beta = 0.1$. We utilize Adam optimizer (Kingma & Ba, 2014) with a fixed learning rate of 0.0001 for training in both *Procurement* and *Deployment* stage (see Appendix for the code). For the implementation of UAN*, we use the hyper-parameter value $w_0 = -0.5$, as specified by the authors for the task **A**→**D** in Office-31 dataset.

4.3 Discussion

a) Comparison with prior arts. We compare our approach with UAN You et al. (2019), and other prior methods. The results are presented in Table 1 and Table 2. Clearly, our framework achieves state-

Table 2: $\mathcal{T}_{avg}$ on **Office-31** (with $|C|/|\mathcal{C}_s \cup \mathcal{C}_t| = 0.32$), **VisDA** (with $|C|/|\mathcal{C}_s \cup \mathcal{C}_t| = 0.50$), and **ImageNet-Caltech** (with $|C|/|\mathcal{C}_s \cup \mathcal{C}_t| = 0.07$). Here, SF denotes support for *source-free* adaptation.

Method	SF	Office-31							VisDA	ImNet-Caltech	
		A→W	D→W	W→D	A→D	D→A	W→A	Avg	S → R	I → C	C → I
ResNet (He et al., 2016)	✗	75.94	89.60	90.91	80.45	78.83	81.42	82.86	52.80	70.28	65.14
IWAN (Zhang et al., 2018b)	✗	85.25	90.09	90.00	84.27	84.22	86.25	86.68	58.72	72.19	66.48
PADA (Zhang et al., 2018b)	✗	85.37	79.26	90.91	81.68	55.32	82.61	79.19	44.98	65.47	58.73
ATI (Busto et al., 2017)	✗	79.38	92.60	90.08	84.40	78.85	81.57	84.48	54.81	71.59	67.36
OSBP (Saito et al., 2018b)	✗	66.13	73.57	85.62	72.92	47.35	60.48	67.68	30.26	62.08	55.48
UAN (You et al., 2019)	✗	85.62	94.77	97.99	86.50	85.45	85.12	89.24	60.83	75.28	70.17
UAN* $\mathcal{T}_{avg}$	✗	83.00±1.8	94.17±0.3	95.40±0.5	83.43±0.7	86.90±1.0	**87.18**±0.6	88.34	54.21	74.77	71.51
Ours ***USFDA-a*** $\mathcal{T}_{avg}$	✓	**85.56**±1.6	95.20±0.3	**97.79**±0.1	**88.47**±0.3	87.50±0.9	86.61±0.6	**90.18**	**63.92**	**76.85**	72.13
Ours ***USFDA-b*** $\mathcal{T}_{avg}$	✓	83.21±1.2	**95.33**±0.3	96.37±0.3	86.84±0.4	**87.91**±0.6	86.74±0.5	89.40	62.77	76.74	**72.25**
UAN* $\mathcal{T}_{unk}$	✗	20.72±11.7	53.53±2.4	51.57±5.0	34.43±3.3	51.88±4.8	43.11±1.3	42.54	19.68	33.43	31.24
Ours ***USFDA-a*** $\mathcal{T}_{unk}$	✓	**73.98**±7.5	85.64±2.2	**80.00**±1.1	82.23±2.7	**78.59**±3.2	**75.52**±1.5	**79.32**	**36.25**	**51.21**	**48.76**
Ours ***USFDA-b*** $\mathcal{T}_{unk}$	✓	70.22±8.8	**85.89**±2.3	78.29±1.7	**84.66**±3.1	76.22±2.8	73.91±1.6	78.19	34.84	51.10	48.20

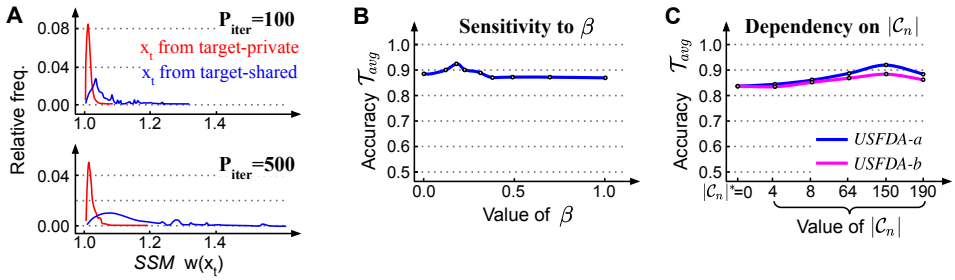

Figure 5: Ablative analysis on the task **A**→**D** in Office-31 dataset. **A)** Histogram of SSM values of x_t separately for target-private and target-shared samples at the *Procurement* iteration 100 (top) and 500 (bottom). **B)** The sensitivity curve for β shows marginally stable adaptation accuracy for a wide-range of values. **C)** A marginal increase in $\mathcal{T}_{avg}$ is observed with increase in $|\mathcal{C}_n|$.

of-the-art results even in a *source-free* setting on several tasks. Particularly in Table 2, we present the target-unknown accuracy $\mathcal{T}_{unk}$ on various dataset. It also holds the mean and standard-deviation for both the accuracy metrics computed over 5 random initializations in the Office-31 dataset (the last six rows). Our method is able to achieve much higher $\mathcal{T}_{unk}$ than UAN* (You et al., 2019), highlighting our superiority as a result of the novel learning approach incorporated in both *Procurement* and *Deployment* stages. Note that, both *USFDA-a* and *USFDA-b* yield similar performance across a wide range of standard benchmarks. We also perform a characteristic comparison of algorithm complexity in terms of the amount of learnable parameters and training time. In contrast to UAN, the proposed framework offers a much simpler adaptation algorithm devoid of utilization of ad-hoc networks like adversarial discriminator and additional finetuning of the ResNet-50 backbone. Parameter size and training time; a) Ours procurement (*USFDA-a*): [11.1M, 380s], b) Ours deployment: [3.5M, 44s], c) UAN (You et al., 2019): [26.7M, 450s] (in a consistent setting). The significant computational advantage in the *Deployment* stage makes *USFDA* highly suitable for real-time adaptation.

b) Does *SSM* satisfy the expected inequality? Effectiveness of the proposed learning algorithm, in case of *source-free* deployment, relies on the formulation of *SSM*, which is expected to satisfy Eq. 1. Fig. 5A shows a histogram of the *SSM* separately for samples from target-shared (blue) and target-private (red) label space. The success of this metric is attributed to the generative nature of *Procurement* stage, which enables the source model to distinguish between the marginally more negative target-private samples as compared to the samples from the shared label space.

c) Sensitivity to hyper-parameters. As we tackle DA in a *source-free* setting simultaneously intending to generalize across varied *category-gaps*, a low sensitivity to hyperparameters would further enhance our practical usability. To this end, we fix certain hyperparameters for all our ablations (also in Fig. 6C) even across datasets (i.e. $\alpha = 0.2$, $\beta = 0.1$). Thus, one can treat them as global-constants with $|\mathcal{C}_n|$ being the only hyperparameter, as variations in one by fixing the others yield complementary effect on regularization in the *Procurement* stage. A thorough analysis reported in the appendix Fig. 8, clearly demonstrates the low-sensitivity of our model to these hyperparameters.

d) Generalization across category-gap. One of the key objectives of the proposed framework is to effectively operate in the absence of the knowledge of label-set relationships. To evaluate it in

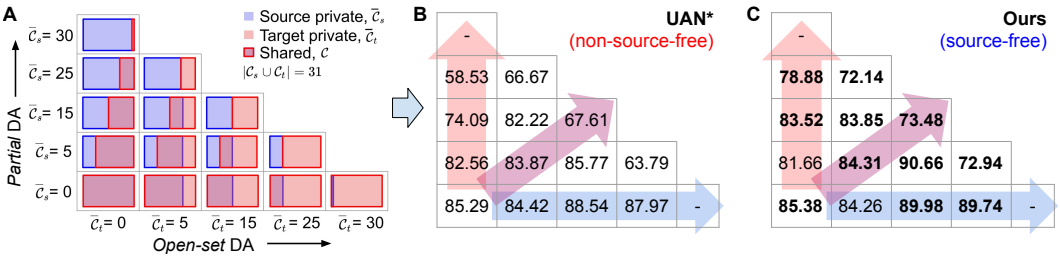

Figure 6: Comparison across varied label-set relationships for the task **A**→**D** in Office-31 dataset. **A)** Visual representation of label-set relationships and $\mathcal{T}_{avg}$ at the corresponding instances for **B)** UAN* (You et al., 2019) and **C)** ours *source-free* model. Effectively, the direction along x-axis (blue horizontal arrow) characterizes increasing *Open-set* complexity. The direction along y-axis (red vertical arrow) shows increasing complexity of *Partial* DA scenario. The pink diagonal arrow denotes the effect of decreasing shared label space.

the most compelling manner, we propose a tabular form shown in Fig. 6A. We vary the number of private classes for target and source along x and y axis respectively, with a fixed $|\mathcal{C}_s \cup \mathcal{C}_t| = 31$. We compare the $\mathcal{T}_{avg}$ metric at the corresponding table instances, shown in Fig. 6B-C. The results clearly highlight superiority of the proposed framework specifically for the more practical scenarios (close to the diagonal instances) as compared to the unrealistic *Closed-set* setting ($|\bar{\mathcal{C}}_s| = |\bar{\mathcal{C}}_t| = 0$).

e) DA in absence of shared categories. In universal adaptation, we seek to transfer the knowledge of "*class-separability criterion*" obtained from the source domain to the deployed target environment. More concretely, it is attributed to the segregation of data samples based on some expected characteristics, such as classification of objects according to their pose, color, or shape etc. To quantify this, we consider an extreme case where $\mathcal{C}_s \cap \mathcal{C}_t = \emptyset$ (**A**→**D** in Office-31 with $|\mathcal{C}_s| = 15$, $|\mathcal{C}_t| = 16$). Allowing access to a single labeled target sample from each category in $\bar{\mathcal{C}}_t = \mathcal{C}_t$, we aim to obtain a one-shot recognition accuracy (assignment of cluster index or class label using the one-shot samples as the cluster center at $F_t \circ M(x_t)$) to quantify the above metric. We obtain 64.72% accuracy for the proposed framework as compared to 13.43% for UAN* (You et al., 2019). This strongly validates our superior knowledge transfer capability as a result of the generative classifier with labeled negative samples complementing for the target-private categories.

f) Dependency on the simulated negative dataset. Conceding that a combinatorial amount of negative labels can be created, we evaluate the scalability of the proposed approach, by varying the number of negative classes in the *Procurement* stage by selecting 0, 4, 8, 64, 150 and 190 negative classes as reported in the X-axis of Fig. 5C. For the case of 0 negative classes, denoted as $|\mathcal{C}_n|^* = 0$ in Fig. 5C, we synthetically generate random negative features at the intermediate level u, which are at least 3-sigma away from each of the positive source priors $P(u_s|c_i)$. We then make use of these feature samples along with positive image samples, to train a $(|\mathcal{C}_s| + 1)$ class *Procurement* model with a single negative class. The results are reported in Fig. 5C on the **A**→**D** task of Office-31 dataset with category relationship inline with the setting in Table 7. We observe an acceptable drop in accuracy with decrease in number of negative classes, hence validating scalability of the approach for large-scale classification datasets (such as ImageNet). Similarly, we also evaluated our framework by combining three or more images to form such negative classes. An increasing number of negative classes (${}^{|\mathcal{C}_s|}C_3 > {}^{|\mathcal{C}_s|}C_2$) attains under-fitting on positive source categories (similar to Fig. 5C, where accuracy reduces beyond a certain limit because of over regularization).

5 Conclusion

We have introduced a novel *source-free*, universal domain adaptation framework, acknowledging practical domain adaptation scenarios devoid of any assumption on the source-target label-set relationship. In the proposed two-stage framework, learning in the *Procurement* stage is found to be highly crucial, as it aims to exploit the knowledge of class-separability in the most general form with enhanced robustness to out-of-distribution samples. Besides this, success in the *Deployment* stage is attributed to the well-designed learning objectives effectively utilizing the source similarity criterion. This work can be served as a pilot study towards learning efficient inheritable models in future.

REFERENCES

Mahsa Baktashmotlagh, Masoud Faraki, Tom Drummond, and Mathieu Salzmann. Learning factorized representations for open-set domain adaptation. In *International Conference on Learning Representations*, 2019. 1

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2, 3

Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018a. 1, 2, 5

Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision*, 2018b. 1, 2, 5

Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1

Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012. 2

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1

ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017. 1

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2012. 6

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014. 2

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, 2005. 6

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 7, 8

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Learning Representations*, 2018. 2

Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 1

Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Advances in neural information processing systems*, 2018. 1

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 2015. 1, 2, 17

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, 2016. 1, 2, 6

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in neural information processing systems*, 2018. 2, 17

Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in neural information processing systems*, 2017. 2

Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 1, 2

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. 2018. 6

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015. 1

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3, 7

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*, 2010. 2, 6, 16, 20

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018a. 3, 5

Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision*, 2018b. 2, 7, 8

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, 2016. 3

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2, 3

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 1

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 4

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 1, 2, 3, 5, 17

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 6

Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In *Advances in neural information processing systems*, 2014. 2

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019. 2

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2019. 2, 5, 6, 7, 8, 9, 16, 17, 18, 19, 20

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018a. 4

Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018b. 1, 2, 7, 8

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, 2013. 2

Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018c. 1

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 16

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2017. 2

A APPENDIX

This appendix is organized as follows,

- Implementation details
  - *Procurement* Stage.
  - *Deployment* Stage.
- Ablation Studies and Additional Results
  - Pretraining the backbone network on Places instead of ImageNet.
  - Space and Time Complexity Analysis.
  - Varying label-set relationship.
  - Sensitivity analysis.
  - Closed-set adaptation.
  - Accuracy on source dataset post *Procurement*.
  - Incremental one-shot classification.
  - Feature Space Visualization.
- Miscellaneous
  - Specification of Computing Resources.
  - References to code

B IMPLEMENTATION DETAILS

In this section, we describe the architecture and the training process used for the *Procurement* and *Deployment* stages of our approach.

B.1 PROCUREMENT STAGE

a) Design of classifier D used in the *Procurement* stage. Keeping in mind the possibility of an additional domain shift after performing adaptation (e.g. encountering domain **W** after performing the adaptation **A** $\rightarrow$ **D** in Office-31 dataset), we design the classifier's architecture in a manner which allows for dynamic modification in the number of negative classes post-procurement. We achieve this by maintaining two separate classifiers during *Procurement* - D_{src}, that operates on the positive source classes, and, D_{neg} that operates on the negative source classes (see architecture in Table 5). The final classification score is obtained by computing softmax over the concatenation of logit vectors produced by D_{src} and D_{neg}. Therefore, the model can be retrained on a different number of negative classes post deployment (using another negative class classifier D'_{neg}), thus preparing it for a subsequent adaptation step to another domain.

b) Negative dataset generation. We propose two methods to generate negative samples for the *Procurement* stage, and name the models trained subsequently as *USFDA-a* and *USFDA-b*. Here, we describe the two processes:

- **Using image-composition for $\mathcal{D}_n^{(a)}$ (*USFDA-a*).** In the presence of domain knowledge (knowledge of the task at hand, i.e. object recognition using images), we generate the negative dataset $\mathcal{D}_n^{(a)}$ by compositing images taken from different classes, as described in Algo. 2. We generate random masks using quadratic splines passing through a central image region (lines 3-9). Using these masks, we merge alternate regions of the images, both horizontally and vertically, resulting in 4 negative images for each pair of images (lines 10-13). To effectively cover the inter-class negative region, we randomly sample image pairs from D_s belonging to different classes, however we do not impose any constraint on how the classes are selected (for e.g. one can composite images from an animal and a non-animal class). We choose 5000 pairs for tasks on Office-31, Office-Home and VisDA datasets, and 12000 for ImageNet-Caltech. Since the input source distribution (p) is fixed we first synthesize a negative dataset offline (instead of creating them on the fly) to ensure finiteness of the training set. The training algorithm for *USFDA-a* is given in Algo. 1.

Algorithm 2 Image-composition algorithm

input: Image pair $(I_1, I_2) \in \mathcal{D}_s$. (image shape HxWx3 = 224x224x3)
$k \leftarrow 30$
$x_1, x_2, y_1, y_2 \leftarrow$ rand$(0, W)$, rand$(0, W)$, rand$(0, H)$, rand$(0, H)$
$c_x, c_y \leftarrow$ rand$(W/2 - k, W/2 + k)$, rand$(H/2 - k/3, H/2 + k/3)$
$d_x, d_y \leftarrow$ rand$(W/2 - k/3, W/2 + k/3)$, rand$(H/2 - k, H/2 + k)$
$s_1 \leftarrow$ quadratic_interpolation$([(0, y_1), (c_x, c_y), (223, y_2)])$ ▷ horizontal splicing
$s_2 \leftarrow$ quadratic_interpolation$([(x_1, 0), (d_x, d_y), (x_2, 223)])$ ▷ vertical splicing
$m_1 \leftarrow$ mask region below s_1
$m_2 \leftarrow$ mask region to the left of s_2
$I_a \leftarrow m_1 * I_1 + (1 - m_1) * I_2$
$I_b \leftarrow m_2 * I_1 + (1 - m_2) * I_2$
$I_c \leftarrow m_1 * I_2 + (1 - m_1) * I_1$
$I_d \leftarrow m_2 * I_2 + (1 - m_2) * I_1$
return I_a, I_b, I_c, I_d

Algorithm 3 Dataset generation using latent-simulated negative samples

input: class-wise source priors $\mathcal{N}(\mu_{c_j}, \Sigma_{c_j})$, global source prior $\mathcal{N}(\mu, \Sigma)$, number of required samples n, source classifier D_{src} ▷ $||$ signifies an Append Operation
$\tilde{\mathcal{U}} \leftarrow \{\}; \tilde{\mathcal{Y}} \leftarrow \{\}$
while $|\tilde{\mathcal{U}}| \leq n$ **do**
    Let λ_{c_j} and l_{c_j} be the maximum eigen value and the corresponding eigen vector of Σ_{c_j}, for each class c_j
    $\tilde{u}_r \sim \mathcal{N}(\mu, \Sigma)$
    if $P(\tilde{u}_r|c_j) < P(\mu_{c_j} + 3 * \sqrt{\lambda_{c_j}} * l_{c_j} \mid c_j)$ for all class c_j **then**
        $\hat{y} \leftarrow \sigma(D_{src}(\tilde{u}_r))$
        $\tilde{y}_r \leftarrow$ assign the negative class based on the top-2 confident classes in $\hat{y}$
        $\tilde{\mathcal{U}} \leftarrow \tilde{\mathcal{U}} \,||\, \tilde{u}_r; \tilde{\mathcal{Y}} \leftarrow \tilde{\mathcal{Y}} \,||\, \tilde{y}_r$
    else
        reject $\tilde{u}_r$
return $\tilde{\mathcal{U}}, \tilde{\mathcal{Y}}$

- **Using latent-simulated negative samples for $\mathcal{D}_n^{(b)}$ (*USFDA-b*)**: Here, we perform rejection sampling as given in Algorithm 3. Here, we obtain a sample from the global source prior $P(u_s) = \mathcal{N}(u_s|\mu, \Sigma)$, where μ and Σ are the mean and covariance computed at u-space over all the positive source image samples.We reject the sample if it lies within the 3-sigma bound of any class (i.e. we keep the sample if it is far away from all source class-priors, $\mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$), as shown in lines 6 to 11 in Algo. 3. A sample selected in this fashion is expected to lie in an intermediate region between the source class priors. The two classes in the vicinity of the sample are then determined by obtaining the two most confident class predictions given by the classifier D_{src} (lines 7 and 8). Using this pair of classes, we assign a unique negative class label to the sample which corresponds to the intermediate region between the pair of classes. Note, to learn the arrangement of positive and negative clusters, the feature extractor F_s must be trained using negative samples. We do this by passing the sampled latent-simulated negative instance ($\tilde{u}_r$) through the decoder-encoder pair, (i.e. $D \circ F_s \circ G(\tilde{u}_r)$), and enforcing the cross-entropy loss to classify them into the respective negative class. The training algorithm for *USFDA-b* is given in Algo. 4.

c) Justification of $\mathcal{L}_p$. The cross-entropy loss on the likelihoods (referred as $\mathcal{L}_p$ in the paper) not only enforces intra-class compactness but also ensures inter-class separability in the embedding space, u. Since the negative samples are only an approximation of future target private classes expected to be encountered during deployment, we choose not to employ this loss for them. Such a training procedure, eventually results in a natural development of bias towards the confident positive source classes. This subsequently leads to the placement of source clusters in a manner which enables *source-free* adaptation (See Fig. 4).

Algorithm 4 Training algorithm for *USFDA-b* in the *Procurement* stage

input: $(x_s, y_s) \in \mathcal{D}_s$; $\theta_{F_s}, \theta_D, \theta_G$: Parameters of F_s, D and G respectively.
initialization: pretrain $\{\theta_{F_s}, \theta_D\}$ using cross-entropy loss on (x_s, y_s), then, compute the sample mean μ_{c_i} and covariance Σ_{c_i} of $F_s \circ M(x_s)$ for x_s from class c_i, for $i = 1, 2, ...|\mathcal{C}_s|$
for $iter < MaxIter$ **do**
    $v_s = M(x_s)$; $u_s = F_s(v_s)$; $\hat{v}_s = G(u_s)$; $u_r \sim \mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$ for $i = 1, 2, ...|\mathcal{C}_s|$; $\hat{u}_r = F_s \circ G(u_r)$
    $(\tilde{u}_r, \tilde{y}_r)$ = sample latent-simulated negative instances from $\mathcal{D}_n^{(b)}$
    $\hat{y}_s^{(k_s)} = \sigma^{(k_s)}(D \circ F_s \circ M(x_s))$, and $\hat{y}_n^{(k_n)} = \sigma^{(k_n)}(D \circ F_s \circ G(\tilde{u}_r))$ where k_s and k_n are the index of ground-truth label y_s and y_n respectively, and σ is the softmax activation.
    $\mathcal{L}_{CE} = -\log \hat{y}_s^{(k_s)} - \alpha \log \hat{y}_n^{(k_n)}$; $\mathcal{L}_v = |v_s - \hat{v}_s|$; $\mathcal{L}_u = |u_r - \hat{u}_r|$
    $\mathcal{L}_p = -\log(\exp(P(u_s|c_{k_s}))/\sum_{i=1}^{|\mathcal{C}_s|}\exp(P(u_s|c_i)))$, where $P(u_s|c_i) = \mathcal{N}(u_s|\mu_{c_i}, \Sigma_{c_i})$
    Update $\theta_{F_s}, \theta_D, \theta_G$ by minimizing $\mathcal{L}_{CE}$, $\mathcal{L}_v$, $\mathcal{L}_u$, and $\mathcal{L}_p$ alternatively using separate optimizers.
    if $(iter \,\%\, UpdateIter == 0)$ **then**
        Recompute μ_{c_i}, Σ_{c_i} for each source class c_i ; Generate $\mathcal{D}_n^{(b)}$ using the updated priors.

d) Minibatch negative sampling strategy. We create an unbiased batch of training samples for a training iteration by sampling equal number of positive and negative samples from the dataset. For *USFDA-a* we sample 32 positive images ($b_{+ve} = 32$) and 32 negative images per training iteration ($b_{-ve} = 32$). Similarly, for *USFDA-b* we sample 32 positive images and 32 latent-simulated negative samples. This gives an effective batch size of $b_{+ve} + b_{-ve} = 64$.

e) Use of multiple optimizers for training. In the presence of multiple loss terms, we subvert a time-consuming loss-weighting scheme search by making use of multiple Adam optimizers during training. Essentially, we define a separate optimizer for each loss term, and optimize only one of the losses (chosen in a round robin fashion) in each iteration of training. We use a learning rate of 0.0001 during training. Intuitively, the higher order moment parameters in the Adam optimizer adaptively scale the gradients as required by the loss landscape.

f) Label-Set Relationships. For Office-31 dataset in the UDA setting, we use the 10 classes shared by Office-31 and Caltech-256 as the shared label-set $\mathcal{C}$. These classes are: *back_pack, calculator, keyboard, monitor, mouse, mug, bike, laptop_computer, headphones, projector*. From the remaining classes, in alphabetical order, we choose the first 10 classes as source-private ($\overline{\mathcal{C}}_s$) classes, and the rest 11 as target-private ($\overline{\mathcal{C}}_t$) classes. For VisDA, alphabetically, the first 6 classes are considered $\mathcal{C}$, the next 3 as $\overline{\mathcal{C}}_s$ and the last 3 comprise $\overline{\mathcal{C}}_t$. The Office-Home dataset has 65 categories, of which we use the first 10 classes as $\mathcal{C}$, the next 5 for $\overline{\mathcal{C}}_s$, and the rest 50 classes as $\overline{\mathcal{C}}_t$.

B.2 Deployment Stage

The details of the architecture used during the *Deployment* stage are given in Table 7. Note that the Feature Decoder G used during the *Procurement* stage, is not available during the *Deployment* stage, restricting complete access to the source data.

Training during the *Deployment* stage. The only trainable component is the Feature Extractor F_t, which is initialized from F_s at *Deployment*. Here, the *SSM* is calculated by passing the target images through the network trained on source data (source model), i.e for each image x_t, we calculate $\hat{y}$ = *softmax*$(D \circ F_s \circ M(x_t))$. Note that the softmax is calculated over all $|\mathcal{C}_s| + |\mathcal{C}_n|$ classes. This is done by concatenating the outputs of D_{src} and D_{neg}, and then calculating softmax. Then, the *SSM* is determined by the exponential confidence of a target sample, where confidence is the highest softmax value in the categories in $|\mathcal{C}_s|$.

C Ablation studies and Additional results

C.1 Pretraining the backbone network on Places instead of ImageNet.

We find that widely adopted standard domain adaptation datasets such as Office-31 and VisDA often share a part or all of their label-set with ImageNet. Therefore, to validate our method's applicability when initialized from a network pretrained on an unrelated dataset, we attempt to solve the adaptation

Table 3: Evaluation of the proposed method on **A**→**D** task of Office-31 (Saenko et al., 2010) dataset, pretraining the ResNet-50 backbone (M) on Places instead of Imagenet. Note that, we set $|C|/|\mathcal{C}_s \cup \mathcal{C}_t| = 0.32$, similiar to the setting used in Table 2 of the main paper. Additionally, the last two columns of the table show a comparison between our method and UAN (You et al., 2019) with regard to the number of trainable parameters and total training time for adaptation

Method	ResNet-50 finetuning	Avg. per-class accuracy, $\mathcal{T}_{avg}$	Number of Trainable Parameters	Training time for Adaptation
UAN*	✓	60.98	26.7 Million	280s
UAN*	✗	52.48	5.6 Million	125s
USFDA-a	✗	**62.74**	**3.5 Million**	**44s**

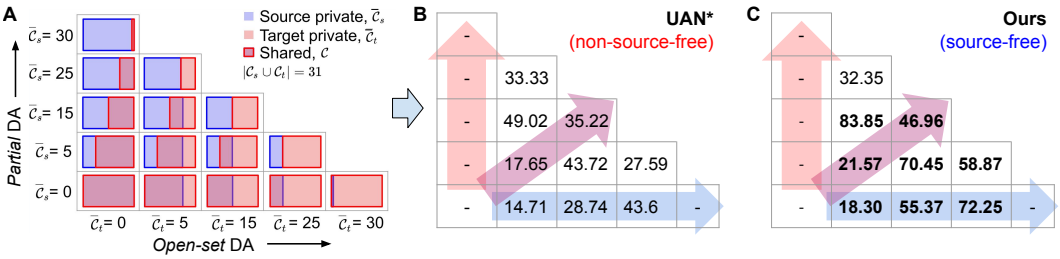

Figure 7: Comparison of target-unknown accuracy $\mathcal{T}_{unk}$ across varied label-set relationships for the task **A**→**D** in Office-31 dataset. **A)** Visual representation of label-set relationships and $\mathcal{T}_{unk}$ at the corresponding instances for **B)** UAN* (You et al., 2019) and **C)** ours *source-free* model. Effectively, the direction along x-axis (blue horizontal arrow) characterizes increasing *Open-set* complexity. The direction along y-axis (red vertical arrow) shows increasing complexity of *Partial* DA scenario. And the pink diagonal arrow denotes the effect of decreasing shared label space.

task **A**→**D** in Office-31 dataset by pretraining the ResNet-50 backbone on **Places** dataset (Zhou et al., 2017). In Table 3 it can be observed that our method outperforms even *source-dependent* methods (e.g. UAN (You et al., 2019), which is also initialized a ResNet-50 backbone pretrained on Places dataset). In contrast to our method, the algorithm in UAN involves ResNet-50 finetuning. Therefore, we also compare against a variant of UAN with a frozen backbone network, by inserting an additional feature extractor that operates on the features extracted from ResNet-50 (similar to F_s in the proposed method). The architecture of the feature extractor used for this variant of UAN is outlined in Table 6. We observe that our method significantly outperforms this variant of UAN with lesser number of trainable parameters (see Table 3).

C.2 Space and Time Complexity Analysis.

On account of keeping the weights of the backbone network frozen throughout the training process, and devoid of ad-hoc networks such as adversarial discriminator our method makes use of significantly lesser trainable parameters when compared to previous methods such as UAN (See Table 3). Devoid of adversarial training, the proposed method also has a significantly lesser total training time for adaptation: 44 sec versus 280 sec in UAN (for the **A**→**D** task of Office-31 and batch size of 32). Therefore, the proposed framework offers a much simpler adaptation pipeline, with a superior time and space complexity and at the same time achieves state-of-the-art domain adaptation performance across different datasets, even without accessing labeled source data at the time of adaptation (See Table 3). This corroborates the superiority of our method in real-time deployment scenarios.

C.3 Varying Label-Set Relationship

In addition to the $\mathcal{T}_{avg}$ reported in Fig. 6 in the paper, we also compare the target-unknown accuracy $\mathcal{T}_{unk}$ for UAN* and our pipeline. The results are presented in Figure 7. Refer the link to the code provided in the submission for details of the chosen class labels for each adaptation scenario shown in Figure 7. Clearly, our method achieves a statistically significant improvement on most of the label-set

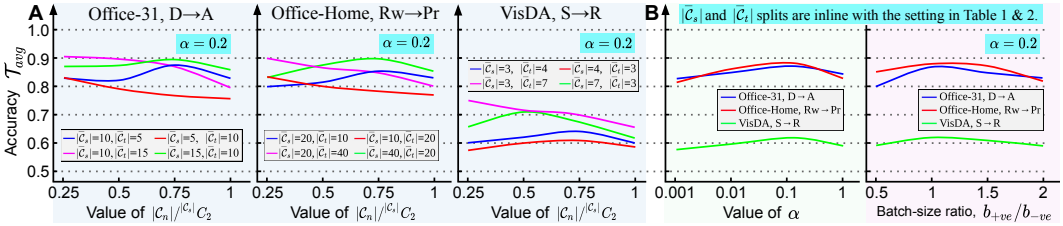

Figure 8: **A.** Sensitivity against $|\mathcal{C}_n|$, represented by $|\mathcal{C}_n|/^{|\mathcal{C}_s|}C_2$ for varying $|\bar{C}_s|$ or $|\bar{C}_t|$ (see fig. legend) by fixing the others (top cyan box), across varied datasets. **B.** Sensitivity against α and batch-size ratio (fixed $b_{+ve} + b_{-ve} = 64$). Note the scale of X and Y-axis.

Table 4: Accuracy (%) on unsupervised closed-set DA (all use *ResNet50*). Ours is w/o hyperparmeter tuning. Refer Section C.5.

Closed-set DA methods	source-free	Universal-DA	Office-31							VisDA
			D→A	A→D	A→W	W→D	W→A	D→W	Avg.	S → R
DAN (ICML'15)	✗	✗	63.6	78.6	80.5	99.6	62.8	97.1	80.4	61.1
ADDA (CVPR'17)	✗	✗	69.5	77.8	86.2	98.4	68.9	96.2	82.8	-
CDAN (NeurIPS'18)	✗	✗	70.1	89.8	93.1	100	68.0	98.2	86.5	66.8
UAN (CVPR'19)	✗	✓	68.4	85.3	81.2	99.1	69.7	98.1	83.6	-
Ours ***USFDA-a*** (*source-free*)	✓	✓	70.4	85.4	81.6	98.0	69.4	98.4	83.9	59.8

relationships over UAN. This demonstrates the capability of our algorithm to detect outlier classes more efficiently than UAN, which can be attributed to the ingeniously developed *Procurement* stage.

C.4 Sensitivity Analysis

In all our experiments (across datasets as in Tables 1 and 2 and across varied label-set relationships as in Fig. 6), we fix the hyperparameters as, $\alpha = 0.2$, $\beta = 0.1$, $|\mathcal{C}_n| = {}^{|\mathcal{C}_s|}C_2$ and $b_{+ve}/b_{-ve} = 1$. As mentioned in Section 4.3, one can treat these hyperparameters as global constants. In Fig. 8 we demonstrate the sensitivity of the model to these hyperparameters. Specifically, in Fig. 8A we show the sensitivity of the adaptation performance, to the choice of $|\mathcal{C}_n|$ during the *Procurement* stage, across a spectrum of label-set relationships. In Fig. 8B we show the sensitivity of the model to α and the batch-size ratio b_{+ve}/b_{-ve}. Sensitivity to β is shown in Fig. 5. Clearly, the model achieves a reasonably low sensitivity to the hyperparameters, even in the challenging *source-free* scenario.

C.5 Closed-set adaptation

We additionally evaluate our method in the unsupervised closed set adaptation scenario. In Table 4 we compare with the closed set domain adaptation methods DAN (Long et al., 2015), ADDA (Tzeng et al., 2017), CDAN (Long et al., 2018) and the universal domain adaptation method UAN (You et al., 2019). Note that, DAN, ADDA and CDAN rely on the assumption of a shared label space between the source and the target, and hence are not suited for a universal setting. Furthermore, all other methods require an explicit retraining on the source data during adaptation to perform well, even in the closed-set scenario. This clearly establishes the superiority of our method in the *source-free* setting.

C.6 Accuracy on source dataset post procurement

We observe in our experiments that the accuracy on the source samples does not drop as a result of the partially generative framework. For the experiments conducted in Fig. 5C, we observe similar classification accuracy on the source validation set, on increasing the number of negative classes from 0 to 190. This effect can be attributed to a carefully chosen $\alpha = 0.2$, which is deliberately biased towards positive source samples to help maintain the discriminative power of the model even in the presence of class imbalance (*i.e.* $|\mathcal{C}_n| \gg |\mathcal{C}_s|$). This enhances the model's generative ability without compromising on the discriminative capacity on the positive source samples.

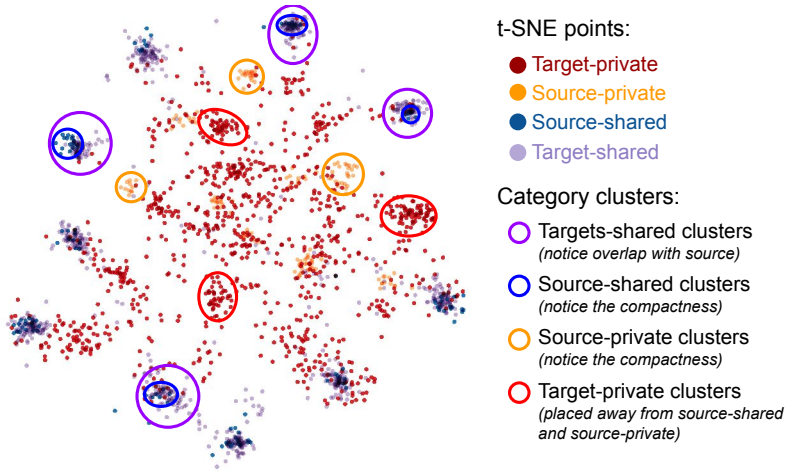

Figure 9: t-SNE plot showing placement of all the four clusters computed after adaptation for the task **A**→**D** in Office-31. It validates our hypothesis in both *Procurement* and *Deployment* stages as shown by the highlighted clusters and the corresponding inferences in the legend under "Category clusters".

C.7 Incremental one-shot classification

In universal adaptation, we seek to transfer the knowledge of "*class separability*" obtained from the source domain to the deployed target environment. More concretely, it is attributed to the segregation of data samples based on an expected characteristics, such as classification of objects according to their pose, color, or shape etc. To quantify this, we consider an extreme case where $\mathcal{C}_s \cap \mathcal{C}_t = \emptyset$ (**A**→**D** in Office-31 with $|\mathcal{C}_s| = 15$, $|\mathcal{C}_t| = 16$). Considering access to a single labeled target sample from each target category in $\overline{\mathcal{C}}_t = \mathcal{C}_t$, which are denoted as $x_t^{c_j}$, where $j = 1, 2, .., |\mathcal{C}_t|$, we perform one-shot Nearest-Neighbour based classification by obtaining the predicted class label as $\hat{c}_t = \text{argmin}_{c_j} ||F_t \circ M(x_t) - F_t \circ M(x_t^{c_j})||_2$. Then, the classification accuracy for the entire target set is computed by comparing $\hat{c}_t$ with the corresponding ground-truth category. We obtain 64.72% accuracy for the proposed framework as compared to 13.43% for UAN* (You et al., 2019). A higher accuracy indicates that, the samples are inherently clustered in the intermediate feature level $M \circ F_t(x_t)$ validating an efficient transfer of "*class separability*" in a fully unsupervised manner.

C.8 Feature Space Visualization

We obtain a t-SNE plot at the intermediate feature level u for both target and source samples (see Figure 9), where the embedding for the target samples is obtained as $u_t = F_t \circ M(x_t)$ and the same for the source samples is obtained as $u_s = F_s \circ M(x_s)$. This is because we aim to learn domain-specific features in contrast to domain-agnostic features as a result of the restriction imposed by the *source-free* scenario ("cannot disturb placement of source clusters"). Firstly we obtain compact clusters for the source-categories as a result of the partially generative *Procurement* stage. Secondly, the target-private clusters are placed away from the source-shared and source-private as expected as a result of the carefully formalized *SSM* weighting scheme in the *Deployment* stage. This plot clearly validates our hypothesis.

D Miscellaneous

D.1 Specifications of Computing Resources

For both *Procurement* and *Deployment* stages, we make use of the machine with the specifications mentioned in Table 8. The architecture is developed and trained in **Python 2.7** with **PyTorch 1.0.0**.

Table 5: Network architecture for *Procurement* stage. Hyperparameter $\alpha = 0.2$

Component	Trainable?	Operation	Notation	Features	Batch Norm?	Non-Linearity
Resnet-50 (Upto AvgPool layer)	✗		M	2048		
Feature Extractor	✓		F_s	256		
		Input		2048	✗	
		Fully connected		1024	✗	ELU
		Fully connected		1024	✓	ELU
		Fully connected		256	✗	ELU
		Fully connected		256	✓	ELU
Feature Decoder	✓		G	2048		
		Input		256	✗	
		Fully connected		1024	✗	ELU
		Fully connected		1024	✓	ELU
		Fully connected		2048	✗	ELU
		Fully connected		2048	✗	-
Classifier	✓		D	$\lvert\mathcal{C}_s\rvert + \lvert\mathcal{C}_n\rvert$		
		Input		256	✗	
		Fully connected	D_{src}	$\lvert\mathcal{C}_s\rvert$	✗	
		Input		256	✗	
		Fully connected	D_{neg}	$\lvert\mathcal{C}_n\rvert$	✗	

Table 6: Feature Extractor Architecture used for training UAN (You et al., 2019) under the "no ResNet-50 finetuning" case (Refer Table 3 and Section C.1)

Operation	Features	Non-Linearity
Input	2048	
Fully connected	512	ReLU
Fully connected	256	ReLU
Fully connected	512	ReLU
Fully connected	2048	ReLU

Table 7: Network architecture for *Deployment* stage. Hyperparameter $\beta = 0.1$

Component	Trainable?	Operation	Notation	Features	Batch Norm?	Non-Linearity
Resnet-50 (Upto AvgPool layer)	✗		M	2048		
Feature Extractor	✓		F_t	256		
		Input		2048	✗	
		Fully connected		1024	✗	ELU
		Fully connected		1024	✓	ELU
		Fully connected		256	✗	ELU
		Fully connected		256	✓	ELU
Classifier	✗		D	$\lvert\mathcal{C}_s\rvert + \lvert\mathcal{C}_n\rvert$		
		Input		256	✗	
		Fully connected	D_{src}	$\lvert\mathcal{C}_s\rvert$	✗	
		Input		256	✗	
		Fully connected	D_{neg}	$\lvert\mathcal{C}_n\rvert$	✗	

Table 8: Specifications of the machine used for both *Procurement* and Deployment stages

CPU	GPU	RAM	VRAM	CUDA
Intel i7-7700K	NVIDIA GeForce GTX 1080 Ti	32 GB	11 GB	V8.0.61

D.2 REFERENCES TO CODE

Proposed Method. Our complete documented code (including data loaders, training pipeline etc.) used for running the experiments is available for reproducibility (refer to the private comment containing the link). Details of dataset splits can be found in Section B.1. For evaluating UAN (You et al., 2019), we execute the official implementation provided by the authors on github[1].

Negative dataset creation. We have provided the complete dataset with augmentations and negative images for the task **A**→**D** in Office-31 (Saenko et al., 2010), along with the negative dataset creation tool (refer to the code link).

[1] UAN (You et al., 2019): https://github.com/thuml/Universal-Domain-Adaptation