Supplementary Material – PUnDA: Probabilistic Unsupervised Domain Adaptation for Knowledge Transfer Across Visual Categories

Behnam Gholami
Department of Computer Science
Rutgers University, Piscataway, NJ, USA
bb510@cs.rutgers.edu

Ognjen (Oggi) Rudovic
MIT Media Lab, Cambridge, MA, USA
orudovic@mit.edu

Vladimir Pavlovic
Department of Computer Science
Rutgers University, Piscataway, NJ, USA
vladimir@cs.rutgers.edu

1. Introduction

This Supplement contains additional materials related to the paper **PUnDA: Probabilistic Unsupervised Domain Adaptation for Knowledge Transfer Across Visual Categories**. In particular, in Sec. 2 we present additional results that qualitatively demonstrate the advantages of our adaptation framework over competing approaches such as the ILS [2]. We contrast 2D embeddings of the features adapted by **PUnDA** to those of the pre-adapted fc6 layer and the ILS. We include detailed derivation of the Variational Bayes algorithm for **PUnDA** in Sec. 4. Finally, in Sec. 5, we provide the computational complexity analysis of our VB algorithm.

2. Visualization experiments

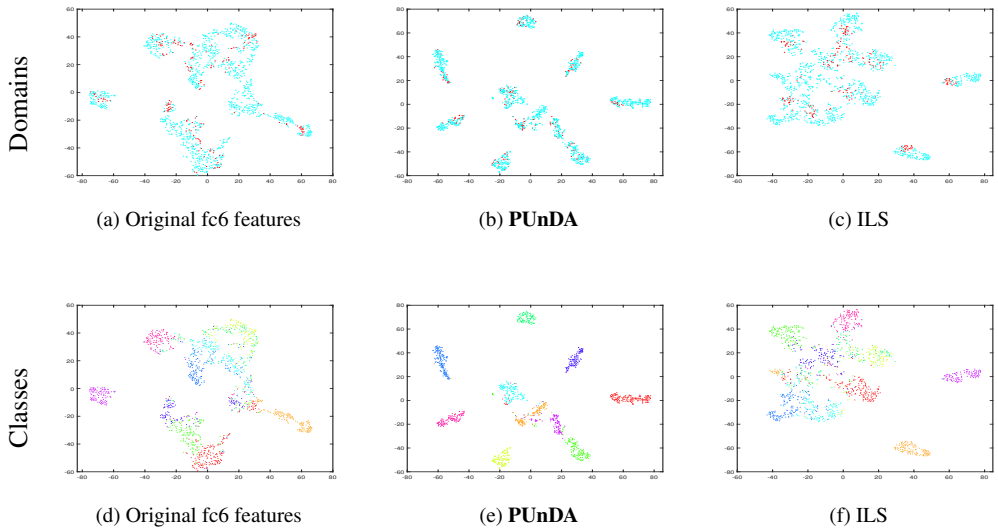
In this Section we provide additional results of experiments described in Sec. 4 of the main paper. Specifically, we show the 2D point clouds of the learned features of **PUnDA**, embedded using the t-SNE algorithm [3], and compare them to those of the ILS features and the standard, non-adapted features. Results will demonstrate that, as supported by the quantitative results in the main paper, joint optimization of the alignment and classification criteria, accomplished through **PUnDA**, leads to qualitatively superior domain matching **and** class separability, compared to competing approaches.

2.1. Office + Caltech10 Results

We refer here to the setting of the Office + Caltech10 experiments introduced in the main paper in Sec. 4.2. Figs 1,2,3 depict the embedded features extracted from the Office+Caltech10 dataset for the cases of $A \rightarrow C$, $W \rightarrow C$ and $D \rightarrow A$ adaptations, respectively. In the figures, blue and red colors indicate the source and target domains, respectively. Colors in the bottom rows correspond to different class instances.

The top row of the Figures illustrates how the features extracted using the MMD criteria (**PUnDA**, ILS) reduce the domain mismatch. Good features for domain adaptation should have a configuration where the red and blue colors are mixed. This effect can be seen in features extracted from the **PUnDA** (b) and ILS algorithms (c), which indicates that the domain mismatch is successfully reduced in the feature space, compared to the pre-aligned features in (a). Note that the domains obtained by **PUnDA** are more compact than those of the ILS, with increasingly matched source-target features.

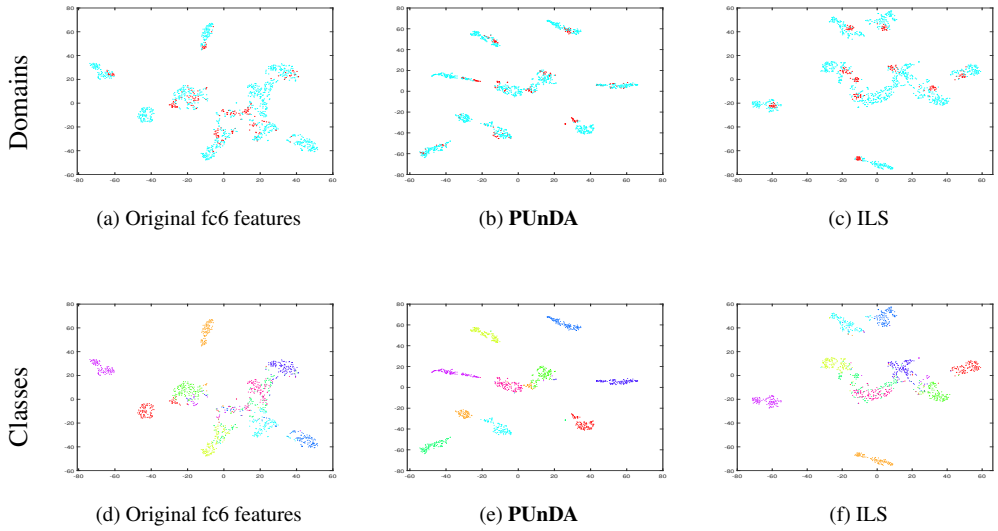
In classification, good domain-adapted features should display large class separability. The bottom row highlights a major difference between **PUnDA** features and the original features and the ILS features in terms of the class separability: the **PUnDA** features are more clustered with respect to the classes than ILS features, with more prominent gaps among clusters. This is partly due to the fact that **PUnDA** exploits the unlabeled data to learn the classifier boundaries for the source domain adapted discriminatively to the target domain by minimizing the expected classification errors on the target domain, something that ILS fails to account for. As a consequence, **PUnDA** framework leads to superior cross-domain classification performance.



(a) Original fc6 features (b) **PUnDA** (c) ILS

(d) Original fc6 features (e) **PUnDA** (f) ILS

Figure 1. Feature visualization for embedding of $A \rightarrow C$ (Office+Caltech10) data samples using t-sne algorithm. The top and bottom rows show the domains and classes respectively. (a,d) Original features. (b,e) **PUnDA** features. (c,f) ILS features.



(a) Original fc6 features (b) **PUnDA** (c) ILS

(d) Original fc6 features (e) **PUnDA** (f) ILS

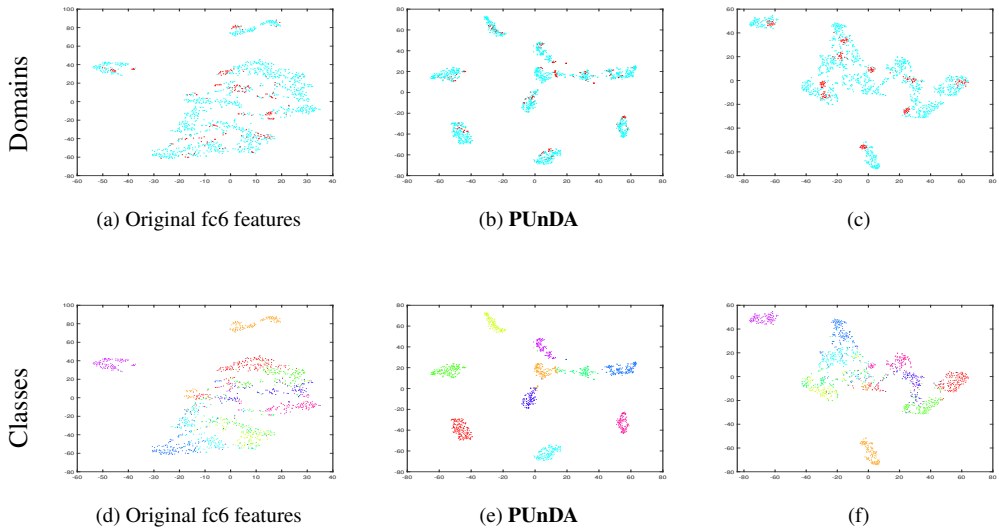
Figure 2. Feature visualization. Embedding of $W \rightarrow C$ (Office+Caltech10) data samples using t-sne algorithm. The top and bottom rows show the domains and classes respectively. (a,d) Original features. (b,e) **PUnDA** features. (c,f) ILS features.

3. Multi-PIE Results

We refer here to the setting of the Multi-PIE experiments introduced in the main paper in Sec. 4.3. Figs 4 and 5 depict the embedded features extracted from the Multi-PIE dataset for the cases of $C27 \rightarrow C05$, and $C27 \rightarrow C37$ adaptations, respectively. In the figures, blue and red colors indicate the source and target domains, respectively. Colors in the bottom rows correspond to different class instances.

The top row of the Figures illustrates how the features extracted using the MMD criteria (**PUnDA**, ILS) reduce the domain mismatch. Again, it can be seen that the domains obtained by **PUnDA** are more compact than those of the ILS, with increasingly matched source-target features.

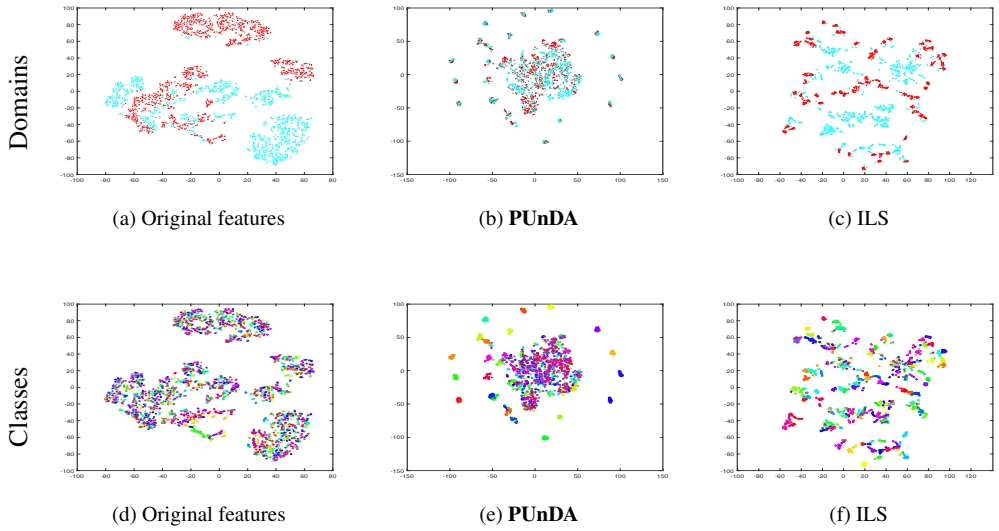
The bottom row highlights a difference between **PUnDA** features and the original features and the ILS features in terms of the class separability: For the Multi-PIE dataset both the **PUnDA** features and **ILS** features are slightly more clustered with



(a) Original fc6 features (b) **PUnDA** (c)

(d) Original fc6 features (e) **PUnDA** (f)

Figure 3. Feature visualization for embedding of $D \rightarrow A$ (Office+Caltech10) data samples using t-sne algorithm. The top and bottom rows show the domains and classes respective). (a,d) Original features. (b,e) **PUnDA** features. (c,f) ILS features.



(a) Original features (b) **PUnDA** (c) ILS

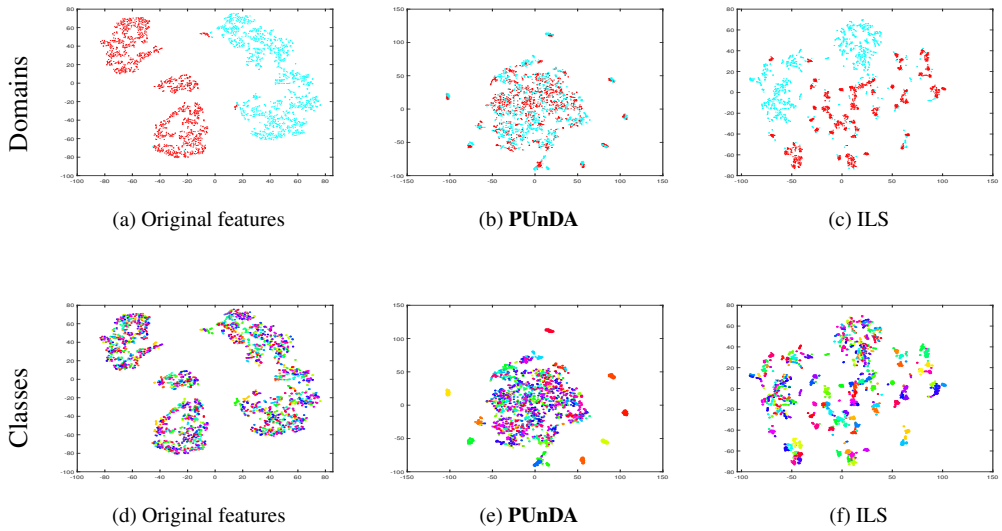
(d) Original features (e) **PUnDA** (f) ILS

Figure 4. Feature visualization for embedding of $C27 \rightarrow C05$ (Multi-PIE) data samples using t-sne algorithm. The top and bottom rows show the domains and classes respective). (a,d) Original features. (b,e) **PUnDA** features. (c,f) ILS features.

respect to the classes than the original features.

4. VB Algorithm for PUnDA

In this section we describe, in detail, the Variational Bayes algorithm at the core of **PUnDA**. We first explicitly define the model and the objective function and then show how to optimize this function by tackling two key challenges of (i) non-conjugacy between the regularizer and the Gaussian distribution used in the model, and (ii) the non-conjugacy between the softmax function and the Gaussian distribution. The optimization approach, formulated as an EM algorithm, utilizes two bounds on the expectation of the softmax function described below.



(a) Original features (b) **PUnDA** (c) ILS

(d) Original features (e) **PUnDA** (f) ILS

Figure 5. Feature visualization for embedding of $C27 \rightarrow C37$ (Multi-PIE) data samples using t-sne algorithm. The top and bottom rows show the domains and classes respective). (a,d) Original features. (b,e) **PUnDA** features. (c,f) ILS features.

The proposed model can be formally defined as

$$P(s_i) \sim \mathcal{N}(0, I_K), \quad i = 1, ..., N \tag{1}$$

$$P(s'_j) \sim \mathcal{N}(0, I_K), \quad j = 1, ..., M \tag{2}$$

$$P(\pi_k|a, b, K) \sim Beta(a/K, b(K-1)/K), \quad k = 1, ..., K \tag{3}$$

$$P(\phi_k) \sim \mathcal{N}(0, I_d), \quad k = 1, ..., K \tag{4}$$

$$P(\phi'_k) \sim \mathcal{N}(0, I_d), \quad k = 1, ..., K \tag{5}$$

$$P(z_k|\pi_k) \sim Bernoulli(\pi_k), \quad k = 1, ..., K \tag{6}$$

$$P(w_c) \sim \mathcal{N}(0, I_K), \quad c = 1, ..., C \tag{7}$$

$$P(\gamma_s|c_1, d_1) \sim Gamma(c_1, d_1) \tag{8}$$

$$P(\gamma_t|c'_1, d'_1) \sim Gamma(c'_1, d'_1) \tag{9}$$

$$P(x_i|\mathbf{\Phi}, Z, s_i, \gamma_s) \sim \mathcal{N}(\mathbf{\Phi}^\top(Z \odot s_i), \gamma_s^{-1} I_d), \quad i = 1, ..., N \tag{10}$$

$$P(x'_j|\mathbf{\Phi}', Z, s'_i, \gamma_s) \sim \mathcal{N}(\mathbf{\Phi}'^\top(Z \odot s'_i), \gamma_t^{-1} I_d), \quad j = 1, ..., M \tag{11}$$

$$P(y_i = c|\boldsymbol{W}, Z, s_i) = \frac{e^{w_c^\top (Z \odot s_i)}}{\sum_{c'} e^{w_{c'}^\top (Z \odot s_i)}}, \quad i = 1, ..., N, \ c = 1, ..., C \tag{12}$$

For our framework to yield a computationally effective inference method, we use a factorized variational distribution with the following forms:

$$q(s_i) \sim \mathcal{N}(\mu_i^s, \beta_{s_i}^{-1} I), \quad i = 1, ..., N \tag{13}$$

$$q(s'_j) \sim \mathcal{N}(\mu_j^{s'}, \beta_{s'_j}^{-1} I) \quad j = 1, ..., M \tag{14}$$

$$q(\pi_k) \sim Beta(a_k, b_k), \quad k = 1, ..., K \tag{15}$$

$$q(\phi_k) \sim \mathcal{N}(\mu_k^\phi, \beta_{\phi_k}^{-1} I_d), \quad k = 1, ..., K \tag{16}$$

$$q(\phi'_k) \sim \mathcal{N}(\mu_k^{\phi'}, \beta_{\phi'_k}^{-1} I_d), \quad k = 1, ..., K \tag{17}$$

$$q(z_k) \sim Bernoulli(\rho_k), \quad k = 1, ..., K \tag{18}$$

$$q(w_c) \sim \mathcal{N}(\mu_c^w, \beta_{w_c}^{-1} I_K), \quad c = 1, ..., C \tag{19}$$

$$q(\gamma_t) \sim Gamma(c_t, d_t) \tag{20}$$

$$q(\gamma_s) \sim Gamma(c_s, d_s) \tag{21}$$

We need to solve the following optimization problem

$$\boldsymbol{\Delta}^* = \arg\max_{\boldsymbol{\Delta}} \mathbb{E}_q\big[\log(\boldsymbol{X}, Y, \boldsymbol{X}', \boldsymbol{\Omega}|\boldsymbol{\Theta})\big] + \boldsymbol{H}[q(\boldsymbol{\Omega})] - \lambda \mathcal{L}(\boldsymbol{S}, \boldsymbol{S}') + \lambda' \mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z), \tag{22}$$

where $\boldsymbol{\Delta} = \{\{\mu_i^s, \beta_{s_i}\}, \{\mu_j^s, \beta_{s'_j}\}, \{a_k, b_k\}, \{\mu_k^\phi\}, \beta_\phi, \{\mu_k^{\phi'}\}, \beta_{\phi'}, \{\rho_k\}, \{\mu_c^w\}, \beta_w, c_t, d_t, c_s, d_s\}$.

The key terms in this objective are defined as follows. $\mathbb{E}_q\big[\log(\boldsymbol{X}, Y, \boldsymbol{X}', \boldsymbol{\Omega}|\boldsymbol{\Theta})\big]$ can be expanded as

$$\begin{aligned}
&\mathbb{E}_q\big[\log(\boldsymbol{X}, Y, \boldsymbol{X}', \boldsymbol{\Omega}|\boldsymbol{\Theta})\big] = -\sum_i \frac{1}{2}\mathbb{E}_q[\gamma_s||x_i - \boldsymbol{\Phi}^\top(Z \odot s_i)||^2] + \frac{dN}{2}\log(\gamma_s) + \frac{dM}{2}\log(\gamma_t) \\
&- \sum_j \frac{1}{2}\mathbb{E}_q[\gamma_s||x'_j - \boldsymbol{\Phi}'^\top(Z \odot s'_j)||^2] + \sum_k \mathbb{E}_q[z_k \log \pi_k] + \mathbb{E}_q[(1 - z_k)\log(1 - \pi_k)] + \sum_i \sum_c \mathbf{1}[y_i = c]\mathbb{E}_q[w_c^\top (Z \odot s_i)] \\
&- \sum_i \sum_c \mathbf{1}[y_i = c]\mathbb{E}_q\left[\log\left(\sum_{c'} e^{w_{c'}^\top (Z \odot s_i)}\right)\right] - \frac{1}{2}\sum_i \mathbb{E}_q||s_i||^2 - \frac{1}{2}\sum_j \mathbb{E}_q||s'_j||^2 - \frac{1}{2}\sum_k \mathbb{E}_q||\phi_k||^2 - \frac{1}{2}\sum_k \mathbb{E}_q||\phi'_k||^2 \\
&- \frac{1}{2}\sum_c \mathbb{E}_q||w_c||^2 + \sum_k \left(\frac{a}{K} - 1\right)\mathbb{E}[\log \pi_k] + \left(\frac{b(K-1)}{K} - 1\right)\mathbb{E}[\log(1 - \pi_k)] + (c_1 - 1)\mathbb{E}_q[\log \gamma_s] - d_1 \mathbb{E}_q[\gamma_s] \\
&+ (c_2 - 1)\mathbb{E}_q[\log \gamma_t] - d_2 \mathbb{E}_q[\gamma_t].
\end{aligned} \tag{23}$$

$\boldsymbol{H}[q(\boldsymbol{\Omega})]$ is subsequently expanded as

$$\begin{aligned}
&\boldsymbol{H}[q(\boldsymbol{\Omega})] = -\frac{k}{2}\sum_i \log \beta_{s_i} - \frac{k}{2}\sum_j \log \beta_{s'_j} - \frac{dK}{2}\log \beta_\phi - \frac{dK}{2}\log \beta_{\phi'} - \frac{KC}{2}\log \beta_w + \log \Gamma(a_k) + \log \Gamma(b_k) \\
&- \log(a_k + b_k) - (a_k - 1)\psi(a_k) - (b_k - 1)\psi(b_k) + (a_k + b_k - 2)\psi(a_k + b_k) + c_s - \log d_s + \log \Gamma(c_s) \\
&+ (1 - c_s)\psi(c_s) + c_t - \log d_t + \log \Gamma(c_t) + (1 - c_t)\psi(c_t) - \sum_k \big[(1 - \rho_k)\log \rho_k + \rho_k \log(1 - \rho_k)\big],
\end{aligned} \tag{24}$$

where ψ denotes the **digamma** function defined as

$$\psi(x) = \frac{d}{dx}\log \Gamma(x), \tag{25}$$

and $\Gamma(x)$ denotes the Gamma function defined as

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du. \tag{26}$$

$\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}')$ is computed as

$$\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}') = \sum_{i,i'} \frac{\mathcal{K}(q(s_i), q(s_j))}{N^2} - 2\sum_{i,j} \frac{\mathcal{K}(q(s_i), q(s'_j))}{NM} + \sum_{j,j'} \frac{\mathcal{K}(q(s'_j), q(s'_{j'}))}{M^2}, \tag{27}$$

where $\mathcal{K}(q(s), q(s'))$ is computed as

$$\mathcal{K}(q(s), q(s')) = \log \int_{\mathbb{R}^K} q(s)^{1/2} q(s')^{1/2}\, ds\, ds'. \tag{28}$$

Since $q(s)$ and $q(s')$ are Gaussian distributions, we can compute (28) in closed form as

$$\mathcal{K}(q(s), q(s')) = \frac{\beta_s + \beta_{s'}}{4\beta_s\beta_{s'}}\|\mu_s - \mu_t\|^2 - \frac{K}{4}\log\beta_s - \frac{K}{4}\log\beta_{s'} + \frac{K}{2}\log(\beta_s + \beta_{s'}). \tag{29}$$

$\mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z)$ is computed as

$$\begin{aligned}\mathcal{L}'(\boldsymbol{W}, \boldsymbol{S}', Z) &= \sum_j \sum_c \mathbb{E}_{P(y'_j|\boldsymbol{W}, s'_j, Z)} \log P(y'_j = c) \\ &= \sum_j \sum_c \left[\mathbb{E}_q[w_c^\top (Z \odot s'_j)] - \mathbb{E}_q[\log \sum_{c'} exp(w_{c'}^\top (Z \odot s'_j))]\right] \left[\frac{exp(w_c^\top (Z \odot s'_j))}{\sum_{c'} exp(w_{c'}^\top (Z \odot s'_j))}\right].\end{aligned} \tag{30}$$

There are two main issues which make deriving the VB update equations in closed-form intractable: (i) Non-conjugacy between the regularizer $\mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z)$ and the Gaussian distribution, and (ii) Non-conjugacy between the softmax function and the Gaussian distribution.

To tackle the first issue, it is worth noting that $\mathcal{L}'(\boldsymbol{W}, \boldsymbol{S}', Z)$ is equivalent to the sum of the negative Shanon Entropy of the C-dimensional probability vectors $P(y'_j|\boldsymbol{W}, Z, s_j) = [p_j^1, ..., p_j^C]$. Hence, we propose to use Renyi entropy [4] rather than Shanon entropy for defining the regularizer $\mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z)$. The Renyi entropy of order α, where $\alpha \geq 0$ and $\alpha \neq 1$ is defined as

$$\boldsymbol{H}_\alpha[y'_j] = \frac{1}{1-\alpha}\log\left(\sum_{c=1}^{C}(p_c^j)^\alpha\right) \tag{31}$$

The intuition behind using the Renyi entropy is that the limiting value of $\boldsymbol{H}_\alpha[y'_j]$ as $\alpha \to 1$ is the Shanon entropy [4]. We set $\alpha = 0.99$ in the experiments to approximate the Shanon entropy with the Renyi entropy. By replacing the Shanon entropy with the Renyi entropy in (30), we have

$$\mathcal{L}'(\boldsymbol{W}, \boldsymbol{S}', Z) = \frac{1}{1-\alpha}\sum_j \mathbb{E}_q[\log \sum_c e^{\alpha w_c^\top (Z \odot s'_j)}] - \frac{\alpha}{1-\alpha}\sum_j \mathbb{E}_q[\log \sum_c e^{w_c^\top (Z \odot s'_j)}]. \tag{32}$$

To tackle the second non-conjugacy problem, we use a linear lower bound and a quadratic upper bound on the expectation of the log-sum function introduced in [1]:

$$\log \sum_{c=1}^{C} e^{w_c^\top (Z \odot s)} \geq \sum_{c=1}^{C} w_c^\top (Z \odot s) \tag{33}$$

$$log \sum_{c=1}^{C} e^{w_c^\top (Z \odot s)} \leq \sum_{c=1}^{C}\left[\eta(\xi_c)\big((w_c^\top (Z \odot s))^2 - \xi_c^2\big) - \log\sigma(\xi_c)\right] + \frac{1}{2}\sum_{c=1}^{C} w_c^\top (Z \odot s) + \xi_c, \tag{34}$$

where

$$\eta(\xi_c) = -\frac{1}{2\xi_c}\left(\frac{1}{1+e^{-\xi_c}} - \frac{1}{2}\right) \tag{35}$$

and $\{\xi_c \in [0, +\infty]\}$ denote the free variational parameters which are optimized to get the tightest possible bound. Hence, we replace $\log\sum_c \exp(\alpha w_c^\top (Z \odot s'_j))$ with its lower bound and $\log\sum_{c'} \exp(w_{c'}^\top (Z \odot s_i))$ and $\log\sum_c \exp(w_c^\top (Z \odot s'_j))$ with their upper bound in the objective function, then we use the EM algorithm to optimize the factorized variational distribution and the free parameters which computes the variational posterior distribution in the E-step and maximizes the free parameters in the M-step, which goes as follows.

E step:

In this step the free variational parameters $\{\xi_c\}$ are fixed, and the variational distributions are updated by maximizing the objective function using a coordinate ascent algorithm.

Update for γ_s:

One can show that the posterior parameters c_s, d_s can be updated as

$$c_s = c_1 + \frac{Nd}{2}, \quad d_s = d_1 + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{N}\epsilon_{ij}, \tag{36}$$

where

$$\epsilon_{ij} = x_{ij}^2 - 2x_{ij}\langle\phi_k\rangle^\top\big(\langle Z\rangle\odot\langle s_j\rangle\big) + \Big(\langle\phi_k\rangle^\top\big(\langle Z\rangle\odot\langle s_j\rangle\big)\Big)^2 + \sum_{k=1}^{K}\langle\phi_{ik}^2\rangle\big(\langle z_k^2\rangle\odot\langle s_{kj}^2\rangle\big) - \sum_{k=1}^{K}\langle\phi_{ik}\rangle^2\big(\langle z_k\rangle^2\odot\langle s_{kj}\rangle^2\big) \quad (37)$$

and $\langle . \rangle$ indicates the expectation operator. **Update for γ_t**:

One can show that the posterior parameters c_t, d_t can be updated as

$$c_t = c_1' + \frac{Md}{2}, \quad d_s = d_1' + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{M}\epsilon_{ij}, \quad (38)$$

where

$$\epsilon_{ij} = {x'_{ij}}^2 - 2x'_{ij}\langle\phi'_k\rangle^\top\big(\langle Z\rangle\odot\langle s'_j\rangle\big) + \Big(\langle\phi'_k\rangle^\top\big(\langle Z\rangle\odot\langle s'_j\rangle\big)\Big)^2 + \sum_{k=1}^{K}\langle{\phi'_{ik}}^2\rangle\big(\langle z_k^2\rangle\odot\langle {s'_{kj}}^2\rangle\big) - \sum_{k=1}^{K}\langle\phi'_{ik}\rangle^2\big(\langle z_k\rangle^2\odot\langle s'_{kj}\rangle^2\big) \quad (39)$$

Update for $\Pi = [\pi_1, ..., \pi_K]$:

One can show that the posterior parameters a_k, b_k can be updated as

$$a_k = \frac{a}{K} + \langle z_k\rangle, \;\; b_k = \frac{b(K-1)}{K} + 1 - \langle z_k\rangle. \quad (40)$$

Update for $\mathbf{\Phi} = [\phi_1, ..., \phi_K]$:

One can show that the posterior parameters $\mu_k^{\phi}, \beta_{\phi_k}$ can be updated as

$$\beta_{\phi_k} = \langle\gamma_s\rangle\sum_{i=1}^{N}\langle z_k\rangle\langle s_{ki}^2\rangle + 1/2, \qquad \mu_k^{\phi} = \frac{\langle\gamma_s\rangle}{\beta_{\phi_k}}\sum_{i=1}^{N}\langle z_k\rangle\langle s_{ki}\rangle\langle x_i^{-k}\rangle, \quad (41)$$

where x_i^{-k} is defined as

$$x_i^{-k} \equiv x_i - \mathbf{\Phi}_{-k}^\top(z^{-k}\odot s_i^{-k}) \quad (42)$$

and Φ_{-k}, z^{-k} and s_i^{-k} are the matrix/vectors with the k-th column/element removed.

Update for $\mathbf{\Phi}' = [\phi'_1, ..., \phi'_K]$:

One can show that the posterior parameters $\mu_k^{\phi'}, \beta_{\phi'_k}$ can be updated as

$$\beta_{\phi'_k} = \langle\gamma_s\rangle\sum_{j=1}^{M}\langle z_k\rangle\langle {s'_{kj}}^2\rangle + 1/2, \qquad \mu_k^{\phi'} = \frac{\langle\gamma_t\rangle}{\beta_{\phi'_k}}\sum_{j=1}^{M}\langle z_k\rangle\langle s_{kj}\rangle\langle {x'_j}^{-k}\rangle \quad (43)$$

where ${x'_j}^{-k}$ is defined as

$$ {x'_j}^{-k} \equiv x'_j - {\mathbf{\Phi}'_{-k}}^\top(z^{-k}\odot {s'_j}^{-k}) \quad (44)$$

Update for $Z = [z_1, ..., z_K]$:

One can show that the posterior parameter ρ_k can be updated as

$$\rho_k = \Big(1 + e^{\langle\log\pi_k\rangle + \langle\log(1-\pi_k)\rangle + \tau_s + \tau_t}\Big)^{-1}, \quad (45)$$

where

$$\begin{aligned}\tau_s = &-\frac{\langle\gamma_s\rangle}{2}\Big[-2\sum_{i=1}^{N}\langle s_{ki}\rangle\sum_{j=1}^{d}\big(\langle\phi_{jk}\rangle\sum_{k'\neq k}^{K}\langle\phi_{jk'}\rangle\langle s_{k'j}\rangle\langle z_{k'}\rangle\big)\Big] + \sum_{i=1}^{N}\sum_{c=1}^{C}\big(1[y_i = c] - \frac{1}{2}\big)\langle w_c\rangle\langle s_{ki}\rangle \\ &- \sum_{i=1}^{N}\sum_{c=1}^{C}\eta(\xi_c)\Big[\langle w_c^2\rangle\langle s_{ki}^2\rangle + 2\langle w_c\rangle\langle s_{ki}\rangle\langle w_c^{-k}\rangle(\langle Z^{-k}\rangle\odot\langle s_i^{-k}\rangle)\Big] \end{aligned} \quad (46)$$

$$\tau_t = -\frac{\langle\gamma_t\rangle}{2}\Big[-2\sum_{j=1}^{M}\langle s'_{kj}\rangle\sum_{i=1}^{d}\big(\langle\phi'_{ik}\rangle\sum_{k'\neq k}^{K}\langle\phi'_{ik'}\rangle\langle s'_{k'j}\rangle\langle z_{k'}\rangle\big)\Big] + \sum_{j=1}^{M}\sum_{c=1}^{C}\frac{\lambda'\alpha}{2-2\alpha}\langle w_c\rangle\langle s_{ki}\rangle$$
$$-\frac{\lambda'\alpha}{1-\alpha}\sum_{j=1}^{M}\sum_{c=1}^{C}\eta(\xi_c)\Big[\langle w_c^2\rangle\langle {s'_{kj}}^2\rangle + 2\langle w_c\rangle\langle s'_{ki}\rangle\langle w_c^{-k}\rangle(\langle Z^{-k}\rangle\odot\langle {s'_j}^{-k}\rangle)\Big] \tag{47}$$

and $\langle\log\pi_k\rangle$ and $\langle\log(1-\pi_k)\rangle$ can be computed as

$$\langle\log\pi_k\rangle = \psi(\frac{a}{K}+\langle z_k\rangle) - \psi(\frac{a+b(K-1)}{K}+1) \tag{48}$$

$$\langle\log(1-\pi_k)\rangle = \psi(\frac{b(K-1)}{K}+1-\langle z_k\rangle) - \psi(\frac{a+b(K-1)}{K}+1). \tag{49}$$

Update for $\boldsymbol{S}=[s_1,...,s_N]$:
One can show that the posterior parameter μ_i^s, β_{s_i} can be updated as

$$\beta_{s_i} = \langle\gamma_s\rangle\langle z_k\rangle\sum_{j=1}^{d}\langle\phi_{jk}^2\rangle + 2\Big(\sum_{c=1}^{C}\eta(\xi_c)(\langle w_c\rangle^\top\langle z_k\rangle)\Big) + 1/2 \tag{50}$$

$$\begin{aligned}(\mu_i^s)_k =& \frac{\langle\gamma_s\rangle}{\beta_{s_i}+\lambda a}\sum_{j=1}^{d}\Big(\langle\phi_{jk}\rangle\langle z_k\rangle\big(x_{ji}-\sum_{k'\neq k}^{K}\langle\phi_{jk'}\rangle\langle z'_k\rangle\langle s_{k'i}\rangle\big)\Big) + \frac{1}{\beta_{s_i}}\sum_{c=1}^{C}\mathbf{1}[y_i=c]\langle w_{kc}\rangle\langle z_k\rangle \\ &-\frac{2}{\beta_{s_i}}\sum_{c=1}^{C}\eta(\xi_c)\Big(\langle w_{kc}\rangle\langle z_k\rangle\sum_{k'\neq k}^{K}\langle w_{k'c}\rangle\langle z'_k\rangle\langle s_{k'i}\rangle\Big) - \frac{1}{2\beta_{s_i}}\big(\langle z_k\rangle\sum_{c=1}^{C}\langle w_{kc}\rangle\big) \\ &+\Big(\frac{\lambda}{N^2}\sum_{i'\neq i}^{N}\frac{\beta_{s_i}+\beta_{s_{i'}}}{\beta_{s_i}\beta_{s_{i'}}}(\mu_{i'}^s)_k\Big) - \Big(\frac{2\lambda}{MN}\sum_{j=1}^{M}\frac{\beta_{s_i}+\beta_{s'_j}}{\beta_{s_i}\beta_{s'_j}}(\mu_j^{s'})_k\Big)\end{aligned} \tag{51}$$

where $(\mu_i^s)_k$ denotes the k-th entry of the vector $(\mu_i^s)_k$, and

$$a = \frac{1}{N^2}\sum_{i'=1}^{N}\frac{\beta_{s_i}+\beta_{s_{i'}}}{\beta_{s_i}\beta_{s_{i'}}} - \frac{2}{MN}\sum_{j=1}^{M}\frac{\beta_{s_i}+\beta_{s_j}}{\beta_{s_i}\beta_{s_j}} \tag{52}$$

Update for $\boldsymbol{S'}=[s'_1,...,s'_N]$:
One can show that the posterior parameter $\mu_i^{s'}, \beta_{s'_i}$ can be updated as

$$\beta_{s'_i} = \langle\gamma_t\rangle\langle z_k\rangle\sum_{j=1}^{d}\langle {\phi'_{jk}}^2\rangle + \frac{2\lambda'\alpha}{1-\alpha}\Big(\sum_{c=1}^{C}\eta(\xi_c)(\langle w_c\rangle^\top\langle z_k\rangle)\Big) + 1/2 \tag{53}$$

$$\begin{aligned}(\mu_i^{s'})_k =& \frac{\langle\gamma_t\rangle}{\beta_{s'_i}+\lambda a}\sum_{j=1}^{d}\Big(\langle\phi'_{jk}\rangle\langle z_k\rangle\big(x'_{ji}-\sum_{k'\neq k}^{K}\langle\phi'_{jk'}\rangle\langle z'_k\rangle\langle s'_{k'i}\rangle\big)\Big) + \frac{\lambda'\alpha}{\beta_{s'_i}(1-\alpha)}\sum_{c=1}^{C}\langle w_{kc}\rangle\langle z_k\rangle \\ &-\frac{2\lambda'\alpha}{\beta_{s'_i}(1-\alpha)}\sum_{c=1}^{C}\eta(\xi_c)\Big(\langle w_{kc}\rangle\langle z_k\rangle\sum_{k'\neq k}^{K}\langle w_{k'c}\rangle\langle z'_k\rangle\langle s'_{k'i}\rangle\Big) - \frac{\lambda'\alpha}{2\beta_{s'_i}(1-\alpha)}\big(\langle z_k\rangle\sum_{c=1}^{C}\langle w_{kc}\rangle\big) \\ &+\Big(\frac{\lambda}{M^2}\sum_{i'\neq i}^{M}\frac{\beta_{s'_i}+\beta_{s'_{i'}}}{\beta_{s'_i}\beta_{s'_{i'}}}(\mu_{i'}^{s'})_k\Big) - \Big(\frac{2\lambda}{MN}\sum_{j=1}^{N}\frac{\beta_{s'_i}+\beta_{s_j}}{\beta_{s'_i}\beta_{s_j}}(\mu_j^s)_k\Big)\Big)\end{aligned} \tag{54}$$

Update for $\boldsymbol{W}=[w_1,...,w_C]$:

One can show that the posterior parameter μ_c^w, β_{w_c} can be updated as

$$\beta_{w_c} = 2\sum_{i=1}^{N}\lambda(\xi_c)trace(F_i) + \frac{2\lambda'\alpha}{1-\alpha}\sum_{j=1}^{M}\lambda(\xi_c)trace(F'_j) + \frac{1}{2} \tag{55}$$

where $F_i = \langle (Z \odot s_i)(Z \odot s_i)^\top$ is a $K \times K$ matrix which its elements are defined as

$$\begin{aligned} F_i(m,n) &= \langle z_m \rangle \langle s_{mi}^2 \rangle, \ \text{ if } \ m = n \\ F_i(m,n) &= \langle z_m \rangle \langle z_n \rangle \langle s_{mi} \rangle \langle s_{ni} \rangle, \ \text{ if } \ m \neq n \end{aligned} \tag{56}$$

and where $F'_j = \langle (Z \odot s'_j)(Z \odot s'_j)^\top$ is a $K \times K$ matrix which its elements are defined as

$$\begin{aligned} F'_j(m,n) &= \langle z_m \rangle \langle s'^2_{mj} \rangle, \ \text{ if } \ m = n \\ F'_j(m,n) &= \langle z_m \rangle \langle z_n \rangle \langle s'_{mj} \rangle \langle s'_{nj} \rangle, \ \text{ if } \ m \neq n \end{aligned} \tag{57}$$

$$\mu_c^w = \big[\text{diag}(\beta_{w_c})\big]^{-1}\bigg(\sum_{i=1}^{N}(\mathbf{1}[y_i = c] - \frac{1}{2})(\langle Z \rangle \odot \langle s_i \rangle) + \frac{\lambda'\alpha}{2-2\alpha}\sum_{j=1}^{M}(\langle Z \rangle \odot \langle s'_j \rangle)\bigg) \tag{58}$$

where $\text{diag}(\beta_{w_c})$ is a $K \times K$ diagonal matrix with entries taken from the elements β_{w_c}.

M step:

In this step, the variational free parameters $\{\xi_c\}$ are computed by maximizing the objective function while keeping the parameters of the posterior distribution $Q(\Omega)$ fixed.

Update for $\boldsymbol{\xi} = [\xi_1, ..., \xi_C]$:
it is easy to show that ξ_c can be computed as

$$\xi_c^2 = \text{trace}\bigg[\sum_{i=1}^{N} F_i + \frac{\lambda'\alpha}{1-\alpha}\sum_{j=1}^{M} F_j \langle w_c w_c^\top \rangle\bigg] \tag{59}$$

5. Computational Complexity of our VB algorithm

Table 1. Computational Complexity of the **PUnDA** VB algorithm.

Step	Complexity
Update γ_s	$O(NKd)$
Update γ_t	$O(MKd)$
Update $\boldsymbol{\Pi}$	$O(K)$
Update $\boldsymbol{\Phi}$	$O(NK^3d))$
Update $\boldsymbol{\Phi'}$	$O(MK^3d)$
Update $\boldsymbol{Z}$	$O(K^2(N+M)d)$
Update $\boldsymbol{S}$	$O(NKd)$
Update $\boldsymbol{S'}$	$O(MKd)$
Update $\boldsymbol{W}$	$O(C(N+M)K)$
Update $\boldsymbol{\xi}$	$O(MK^2)$

In this section, we analyze the computational complexity of each iteration of the proposed VB algorithm. Each iteration in our VB algorithm, mainly includes matrix multiplications. The computational complexity of each parameter's updating is

summarized in Table 1. As can be seen from Table 1, the total complexity for VB algorithm is $O\big(T \times (N+M)K^2 d\big)$ and T is the total number of iterations. It is clear that the computational complexity of the proposed VB algorithm, for training, in one iteration is $O((N+M)K^2 d)$, i.e., linear in the size of the source+target data $N+M$, the data dimensionality d, and quadratic in the dimensionality of the shared space K, $K \ll d$. For classifying a test data point the computational complexity is $O(CK^2)$, for C class instances.

References

[1] G. Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models. In *NIPS 2007 workshop for approximate Bayesian inference in continuous/hybrid systems*, 2007. 6

[2] S. Herath, M. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[3] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 1

[4] R. Renner and S. Wolf. Smooth rényi entropy and applications. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 233. IEEE, 2004. 6