SimProF: A Simple Probabilistic Framework for Unsupervised Domain Adaptation

Mengzhu Wang

Hebei University of Technology, School of Artificial Intelligence
wangmz@hebut.edu.cn

Abstract

Unsupervised domain adaptation (UDA) aims at knowledge transfer from a labeled source domain to an unlabeled target domain. Most UDA techniques achieve this by reducing feature discrepancies between the two domains to learn domain-invariant feature representations. In this paper, we enhance this approach by proposing a simple yet powerful probabilistic framework (SimProF) for UDA to minimize the domain gap between the two domains. SimProF estimates the feature space distribution for each class and generates contrastive pairs by leveraging the shared categories between the source and target domains. The concept behind SimProF is inspired by the observation that normalized features in contrastive learning tend to follow a mixture of von Mises-Fisher (vMF) distributions on the unit sphere. This characteristic allows for the generation of an infinite number of contrastive pairs and facilitates an efficient optimization method using a closed-form expression for the expected contrastive loss. As a result, target semantics can be effectively used to augment source features. To implement this, we create vMF distributions based on the inter-domain feature mean difference for each class. Notably, we derive and minimize an upper bound of the expected loss, which is implicitly achieved through an estimated supervised contrastive learning loss applied to the augmented source distribution. Comprehensive experiments on cross-domain benchmarks confirm the efficacy of the proposed method.

Introduction

Deep learning models typically exhibit good performance on comparable datasets after being trained on a particular dataset. Nevertheless, these models frequently exhibit markedly worse performance when applied to data from a different distribution. Domain adaptation (Wang et al. 2024c,b,a) offers a way around this problem by allowing a model that was trained on a labeled source domain to adapt to an unlabeled or sparsely labeled target domain.

Generally, solving the domain shift issue is the fundamental challenge to domain adaptation. To tackle this, a substantial subset of current UDA techniques concentrates on aligning the feature distributions in the source and target domains such that a classifier trained on the former can be successfully extended to the latter (Liang, Hu, and Feng 2021). Although these techniques show promising performance, they rely on modifying features using a shared source-supervised classifier. However, this approach carries a substantial risk: the classifier, being supervised by source features, may become biased towards the source domain. As a result, directly applying this classifier to the target domain can limit its generalization ability, especially when cross-domain features are not perfectly aligned. To address this limitation, some studies (Xie et al. 2024; Li et al. 2021b) focus on classifier adaptation, introducing specialized network modules to facilitate the learning of a target-specific classifier.

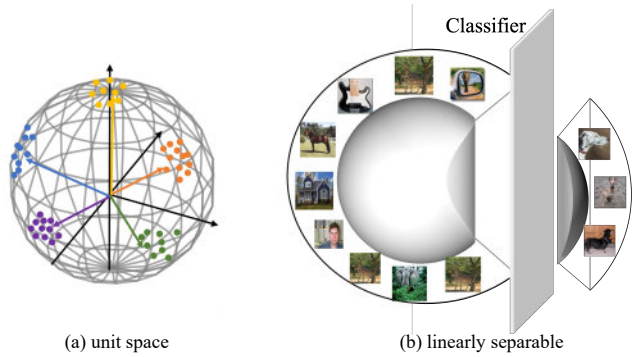

Figure 1: Each colored dot represents a class, and the different classes can be effectively separated by using von Mises-Fisher distributions.

To advance classifier adaptation, we propose a simple probabilistic framework for unsupervised domain adaptation (SimProF). This approach involves semantically augmenting source features toward the target domain, enabling the classifier to adapt to target data within an augmented feature space enriched with target-specific semantics. Our key insight is to directly sample an infinite number of contrastive pairs from the data distribution and then minimize the predicted loss to establish the optimization objective. This approach eliminates the need for large batch sizes by explicitly predicting and minimizing the expected loss. The concept of semantic augmentation draws inspiration from the

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

exceptional ability of deep neural networks to linearize features and disentangle the underlying variations in the data. By modifying features in the deep feature space along certain axes, we can induce significant semantic changes in the original input space (Li et al. 2021a; Wang et al. 2021b). Traditional methods often model unconstrained features using a normal distribution for data augmentation, offering an upper bound on the estimated cross-entropy loss for optimization. However, direct modeling with a normal distribution is impractical in contrastive learning due to the need for feature normalization, and estimating the distribution of each class within small batches is challenging. To address these issues, we model the feature distribution using the von Mises-Fisher distribution on the unit sphere (see Figure. 1), which extends the normal distribution to the hypersphere. By adding directions sampled from this distribution to each source data point, we create a new target-like domain. This augmentation preserves the class identity of the data, allowing us to monitor the classifier's training within the newly generated target-like domain. This method bypasses the need to directly sample a large number of contrastive pairs by instead focusing on minimizing a surrogate loss function, which can be efficiently optimized without adding complexity during inference.

The following is a summary of the SimProF's main contributions:

- For unsupervised domain adaptation, we provide SimProF, a novel probabilistic contrastive learning technique. SimProF effectively estimates parameters across batches by modeling feature distributions using the von Mises-Fisher (vMF) distribution. This strategy directly samples contrastive pairs from the predicted distribution, effectively overcoming the inherent limitation of contrastive learning, which typically requires large batch sizes.
- We minimize the upper bound loss in the limiting case to obtain a closed-form solution. It is possible to efficiently optimize the resulting surrogate loss function without increasing any additional overhead during inference.
- We investigate the efficacy of the suggested SimProF method on a number of datasets, including VisDA-2017, Office-31, Office-Home, and DomainNet.

Method

Motivation and Preliminaries

This work explores the use of von Mises-Fisher (vMF) distributions on unit space for unsupervised domain adaptation (UDA), an area that has not been extensively investigated. We propose a straightforward probabilistic framework for UDA, termed SimProF, which captures the semantics of the target domain while enhancing discriminability. SimProF is inspired by prior research demonstrating that deep neural networks can effectively manipulate features along specific directions (Wang et al. 2019b) in the deep feature space to alter semantics, given their ability to disentangle underlying data variations. Figure. 2 provides an overview of SimProF, with further details described below.

Unsupervised domain adaptation aims to transfer models trained on a labeled source domain to an unlabeled target domain with a differing data distribution which depends on having access to both unlabeled data from the target domain and all labeled samples from the source domain during training. Formally, let $D_s = (\mathcal{X}_s, \mathcal{Y}_s) = \{(x_{si}, y_{si})\}_{i=1}^{N_s}$ represents a fully labeled source domain with N_s image-label pairs, and $D_t = \mathcal{X}_t = \{x_{ti}\}_{i=1}^{N_t}$ denote an unlabeled dataset from the target domain with N_t images. Both $\{x_{si}\}$ and $\{x_{ti}\}$ are drawn from the same set of M predefined categories.

Contrastive Learning (CL)

In CL (Khosla et al. 2020), a feature extractor F is trained to discriminate between negative pairings (x_i, x_j) with distinct labels $y_i \neq y_j$ and positive pairs (x_i, x_j) with the same label $y_i = y_j$. In embedding space, clusters of points from the same class are drawn together, while clusters of samples from different classes are pushed apart. Given every batch of sample-label pairings $B = \{(x_i, y_i)_{i=1}^{N^B}\}$ and a temperature parameter τ, there are two standard methods to construct the CL loss:

$$\mathcal{L}_{in}(\boldsymbol{x}_i, y_i) = -\log\left\{\frac{1}{N_{y_i}^B}\sum_{p\in P(y_i)}\frac{e^{\boldsymbol{x}_i\cdot\boldsymbol{x}_p/\tau}}{\sum_{j=1}^{M}\sum_{b\in A(j)}e^{\boldsymbol{x}_i\cdot\boldsymbol{x}_a/\tau}}\right\}, \tag{1}$$

$$\mathcal{L}_{out}(\boldsymbol{x}_i, y_i) = \frac{-1}{N_{y_i}^B}\sum_{p\in A(y_i)}\log\frac{e^{\boldsymbol{x}_i\cdot\boldsymbol{x}_p/\tau}}{\sum_{j=1}^{M}\sum_{b\in A(j)}e^{\boldsymbol{x}_i\cdot\boldsymbol{x}_a/\tau}}, \tag{2}$$

where $P(y_i) = \{b \in A(j) : y_p = y_j\}$ is the set of indices of all positives in the multi-viewed batch distinct from i, and $N_{y_i}^B = |A(y_i)|$ is the cardinality. $\boldsymbol{x}$ denotes the normalized features of x extracted by F:

$$\boldsymbol{x}_i = \frac{F(x_i)}{\|F(x_i)\|}, \quad \boldsymbol{x}_p = \frac{F(x_p)}{\|F(x_p)\|}, \quad \boldsymbol{x}_a = \frac{F(x_a)}{\|F(x_a)\|}.$$

Furthermore, with respect to the position of the log function, the values $\mathcal{L}_{out}$ and $\mathcal{L}_{in}$ represent the sums over the positive pairs. The two loss formulations are not equivalent, as shown in (Khosla et al. 2020), and $\mathcal{L}_{in} \leq \mathcal{L}_{out}$, according to Jensen's inequality (Jensen 1906). Since $\mathcal{L}_{out}$ offers an upper bound for $\mathcal{L}_{in}$, CL utilizes it as the loss function.

von Mises–Fisher (vMF) Distribution

Features in contrastive learning are restricted to lie on the unit hypersphere, as was previously mentioned. Since vMF distributions extend the normal distribution to hyperspherical spaces, we utilize a mixture of von Mises–Fisher (vMF) distributions (Mardia, Jupp, and Mardia 2000) to simulate these properties. For a random p-dimensional unit vector z,

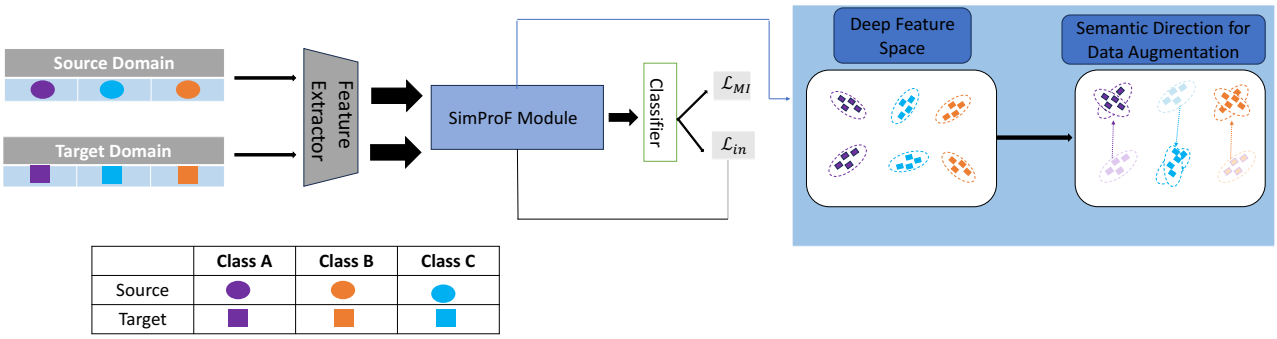

Figure 2: An example of SimProF is as follows: SimProF creates contrastive pairs from the sample distribution by estimating it using characteristics from different batches. By sampling an infinite number of contrastive pairs, it also obtains a closed-form expression for the predicted contrastive loss. This method overcomes contrastive learning's intrinsic drawback with respect to high batch sizes.

the probability density function of the vMF distribution can be written as follows:

$$f_p(\boldsymbol{x};\mu,\kappa_y) = \frac{1}{C_p(\kappa_y)} e^{\kappa_y \mu^\top \boldsymbol{x}}, \tag{3}$$

$$C_p(\kappa_y) = \frac{(2\pi)^{p/2} I_{(p/2-1)}(\kappa_y)}{\kappa_y^{p/2-1}}, \tag{4}$$

where $I_{(p/2-1)}$, $\kappa \geq 0$, $\|\mu\|_2 = 1$, and x are p-dimensional unit vectors. indicates the first-kind modified Bessel function at order $p/2-1$. It has the following definition:

$$I_{(p/2-1)}(\boldsymbol{x}) = \sum_{k=0}^{\infty} \frac{1}{k!\Gamma(p/2-1+k+1)} (\frac{\boldsymbol{x}}{2})^{2k+p/2-1}. \tag{5}$$

The concentration parameter and the mean direction are denoted by the parameters κ_y and μ, respectively. The distribution becomes more concentrated around the mean direction μ as κ_y grows. On the other hand, the distribution becomes uniform throughout the sphere when $\kappa_y = 0$.

Based on the assumption above, we model the feature distribution using a mixture of vMF distributions:

$$P(\boldsymbol{x}) = \sum_{y=1}^{M} P(y)P(\boldsymbol{x}|y) = \sum_{y=1}^{M} \pi_y \frac{\kappa_y^{p/2-1} e^{\kappa_y \mu_y^\top \boldsymbol{x}}}{(2\pi)^{p/2} I_{p/2-1}(\kappa_y)}, \tag{6}$$

where π_y, which represents the probability of a class y, indicates how frequently class y occurs in the training dataset. Next, we use maximum likelihood estimation to estimate the mean vector μ_y and the concentration parameter κ_y for the feature distribution.

Maximum Likelihood Estimation. To get the mean vector $\boldsymbol{\mu}_y$ and the concentration parameter κ_y, we perform maximum likelihood estimation on (κ_y, μ_y) as follows:

$$\begin{aligned} L(\mu_y; p, \kappa_y, \lambda) =& n \log\left(C_p(\kappa_y)\right) \\ &+ \kappa_y \mu_y^\top \sum_{i=1}^{n} \boldsymbol{x}_i + \lambda(\|\mu_y\| - 1), \end{aligned} \tag{7}$$

Based on the maximum likelihood estimation from Eq. 7, the partial derivative of L can be expressed as follows:

$$\frac{\partial L}{\partial \mu_y} = \kappa_y \sum_{i=1}^{n} \boldsymbol{x}_i + \lambda \frac{\mu_y}{\|\mu_y\|} \quad = 0, \tag{8}$$

$$\frac{\partial L}{\partial \kappa_y} = n \frac{C_p'(\kappa_y)}{C_p(\kappa_y)} + \mu_y{}^\top \sum_{i=1}^{n} \boldsymbol{x}_i = 0, \tag{9}$$

$$\frac{\partial L}{\partial \lambda} = \|\mu_y\| - 1 \quad = 0, \tag{10}$$

Due to $C_p(\kappa_y) = \frac{\kappa_y^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa_y)} = (2\pi)^{-\frac{p}{2}} \frac{\kappa_y^{\frac{p}{2}-1}}{I_{\frac{p}{2}-1}(\kappa_y)}$, then from the Eq. 9, $C_p'(\kappa_y)$ is given by:

$$C_p'(\kappa_y) = (2\pi)^{-\frac{p}{2}} \frac{\left(\frac{p}{2}-1\right)\kappa_y^{\frac{p}{2}-2} I_{\frac{p}{2}-1}(\kappa_y) - \kappa_y^{\frac{p}{2}-1} I'_{\frac{p}{2}-1}(\kappa_y)}{I^2_{\frac{p}{2}-1}(\kappa_y)} \tag{11}$$

According to the results of $C_p(\kappa_y)$ and $C_p'(\kappa_y)$, the $\frac{C_p'(\kappa_y)}{C_p(\kappa_y)}$ is derived as follows:

$$\begin{aligned} \frac{C_p'(\kappa_y)}{C_p(\kappa_y)} &= \frac{(2\pi)^{-\frac{p}{2}} \frac{\left(\frac{p}{2}-1\right)\kappa_y^{\frac{p}{2}-2} I_{\frac{p}{2}-1}(\kappa_y) - \kappa_y^{\frac{p}{2}-1} I'_{\frac{p}{2}-1}(\kappa_y)}{I^2_{\frac{p}{2}-1}(\kappa_y)}}{(2\pi)^{-\frac{p}{2}} \frac{\kappa_y^{\frac{p}{2}-1}}{I_{\frac{p}{2}-1}}(\kappa_y)} \\ &= \frac{\left(\frac{p}{2}-1\right)\kappa_y^{\frac{p}{2}-2} I_{\frac{p}{2}-1}(\kappa_y) - \kappa_y^{\frac{p}{2}-1} I'_{\frac{p}{2}-1}(\kappa_y)}{\kappa_y^{\frac{p}{2}-1} I_{\frac{p}{2}-1}(\kappa_y)} \\ &= \frac{\left(\frac{p}{2}-1\right)\kappa_y^{-1} I_{\frac{p}{2}-1}(\kappa_y) - I'_{\frac{p}{2}-1}(\kappa_y)}{I_{\frac{p}{2}-1}(\kappa_y)} \\ &= \frac{\left(\frac{p}{2}-1\right)\kappa_y^{-1} I_{\frac{p}{2}-1}(\kappa_y) - \left[I_{\frac{p}{2}}(\kappa_y) + \left(\frac{p}{2}-1\right)\kappa_y^{-1} I_{\frac{p}{2}-1}(\kappa_y)\right]}{I_{\frac{p}{2}-1}(\kappa_y)} \\ &= -\frac{I_{\frac{p}{2}}(\kappa_y)}{I_{\frac{p}{2}-1}(\kappa_y)} \end{aligned} \tag{12}$$

Suppose that $A_p(\kappa_y) = -\frac{C'_p(\kappa_y)}{C_p(\kappa_y)} = \frac{I_{\frac{p}{2}}(\kappa_y)}{I_{\frac{p}{2}-1}(\kappa_y)}$, then the maximum likelihood estimates of the mean direction $\boldsymbol{\mu}_y$ satisfy the following equations:

$$\left(\kappa_y \sum_{i=1}^{n} x_i\right)^\top \left(\kappa_y \sum_{i=1}^{n} x_i\right) = (-\lambda\mu_y)^\top(-\lambda\mu_y) \quad (13)$$

$$\begin{aligned} & n\kappa_y \frac{C'_p(\kappa_y)}{C_p(\kappa_y)} + \kappa_y \boldsymbol{\mu}_y^\top \sum_{i=1}^{n} \boldsymbol{x}_i \\ =& \kappa_y \mu_y^\top \sum_{i=1}^{n} x_i + n\kappa_y \frac{C'_p(\kappa_y)}{C_p(\kappa_y)} \\ =& 0 \end{aligned} \quad (14)$$

$$\begin{aligned} A_p(\kappa_y) &= \frac{1}{n}\left\|\sum_{i=1}^{n} \boldsymbol{x}_i\right\| \\ &= \|\overline{x}\| \end{aligned} \quad (15)$$

where $n\kappa_y \frac{C'_p(\kappa_y)}{C_p(\kappa_y)} = \lambda$ and $\bar{R} = \|\bar{x}\|$ is the length of sample mean. A simple approximation (Wang et al. 2021a) to κ_y is:

$$\hat{\kappa}_y = \frac{\bar{R}(p - \bar{R}^2)}{1 - \bar{R}^2}. \quad (16)$$

Furthermore, an online method is employed to estimate the sample mean for each class by incorporating data from both the current and previous mini-batches. Specifically, we utilize the online estimation algorithm outlined below to continuously update the sample mean, starting from zero initialization at the beginning of the current epoch. The sample mean estimated from the previous epoch is then used for maximum likelihood estimation.

Implicit Augmentation by vMF distribution

On the unit hypersphere $\mathcal{S}^{p-1}$, consider a set of N independent unit vectors $(x_i)_i^N$ sampled from a von Mises–Fisher (vMF) (Sra 2012; Mardia and Jupp 2009) distribution corresponding to class y. The following equations provide the maximum likelihood estimates for the concentration parameter κ_y and the mean direction μ_y:

$$\mu_y = \bar{x}/\bar{R}, \quad (17)$$

$$A_p(\kappa_y) = \frac{I_{p/2}(\kappa_y)}{I_{p/2-1}(\kappa_y)} = \bar{R}, \quad (18)$$

where the sample mean is represented by $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, and the sample mean vector's length is indicated by $\bar{R} = |\bar{x}|_2$. κ_y has an approximation expression that is as follows:

$$\hat{\kappa}_y = \frac{\bar{R}(p - \bar{R}^2)}{1 - \bar{R}^2}. \quad (19)$$

Additionally, by combining data from the previous and current mini-batches, a real-time estimate of the sample mean for each class is computed. More specifically, we use the sample mean estimated from the previous epoch for maximum likelihood estimation and update the sample mean from zero initialization in the current epoch using the online estimation algorithm below:

$$\bar{x}_j^{(t)} = \frac{n_j^{(t-1)}\bar{x}_j^{(t-1)} + m_j^{(t)}\bar{x}_j'^{(t)}}{n_j^{(t-1)} + m_j^{(t)}}, \quad (20)$$

where $\bar{x}_j'^{(t)}$ represents the sample mean of class j in the current mini-batch, and $\bar{x}_j^{(t)}$ indicates the estimated sample mean of class j at step t. The number of samples from the prior mini-batches and the current mini-batch are indicated by the variables $n_j^{(t-1)}$ and $m_j^{(t)}$, respectively.

Semantic Directions for Domain Augmentation

Every class has an average feature representation that can be thought of as its semantic prototype, from which all samples are created. For example, take a bicycle. Variants of this prototype provide particular samples, but the semantic prototype of the bicycle may comprise essential structural components like wheels, handlebars, and a seat. The average representations, or semantic prototypes, of source and target samples for the same class frequently vary as a result of domain shift. To fix this prototype mismatch, we translate each source feature along the direction given by the prototype vector corresponding to its class. More specifically, μ_s^c and μ_t^c represent the estimated mean feature for class c in the source and target domains, respectively.

A straightforward method that builds on the predicted parameters would be to select contrastive pairs from the mixture of vMF distributions. During each training iteration, directly sampling a large number of data points from these distributions might not be as beneficial. We overcome this by expanding the number of samples to infinity and using mathematical analysis to obtain a closed-form formula for the anticipated contrastive loss function.

Proposition 1. *Assuming the parameters of the mixture of vMF distributions are π_y, μ_y, and κ_y for $y = 1, \ldots, M$, and letting the sample size N approach infinity, the projected contrastive loss function can be expressed as follows:*

$$\mathcal{L}_{\text{out}}(\boldsymbol{x_i}, y_i) = \frac{-\boldsymbol{x}_i \cdot A_p(\kappa_{y_i})\mu_{y_i}}{\tau} + \log\left(\sum_{j=1}^{M} \pi_j \frac{C_p(\tilde{\kappa}_j)}{C_p(\kappa_j)}\right), \quad (21)$$

$$\mathcal{L}_{\text{in}}(\boldsymbol{x_i}, y_i) = -\log\left(\pi_{y_i}\frac{C_p(\tilde{\kappa}_{y_i})}{C_p(\kappa_{y_i})}\right) + \log\left(\sum_{j=1}^{M} \pi_j \frac{C_p(\tilde{\kappa}_j)}{C_p(\kappa_j)}\right), \quad (22)$$

where $\tilde{\boldsymbol{x}}_j \sim \text{vMF}(\mu_j, \kappa_j)$, $\tilde{\kappa}_j = ||\kappa_j\mu_j + \boldsymbol{x}_i/\tau||_2$.

Proof. As per Eq. 2, which defines supervised contrastive loss, we obtain

Methods	D→A	W→A	A→W	D→W	W→D	A→D	Avg.
ResNet-50	62.5	60.7	68.4	96.7	99.3	68.9	76.1
SimNet	73.4	71.6	88.6	98.2	99.7	85.3	86.2
CyCADA	72.8	71.4	89.5	97.9	99.8	87.7	86.5
CDAN	71.0	69.3	94.1	98.6	**100.0**	92.9	87.7
TADA	72.9	73.0	94.3	98.7	99.8	91.6	88.4
BSP	73.6	72.6	93.3	98.2	**100.0**	93.0	88.5
T^2SA	78.2	78.5	94.6	97.2	99.8	92.4	90.1
SimProF	**81.7**	**84.5**	**97.2**	**100.0**	**100.0**	**97.3**	**93.5**

Table 1: Recognition accuracy (%) using the Office-31 dataset.

Method	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-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SAFN	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
TADA	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SymNet	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
T^2SA	61.0	78.5	83.1	71.4	80.1	79.9	70.3	60.1	83.5	75.6	62.6	86.6	74.4
SimProF	**61.9**	**78.9**	**84.5**	**72.9**	**78.4**	**81.1**	**72.3**	**62.2**	**84.8**	**81.9**	**69.4**	**92.2**	**76.7**

Table 2: Recognition accuracy (%) using the Office-Home dataset.

$$\mathcal{L}_{out} = \frac{-1}{N_{y_i}} \sum_{p \in A(y_i)} \boldsymbol{x}_i \cdot \boldsymbol{x}_p / \tau + \log \left(\sum_{j=1}^{M} N \frac{N_j}{N} \frac{1}{N_j} \sum_{a \in A(j)} e^{\boldsymbol{x}_i \cdot \boldsymbol{x}_a / \tau} \right), \quad (23)$$

where N_j represents the sampling number of class j and $\lim_{N\to\infty} N_j/N = \pi_j$ is satisfied.

The loss function is as follows once $N \to \infty$ and the constant term $\log N$ are eliminated:

$$\mathcal{L}_{out} = \frac{-\boldsymbol{x}_i \cdot \mathbb{E}[\tilde{\boldsymbol{x}}_{y_i}]}{\tau} + \log \left(\sum_{j=1}^{M} \pi_j \mathbb{E}[e^{\boldsymbol{x}_i \cdot \tilde{\boldsymbol{x}}_j / \tau}] \right) \quad (24)$$

$$= \frac{-\boldsymbol{x}_i \cdot A_p(\kappa_{y_i}) \mu_{y_i}}{\tau} + \log \left(\sum_{j=1}^{M} \pi_j \frac{C_p(\tilde{\kappa}_j)}{C_p(\kappa_j)} \right). \quad (25)$$

By utilizing the expectation and moment-generating function of the vMF distribution, Eq. 25 is obtained:

$$\mathbb{E}(\boldsymbol{x}) = A_p(\kappa)\mu, A_p(\kappa) = \frac{I_{p/2}(\kappa)}{I_{p/2-1}(\kappa)}, \quad (26)$$

$$\mathbb{E}\left(e^{t^{\mathrm{T}}\boldsymbol{x}}\right) = \frac{C_p(\tilde{\kappa})}{C_p(\kappa)}, \tilde{\kappa} = ||\kappa\mu + t||_2. \quad (27)$$

We can derive the other loss function from Eq. 1 in a manner similar to that of Eq. 21.

Methods	Synthetic → Real
DAN	53.0
DANN	57.4
SimNet	69.6
CDAN	70.0
MCD	73.7
DSAN	75.1
T^2SA	83.2
CDAN+InterBN	**87.9**

Table 3: Recognition accuracy (%) using the VisDA-2017 dataset.

$$\mathcal{L}_{in} = -\log \left(\pi_{y_i} \frac{C_p(\tilde{\kappa}_{y_i})}{C_p(\kappa_{y_i})} \right) + \log \left(\sum_{j=1}^{K} \pi_j \frac{C_p(\tilde{\kappa}_j)}{C_p(\kappa_j)} \right). \quad (28)$$

□

Overall Formulation

Mutual information $I(X;Y)$ (Csiszár, Shields et al. 2004) measures the degree of dependency between two random variables, X and Y, in information theory. Strong correlations between target features and predictions improve our semantic augmentations by guaranteeing that important semantics relevant to predictions are captured in the extracted

CDAN	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	36.6	9.0	50.7	42.3	31.8
inf	27.5	-	25.7	1.8	34.7	20.1	22.0
pnt	42.6	20.0	-	2.5	55.6	38.5	31.8
qdr	21.0	4.5	8.1	-	14.3	15.7	12.7
rel	51.9	23.3	50.4	5.4	-	41.4	34.5
skt	50.8	20.3	43.0	2.9	50.8	-	33.6
Avg.	38.8	17.7	32.8	4.3	41.2	31.6	27.7

T^2SA	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	44.5	16.0	61.7	47.4	38.0
inf	42.7	-	43.3	8.8	56.5	35.6	37.4
pnt	50.5	21.3	-	8.8	63.3	45.1	37.8
qdr	35.3	8.2	17.3	-	28.8	24.9	22.9
rel	57.2	23.6	53.9	9.7	-	44.3	37.8
skt	60.4	21.2	50.4	16.7	61.4	-	42.0
Avg.	49.2	19.0	41.9	12.0	54.3	39.5	36.0

SimProF	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	22.3	46.3	17.8	63.5	49.7	**39.9**
inf	45.2	-	46.1	9.9	58.3	37.9	**39.5**
pnt	51.8	22.5	-	9.6	64.5	46.8	**39.0**
qdr	36.4	8.6	19.7	-	31.2	26.1	**24.4**
rel	59.4	24.5	55.2	10.6	-	46.8	**39.3**
skt	62.4	23.7	51.3	18.8	62.5	-	**43.7**
Avg.	**51.0**	**20.3**	**43.7**	**13.3**	**56.0**	**41.5**	**37.6**

Table 4: Recognition accuracy (%) using the DomainNet dataset.

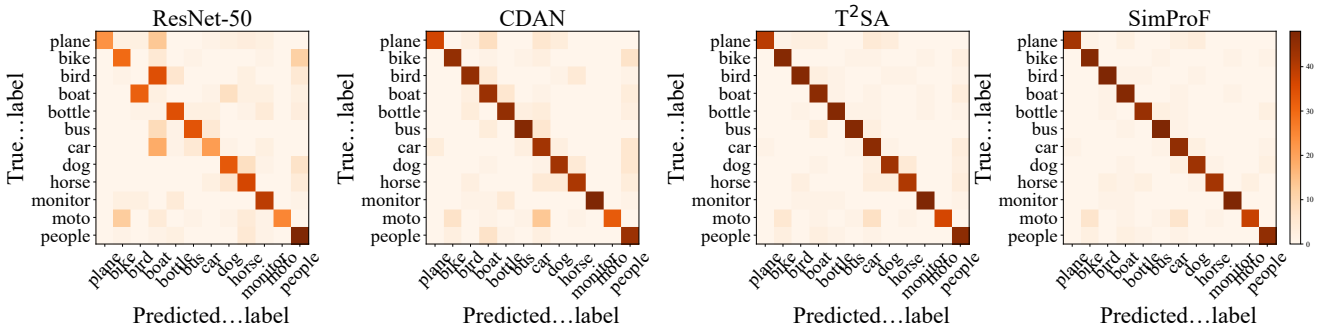

Figure 3: For the task C→P in ImageCLEF-DA, the confusion matrices of the target samples obtained from various methods are displayed. (For optimal clarity, please magnify the image.)

features, instead of insignificant information. Therefore, by reducing the loss function provided in Eq. 29, we use mutual information maximization for the goal data.

$$\mathcal{L}_{MI} = \sum_{c=1}^{C} \hat{P}^c \log \hat{P}^c - \frac{1}{n_t} \sum_{j=1}^{n_t} \sum_{c=1}^{C} P_{tj}^c \log P_{tj}^c, \quad (29)$$

The expression $\hat{P} = \frac{1}{n_t} \sum_{j=1}^{n_t} P_{tj}$ is used. We use the average of the target predictions to approximate the ground-truth distribution because the target domain is unlabeled.

In conclusion, the general goal function of SimProF is:

$$\mathcal{L}_{SimProF} = \mathcal{L}_{in} + \beta \mathcal{L}_{MI}, \quad (30)$$

where the trade-off parameter is β. Experiments will be conducted to thoroughly investigate the effects of various SimProF components.

Experiments

Datasets

Office-31 (Saenko et al. 2010) is a well-known cross-domain dataset used in office environments, consisting of images across 31 classes from three distinct domains: Amazon (A), Webcam (W), and DSLR (D). The dataset is imbalanced, with 2,817 images from Amazon, 795 images from Webcam, and 498 images from DSLR.

Office-Home (Venkateswara et al. 2017) is a collection of 15,500 photos from both home and business settings, organized into 65 categories. The dataset consists of four distinct domains: Product (Pr), Clipart (Cl), Real-World (Rw), and Art (Ar). Real-world camera catches, product photos, clipart images, and artistic renderings are included in these categories in that order.

VisDA-2017 (Peng et al. 2017) comprises more than 280,000 photos in 12 categories and presents a difficult benchmark for unsupervised domain adaptation (UDA). We employ the validation set as the target domain and the training set as the source domain.

DomainNet (Peng et al. 2019) comprise over 590,000 images in six domains Painting (pnt), Infograph (inf), Real (rel), Quickdraw (qdr), (Clipart (clp) and Sketch (skt)), is the largest image benchmark for domain adaptation.

Implementation Details

We use PyTorch to implement our technique, and the backbone network for all datasets is ResNet (He et al. 2016), which has been pre-trained on ImageNet (Russakovsky et al. 2015). We used Pytorch (Paszke et al. 2019) to implement all of the experiments. Our Stochastic Gradient Descent (SGD) algorithm is used with momentum set to 0.9 and weight decay set to 0.001. For model optimization, we follow the learning rate annealing technique described in (Ganin et al. 2016). In this study, we specifically specify $\beta = 0.5$ and perform a sensitivity analysis to find out how hyper-parameter selection affects the results. We present the average accuracy from three random trials for each task. We compared with ResNet-50 (He et al. 2016), DDC (Tzeng et al. 2014), DAN (Long et al.

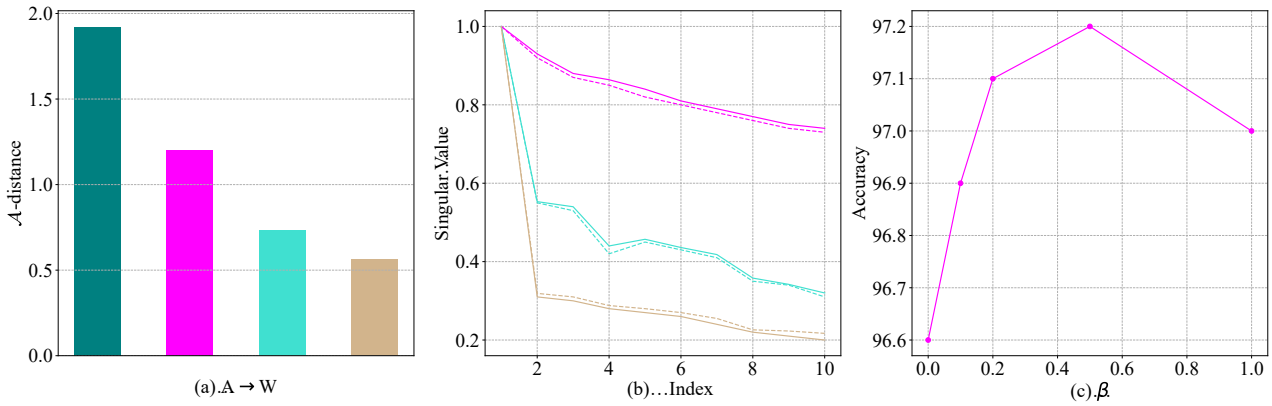

Figure 4: (a): $\mathcal{A}$-distance of different methods, (b): SVD Analysis, (c): Sensitivity of β, zoom in to see the details.

2015), DANN (Ganin and Lempitsky 2015), SimNet (Pinheiro 2018), CyCADA (Hoffman et al. 2018), CDAN (Long et al. 2017), TADA (Wang et al. 2019a), BSP (Chen et al. 2019), MCD (Saito et al. 2018), SAFN (Xu et al. 2019), DSAN (Zhu et al. 2020),TADA (Wang et al. 2019a), SymNet (Pinheiro 2018),ATM (Li et al. 2020), TSA (Li et al. 2021b), T^{2}SA (Xie et al. 2024).

Results

We present our results in Table. 1 to Table. 4. Our SimProF method significantly enhances the performance of T^{2}SA, achieving average accuracy improvements of 3.4%, 2.3%, 4.7%, and 1.6% on the Office-31, Office-Home, VisDA-2017, and DomainNet datasets, respectively. Notably, we observe substantial accuracy gains on some of the most challenging tasks, such as in the Office-31 dataset, where D→A improved from 78.2% to 81.7% and W→A improved from 78.5% to 84.5%. On the largest dataset, DomainNet, our SimProF method also demonstrates impressive performance with a 1.6% increase compared to T^{2}SA.

Empirical Analysis

Confusion Matrices. We present the confusion matrices in Figure 3 to visually demonstrate the effectiveness of our proposed method. For the ResNet-50 model, the categories "aeroplane," "car," and "motorbike" are particularly challenging to predict. Specifically, there are numerous misclassifications, such as many samples from the "car" class being incorrectly predicted as "boat." In comparison to the domain adversarial method CDAN and the semantic augmentation method T^{2}SA, our SimProF method shows a significant improvement, with a greater number of correct predictions appearing on the diagonal.

Distribution Discrepancy. The $\mathcal{A}$-distance (Ben-David et al. 2010) is a widely used metric for measuring the distance between two distributions, with a larger $\mathcal{A}$-distance indicating a greater distinction between the source and target domains. However, the complexity of directly computing the $\mathcal{A}$-distance prompts us to use the proxy $\mathcal{A}$-distance, $\hat{d}_{\mathcal{A}}$, defined as $\hat{d}_{\mathcal{A}} = 2(1 - 2\varepsilon)$. In our analysis, we compute this proxy $\mathcal{A}$-distance for various models, including ResNet, CDAN, T^{2}SA, and SimProF, specifically focusing on the feature representations for the A → W task in Office-31. The results, shown in Figure 4(a), reveal that SimProF has the lowest proxy $\mathcal{A}$-distance among the evaluated methods.

SVD Analysis. Singular value decomposition (SVD) offers valuable insights into unsupervised domain adaptation (UDA) through the analysis of singular value distributions (Chen et al. 2019). To explore this, we plot the singular values (using max-normalization) of features extracted by T^{2}SA and SimProF in Figure 4(b). The figure shows that SimProF effectively reduces the disparity between the largest and the remaining singular values, suggesting improved domain adaptation capability.

Hyper-parameter Sensitivity. Figure 4(c) illustrates the sensitivity of SimProF to the hyper-parameter β, with values varying from $0.0, 0.1, 0.2, 0.5, 1.0$. The results indicate that SimProF performs optimally when $\beta = 0.5$, demonstrating the robustness and stability of our method.

Conclusions

In this paper, we introduce SimProF, a straightforward probabilistic framework for UDA to ameliorate the adaptation ability of the classifier by optimizing the derived upper bound of the expected. We utilize a von Mises-Fisher distribution to model the normalized feature space of samples within the context of contrastive learning. Besides. it not only allow for efficient parameter estimation across different batches using maximum likelihood estimation, but also derive a closed-form expression for the expected supervised contrastive loss by sampling an infinite number of samples from the estimated distribution, overcoming the typical limitation of supervised contrastive learning that requires a large sample size to achieve satisfactory performance.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grants No. 62406100. Tianjin Natural Science Foundation under Grants No. 24JCQNJC00320.

References

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

Chen, X.; Wang, S.; Long, M.; and Wang, J. 2019. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, 1081–1090.

Csiszár, I.; Shields, P. C.; et al. 2004. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4): 417–528.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *JMLR*, 17(59): 1–35.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 1989–1998.

Jensen, J. L. W. V. 1906. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NeurIPS*.

Li, J.; Chen, E.; Ding, Z.; Zhu, L.; Lu, K.; and Shen, H. T. 2020. Maximum density divergence for domain adaptation. *TPAMI*.

Li, S.; Gong, K.; Liu, C. H.; Wang, Y.; Qiao, F.; and Cheng, X. 2021a. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *CVPR*, 5212–5221.

Li, S.; Xie, M.; Gong, K.; Liu, C. H.; Wang, Y.; and Li, W. 2021b. Transferable semantic augmentation for domain adaptation. In *CVPR*, 11516–11525.

Liang, J.; Hu, D.; and Feng, J. 2021. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, 16632–16642.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2017. Conditional adversarial domain adaptation. *NeurIPS*.

Mardia, K. V.; and Jupp, P. E. 2009. *Directional statistics*. John Wiley & Sons.

Mardia, K. V.; Jupp, P. E.; and Mardia, K. 2000. *Directional statistics*. Wiley Online Library.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *ICCV*, 1406–1415.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. Visda: The visual domain adaptation challenge. *arXiv:1710.06924*.

Pinheiro, P. O. 2018. Unsupervised domain adaptation with similarity learning. In *CVPR*, 8004–8013.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*, 115: 211–252.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 3723–3732.

Sra, S. 2012. A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of I s (x). *Computational Statistics*, 27: 177–190.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 5018–5027.

Wang, J.; Zhang, W.; Zang, Y.; Cao, Y.; Pang, J.; Gong, T.; Chen, K.; Liu, Z.; Loy, C. C.; and Lin, D. 2021a. Seesaw loss for long-tailed instance segmentation. In *CVPR*, 9695–9704.

Wang, M.; Liu, J.; Luo, G.; Wang, S.; Wang, W.; Lan, L.; Wang, Y.; and Nie, F. 2024a. Smooth-Guided Implicit Data Augmentation for Domain Generalization. *TNNLS*.

Wang, M.; Liu, Y.; Yuan, J.; Wang, S.; Wang, Z.; and Wang, W. 2024b. Inter-class and inter-domain semantic augmentation for domain generalization. *IEEE TIP*.

Wang, M.; Wang, S.; Yang, X.; Yuan, J.; and Zhang, W. 2024c. Equity in unsupervised domain adaptation by nuclear norm maximization. *TCSVT*.

Wang, X.; Jin, Y.; Long, M.; Wang, J.; and Jordan, M. 2019a. Transferable normalization: Towards improving transferability of deep neural networks. *NeurIPS*.

Wang, Y.; Huang, G.; Song, S.; Pan, X.; Xia, Y.; and Wu, C. 2021b. Regularizing deep networks with semantic data augmentation. *TPAMI*, 44(7): 3733–3748.

Wang, Y.; Pan, X.; Song, S.; Zhang, H.; Huang, G.; and Wu, C. 2019b. Implicit semantic data augmentation for deep networks. *NeurIPS*, 32.

Xie, M.; Li, S.; Gong, K.; Wang, Y.; and Huang, G. 2024. Adapting Across Domains via Target-Oriented Transferable

Semantic Augmentation Under Prototype Constraint. *IJCV*, 132(4): 1417–1441.

Xu, R.; Li, G.; Yang, J.; and Lin, L. 2019. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 1426–1435.

Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; and He, Q. 2020. Deep subdomain adaptation network for image classification. *TNNLS*, 32(4): 1713–1722.