Confidence-based Visual Dispersal for Few-shot Unsupervised Domain Adaptation

Table 1: Overall statistics of datasets used to evaluate the proposed C-VisDiT method.

Dataset	Domain	Total images	Labeled images		Classes
			1-shot	3-shots	
Office-31 [11]	Amazon (A)	2817	31	93	31
	DSLR (D)	498	31	93	
	Webcam (W)	795	31	93	
			3%	6%	
Office-Home [13]	Art (Ar)	2427	73	146	65
	Clipart (Cl)	4365	131	262	
	Product (Pr)	4439	133	266	
	Real (Rw)	4357	131	261	
			1%		
VisDA-C [9]	Train	152397	1531		12
	Validation	55388	-		
			1-shot	3-shots	
DomainNet [8]	Clipart (C)	18703	126	378	126
	Painting (P)	31502	126	378	
	Real (R)	70358	126	378	
	Sketch (S)	24582	126	378	

Appendix

A. Additional Datasets Details

We present the overall statistics of the datasets that are used to evaluate our proposed C-VisDiT in Tab. 1. **Office-31** [11] is a dataset of office images containing 3 domains (Amazon, DSLR, and Webcam) with 31 classes. **Office-Home** [13] also contains office images across 4 domains (Art, Clipart, Product, Real) with 65 classes in each domain. **VisDA-C** [9] is a large simulation-to-real dataset with over 150K images in the training domain and over 55K images in the validation domain. **DomainNet** [8] is the most diverse and recent cross-domain benchmark to-date. Following [14, 2], we use a subset of DomainNet containing 4 domains (Clipart, Real, Painting and Sketch) and 126 classes. We follow the same settings in [5, 14, 2] and conduct experiments with corresponding numbers of labeled images on the aforementioned datasets. For labeled image sampling, we directly leverage the split files[1] published by [14] for a fair comparison with the existing FUDA methods.

[1] https://github.com/zhengzangw/PCS-FUDA

B. Additional Implementation Details

B.1. Details for the Baseline Model

To enhance the cross-domain feature alignment, we adopt the prototypical self-supervised learning method in [14] to optimize the baseline model and denote the objective as $\mathcal{L}_{self}$ in the article. Following [14], we utilize the k-means clustering to obtain normalized source class centers $\{\mu_i^s\}_{i=1}^c$ and normalized target class centers $\{\mu_i^t\}_{i=1}^c$. The class centers are updated via momentum in each training epoch. For each target image $\mathbf{x}_i^{ut} \in \mathcal{D}_{ut}$, we calculate two similarity distributions P_i^t and $P_i^{t\to s}$ as:

$$P_{i,j}^t = \frac{\exp(\mu_j^t \cdot F(\mathbf{x}_i^{ut})/t)}{\sum_{k=1}^c \exp(\mu_k^t \cdot F(\mathbf{x}_i^{ut})/t)} \tag{1}$$

$$P_{i,j}^{t\to s} = \frac{\exp(\mu_j^s \cdot F(\mathbf{x}_i^{ut})/t)}{\sum_{k=1}^c \exp(\mu_k^s \cdot F(\mathbf{x}_i^{ut})/t)}, \tag{2}$$

where t is a temperature value. Similarly, we can calculate P_i^s and $P_i^{s\to t}$ for each source image $\mathbf{x}_i^s \in \mathcal{D}_{ls} \cup \mathcal{D}_{us}$. Therefore, the $\mathcal{L}_{self}$ objective in the baseline model can be formulated as:

$$\begin{aligned}\mathcal{L}_{self} = &\sum_{i=1}^{N_{ls}+N_{us}} (\mathcal{L}_{CE}(P_i^s, c_s(i)) + \mathcal{H}(P_i^{s\to t})) \\ &+ \sum_{i=1}^{N_{ut}} (\mathcal{L}_{CE}(P_i^t, c_t(i)) + \mathcal{H}(P_i^{t\to s})),\end{aligned} \tag{3}$$

where $\mathcal{H}(\cdot)$ is the entropy metric and $c(\cdot)$ denotes the cluster index of the corresponding sample.

B.2. Details for Model Training

Backbone Choices. We use ResNet-50 [3] pretrained on ImageNet [10] as our backbone $F(\cdot)$ when validating our method on the Office-31, the Office-Home and the VisDA-C datasets following [5, 14]. On the DomainNet dataset, we use ResNet-101 pretrained on ImageNet following PCS [14] for a fair comparison. In the meantime, we use ResNet-50 with pre-trained generalization weights provided by BrAD [2] for fair comparisons with FUDA results in BrAD.

Model Structure Details. For fair comparison with [5, 14, 2], we replace the last fully-connected layer with a randomly initialized linear layer and set the output feature dimension as 512. We perform the L2-normalization on the output features before sending them to the classifier $\phi(\cdot)$.

Hyper-parameter Choices. During model training, we use the SGD optimizer with a momentum of 0.9. As for the model learning rate, we choose 1e-2 for Office-31, Office-Home, DomainNet (comparing with PCS), 1e-3 for VisDA-C, and 1e-4 for DomainNet (comparing with BrAD). Throughout the training, we fix the batch size at 64. We simply set the values of λ_{MI} and λ_{self} as in [14]. For visual dispersal objectives, we empirically set $\alpha = 0.75$, $\lambda_{\text{X-VD}} = 1.0$, $\lambda_{\text{I-VD}} = 0.1$, $r_h^{\text{X}} \in \{0.75, 0.85\}$, $r_E^{\text{I}} = 0.1$ and $r_H^{\text{I}} \in \{0.65, 0.75\}$.

Training Device Choices. We use one NVIDIA GeForce RTX 3090 GPU for training and evaluation.

B.3. Implementation on the VisDA-C dataset

On the Office-31 [11], the Office-Home [13], and the DomainNet [8] datasets, we follow [14] and utilize the k-means clustering for $\mathcal{L}_{self}$ when training the baseline model. On the significantly bigger VisDA-C dataset, to improve the training efficiency, we substitute the time-consuming k-means clustering with the attention-based prototype generating strategy in [6] and conduct only source-to-target transfer learning in loss $\mathcal{L}_{self}$.

Simplified $\mathcal{L}_{self}$ in the baseline model. As shown in Tab. 1, the VisDA-C [9] dataset is significantly bigger. Training with k-means clustering on the VisDA-C dataset leads to unacceptable time costs. In order to improve the training efficiency of our proposed C-VisDiT, we substitute the time-consuming k-means approach with an attention-based strategy similar to [6]. We also provide an easier form of $\mathcal{L}_{self}$ to further reduce the calculation complexity. Specifically, for each source sample $\mathbf{x}_i^s \in \mathcal{D}_{ls} \cup \mathcal{D}_{us}$, we construct a memory bank to store the source sample features and update it by momentum in each epoch in order to reduce the impact of training fluctuation. The memory bank is denoted as $B_s = [\mathbf{f}_1^s, \mathbf{f}_2^s, \cdots, \mathbf{f}_{N_s}^s]$ where $\mathbf{f}_i^s$ is the feature of $\mathbf{x}_i^s$ inside the memory bank. We obtain the semantic distribution $p(y|\mathbf{f}_i^s) = [p_1, p_2, \cdots, p_c]$ for each source sample using the feature stored inside the memory bank. We concatenate the semantic distributions for all source samples as $K_s = [p(y|\mathbf{f}_1^s)^T, p(y|\mathbf{f}_2^s)^T, \cdots, p(y|\mathbf{f}_{N_s}^s)^T]$, where $N_s = N_{ls} + N_{us}$. We then construct the source class centers in an attention-based manner:

$$[\hat{\mu}_1^s, \hat{\mu}_2^s, \cdots, \hat{\mu}_c^s] = B_s K_s^T. \tag{4}$$

With the attention-based source class centers $\hat{\mu}_j^s$, we calculate the similarity distribution for each target sample $\hat{P}_i^{t\to s}$ following Eq. (2). We formulate the simplified $\mathcal{L}_{self}$ objective as:

$$\mathcal{L}_{self} = \sum_{i=1}^{N_{ut}} (\mathcal{H}(\hat{P}_i^{t\to s})). \tag{5}$$

Table 2: Adaptation accuracy (%) comparison on 1-shot and 3-shots labeled source per class on the DomainNet dataset.

Method	DomainNet: Target Acc.							
	R→C	R→P	R→S	P→C	P→R	C→S	S→P	Avg
1-shot labeled source								
Source Only	15.9	22.1	10.5	12.8	18.6	5.7	5.8	13.1
MME [12]	13.8	29.2	9.7	16.0	26.0	13.4	14.4	17.5
CDAN [7]	16.0	25.7	12.9	12.6	19.5	7.2	8.0	14.6
MDDIA [4]	18.0	30.6	15.9	15.4	27.4	9.3	10.2	18.1
CDS [5]	21.7	30.1	18.2	17.4	20.5	18.6	22.7	21.5
PCS [14]	39.0	51.7	**39.8**	26.4	38.8	**23.7**	23.6	34.7
C-VisDiT (Ours)	**39.1**	**52.2**	38.1	**27.6**	**43.8**	23.3	**27.1**	**35.9**
BrAD [2]	48.6	55.1	52.8	44.6	47.8	47.9	51.0	49.7
+C-VisDiT (Ours)	**51.0**	**55.8**	**54.2**	**45.6**	**47.9**	**49.9**	**52.3**	**51.0**
3-shot labeled source								
Source Only	23.7	40.3	22.9	19.3	48.3	19.1	15.8	27.1
MME [12]	22.8	46.5	14.5	25.1	50.0	20.1	24.9	29.1
CDAN [7]	30.0	40.1	21.7	21.4	40.8	17.1	19.7	27.3
MDDIA [4]	41.4	50.7	37.4	31.4	52.9	23.1	24.1	37.3
CDS [5]	44.5	52.2	40.9	40.0	47.2	33.0	40.1	42.5
PCS [14]	45.2	**59.1**	41.9	41.0	66.6	31.9	37.4	46.1
C-VisDiT (Ours)	**48.2**	58.7	**42.1**	**41.6**	**68.3**	**32.5**	**45.3**	**48.1**
BrAD [2]	60.6	62.8	61.6	56.6	63.6	59.8	61.0	60.9
+C-VisDiT (Ours)	**64.0**	**65.0**	**63.9**	**60.6**	**65.6**	**61.6**	**63.1**	**63.4**

Other implementation details. It is important to choose a decent initialization for training on the VisDA-C dataset. We train the model backbone with labeled source samples in a fully-supervised manner and utilize the weights of the trained backbone as the initialization for the training. Besides, during the first training epoch, we freeze the classification head $\phi(\cdot)$ and train our model with only $\mathcal{L}_{cls} + \lambda_{MI} \cdot \mathcal{L}_{MI}$.

C. Detailed Results for each Adaptation Setting on the DomainNet Dataset

DomainNet [8] is a challenging large-scale domain adaptation benchmark featuring 126 object classes. We present the overall performance of our C-VisDiT model on the DomainNet dataset in Tab. 4, Page 6 of the article. Here we present the detailed performance for each adaptation setting in Tab. 2. According to Tab. 2, our C-VisDiT can achieve 1.2%/2.0% and 1.3%/2.5% accuracy gain (1-shot/3-shots) comparing with PCS [14] and BrAD [2], respectively. Looking into each adaptation setting, our C-VisDiT can realize state-of-the-art results on 25 out of 28 settings. These results show that our proposed C-VisDiT establishes new state-of-the-art performance on the most challenging benchmark for FUDA, well demonstrating its superiority.

D. Additional Analysis Experiment

Ablation Studies on the Office-Home, the VisDA-C, and the DomainNet datasets. We present the result of ab-

Table 3: Performance contribution of each component on the Office-Home (3% / 6% labeled source), the VisDA-C (1% labeled source), and the DomainNet (1-shot / 3-shots labeled source) datasets in terms of adaptation accuracy (%).

Method	$\mathcal{L}_{\text{X-VD}}$	$\mathcal{L}_{\text{I-VD}}$	Office-Home	VisDA-C	DomainNet
Baseline	×	×	60.0 / 63.0	78.9	48.0 / 61.3
C-VisDiT-X	√	×	61.5 / 64.5	79.6	50.5 / 62.8
C-VisDiT-I	×	√	60.8 / 63.9	80.0	50.4 / 63.0
C-VisDiT	√	√	**62.3 / 65.4**	**80.5**	**51.0 / 63.4**

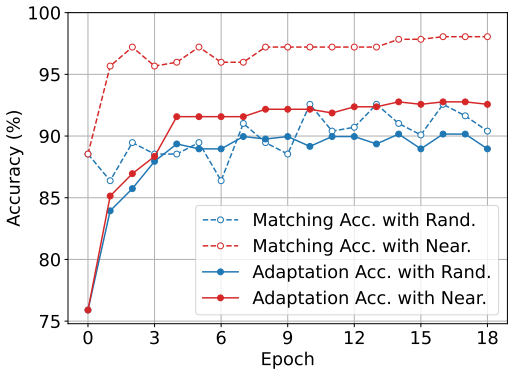

Figure 1: Analysis on the matching accuracy and the adaptation accuracy for both nearest source matching ("Near.") and random source matching ("Rand.") strategies in the W→D (1-shot) setting on the Office-31 dataset.

Table 4: Performance comparison of X-VD between different sample similarity measurement choices on the Office-31 dataset (%).

Method	1-shot / 3-shots
Cosine similarity	79.1 / 84.7
Euclidean distance (Ours)	**79.2 / 84.9**

Table 5: Adaptation accuracy when $\mathcal{D}_{ut}^{E}$ is added to $\mathcal{T}_{ut}^{H}$ for C-VisDiT-I on the Office-31 dataset (%).

Method	$\mathcal{T}_{ut}^{E}$	$\mathcal{T}_{ut}^{H}$	1-shot / 3-shots
Baseline	-	-	77.6 / 83.8
C-VisDiT-I	$\mathcal{D}_{ls} \cup \mathcal{D}_{ut}^{E}$ **(Ours)**	$\mathcal{D}_{ut}^{E} \cup \mathcal{D}_{ut}^{H}$	78.7 / **85.0**
	$\mathcal{D}_{ls} \cup \mathcal{D}_{ut}^{E}$ **(Ours)**	$\mathcal{D}_{ut}^{H}$ **(Ours)**	**78.8 / 85.0**

Table 6: Comparison between C-VisDiT and PCS [14] in terms of the training efficiency on the VisDA-C dataset.

	PCS [14]	C-VisDiT (Ours)
Training time (h)	16.21	3.49
Target accuracy (%)	79.0	**80.5**
Speedup	1	**4.64**

lation studies on the Office-Home [13], the VisDA-C [9], and the DomainNet [8] datasets in Tab. 3. We can see that both C-VisDiT-X and C-VisDiT-I can achieve better results compared to the baseline. Combining C-VisDiT-X and C-VisDiT-I, *i.e.*, adding both X-VD and I-VD to the baseline model, results in the highest performance gain. These results, again, show that our X-VD and I-VD strategies are effective and complementary.

Table 7: Performance comparison with other related works on the Office-31 dataset (%).

Method	1-shot / 3-shots
Baseline	77.6 / 83.8
MixStyle [16]	78.3 / 84.3
FlexMatch [15]	77.8 / 83.9
C-VisDiT (Ours)	**81.0 / 85.7**

Table 8: Adaptation accuracy with UDA methods using full source labels on the Office-31 dataset (%).

Method	Origin	+C-VisDiT
DANN [1]	82.2	**82.9**
DSAN [17]	88.4	**88.8**

Analysis on the nearest source matching in X-VD. In our proposed X-VD strategy, we conduct nearest source matching, where we match a given target sample to its nearest labeled source sample. To verify the effect of our nearest source matching, we analyze the matching accuracy, which reveals the consistency between the target sample and its neighboring labeled source sample. We compare our nearest source matching ("Near.") with random source matching ("Rand."), in which we match a given target sample to a random labeled source sample. As illustrated in Fig. 1, the employed nearest source matching can consistently obtain better matching accuracy than the random source matching (see dashed lines) during the training, which is in accordance with the comparison of the adaptation accuracy (see solid lines). These observations indicate that our nearest source matching can pull two samples with similar semantics but from different domains closer to each other, thus greatly boosting the adaptation.

Comparison to other sample similarity measurement metrics. In our proposed X-VD strategy, we utilize Euclidean distance to find similar samples across domains. To verify the effect of measurement choices, we replace Euclidean distance with the cosine similarity metric between feature vectors. According to Tab. 4, both metrics yield similar performance. This suggests that the effectiveness of our method is not affected much by the sample similarity measurement metrics.

Necessity to train on easy target samples in I-VD. Fig. 4 in Page 8 of the article indicates that easy target samples suffer from slight accuracy loss during training. To show that it is unnecessary to train on easy target samples as a compensation for the accuracy loss, we further add easy target samples $\mathcal{D}_{ut}^{E}$ to the training hard sample set $\mathcal{T}_{ut}^{H} = \mathcal{D}_{ut}^{E} \cup \mathcal{D}_{ut}^{H}$. As shown in Tab. 5, adding $\mathcal{D}_{ut}^{E}$ to $\mathcal{T}_{ut}^{H}$ yields comparable results with the standard implementation $\mathcal{T}_{ut}^{H} = \mathcal{D}_{ut}^{H}$, indicating that it is not necessary to train on the easy target samples.

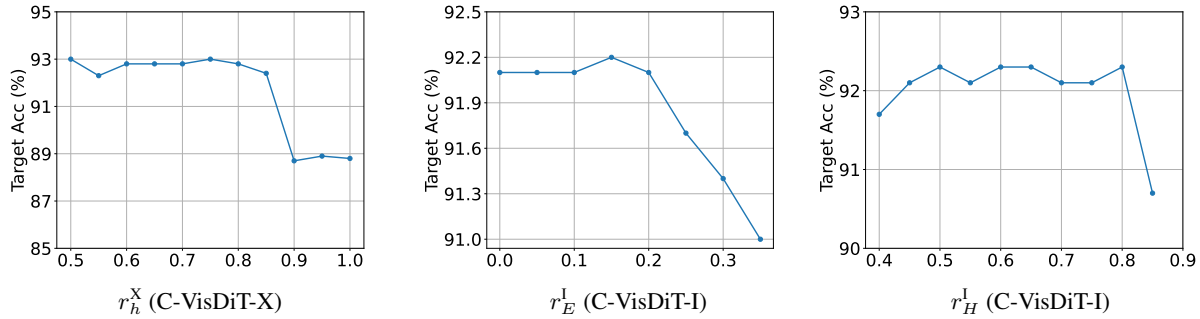

Figure 2: Analysis of hyper-parameter r_h^{X}, r_E^{I}, and r_H^{I} in D→W (1-shot) on the Office-31 dataset. To reach promising performance, both r_h^{X}, r_E^{I} and r_H^{I} should be smaller than a value threshold (0.85 for r_h^{X}, 0.2 for r_E^{I}, and 0.8 for r_H^{I}).

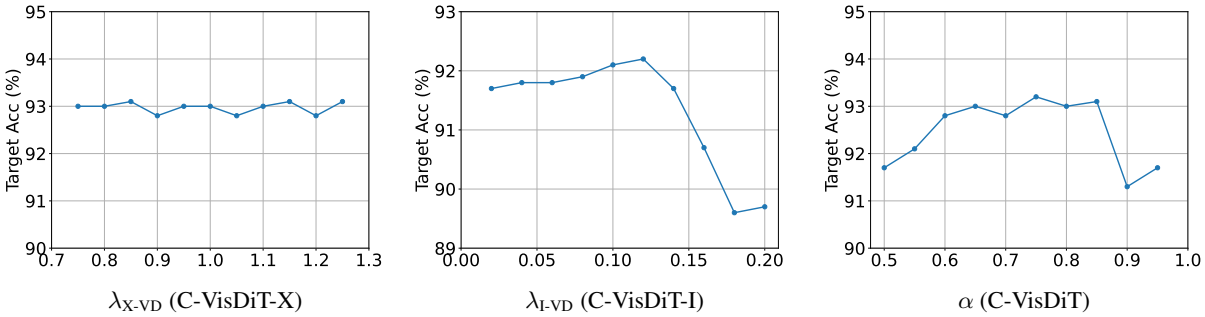

Figure 3: Analysis of hyper-parameter $\lambda_{\text{X-VD}}$, $\lambda_{\text{I-VD}}$, and α in D→W (1-shot) on the Office-31 dataset. The value of $\lambda_{\text{X-VD}}$ hardly affects adaptation accuracy on the target domain, while the value of $\lambda_{\text{I-VD}}$ should be smaller than a value threshold (around 0.15). The value of α should be between a lower threshold (around 0.6) and an upper threshold (around 0.85).

Efficiency Comparison with PCS on VisDA-C. In Appendix B.3, we claim that our simplified training pipeline on the VisDA-C dataset could effectively improve the training efficiency. To show the effect after simplifying the baseline model, we compare our simplified C-VisDiT with the existing state-of-the-art PCS [14] in terms of training efficiency. We train both methods on the VisDA-C dataset for 20 epochs from scratch, and report the training time and accuracy results. As indicated by Tab. 6, the proposed method can achieve significant speedup, *i.e.*, 4.6 times compared with PCS, while enjoying a superior performance by 1.5%. These results demonstrate that our simplifying strategies are especially effective on large datasets.

Comparison to other related works. We provide performance comparisons with other related works in related fields, namely MixStyle [16] and FlexMatch [15]. For MixStyle, we incorporate it into our baseline model using its published code. For FlexMatch, we implement it to FUDA by applying it to target samples based on the baseline model. The results are shown in Tab. 7. Comparing to our C-VisDiT, both MixStyle and FlexMatch achieve inferior performance, and even worse than our C-VisDiT-X (79.2 / 84.9). Again, this demonstrates the superiority of our C-VisDiT, especially in the field of FUDA.

E. Results on the UDA Problem

To demonstrate the exclusive advantage of our C-VisDiT for FUDA, we further investigate the effect of C-VisDiT on the UDA problem using full source labels. We formally choose two remarkable UDA methods, *i.e.*, DANN [1] and DSAN [17], as the baseline model, and equip them with our C-VisDiT. As shown in Tab. 8, our strategies still lead to performance gains, albeit limited (0.7% on DANN and 0.4% on DSAN). This can be attributed to that the knowledge is more confident in UDA due to its abundant supervision. As a result, the reliability of knowledge transfer can be guaranteed, leading to better adaptation on target samples. Therefore, on the UDA problem, our confidence-based strategies cannot have significant effect as in FUDA. These results confirm that our confidence-based strategies have a distinctive edge for FUDA.

F. Hyper-parameter Analysis

r analysis. In our proposed C-VisDiT, we manually choose the ratios controlling the proportion of different confidence-level samples. The choice of r hyper-parameters is correlated to the specific settings, as our model achieves different adaptation accuracy in different

Table 9: Performance comparison of different hyper-paramameter β choices on the Office-31 dataset (%).

Method	$\beta > 0.5$	random β	$\beta < 0.5$
C-VisDiT-X	**79.2%/84.9%**(Ours)	78.8%/84.8%	78.5%/84.3%
C-VisDiT-I	78.4%/84.6%	78.6%/84.7%	**78.8%/85.0%**(Ours)

settings. Here we provide a detailed hyper-parameter analysis for the D→W (1-shot) setting on the Office-31 dataset. We investigate the effect of r_h^{X}, r_E^{I}, and r_H^{I} via adaptation accuracy. As shown in Fig. 2, the adaptation performance basically remains stable in a wide range of r values in the D→W (1-shot) setting, *e.g.*, $r_h^{\mathrm{X}} \leq 0.85$, $r_E^{\mathrm{I}} \leq 0.2$, and $r_H^{\mathrm{I}} \leq 0.8$. As a result, we choose $r_h^{\mathrm{X}} = 0.75$, $r_E^{\mathrm{I}} = 0.1$ and $r_H^{\mathrm{I}} \leq 0.65$ for the D→W (1-shot) setting. For other settings, we empirically choose $r_h^{\mathrm{X}} \in \{0.75, 0.85\}$, $r_E^{\mathrm{I}} = 0.1$ and $r_H^{\mathrm{I}} \in \{0.65, 0.75\}$.

$\lambda_{\text{X-VD}}$ and $\lambda_{\text{I-VD}}$ analysis. We investigate the effect of $\lambda_{\text{X-VD}}$ and $\lambda_{\text{I-VD}}$ via adaptation accuracy in the D→W (1-shot) setting on the Office-31 dataset. The results are shown in Fig. 3. While the adaptation performance is not sensitive to $\lambda_{\text{X-VD}}$, it suffers from bigger values of $\lambda_{\text{I-VD}}$ in the D→W (1-shot) setting. We empirically choose $\lambda_{\text{X-VD}} = 1.0$ and $\lambda_{\text{X-VD}} = 0.1$ in other settings for model evaluation experiments.

α analysis. We investigate the effect of α utilized to sample β in visual dispersal strategies, where $\beta \sim \text{Beta}(\alpha, \alpha)$. Similarly, we conduct experiments in the D→W (1-shot) setting on the Office-31 dataset. The results are shown in Fig. 3. Empirically, we choose $\alpha = 0.75$ for the proposed C-VisDiT method in all settings.

β analysis. In Equation 10 and 14, Page 4 and 5 of the article, we use hyper-parameter β to control the sample importance inside hybrid samples. In X-VD, we ensure that $\beta > 0.5$ to put more importance on target samples to be learned. In I-VD, we guarantee that $\beta < 0.5$ to put more importance on har target samples to be learned. As shown in Tab. 9, our choice of β yields the best performance, implying that it is beneficial towards both X-VD and I-VD strategies.

G. Image Retrieval Results

To qualitatively present that our proposed C-VisDiT can align semantically similar images across domains, we analyze our proposed C-VisDiT and the existing state-of-the-art, PCS [14], via image retrieval. Given an unlabeled source sample $\mathbf{x}_i^{us}$, we find the three closest target domain samples measured by Euclidean distance in the feature space. As shown in Fig. 4, some features trained with PCS are aligned via visual textures and patterns instead of actual image semantics. For example, PCS tends to match letter trays (row 3) with bookcases, as both objects have similar layer structures. As a comparison, our proposed C-VisDiT aligns features that are semantically similar across domains, providing correct retrieval results for unlabeled source samples.

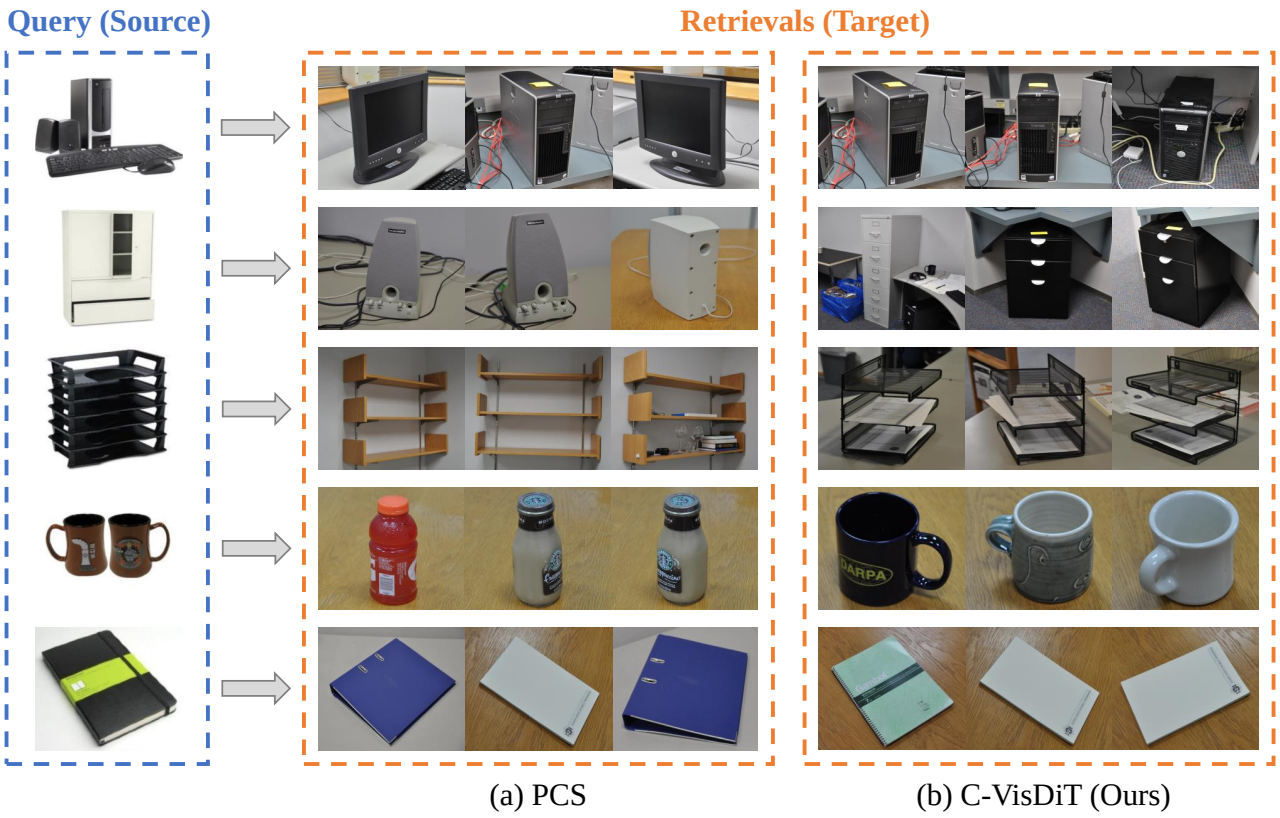

Figure 4: Image retrieval examples of the closest target domain samples given an unlabeled sample from the source domain, using PCS [14] (a) and our proposed C-VisDiT (b). The query images (from top to bottom) belong to the following categories: desktop computer, file cabinet, letter tray, mug, and paper notebook. PCS features tend to match images with similar visual patterns and textures. (Row 1: PCS matches monitors to the desktop computer query. Row 2: PCS matches speakers to the file cabinet query. Row 3: PCS matches bookcases to the letter tray query. Row 4: PCS matches bottles to the mug query. Row 5: PCS matches ring binders to the paper notebook query.) As a comparison, our C-VisDiT correctly matches images with similar semantics.

References

[1] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 3, 4

[2] Sivan Harary, Eli Schwartz, Assaf Arbelle, Peter Staar, Shady Abu-Hussein, Elad Amrani, Roei Herzig, Amit Alfassy, Raja Giryes, Hilde Kuehne, et al. Unsupervised domain generalization by learning a bridge across domains. In *Computer Vision and Pattern Recognition*, pages 5280–5290, 2022. 1, 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[4] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International conference on machine learning*, pages 4816–4827, 2020. 2

[5] Donghyun Kim, Kuniaki Saito, Tae-Hyun Oh, Bryan A. Plummer, Stan Sclaroff, and Kate Saenko. Cross-domain self-supervised learning for domain adaptation with few source labels. *CoRR*, abs/2003.08264, 2020. 1, 2

[6] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 2

[7] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[8] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision*, pages 1406–1415, 2019. 1, 2, 3

[9] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 1, 2, 3

[10] 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. 1

[11] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226, 2010. 1, 2

[12] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *International Conference on Computer Vision*, pages 8050–8058, 2019. 2

[13] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 1, 2, 3

[14] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto L. Sangiovanni-Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Computer Vision and Pattern Recognition*, pages 13834–13844, 2021. 1, 2, 3, 4, 5, 6

[15] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 3, 4

[16] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 3, 4

[17] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1713–1722, 2020. 3, 4