Independent Feature Decomposition and Instance Alignment for Unsupervised Domain Adaptation
Technical Appendix

1 More details about IndUDA

1.1 The structure of invertible flow

Similar to the method in [Liu *et al.*, 2021], the structure of our invertible flow model consists of N blocks. Suppose that the input of i-th block is $\mathbf{u}^i_{1:2d}$, where $2d$ denotes the dimension size. The process of the i-th block with residual can be written as:

$$\begin{aligned}
[\mathbf{u}^i_{1:d}, \mathbf{u}^i_{d:2d}] &= \text{Split}\left(\mathbf{u}^i_{1:2d}\right), \\
\mathbf{u}^{i+1}_{1:d} &= \mathbf{u}^i_{1:d} + \phi_1\left(\mathbf{u}^i_{d:2d}\right), \\
\mathbf{u}^{i+1}_{d:2d} &= \mathbf{u}^i_{d:2d} \odot \exp\left(\phi_2(\mathbf{u}^{i+1}_{1:d})\right) + \phi_3\left(\mathbf{u}^{i+1}_{1:d}\right), \\
\mathbf{u}^{i+1}_{1:2d} &= \text{Concat}\left(\mathbf{u}^{i+1}_{1:d}, \mathbf{u}^{i+1}_{d:2d}\right),
\end{aligned}$$

where ϕ_1, ϕ_2, ϕ_3 are three different networks with the same structure. The output $\mathbf{u}^{i+1}_{1:2d}$ will be the input of the next block. The inverse projection that maps $\mathbf{u}^{i+1}_{1:2d}$ back to $\mathbf{u}^i_{1:2d}$ is as follows:

$$\begin{aligned}
\mathbf{u}^{i+1}_{1:d}, \mathbf{u}^{i+1}_{d:2d} &= \text{Split}\left(\mathbf{u}^{i+1}_{1:2d}\right), \\
\mathbf{u}^i_{d:2d} &= \left(\mathbf{u}^{i+1}_{d:2d} - \phi_3\left(\mathbf{u}^{i+1}_{1:d}\right)\right) / \exp(\phi_2\left(\mathbf{u}^{i+1}_{1:d}\right), \\
\mathbf{u}^i_{1:d} &= \mathbf{u}^{i+1}_{1:d} - \phi_1\left(\mathbf{u}^i_{d:2d}\right), \\
\mathbf{u}^i_{1:2d} &= \text{Concat}\left(\mathbf{u}^i_{1:d}, \mathbf{u}^i_{d:2d}\right).
\end{aligned}$$

Obviously, the above process illustrates that the feature mapping is invertible. The input and output of this mapping can be transformed into each other through forward and reverse processes.

1.2 Contrastive domain discrepancy loss

Contrastive domain discrepancy (CDD) loss [Kang *et al.*, 2019] is defined as

$$\mathcal{D}_H(m,n) = \frac{\sum_{c\in\mathcal{C}'} \mathcal{D}^{c,c}(m^c, n^c)}{|\mathcal{C}'|} - \frac{\sum_{c_1,c_2\in\mathcal{C}'}^{c_1\neq c_2} \mathcal{D}^{c_1,c_2}(m^{c_1}, n^{c_2})}{|\mathcal{C}'|(|\mathcal{C}'|-1)}, \tag{1}$$

where $m^c \in Q$, $n^c \in T$ mean the c category samples from the Q and T sets respectively. $|\mathcal{C}'|$ is the category number. The first term minimizes the intra-class discrepancy, while the second term maximizes the inter-class discrepancy between two sets. The distribution measure between two category samples from two sets is defined as

$$\begin{aligned}
\mathcal{D}^{c_1,c_2}(m^{c_1}, n^{c_2}) = &\sum_{i,j} \frac{k(m^{i,c_1}, m^{j,c_1})}{|Q_{c1}|^2} + \sum_{i,j} \frac{k(n^{i,c_2}, n^{j,c_2})}{|T_{c_2}|^2} \\
&- 2\sum_{i,j} \frac{k(m^{i,c_1}, n^{j,c_2})}{|Q_{c1}|\cdot|T_{c_2}|},
\end{aligned}$$

where m^{i,c_1} means the i-th sample from the set Q belonging to the category c_1 and n^{i,c_2} means the i-th sample from the set Q belonging to the category c_1. $|Q_{c_1}|$ and $|T_{c_2}|$ denote the c_1 category sample number in the set Q and c_2 category sample number in the set T. $k(\cdot,\cdot)$ represents the kernel function which embeds feature in Reproducing Kernel Hilbert Space (RKHS).

1.3 Algorithm

The overall algorithm is given as Algorithm 1.

Algorithm 1: Pseudo code of IndUDA

```
Input: Unlabeled target data $\mathcal{T}$, Labeled source data
       $\mathcal{S}$, Model $\{F, C, \text{INN}\}$, max epoch $E$, iteration
       number $K$ per epoch
Output: $\theta$ of the adapted model $\{F, C\}$
1  for $e = 1$ to $E$ do
2      obtain pseudo label $\hat{y}^i_t$ on target data $\mathcal{T}$ by source
       model
3      for $k = 1$ to $K$ do
4          Sample batch $(x^i_s, y^i_s)$ from $\mathcal{S}$ and compute
           $\mathcal{L}_{ce}$ by Eq.(10) in Main text
5          Use Eqs.(1-3) in Main text to compute
           gradient mask for each category by $(x^i_s, y^i_s)$
6          Sample batch $(x^j_s, y^j_s)$ from $\mathcal{S}$ and $(x^j_t, \hat{y}^j_t)$ of
           same categories from $\mathcal{T}$
7          Compute $\mathcal{L}_{swap}$ using Eq.(5) in Main text
8          Compute $\mathcal{L}_{align}$ using Eq.(8) in Main text
9          Back-propagate and update model
10     end
11 end
```

Method	Venue	A→ D	A→ W	D→ A	D→ W	W→A	W→D	*Avg.*
ResNet50	CVPR16	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DAN	ICML15	78.6	80.5	63.6	97.1	62.8	99.6	80.4
CAN	CVPR19	94.5	95.0	78.0	99.1	77.0	99.8	90.6
TSA	CVPR21	92.6	94.8	74.9	99.1	74.4	**100.0**	89.3
ATDOC	CVPR21	94.4	94.5	75.6	98.9	75.2	99.6	89.7
SIDA	IJCAI22	95.7	94.5	76.6	**99.2**	76.2	100.0	90.4
SUDA	CVPR22	91.2	90.8	72.2	98.7	71.4	**100.0**	87.4
CaCo	CVPR22	91.7	89.7	73.1	98.4	72.8	**100.0**	87.6
DANN	JMLR16	79.7	82.0	68.2	96.9	67.4	99.1	82.2
CDAN	NIPS18	89.8	93.1	70.1	98.2	68.0	**100.0**	86.5
MetaAlign	CVPR21	94.5	93.0	75.0	98.6	73.6	**100.0**	89.2
DWL	CVPR21	91.2	89.2	73.1	**99.2**	69.8	**100.0**	87.1
SRADA	IJCAI20	91.7	95.2	74.5	98.6	73.7	**100.0**	89.0
SDAT+ELS	ICLR23	93.4	93.6	78.7	99.0	77.5	**100.0**	90.4
DSR	IJCAI19	92.4	93.1	73.5	98.7	73.9	99.8	88.6
IC^{2}FA	ACMMM21	95.4	94.6	77.3	**99.2**	77.6	**100.0**	90.7
IndUDA	Ours	**96.6** ±0.3	**95.8** ±0.2	**78.9** ±0.3	**99.2** ±0.1	**78.1** ±0.1	**100.0** ±0.0	**91.4**

Table 1: Classification accuracy for all six tasks on *Office-31* dataset based on ResNet50.

2 Further experiments

2.1 Experimental setup

Datasets. We further conduct experiments on two standard domain adaptation datasets. ***ImageClef-DA*** [Long *et al.*, 2017] contains three domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P), which serves as a baseline for the ImageCLEF 2014 domain adaptation challenge. There are 50 photos in each of the 12 categories for each domain. ***Domainnet*** [Peng *et al.*, 2019] is a dataset with six domains in common object and 345 categories (classes) of items, including cello, plane, bird, and bracelet, are present throughout all domains. The domains include painting(P): artistic representations of objects in the form of paintings; quickdraw(Q): drawings of the global players of the game "Quick Draw!"; real(R): photos and real world images; sketch(S): sketches of specific objects; infograph(I): infographic images with specific object, and clipart(C): collection of clipart images".

Implementation details. The settings are the same as the main paper except that for the learning rate scheduler $\eta_p = \eta_0(1+ap)^{-b}$, we set $a = 10$ and $b = 0.75$ on Imageclef-DA and DomainNet as Office-31 and Office-home.

2.2 Further comparison results

Tables 1-3 show the experiment results of our IndUDA and other state-of-the-art methods on datasets *Office-31, Imageclef-DA, DomainNet*, respectively. We reproduce the result of CAN on Imageclef-DA, and other results come from the corresponding papers. On all datasets, it is obvious that all domain adaptation methods have significantly improvements compared with the source-only method. This demonstrates the effectiveness of feature alignment process. On *Imageclef-DA*, IndUDA offers classification accuracy of 91.0% on six task. In addition, our method achieves the best results for each sub task on above two datasets. For the difficult dataset *DomainNet*, our result are 31.4% in average sense, which is 7.8% higher than the second best method.

2.3 Further ablation results

Ablation on random variable r. To prove that the random variable r can help improve the generalization ability. We do experiments with and without the random sampling variable r. From Tab.4, the results with random sampling are better than those without sampling, i.e., 0.7% higher in average on *Office-31* and 0.8% higher on *Office-home*. This shows that simply reconstructing feature by domain-invariant part will lose the representation diversity; while random sampling can increase variance and keep the mathematical expectation unchanged.

Ablation on block number in INN. To show the performances of using different blocks in INN, the cases with 1-5 block numbers are evaluated, which is shown in Tab. 5. We can find that only using one or two blocks do not work because the network has not a good fitting ability to reach an ideal latent space. The performance is increased with the amount of INN blocks increase, but becomes saturated when the block number is above three. Further, it should be noted that the results on task $A \rightarrow W$ and $A \rightarrow D$ in *Office-31* change little with the increase of block numbers. It is due to that the feature of domain amazon on *Office-31* originally contains little domain-invariant noise, which means the denoising effect is not particularly obvious in these two tasks. Since INN loses some representation ability while considering the strictly invertible attribute, considering fitting ability and computational efficiency comprehensively, we set the block number as 5.

Ablation on different kind of loss functions on D_H. To show the performance of different kind of loss functions on D_H in the losses $\mathcal{L}_{swap}$ and $\mathcal{L}_{align}$, we compare CDD loss with L_2 loss, which is shown in Table 6. For $\mathcal{L}_{swap}$, applying CDD loss performs better than L_2 loss. Because domain-invariant parts cannot be exactly the same for each category as we analyzed in the main text and the category does not change after swap, CDD loss is more suitable to measure the distribution discrepancy. While for $\mathcal{L}_{align}$, the performance

Method	Venue	I→ P	P→ I	I→ C	C→ I	C→ P	P→ C	*Avg.*
ResNet50	CVPR16	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN	ICML15	74.5	82.2	92.8	86.3	69.2	89.8	82.5
CAN	CVPR19	78.7	94.0	97.5	93.5	79.9	97.7	90.3
TSA	CVPR21	78.6	92.8	97.0	92.8	79.0	95.2	89.2
DLAD	TMM22	79.8	92.8	95.7	92.8	78.8	95.3	89.2
DANN	JMLR16	75.0	86.0	96.2	87.0	74.3	91.5	85.0
CDAN	NIPS18	77.7	90.7	97.7	91.3	74.2	94.3	87.7
CDAL	ACMMM22	**80.4**	93.7	97.8	93.3	80.2	97.5	90.5
AMRC	TMM22	80.3	94.0	97.5	93.7	79.5	96.8	90.3
IndUDA	Ours	**80.4** ±0.1	**94.8** ±0.2	**98.0** ±0.0	**93.8** ±0.1	**80.9** ±0.3	**97.8** ±0.1	**91.0**

Table 2: Classification accuracy for all six tasks on Image-clef dataset based on ResNet50.

ResNet-50	clp	inf	pnt	qdr	rel	skt	*Avg.*	MCD	clp	inf	pnt	qdr	rel	skt	*Avg.*	BNM	clp	inf	pnt	qdr	rel	skt	*Avg.*
clp	-	14.2	29.6	9.5	43.8	34.3	26.3	clp	-	15.4	25.5	3.3	44.6	31.2	24.0	clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	21.8	-	23.2	2.3	40.6	20.8	21.7	inf	24.1	-	24.0	1.6	35.2	19.7	20.9	inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	24.1	15.0	-	4.6	45.0	29.0	23.5	pnt	31.1	14.8	-	1.7	48.1	22.8	23.7	pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	12.2	1.5	4.9	-	5.6	5.7	6.0	qdr	8.5	2.1	4.6	-	7.9	7.1	6.0	qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	32.1	17.0	36.7	3.6	-	26.2	23.1	rel	39.4	17.8	41.2	1.5	-	25.2	25.0	rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	30.4	11.3	27.8	3.4	32.9	-	21.2	skt	37.3	12.6	27.2	4.1	34.5	-	23.1	skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg.	24.1	11.8	24.4	4.7	33.6	23.2	20.3	Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5	Avg.	37.6	10.3	31.4	4.2	40.9	27.3	25.3
CDAN	clp	inf	pnt	qdr	rel	skt	*Avg.*	SWD	clp	inf	pnt	qdr	rel	skt	*Avg.*	IndUDA	clp	inf	pnt	qdr	rel	skt	*Avg.*
clp	-	13.5	28.3	9.3	43.8	30.2	25.0	clp	-	14.7	31.9	10.1	45.3	36.5	27.7	clp	-	17.8	43.2	17.8	57.2	49.2	37.0
inf	18.9	-	21.4	1.9	36.3	21.3	20.0	inf	22.9	-	24.2	2.5	33.2	21.3	20.0	inf	30.6	-	32.0	2.4	49.1	29.1	28.6
pnt	29.6	14.4	-	4.1	45.2	27.4	24.2	pnt	33.6	15.3	-	4.4	46.1	30.7	26.0	pnt	44.9	18.7	-	7.1	57.8	38.6	33.4
qdr	11.8	1.2	4.0	-	9.4	9.5	7.2	qdr	15.5	2.2	6.4	-	11.1	10.2	9.1	qdr	24.0	4.9	14.0	-	23.0	17.4	16.7
rel	36.4	18.3	40.9	3.4	-	24.6	24.7	rel	41.2	18.1	44.2	4.6	-	31.6	27.9	rel	51.1	22.5	52.1	7.1	-	40.8	34.7
skt	38.2	14.7	33.9	7.0	36.6	-	26.1	skt	44.2	15.2	37.3	10.3	44.7	-	30.3	skt	55.9	18.6	45.9	14.6	55.7	-	38.1
Avg.	27.0	12.4	25.7	5.1	34.3	22.6	21.2	Avg.	31.5	13.1	28.8	6.4	36.1	26.1	23.6	Avg.	41.3	16.5	37.4	9.8	48.6	35.0	**31.4**

Table 3: Comparison with the state-of-the-art methods on *DomainNet* dataset. The column-wise fields denote the source domain while the row-wise fields represent the target domain. Backbone: Resnet-50.

r	Office-31				Office-home
	A→D	A→W	D→A	W→A	*Avg.*
✓	**96.6**	**95.8**	**78.9**	**78.1**	**74.3**
×	94.9	95.4	78.2	77.4	73.4

Table 4: Ablation study on random variable r.

Block Number	1	2	3	4	5
A→D	95.0	95.8	96.0	95.8	**96.6**
A→W	93.8	95.2	95.5	**95.8**	**95.8**
D→A	78.3	76.7	78.1	78.5	**78.9**
W→A	74.0	74.6	78.0	77.4	**78.1**
Office-home	72.4	73.2	74.1	74.0	**74.3**

Table 5: Ablation study on different number of blocks.

$\mathcal{L}_{swap}$	$\mathcal{L}_{align}$	Office-31				Office-home
		A→D	A→W	D→A	W→A	*Avg.*
L_2	L_2	95.6	94.8	76.9	75.7	73.1
CDD	L_2	96.0	95.1	78.5	77.2	73.7
L_2	CDD	95.6	95.1	78.1	77.4	73.4
CDD	CDD	**96.6**	**95.8**	**78.9**	**78.1**	**74.3**

Table 6: Ablation study on different kind of loss functions on D_H.

of using CDD is also better than L_2 loss. Although L_2 loss can help finding domain-invariant part to some extent, it is not discriminative enough, which means it cannot guarantee the consistency of class distribution compared with CDD loss.

Analysis on computational complexity.

2.4 Visualization

To further verify the effectiveness of the proposed IndUDA, we employ t-SNE [Van der Maaten and Hinton, 2008] and Grad-CAM[Selvaraju *et al.*, 2017] on *Office-31* tasks $A \to D$ and $A \to W$ to visualize the effectiveness of our method. We compare our method with Direct alignment and variants IndUDA (w/o L_{swap}) and IndUDA (Doublemap). Direct alignment means the method using statistical matching method [Kang *et al.*, 2019]. IndUDA (w/o $\mathcal{L}_{swap}$) means our proposed method does not use the loss $\mathcal{L}_{swap}$ and only the loss $\mathcal{L}_{con}$ is used to train the invertible flow; while IndUDA (Doublemap) means domain adaptation based on only the loss $\mathcal{L}_{align}$ i.e., the invertible flow is changed by double mapping.

Visualization on T-SNE. The visual results on *Office-31* dataset, are shown in Fig.1 and Fig.2, respectively. Fig.1a and Fig.2a show the visualized results of Direct alignment. The feature is not denoised, so it is aligned with domain-specific noise, leading to relatively dispersed and more noise feature. Fig.1b and Fig.2b show the visualized results of IndUDA (w/o $\mathcal{L}_{swap}$). Since there is no operation of feature

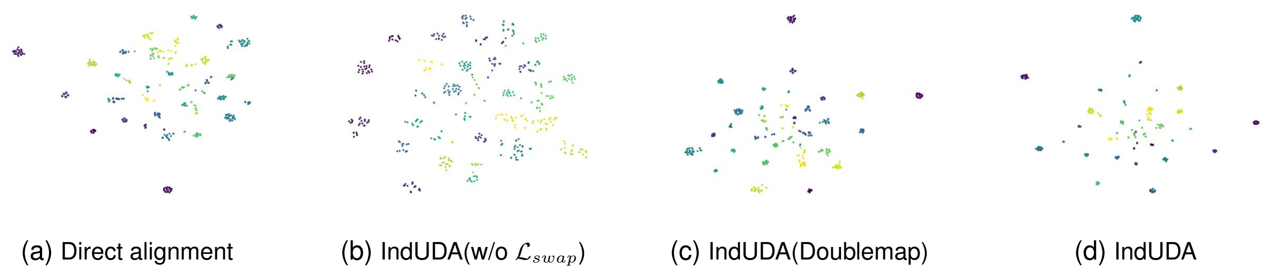

Figure 1: Visualization results on T-sne of task A$\rightarrow$D

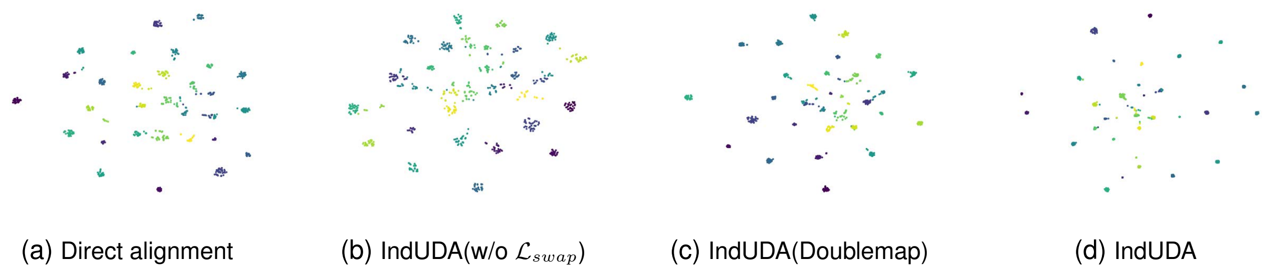

Figure 2: Visualization results on T-sne of task A$\rightarrow$W

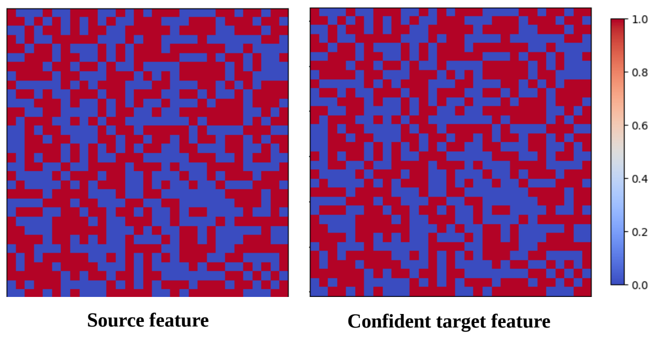

Figure 3: Visualization results of channel-mask.

swap, which means no compact constrain added in the latent space, the separated domain-invariant features contain more noise, resulting in the dispersed extracted features. Fig.1c and Fig.2c demonstrate the results of IndUDA (Doublemap), which cannot reconstruct the corresponding features perfectly and bring more deviations and some categories are still relatively dispersed. Compared with the above mentioned methods, the proposed IndUDA shows the best results, as shown in Fig.1d and Fig.2d. IndUDA not only considers denoising the domain-specific part by swapping features, but also gathers each category samples by invertible feature mapping. The use of the swap feature strategy and contrastive domain discrepancy constraint force the in-class features more compact and factor out the domain-invariant features.

Visualization on GRAD-CAM. We randomly select four categories, and then select one image for activate mapping visualization for each category. From top to bottom, the figures represent the visualization results of Direct alignment, IndUDA (w/o $\mathcal{L}_{swap}$), IndUDA (Doublemap) and IndUDA. The first column in Fig.4 is the visualization results of the original image. The results of Direct alignment are shown in the second column of Fig.4. This approach does find some important features, but it ignores the characteristics of the whole object and cannot completely mark the exact local points, indicating that it still contains domain noises which affect the final classification performance. The results of IndUDA (w/o $\mathcal{L}_{swap}$) method are shown in the third column of Fig.4. Without swapping domain-invariant parts, the mined domain-invariant part is separated with domain-specific noises. So some category characteristics are also missed. The results of IndUDA (Doublemap) are shown in the fourth column of Fig.4. Without invertible flow, we cannot achieve perfect feature reconstruction through double mapping. As a result, the feature reconstructed from domain-invariant part will deviate to some extent. So some important characteristics are not marked obviously. That means we cannot guarantee the original feature space can be compact enough simply via feature extractor. The results of IndUDA are shown in the last column of Fig.4. Compared with the variants, IndUDA can better estimate both of category specific local and global attentions in the image. It is due to that our method can obtain more accurate domain-invariant part with less domain-specific noises and make the intra-class samples more compact.

Visualization on channel-mask. In this part, we will show the channel masks of confident pseudo-labelled target sample

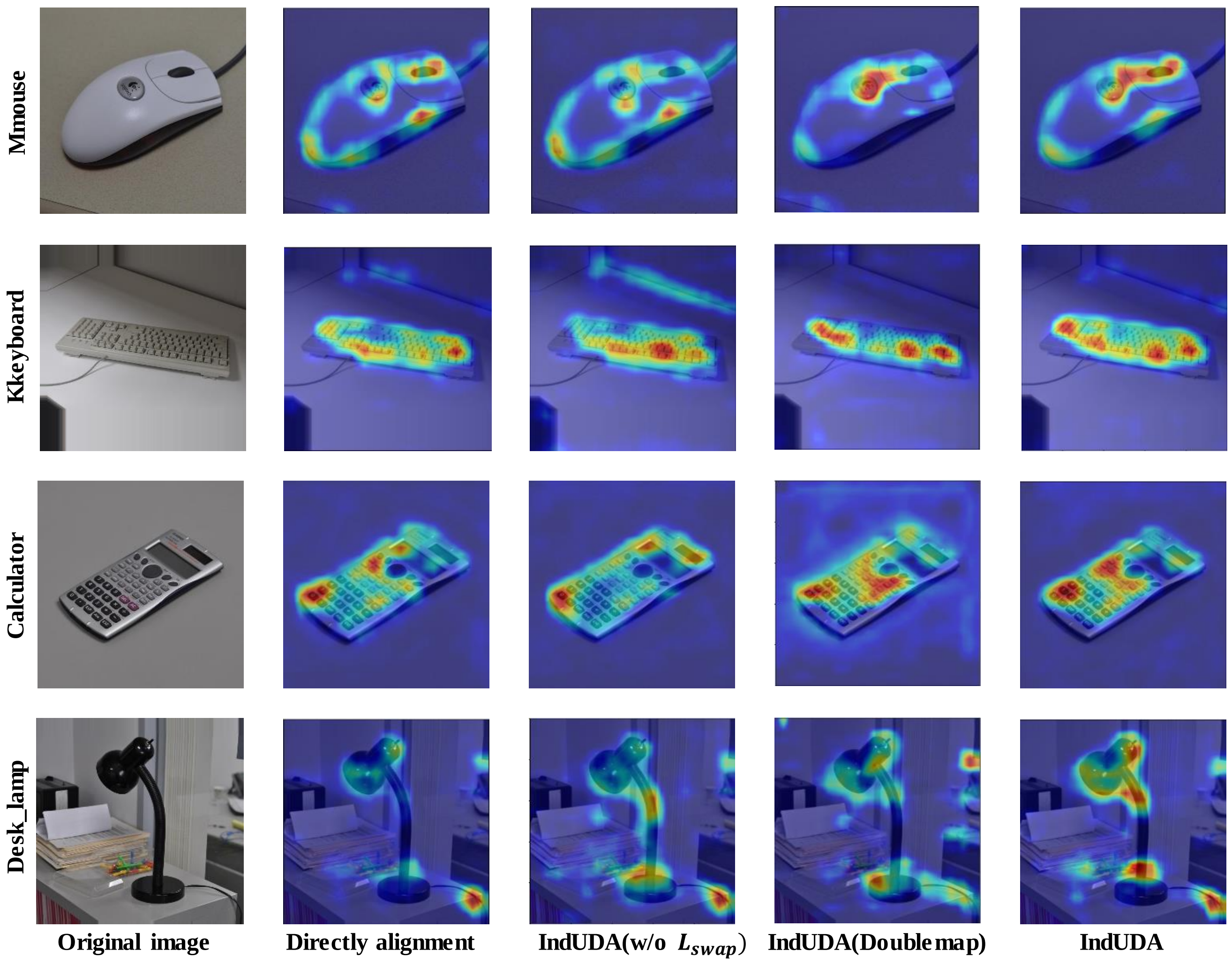

Figure 4: The GRAD-CAM visualization on $A \to D$ on *Office-31* dataset.

feature are the same as the source features. The experiment is performance on on *Office-31*. Because the channel number is huge, if we show everything, we will see nothing. So we randomly select 32 channels in 31 categories to construct a heat-map. As shown in Fig.3, the horizontal and vertical axes correspond to the channel and category respectively. The red square means this channel is selected respect to a category. The visualization results show that the masks of source feature and confident target sample feature are exactly the same. Because the confident sample target feature is quite similar to the source feature respect to the corresponding category, so their masks should also be same.

References

[Kang *et al.*, 2019] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[Liu *et al.*, 2021] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 13365–13374, 2021.

[Long *et al.*, 2017] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217. PMLR, 2017.

[Peng *et al.*, 2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization.

In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.