Research Paper Summary - Tuan Nguyen

Unsupervised Domain Adaptation without Source Data by Casting a BAIT

1. Main contribution

- Does not require the usage of pseudo-labeling.
- Used in source-free setting (SFDA) by building an additional classifer with corresponding class prototypes of source classifier.
- Achieve similar results or outperforms existing UDA and SFDA. Achieved SOTA performance on VisDA.

2. Proposed method

2.1. Prototype of source classifier as anchor

Model structure: a feature extractor f, and a classifier head C_1 (contains only one fully connected layer with weight normalization)
Phase 1: train the baseline model on labeled source data $\mathcal{D}_s$ with standard cross-entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{K} I_{\left[k=y_i^s\right]} \log p_k\left(x_i^s\right), \tag{1}$$

where K is the number of classes, p_k is the k-th element of the softmax output, and $I_{[z]}$ is the indicator function which is 1 if z is true, and 0 otherwise.

2.2. Prototype of second classifier as bait

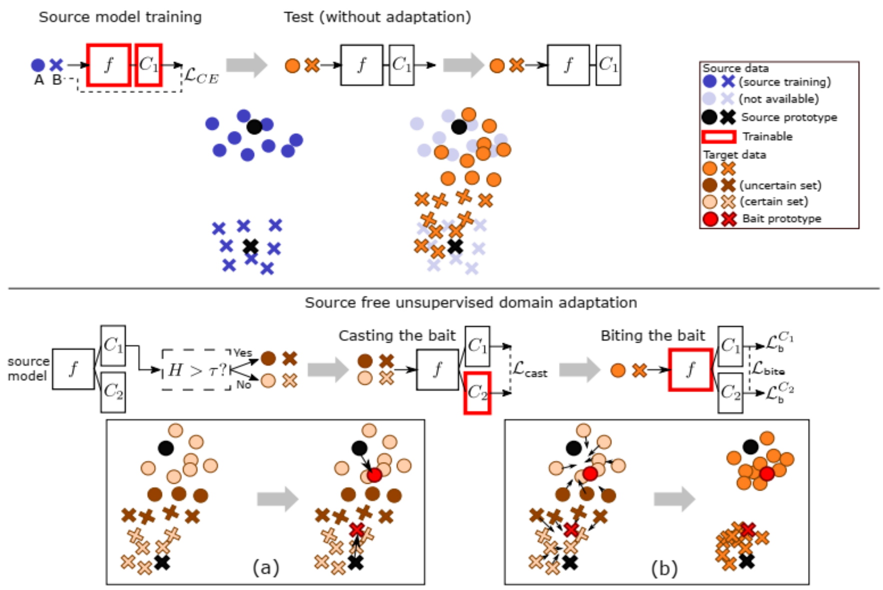

Figure 2: Illustration of training process. The top shows that the source-training model fails on target domain due to domain shift. The bottom illustrates our adaptation process. Bottom (a): splitting feature into certain set and uncertain set by whether the prediction entropy H is above the threshold τ, then casting the bait prototype towards uncertain set while also staying close to certain set. Bottom (b): training feature extractor push all features towards both prototypes of C_1 and C_2, thus achieving aligning target features with source classifier.

Phase 2: freeze the source trained classifier C_1, then update another classifier C_2 and f.
Convention:

- classifier C1 on target data: *anchor classifier*, and its class prototype as *anchor prototype*.
- classifier C2 on target data: *bait classifier*, and its class prototype as *bait prototype*.

Phase 2 consists of two steps:

Step 1: "casting the bait". Train C_2 only.

First, we split the features of the current batch of data into two sets: the uncertain $\mathcal{U}$ and certain set C, as shown in Fig. 2 (a), according to their prediction entropy

$$\mathcal{U} = \left\{x | x \in \mathcal{D}_t, H\left(p^{(1)}(x)\right) > \tau\right\}$$
$$\mathrm{C} = \left\{x | x \in \mathcal{D}_t, H\left(p^{(1)}(x)\right) \leqslant \tau\right\}, \tag{2}$$

where $p^{(1)}(x) = \sigma(C_1 f(x))$ is the prediction of the anchor classifier (σ represents the softmax operation) and

$H(p(x)) = -\sum_{i=1}^{K} p_i \log p_i$. The threshold τ is estimated as a percentile of the entropy of $p^{(1)}(x)$ in sample batch $\mathcal{T}$, set to 50% (i.e. the median). Then we make the **bait prototype move toward** those **higher entropy** features, but still **stay nearby** target features with **lower entropy** (Figure 2a)

$$\mathcal{L}_{cast}(C_2) = \sum_{x \in \mathcal{C}} D_{SKL}\left(p^{(1)}(x), p^{(2)}(x)\right) - \sum_{x \in \mathcal{U}} D_{SKL}\left(p^{(1)}(x), p^{(2)}(x)\right), \tag{3}$$

where D_{SKL} is the symmetric KL divergence: $D_{SKL}(a,b) = \frac{1}{2}(D_{KL}(a|b) + D_{KL}(b|a))$

Step 2: "biting the bait".
Remark: it is hard to directly drive all target features to the correct anchor prototype, we seek to make **target features cluster** around **both** anchor and bait prototypes.

$$\mathcal{L}_{bite}(f) = \sum_{i=1}^{n_t} \sum_{k=1}^{K} \left[-p_{i,k}^{(2)} \log p_{i,k}^{(1)} - p_{i,k}^{(1)} \log p_{i,k}^{(2)} \right] \tag{4}$$

By minimizing this loss, the prediction distribution of the bait classifier should be similar to that of the anchor classifier and vice verse, which means target features are excepted to get closer to both bait and anchor prototypes.
Moreover, in order to avoid the degenerate solutions [1], which allocate all uncertain features to a few anchor class prototype, we adopt the class balance loss (CB loss) $\mathcal{L}_b$ to regularize the feature extractor:

$$L_b(f) = \sum_{k=1}^{K} \left[KL\left(\overline{p}_k^{(1)}(x) || \boldsymbol{q}_k\right) + KL\left(\overline{p}_k^{(2)}(x) || \boldsymbol{q}_k\right) \right] \tag{5}$$

where $\overline{p}_k = \frac{1}{n_t} \sum_{x \in D_t} p_k(x)$ is the empirical label distribution, and $\boldsymbol{q}$ is uniform distribution $\boldsymbol{q}_k = \frac{1}{K}, \sum_{k=1}^{K} \boldsymbol{q}_k = 1$.

With the class balance loss Lb, the model is expected to have more balanced prediction.
* Note that this 2-step training happens in every mini-batch iteration during adaptation (see Algorithm 1)

Algorithm 1 Unsupervised domain adaptation with BAIT

Require: $\mathcal{D}_t$ ▷ unlabeled target data
Require: f, C_1 ▷ network trained with source data $\mathcal{D}_s$
$C_2 \leftarrow C_1$
while not done **do**
  Sample batch $\mathcal{T}$ from $\mathcal{D}_t$
  Entropy based splitting: $\mathcal{U}$ and $\mathcal{C}$ ▷ Eq. 2
  $C_2 \leftarrow \mathrm{argmin}_{C_2}\, \mathcal{L}_{\mathrm{cast}}(C_2)$ ▷ Eq. 3
  $f \leftarrow \mathrm{argmin}_f\, \mathcal{L}_{\mathrm{bite}}(f) + \mathcal{L}_b(f)$ ▷ Eq. 4& 5
end while

3. Experimental results

Method (Synthesis → Real)	**Source-free**	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
ADR [30]	×	94.2	48.5	84.0	72.9	90.1	74.2	**92.6**	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [21]	×	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [3]	×	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SWD [16]	×	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [43]	×	-	-	-	-	-	-	-	-	-	-	-	-	74.6
IA [10]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.8
DMRL [39]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [11]	×	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
SHOT [18]	√	92.6	81.1	80.1	58.5	89.7	86.1	81.5	77.8	89.5	84.9	84.3	**49.3**	79.6
SFDA [12]	√	86.9	81.7	**84.6**	63.9	93.1	91.4	86.6	71.9	84.5	58.2	74.5	42.7	76.7
*MA [17]	√	**94.8**	73.4	68.8	**74.8**	93.1	**95.4**	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
BAIT (ours)	√	93.7	**83.2**	84.5	65.0	92.9	**95.4**	88.1	80.8	**90.0**	**89.0**	84.0	45.3	**82.7**

Table 2: Accuracies (%) on VisDA-C for ResNet101-based unsupervised domain adaptation methods. **Source-free** means setting without access to source data during adaptation. Underlined results are second highest result. * means method needs to generate extra images.

Method	**Source-free**	A→D	A→W	D→W	W→D	D→A	W→A	Avg
MCD [31]	×	92.2	88.6	98.5	**100.0**	69.5	69.7	86.5
CDAN [21]	×	92.9	94.1	98.6	**100.0**	71.0	69.3	87.7
MDD [43]	×	90.4	90.4	98.7	99.9	75.0	73.7	88.0
MDD+IA [10]	×	92.1	90.3	98.7	99.8	**75.3**	74.9	88.8
BNM [5]	×	90.3	91.5	98.5	**100.0**	70.9	71.6	87.1
DMRL [39]	×	93.4	90.8	99.0	100.0	73.0	71.2	87.9
BDG [41]	×	93.6	93.6	99.0	100.0	73.2	72.0	88.5
MCC [11]	×	95.6	95.4	98.6	100.0	72.6	73.9	89.4
SRDC [35]	×	**95.8**	**95.7**	**99.2**	100.0	76.7	77.1	**90.8**
*USFDA [13]	√	-	-	-	-	-	-	85.4
SHOT [18]	√	93.1	90.9	98.8	99.9	74.5	74.8	88.7
SFDA [12]	√	92.2	91.1	98.2	99.5	71.0	71.2	87.2
*MA [17]	√	92.7	93.7	98.5	99.8	**75.3**	**77.8**	89.6
BAIT (ours)	√	92.0	94.6	98.1	**100.0**	74.6	75.2	89.1

Table 3: Accuracies (%) on Office-31 for ResNet50-based unsupervised domain adaptation methods. **Source-free** means setting without access to source data during adaptation. Underline means the second highest result. * means method needs to generate extra images.

Method	**Source-free**	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
MCD [31]	×	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [21]	×	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
MDD [43]	×	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
MDD+IA [10]	×	56.0	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
BNM [5]	×	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [41]	×	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [35]	×	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	**85.0**	71.3
SHOT [18]	√	56.9	**78.1**	81.0	67.9	**78.4**	**78.1**	67.0	54.6	81.8	73.4	58.1	84.5	**71.6**
SFDA [12]	√	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
BAIT (ours)	√	**57.4**	77.5	**82.4**	68.0	77.2	75.1	67.1	**55.5**	**81.9**	73.9	**59.5**	84.2	**71.6**

Table 4: Accuracies (%) on Office-Home for ResNet50-based unsupervised domain adaptation methods. **Source-free** means source-free setting without access to source data during adaptation. Underline means the second highest result.

Method	Avg.
Source only	46.1
Single classifier (w/ $\mathcal{L}_b$)	52.4
BAIT (w/o $\mathcal{L}_b$, w/ splitting)	64.5
BAIT (w/ $\mathcal{L}_b$, w/o splitting)	70.6
BAIT	**71.6**

Table 5: Ablation study on Office-Home dataset in the source-free setting. *Single classifier* (w-$\mathcal{L}_b$) is to adapt the source model to target domain by optimizing $\mathcal{L}_b$. Splitting is performed according to Eq. 2.

References

[0] Unsupervised Domain Adaptation without Source Data by Casting a BAIT, S. Yang et al., https://arxiv.org/abs/2010.12427

[1] Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. K. G. Dizaji et al. In Proceedings of the IEEE international conference on computer vision, pages 5736–5745, 2017.