Federated Adversarial Domain Adaption (FADA)

Xingchao Peng, Zijun Huang, Yizhe Zhu, Kate Saenko

Compiled by Hongming Shan

Overview

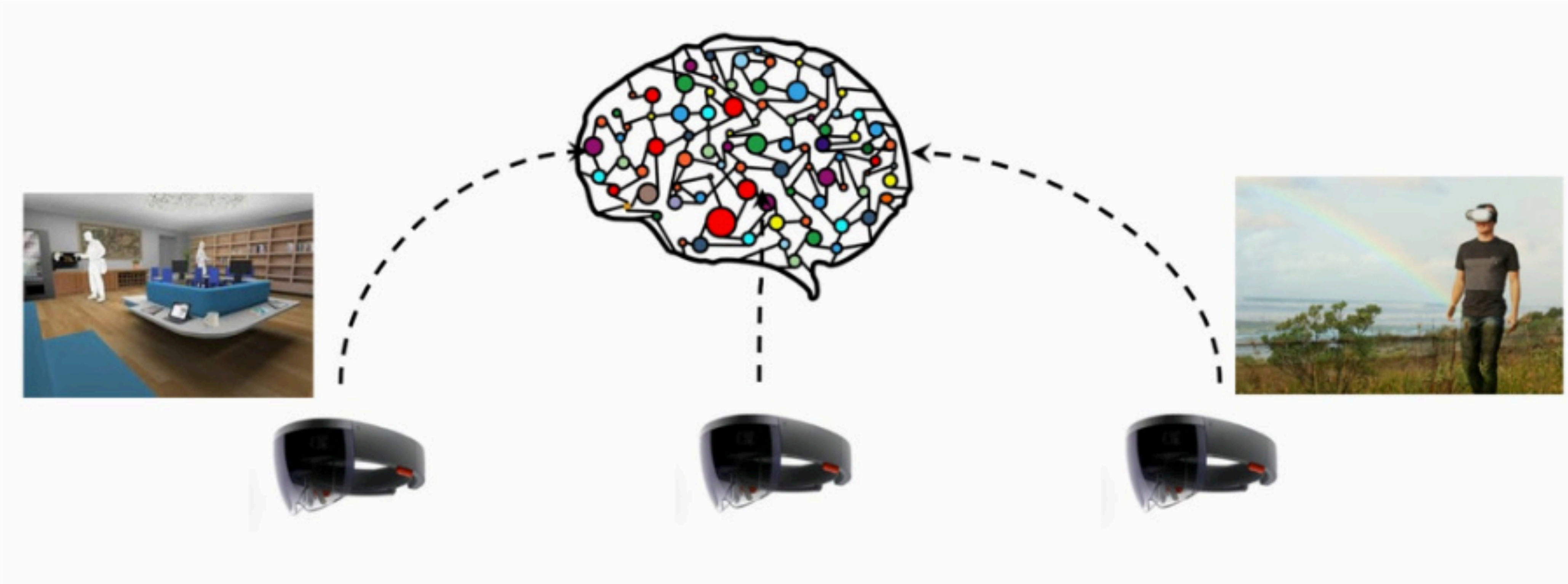
- Federated learning improves data privacy and efficiency in machine learning performed over networks of distributed devices, such as mobile phones, IoT and wearable devices, etc.
- Yet models trained with federated learning can still fail to generalize to new devices due to the problem of domain shift. Domain shift occurs when the labeled data collected by source nodes statistically differs from the target node's unlabeled data.
- In this work, we present a principled approach to the problem of federated domain adaptation, which aims to align the representations learned among the different nodes with the data distribution of the target node.
- Our approach extends adversarial adaptation techniques to the constraints of the federated setting. In addition, we devise a dynamic attention mechanism and leverage feature disentanglement to enhance knowledge transfer.

Federated Learning

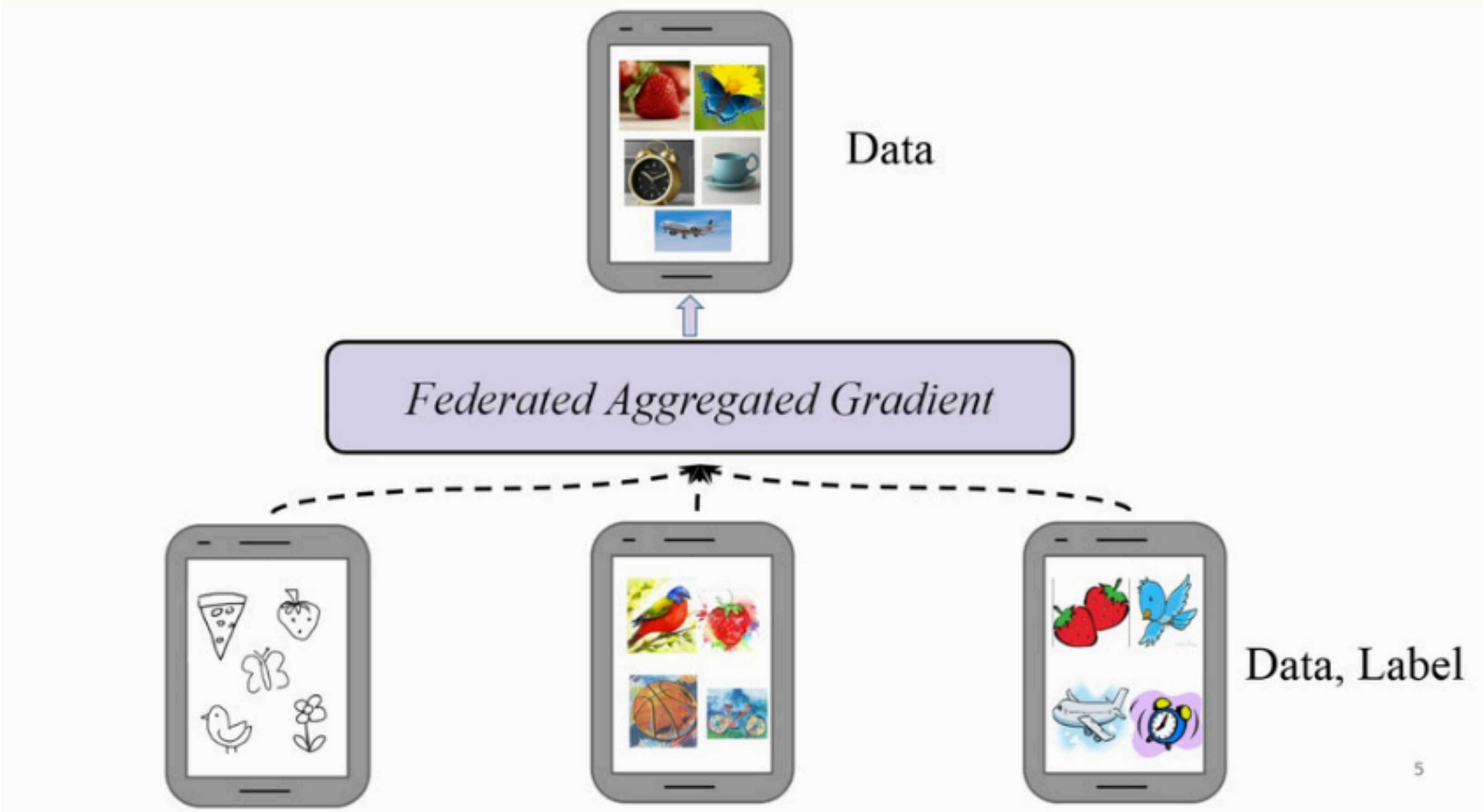
- Main idea is to have each node learn on its own local data and not share either the data or the model parameters

Domain shift

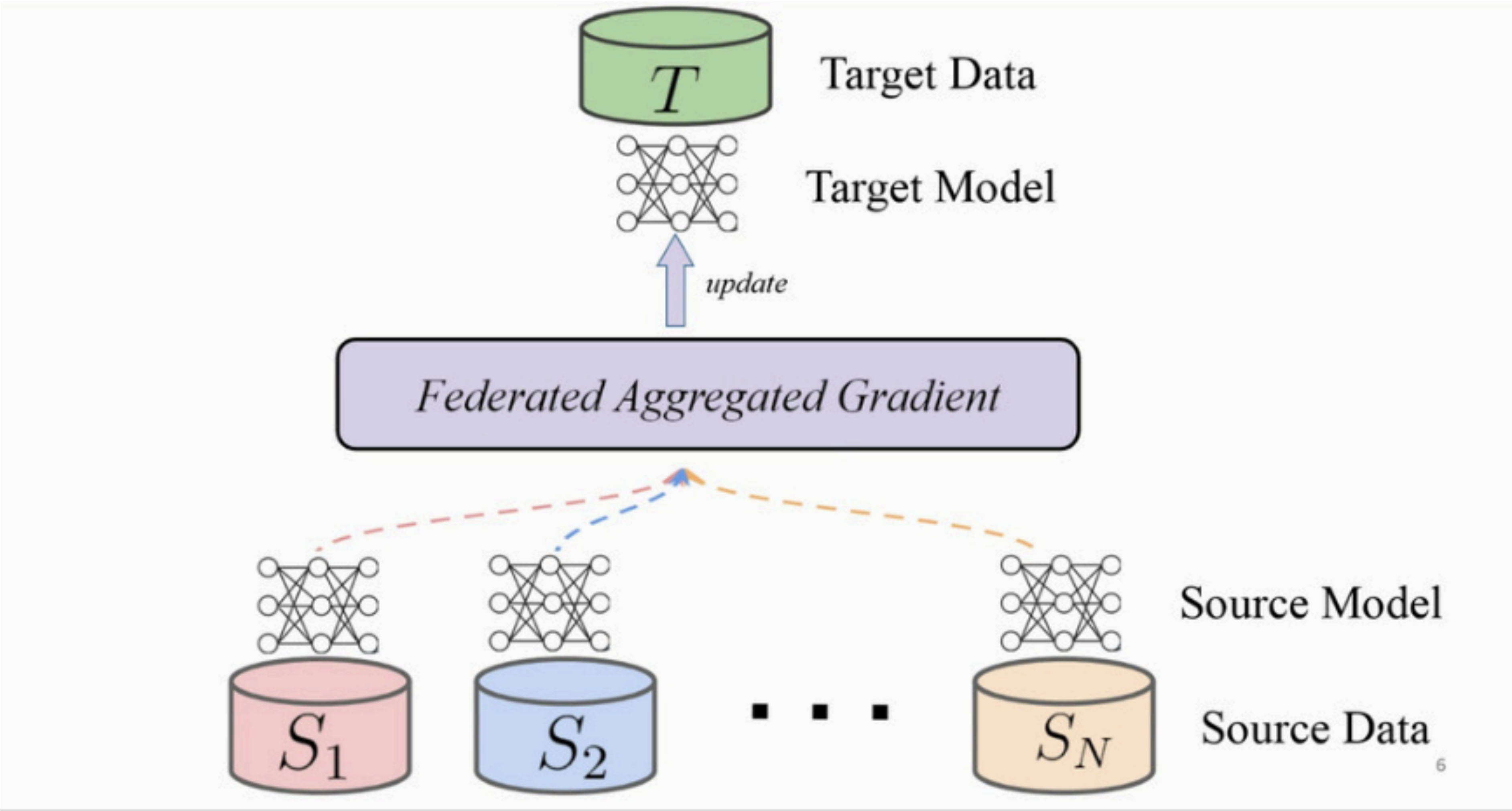
- Federated learning with non-iid data



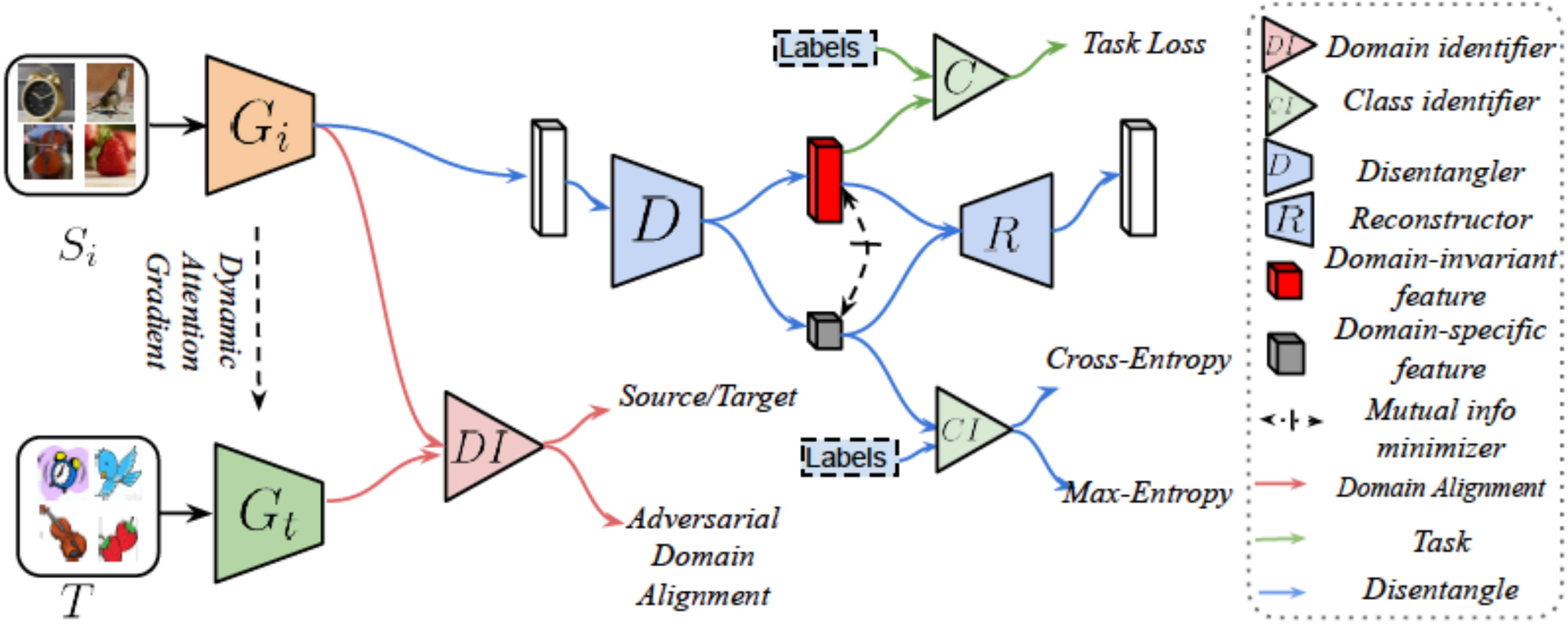
Unsupervised Federated Domain Adaption



Unsupervised Federated Domain Adaption



Federated Adversarial Domain Adaption

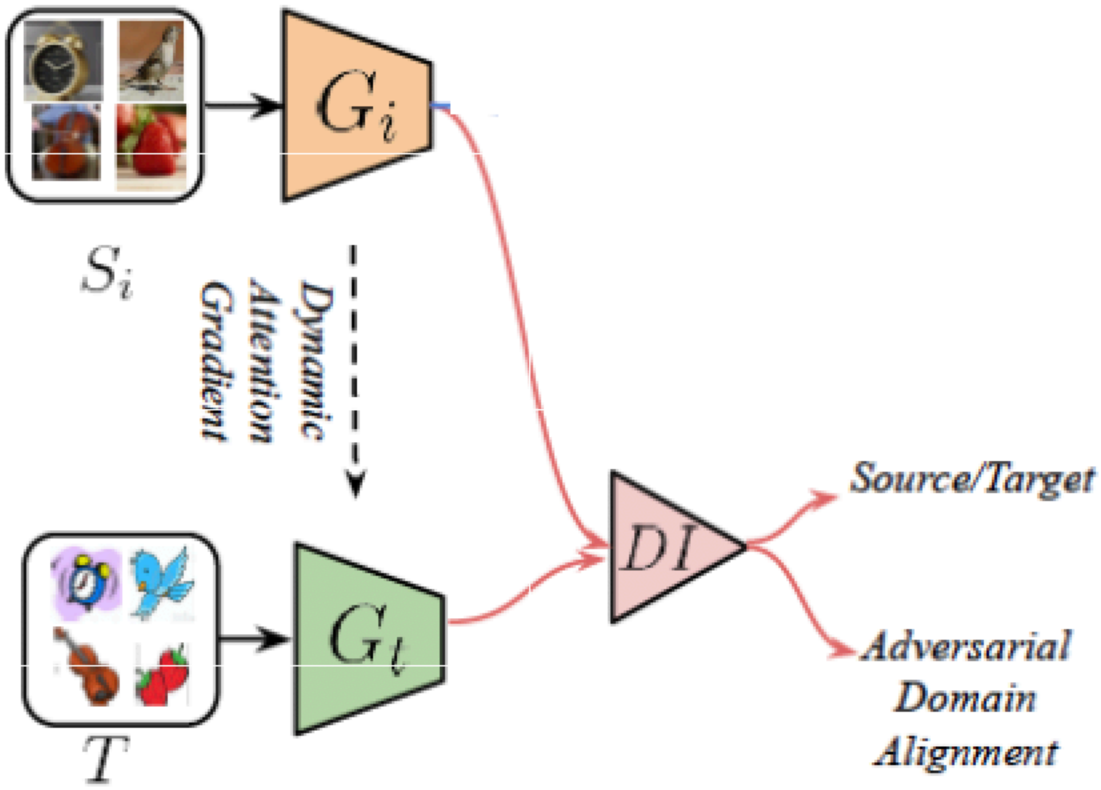


(b) Federated Adversarial Domain Adaptation

Four key steps

- Domain Alignment (Federated Adversarial Alignment)
- Domain Disentanglement (Representation Disentanglement)
- Mutual Information Minimization
- Dynamic Attention (Dynamic Weights)

1. Domain Alignment



- Divide optimization into two independent steps, a domain-specific local feature extractor and a global discriminator

1) for each domain, we train a local feature extractor, Gi for Di and Gt for Dt;
2) for each (Di , Dt) source-target domain pair, we train an adversarial domain identifier DI to align the distributions in an adversarial manner

1. Domain Alignment

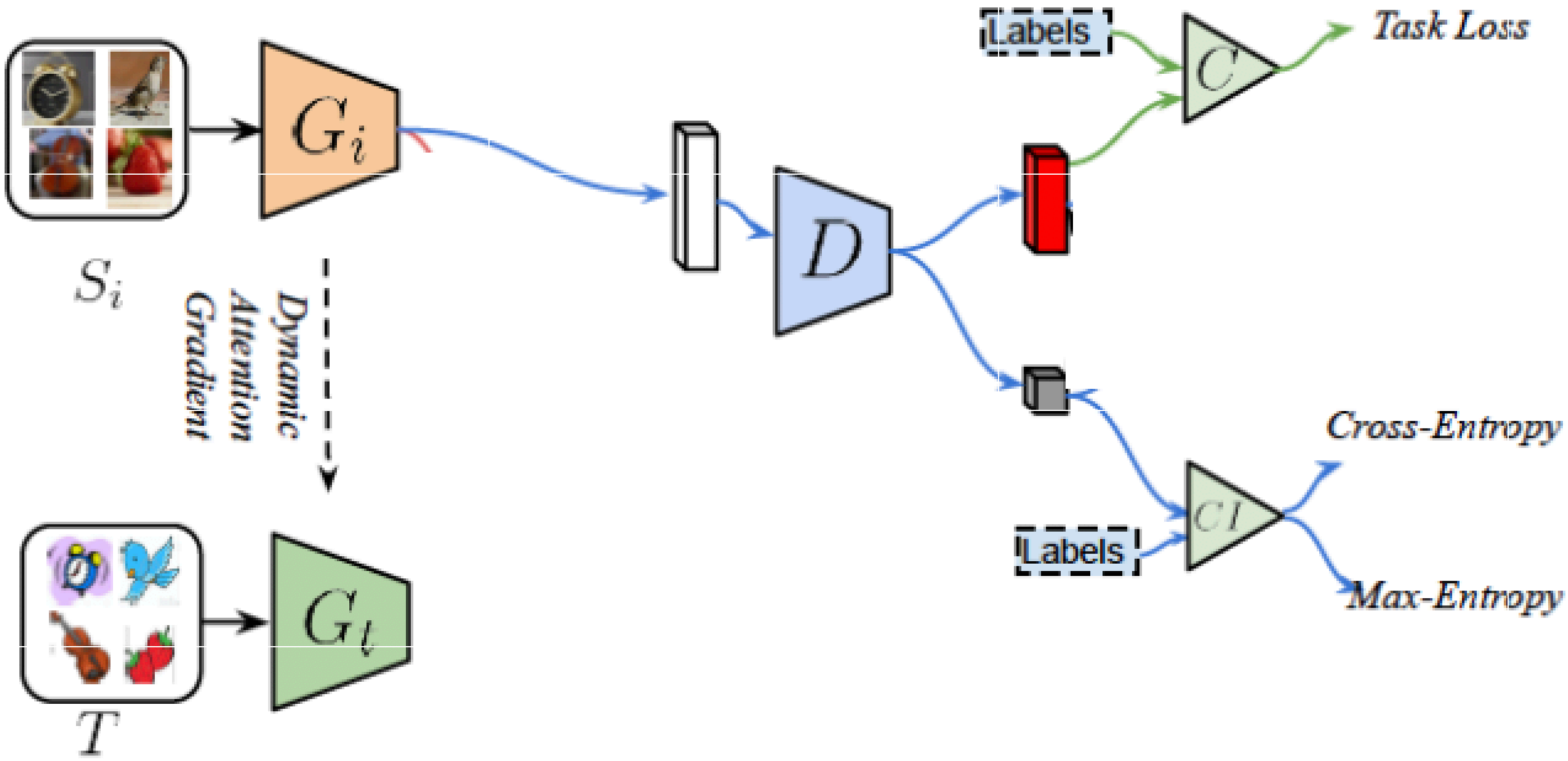
- We first train DI to identify which domain the features come from,

$$\min_{\Theta^{DI_i}} L_{adv_{DI_i}}(\mathbf{X}^{S_i}, \mathbf{X}^T, G_i, G_t) = -\mathbb{E}_{\mathbf{x}^{s_i} \sim \mathbf{X}^{s_i}} \left[\log DI_i(G_i(\mathbf{x}^{s_i}))\right] - \mathbb{E}_{\mathbf{x}^t \sim \mathbf{X}^t}[\log(1 - DI_i(G_t(\mathbf{x}^t)))]. \quad (4)$$

- Then we train the generator (Gi , Gt) to confuse the DI .

$$\min_{\Theta^{G_i}, \Theta^{G_t}} L_{adv_G}(\mathbf{X}^{S_i}, \mathbf{X}^T, DI_i) = -\mathbb{E}_{\mathbf{x}^{s_i} \sim \mathbf{X}^{s_i}}[\log DI_i(G_i(\mathbf{x}^{s_i}))] - \mathbb{E}_{\mathbf{x}^t \sim \mathbf{X}^t}[\log DI_i(G_t(\mathbf{x}^t))] \quad (5)$$

2. Domain Disentanglement



- The high-level intuition is to disentangle the features extracted by (Gi , Gt) into domain-invariant and domain-specific features

2. Domain Disentanglement

- We train the K-way classifier Ci and K-way class identifier Cii to correctly predict the labels with a cross-entropy loss, based on domain-invariant feature fdi and domain-specific feature fds.

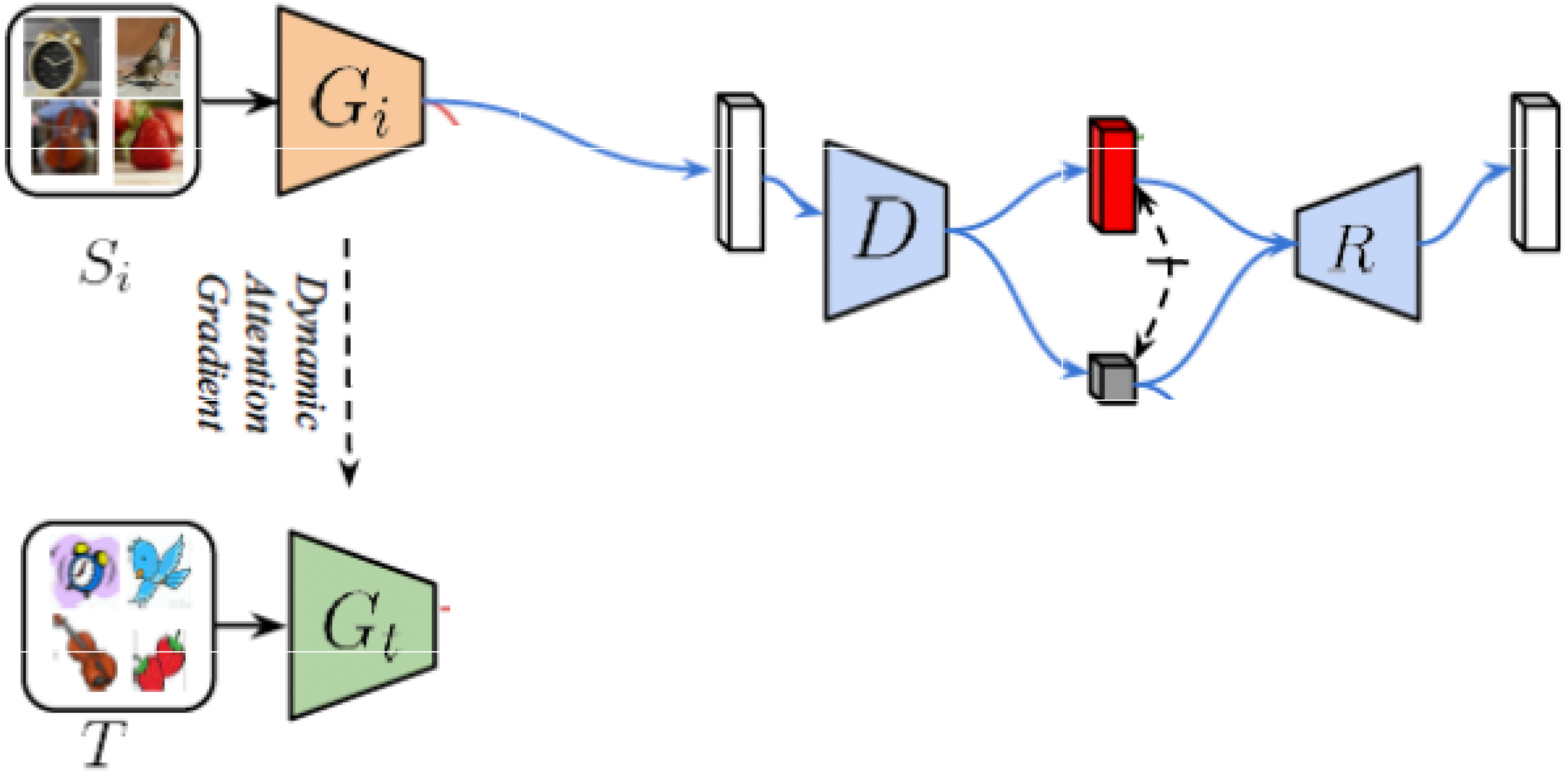
$$\underset{\Theta^{G_t},\Theta^{D_t},\Theta^{C_t},\Theta^{CI_t}}{L_{cross-entropy}} = -\mathbb{E}_{(\mathbf{x}^{s_t},\mathbf{y}^{s_t})\sim\hat{\mathcal{D}}_{s_t}} \sum_{k=1}^{K} \mathbb{1}[k=\mathbf{y}^{s_t}]log(C_i(f_{di})) - \mathbb{E}_{(\mathbf{x}^{s_t},\mathbf{y}^{s_t})\sim\hat{\mathcal{D}}_{s_t}} \sum_{k=1}^{K} \mathbb{1}[k=\mathbf{y}^{s_t}]log(CI_i(f_{ds})) \quad (6)$$

- Freeze the class identifier CIi and only train the feature disentangle to confuse the class identifier CIi by generating the domain-specific features fds (not related to class information)

$$\underset{\Theta^{D_t},\Theta^{G_t}}{L_{ent}} = -\frac{1}{N_{s_t}} \sum_{j=1}^{N_{s_t}} \log CI_i(f_{ds}^j) = -\frac{1}{N_{s_t}} \sum_{j=1}^{N_{s_t}} \log CI_i(D_i(G_i(\mathbf{x}^{s_t}))) \quad (7)$$

- Feature disentanglement facilitates the knowledge transfer by reserving fdi and dispelling fds.

3. Mutual Information Minimization



- Mutual Information Minimization
- Reconstruction Loss (L2)

3. Mutual Information Minimization

- Mutual Information Minimization through MINE(Mutual Information Neural Estimator)

$$I(\mathcal{P}, \mathcal{Q}) = \frac{1}{n}\sum_{i=1}^{n} T(p, q, \theta) - \log(\frac{1}{n}\sum_{i=1}^{n} e^{T(p, q', \theta)}) \qquad (8)$$

- Reconstruction Loss

4. Dynamic Attention

- We use the gap statistics to evaluate how well the target features ft can be clustered with unsupervised clustering algorithms (K-means). Defined as

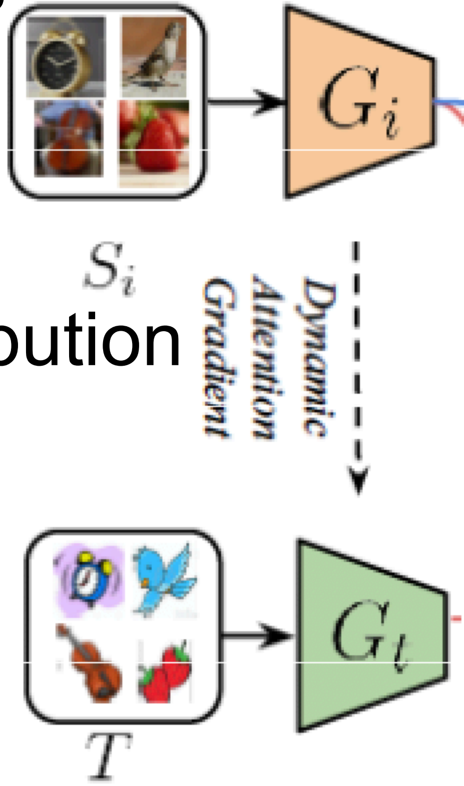
$$I = \sum_{r=1}^{k} \frac{1}{2n_r} \sum_{i,j \in C_r} ||f_i^t - f_j^t||_2$$

- A smaller gap statistics value indicates the feature distribution has smaller intra-class variance.
- Gap statistics gain between two consecutive iterations

$I_i^{gain} = I_i^{p-1} - I_i^p$ (p indicating training step),

- Attention

$$Softmax(I_1^{gain}, I_2^{gain}, \ldots, I_N^{gain}).$$



Algorithm 1 Federated Adversarial Domain Adaptation

Input: N source domains $\mathcal{D}_S=\{\mathcal{D}_{S_i}\}_{i=1}^{N}$; a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$; N feature extractors $\{\Theta^{G_1}, \Theta^{G_2}, ...\Theta^{G_N}\}$, N disentanglers $\{\Theta^{D_1}, \Theta^{D_2}, ...\Theta^{D_N}\}$, N classifiers $\{\Theta^{C_1}, \Theta^{C_2}, ...\Theta^{C_N}\}$, N class identifiers $\{\Theta^{CI_1}, \Theta^{CI_2}, ...\Theta^{CI_N}\}$, N mutual information estimators $\{\Theta^{M_1}, \Theta^{M_2}, ...\Theta^{M_N}\}$ trained on source domains. Target feature extractor Θ^{G_t}, classifier Θ^{C_t}. N domain identifiers $\{\Theta^{DI_1}, \Theta^{DI_2}, ..., \Theta^{DI_N}\}$
Output: well-trained target feature extractor $\hat{\Theta}^{G_t}$, target classifier $\hat{\Theta}^{C_t}$.
Model Initialization .

while not converged **do**
  for i **do**=1:N
    Sample mini-batch from from $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ and $\{\mathbf{x}_j^t\}_{j=1}^{n_t}$;
    Compute gradient with cross-entropy classification loss, update Θ^{G_t}, Θ^{C_t}.
    Domain Alignment:
    Update Θ^{DI_i}, $\{\Theta^{G_i}, \Theta^{G_t}\}$ with Eq. 4 and Eq. 5 respectively to align the domain distribution;
    Domain Disentangle:
    update $\Theta^{G_i}, \Theta^{D_i}, \Theta^{C_i}, \Theta^{CI_i}$ with Eq. 6
    update Θ^{D_i} and $\{\Theta^{G_i}\}$ with Eq. 7
    Mutual Information Minimization:
    Calculate mutual information between the disentangled feature pair (f_{di}, f_{ds}) with M_i;
    Update $\Theta^{D_i}, \Theta^{M_i}$ by Eq.8;
  end for
  Dynamic weight:
  Calculate dynamic weight by Eq. 3
  Update $\Theta^{G_t}, \Theta^{C_t}$ by aggregated $\{\Theta^{G_1}, \Theta^{G_2}, ..., \Theta^{G_N}\}$, $\{\Theta^{C_1}, \Theta^{C_2}, ...\Theta^{C_N}\}$ respectively with the computed dynamic weight;
end while
return $\Theta^{G_t}, \Theta^{C_t}$

Four datasets – domain adaption

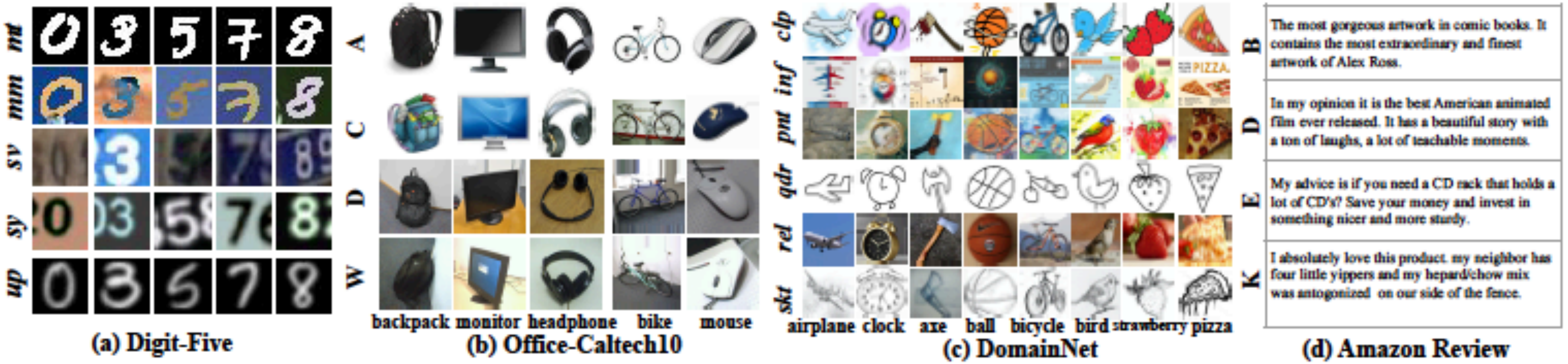


Figure 2: We demonstrate the effectiveness of FADA on four datasets: (1) "Digit-Five", which includes MNIST (*mt*), MNIST-M (*mm*), SVHN (*sv*), Synthetic (*syn*), and USPS (*up*). (2) Office-Caltech10 dataset, which contains *Amazon* (**A**), *Caltech* (**C**), *DSLR* (**D**), and *Webcam* (**W**). (3) DomainNet dataset, which includes: *clipart* (*clp*), *infograph* (*inf*), *painting* (*pnt*), *quickdraw* (*qdr*), *real* (*rel*), and *sktech* (*skt*). (4) Amazon Review dataset, which contains review for *Books* (**B**), *DVDs* (**D**), *Electronics* (**E**), and *Kitchen & housewares* (**K**).

Experiments on Digit Recognition

Models	*mt,sv,sy,up*→*mm*	*mm,sv,sy,up*→*mt*	*mt,mm,sy,up*→*sv*	*mt,mm,sv,up*→*sy*	*mt,mm,sv,sy*→*up*	Avg
Source Only	63.3±0.7	90.5±0.8	88.7±0.8	63.5±0.9	82.4±0.6	77.7
DAN	63.7±0.7	96.3±0.5	94.2±0.8	62.4±0.7	85.4±0.7	80.4
DANN	71.3±0.5	97.6±0.7	92.3±0.8	63.4±0.7	85.3±0.8	82.1
Source Only	49.6±0.8	75.4±1.3	22.7±0.9	44.3±0.7	75.5±1.4	53.5
AdaBN	59.3±0.8	75.3±0.7	34.2±0.6	59.7±0.7	87.1±0.9	61.3
AutoDIAL	60.7±1.6	76.8±0.9	32.4±0.5	58.7±1.2	90.3±0.9	65.8
f-DANN	59.5±0.6	86.1±1.1	44.3 ±0.6	53.4±0.9	89.7±0.9	66.6
f-DAN	57.5 ±0.8	86.4 ±0.7	45.3±0.7	58.4±0.7	90.8 ±1.1	67.7
FADA+*attention* (I)	44.2±0.7	90.5±0.8	27.8±0.5	55.6±0.8	88.3±1.2	61.3
FADA+*adversarial* (II)	58.2±0.8	**92.5**± 0.9	48.3±0.6	62.1±0.5	90.6±1.1	70.3
FADA+*disentangle* (III)	**62.5**±0.7	91.4 ±0.7	**50.5** ±0.3	**71.8**±0.5	**91.7**±1.0	**73.6**

Table 1: Accuracy (%) on "Digit-Five" dataset with UFDA protocol. FADA achieves **73.6%**, outperforming other baselines. We incrementally add each component t our model, aiming to study their effectiveness on the final results. (model **I**: with *dynamic attention*; model **II**: **I**+*adversarial alignment*; model **III**: **II**+*representation disentanglement*. *mt*, *up*, *sv*, *sy*, *mm* are abbreviations for *MNIST*, *USPS*, *SVHN*, *Synthetic Digits*, *MNIST-M*.)

Feature Visualization

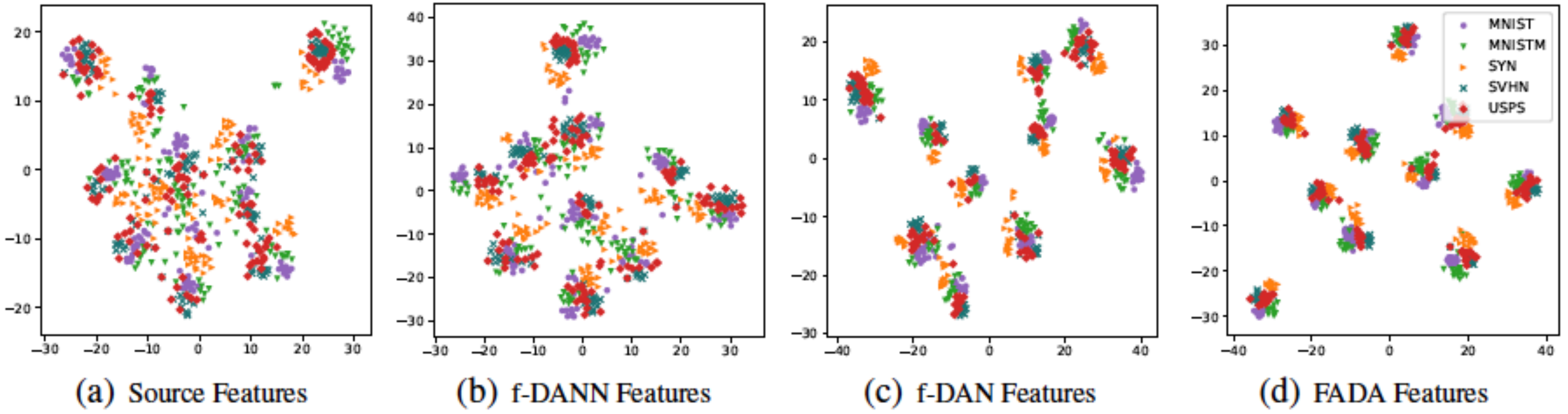


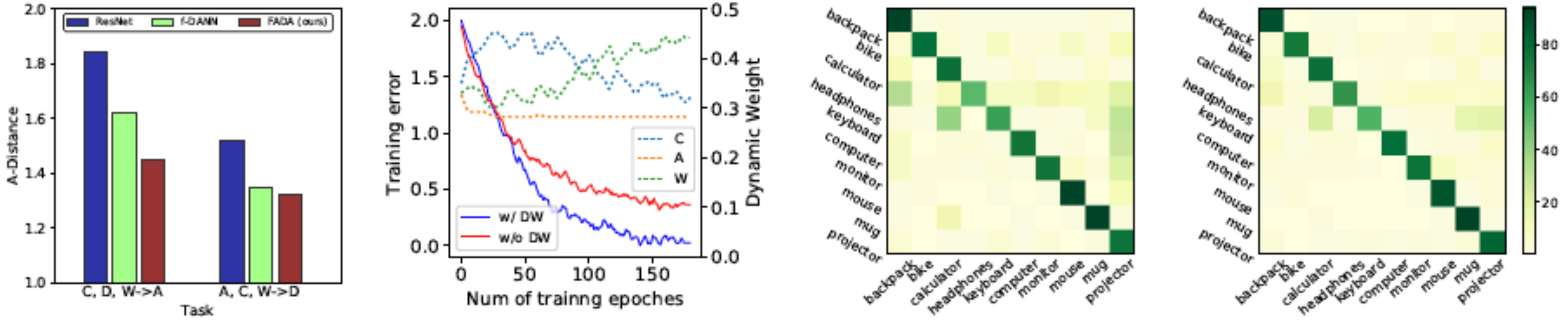
Figure 3: Feature visualization: t-SNE plot of source-only features, f-DANN (Ganin & Lempitsky, 2015) features, f-DAN (Long et al., 2015) features and FADA features in *sv,mm,mt,sy*→*up* setting. We use different markers and colors to denote different domains. The data points from target domain have been denoted by red for better visual effect. (Best viewed in color.)

Experiments on Office-Caltech10

Method	**C,D,W → A**	**A,D,W → C**	**A,C,W → D**	**A,C,D → W**	Average
AlexNet	80.1±0.4	86.9±0.3	82.7±0.5	85.1±0.3	83.7
f-DAN	82.5±0.5	87.2±0.4	85.6±0.4	86.1±0.3	85.4
f-DANN	83.1±0.4	86.5±0.5	84.8±0.5	86.4±0.5	85.2
FADA+*attention* **(I)**	81.2±0.3	87.1±0.6	83.5±0.5	85.9±0.4	84.4
FADA+*adversarial* **(II)**	83.1±0.6	87.8±0.4	85.4±0.4	86.8±0.5	85.8
FADA+*disentangle* **(III)**	**84.3**±0.6	88.4±0.5	86.1±0.4	87.3±0.5	86.5
ResNet101	81.9±0.5	87.9±0.3	85.7±0.5	86.9±0.4	85.6
AdaBN	82.2±0.4	88.2±0.6	85.9±0.7	87.4±0.8	85.7
AutoDIAL	83.3±0.6	87.7±0.8	85.6±0.7	87.1±0.6	85.9
f-DAN	82.7±0.3	88.1±0.5	86.5±0.3	86.5±0.3	85.9
f-DANN	83.5±0.4	88.5±0.3	85.9±0.5	87.1±0.4	86.3
FADA+*attention* **(I)**	82.1±0.5	87.5±0.3	85.8±0.4	87.3±0.5	85.7
FADA+*adversarial* **(II)**	83.2±0.4	88.4±0.3	86.4±0.5	87.8±0.4	86.5
FADA+*disentangle* **(III)**	84.2±0.5	**88.7**±0.5	**87.1**±0.6	**88.1**±0.4	**87.1**

Table 2: Accuracy on *Office-Caltech10* dataset with unsupervised federated domain adaptation protocol. The upper table shows the results for AlexNet backbone and the table below shows the results for ResNet backbone.

Experiments on Office-Caltech10



(a) $\mathcal{A}$-Distance (b) Training loss for **A,C,W→D** (c) f-DAN confusion matrix (d) FADA confusion matrix

Figure 4: (**a**)$\mathcal{A}$-Distance of ResNet, f-DANN, and FADA features on two different tasks. (**b**) training errors and dynamic weight on **A,C,W→D** task. (**c**)-(**d**) confusion matrices of f-DAN, and FADA on **A,C,D→W** task.

Experiments on DomainNet

Models	*inf,pnt,qdr, rel,skt→clp*	*clp,pnt,qdr, rel,skt→inf*	*clp,inf,qdr, rel,skt→pnt*	*clp,inf,pnt, rel,skt→qdr*	*clp,inf,pnt, qdr,skt→rel*	*clp,inf,pnt, qdr,rel→skt*	Avg
AlexNet	39.2±0.7	12.7±0.4	32.7±0.4	5.9±0.7	40.3±0.5	22.7±0.6	25.6
f-DAN	41.6±0.6	13.7±0.5	36.3±0.5	6.5±0.5	43.5±0.8	22.9±0.5	27.4
f-DANN	42.6±0.8	14.1±0.7	35.2±0.3	6.2±0.7	42.9±0.5	22.7±0.7	27.2
FADA+*disentangle* (III)	<u>44.9</u>±0.7	<u>15.9</u>±0.6	36.3±0.8	**8.6**±0.8	44.5±0.6	23.2±0.8	28.9
ResNet101	41.6 ±0.6	14.5±0.7	35.7±0.7	<u>8.4</u>±0.7	43.5±0.7	23.3±0.7	27.7
f-DAN	43.5±0.7	14.1±0.6	<u>37.6</u>±0.7	8.3±0.6	44.5±0.5	25.1±0.5	28.9
f-DANN	43.1±0.8	15.2±0.9	35.7±0.4	8.2±0.6	<u>45.2</u>±0.7	**27.1**±0.6	29.1
FADA+*disentangle* (III)	**45.3**±0.7	**16.3**±0.8	**38.9** ±0.7	7.9±0.4	**46.7**±0.4	<u>26.8</u>±0.4	**30.3**

Table 3: Accuracy (%) on the DomainNet dataset (Peng et al., 2018) dataset under UFDA protocol. The upper table shows the results based on AlexNet (Krizhevsky et al., 2012) backbone and the table below are the results based on ResNet (He et al., 2016) backbone.

Experiments on Amazon Review

Method	D,E,K $\rightarrow$ B	B,E,K $\rightarrow$ D	B,D,K $\rightarrow$ E	B,D,E $\rightarrow$ K	Average
Source Only	74.4±0.3	79.2±0.4	73.5 ±0.2	71.4±0.1	74.6
f-DANN	75.2±0.3	**82.7**±0.2	76.5±0.3	72.8±0.4	76.8
AdaBN	76.7±0.3	80.9±0.3	75.7±0.2	74.6±0.3	76.9
AutoDIAL	76.3±0.4	81.3±0.5	74.8±0.4	75.6±0.2	77.1
f-DAN	75.6±0.2	<u>81.6</u>±0.3	**77.9**±0.1	73.2±0.2	77.6
FADA+*attention* **(I)**	74.8±0.2	78.9±0.2	74.5±0.3	72.5±0.2	75.2
FADA+*adversarial* **(II)**	**79.7**±0.2	81.1±0.1	77.3±0.2	<u>76.4</u>±0.2	78.6
FADA+*disentangle* **(III)**	<u>78.1</u>±0.2	**82.7**±0.1	<u>77.4</u>±0.2	**77.5**±0.3	**78.9**

Table 4: Accuracy (%) on "Amazon Review" dataset with unsupervised federated domain adaptation protocol.

Ablation Study - Attention

target	*mm*	*mt*	*sv*	*sy*	*up*	Avg	A	C	D	W	Avg	B	D	E	K	Avg
FADA *w/o. attention*	60.1	91.2	49.2	69.1	90.2	**71.9**	83.3	85.7	86.2	88.3	**85.8**	77.2	82.8	77.2	76.3	**78.3**
FADA *w. attention*	62.5	91.4	50.5	71.8	91.7	**73.6**	84.2	88.7	87.1	88.1	**87.1**	78.1	82.7	77.4	77.5	**78.9**

Table 5: The ablation study results show that the dynamic attention module is essential for our model.

Conclusion

- Proposed a novel unsupervised federated domain adaption problem
- Proposed a novel model called Federated Adversarial Domain Adaption (FADA) to transfer knowledge learned from distributed source domains to an unlabeled target domain with a novel dynamic attention schema
- Experimental results show that feature disentanglement boosts the performance of FADA in UFDA tasks
- Extensive results demonstrated the efficacy of FADA against several domain adaptation baselines.