ZHEJIANG UNIVERSITY

ICML|2021
Thirty-eighth International
Conference on Machine Learning

KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation

Hao-zhe Feng[1]✉, Zhaoyang You[2], Minghao Chen[1], Tianye Zhang[1], Minfeng Zhu[1], Fei Wu[2], Chao Wu[3], Wei Chen[1*]

1 State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

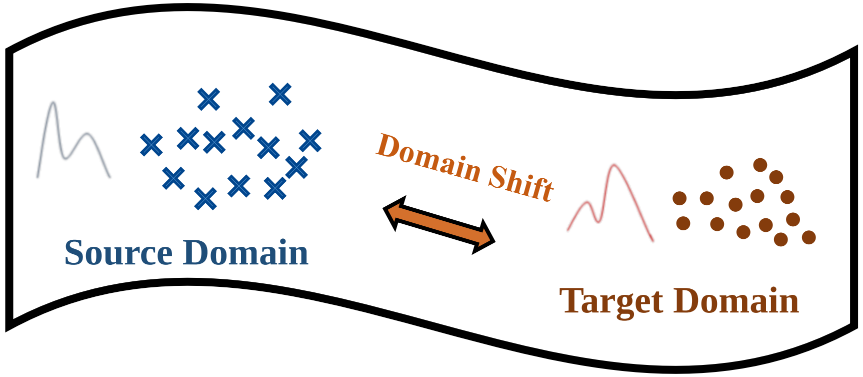
2 College of Computer Science and Technology, Zhejiang University, Hangzhou, China

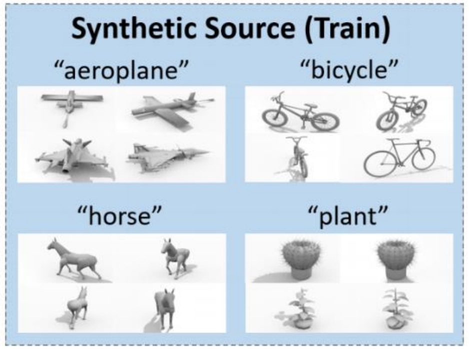
3 School of Public Affairs, Zhejiang University, Hangzhou, China

✉fenghz@zju.edu.cn

*Corresponding author

Unsupervised Multi-Source Domain Adaptation (UMDA)

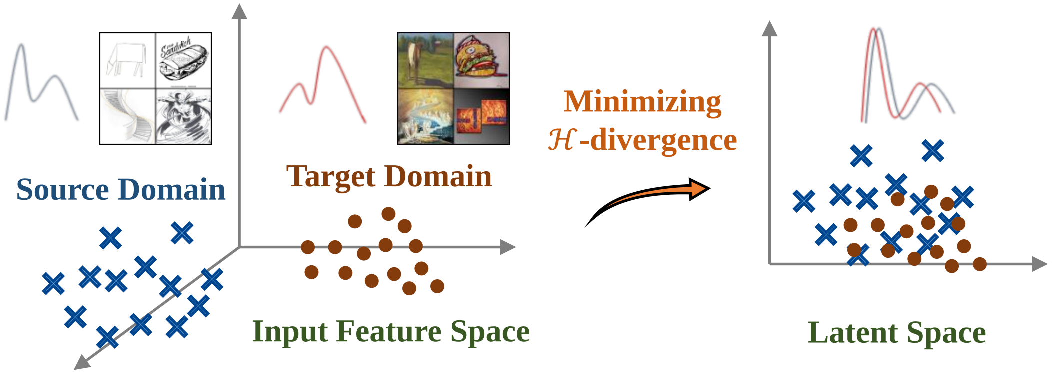




Synthesized Data: Cheap and Abundant

Real Data: Expensive and Scarce

[Peng X et al. VisDA: The Visual Domain Adaptation Challenge, 2017.]



[The prevailing unsupervised multi-source domain adaptation (UMDA) paradigm.]

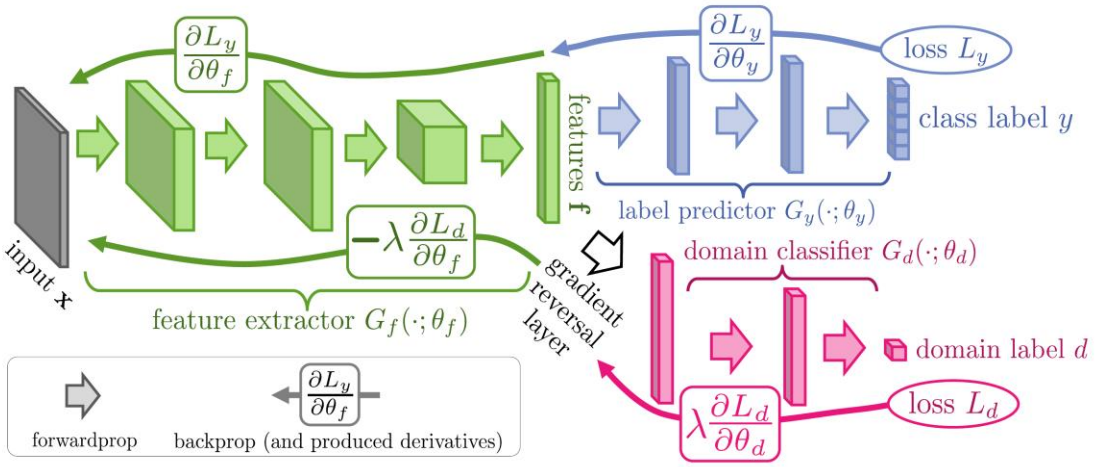
Theorem 1 *Let $\mathcal{H}$ be the model space, $\{\epsilon_{\mathbb{D}_S^k}(h)\}_{k=1}^K$ and $\epsilon_{\mathbb{D}_T}(h)$ be the task risks of source domains $\{\mathbb{D}_S^k\}_{k=1}^K$ and the target domain $\mathbb{D}_T$, and $\boldsymbol{\alpha} \in \mathcal{R}_+^K, \sum_{k=1}^K \boldsymbol{\alpha}_k = 1$ be the domain weights. Then for all $h \in \mathcal{H}$ we have:*

$$\epsilon_{\mathbb{D}_T}(h) \leq \sum_{k=1}^{K} \boldsymbol{\alpha}_k \left(\overset{\text{Task loss}}{\epsilon_{\mathbb{D}_S^k}(h)} + \overset{\mathcal{H}\text{-divergence}}{d_{\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T)} \right) + \lambda_0 \qquad (3)$$

where $\lambda_0 = \min_{h\in\mathcal{H}} \sum_{k=1}^K \alpha_k \epsilon_{\mathbb{D}_S^k}(h) + \epsilon_{\mathbb{D}_T}(h)$ is a constant according to the task risk of the optimal model on the source domains and target domain.

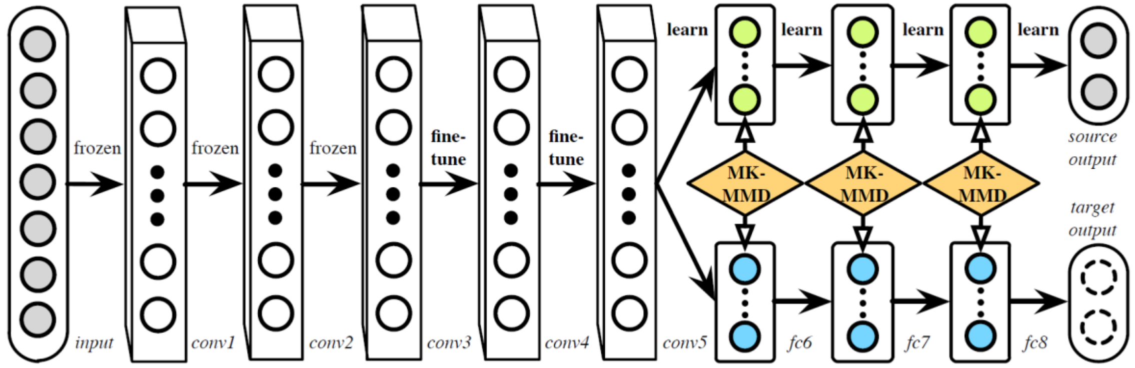
[The Generalization Bound of UMDA.]

Unsupervised Multi-Source Domain Adaptation (UMDA)



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
input $\mathbf{x}$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
feature extractor $G_f(\cdot;\theta_f)$
domain label d
loss L_d
$\lambda \frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

[Ganin Y et al. Unsupervised Domain Adaptation by Backpropagation (DANN), ICML 2015.]



[Long M et al. Learning Transferable Features with Deep Adaptation Networks, ICML 2015.]

How to optimize the $\mathcal{H}$-divergence?

- ### Adversarial Learning

Making features from different domains undistinguished

1. Build domain classifier $h \in \mathcal{H}$ with:

$$d_{\mathcal{H}}(\mathbb{D}_S, \mathbb{D}_T) = \sup_{h \in \mathcal{H}} \left| \mathbb{E}_{\mathbf{X}_i \sim \mathbb{D}_S} \mathbf{I}[h(\mathbf{X}_i) = 1] - \mathbb{E}_{\mathbf{X}_i \sim \mathbb{D}_T} \mathbf{I}[h(\mathbf{X}_i) = 1] \right|$$

2. Seek the classifier $G_d(\cdot, \boldsymbol{\theta_d})$ that minimizing the loss of domain classifier: $\lambda \frac{\partial L_d}{\partial \theta_d}$
3. Seek the feature mapping $G_f(\cdot, \boldsymbol{\theta_f})$ that maximizing the loss of domain classifier: $-\lambda \frac{\partial L_d}{\partial \theta_f}$

- ### Maximum Mean Discrepancy (MMD)

1. Build reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel κ: $h(\boldsymbol{X}) = \langle h(\cdot), \kappa(\cdot, \boldsymbol{X}) \rangle$
2. Write the equivalence $\mathcal{H}$-divergence as

$$d_{\mathcal{H}}(\mathbb{D}_S, \mathbb{D}_T) = \sup_{h \in \mathcal{H}} \langle h, \int_{\mathbf{X}} \kappa(\cdot, \mathbf{X}) d\mathbb{P}_S(\mathbf{X}) \rangle - \langle h, \int_{\mathbf{X}} \kappa(\cdot, \mathbf{X}) d\mathbb{P}_T(\mathbf{X}) \rangle$$

3. Minimize kernel MMD

$$d^{\kappa}_{\text{MMD}}(\mathbb{D}_S, \mathbb{D}_T) = -2\mathbb{E}_{\mathbf{X}_S, \mathbf{X}_T \sim \mathbb{D}_S, \mathbb{D}_T} \kappa(\mathbf{X}_S, \mathbf{X}_T) \\ + \mathbb{E}_{\mathbf{X}_S, \mathbf{X}'_S \sim \mathbb{D}_S} \kappa(\mathbf{X}_S, \mathbf{X}'_S) + \mathbb{E}_{\mathbf{X}_T, \mathbf{X}'_T \sim \mathbb{D}_T} \kappa(\mathbf{X}_T, \mathbf{X}'_T)$$

Domain Adaptation with Privacy-preserving Policy

General Data Protection Regulation (GDPR)

- **Individual Rights**

Simply combining source data is forbidden.

1. **Right to be informed and accessible.** Individuals have the right to be informed about the collection and use of their personal data, such as the purposes of data analysis, with whom the data is shared, and the data collecting methods. **If the data is obtained from other sources, the privacy information must be provided within one month**. Individuals also have the right to access and receive a copy of their personal data and other supplementary information

2. **Right to object.** Individuals have the right to object **the directly usage in market**.

3. **Right to rectification and erasure.** Individuals have the right to rectify inaccurate personal data. They also have rights to 'forget' the influence of their data, such as to be total deleted in databases and to **control the impact in data-mining models**.

4. **Right to restrict processing and automated decision making.** Individuals have the right to restrict the processing their personal data. **The automated profiling about evaluating certain things of individuals** or making automated decisions about individuals without human involvement **must be authorized**.

[Individual rights in general data protection regulation, https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/individual-rights.]

Domain Adaptation with Privacy-preserving Policy

Data Residency Laws and Federated Learning

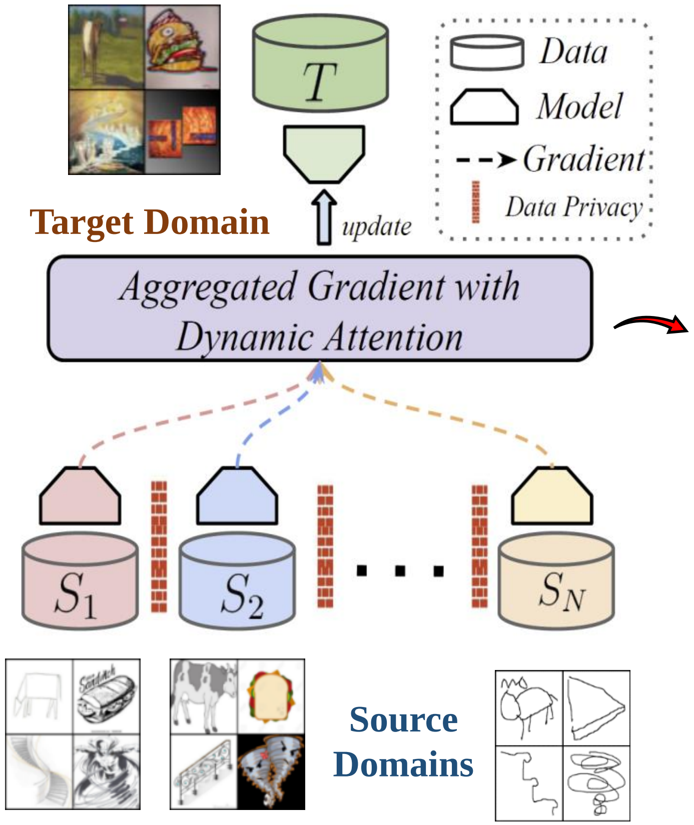
- **Corporate Responsibilities**

Data localization or residency refers to the data location's physical storage within a country's national boundaries. For European Union**,** **Companies have to keep the data inside the EU**. Data can only be conditionally transferred to countries and organizations that have signed up to equivalent privacy protection as EU.

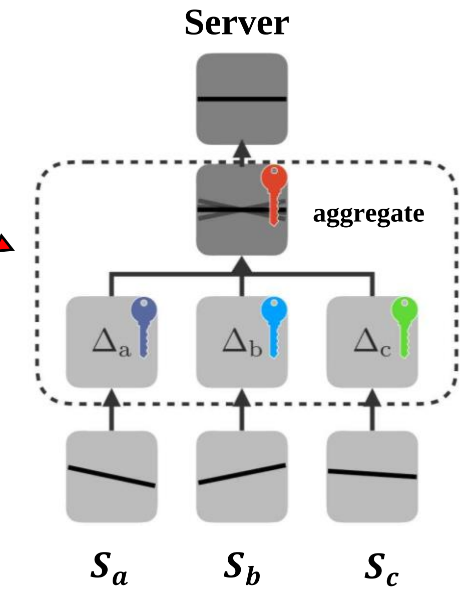
Regulated data types: **Profile; Finance; Employment; Health, Payment.**

- **Solutions: Federated Learning**

Federated learning (FL) is a machine learning setting where many clients (e.g. mobile devices or whole organizations) **collaboratively train a model** under the orchestration of a central server (e.g. service provider), **while keeping the training data decentralized**.



[Peng X, et al. Federated adversarial domain adaptation (FADA), ICLR 2020.]



[Federated Average.]

Domain Adaptation with Privacy-preserving Policy

Problem Formulation for Decentralized Scenario

1. **All the data and computations** on source domains must be kept **localized.**

2. **Available information** in each communication round**:**

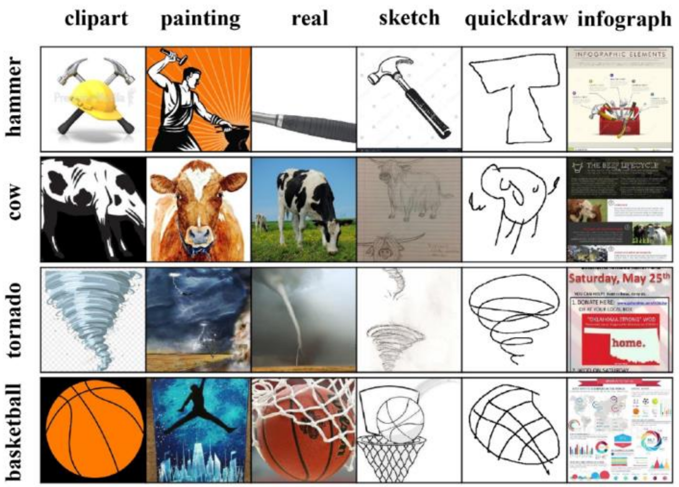
- The size of the training sets $\{N_S^k\}_{k=1}^K$on source domains.
- The parameters of K models $\{h_S^k\}_{k=1}^K$ trained on source domains.
- The target domain data containing N_T unlabeled examples $\mathbb{D}_T \coloneqq \{\boldsymbol{X}_i^T\}_{i=1}^{N_T}$.

Challenges for Decentralized UMDA

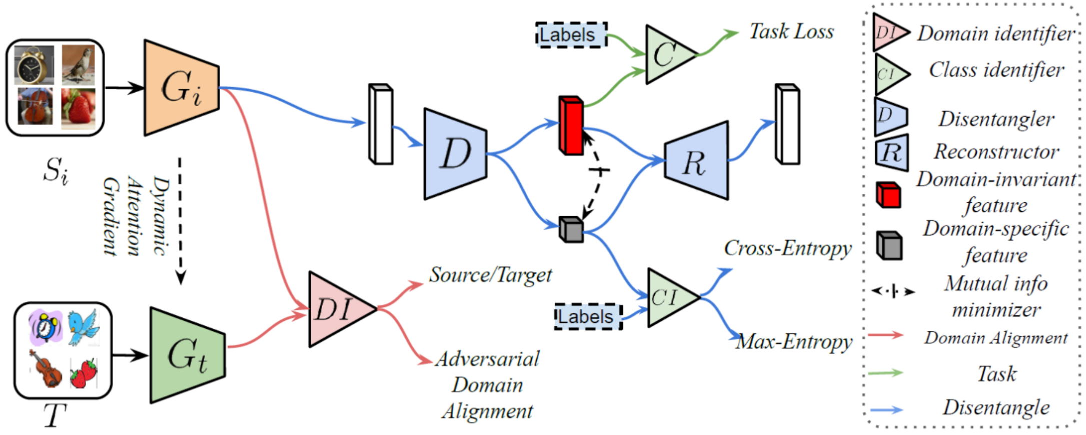
1. **Minimizing the $\mathcal{H}$-divergence requires pairwise calculation of data.**

2. **The communication cost and privacy security.**

3. **The negative transfer problem.**



[DomainNet Dataset.]



[Peng X, et al. Federated adversarial domain adaptation (FADA), ICLR 2020.]

Domain Adaptation with Privacy-preserving Policy

Problem Formulation for Decentralized Scenario

1. **All the data and computations** on source domains must be kept **localized.**

2. **Available information** in each communication round:

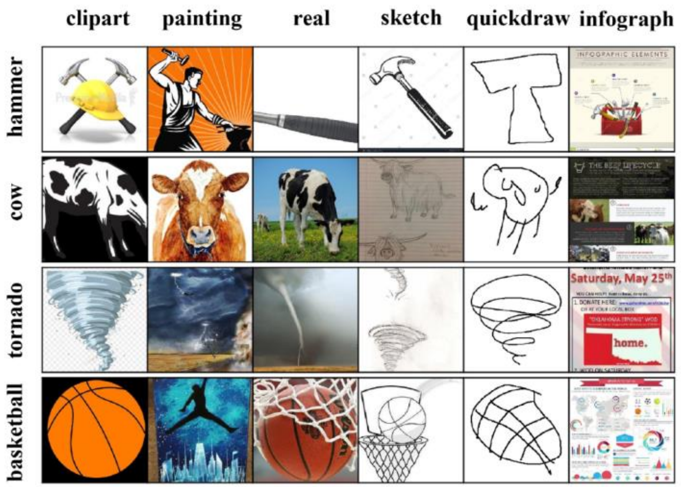
- The size of the training sets $\{N_S^k\}_{k=1}^{K}$ on source domains.
- The parameters of K models $\{h_S^k\}_{k=1}^{K}$ trained on source domains.
- The target domain data containing N_T unlabeled examples $\mathbb{D}_T \coloneqq \{\boldsymbol{X}_i^T\}_{i=1}^{N_T}$.

Challenges for Decentralized UMDA

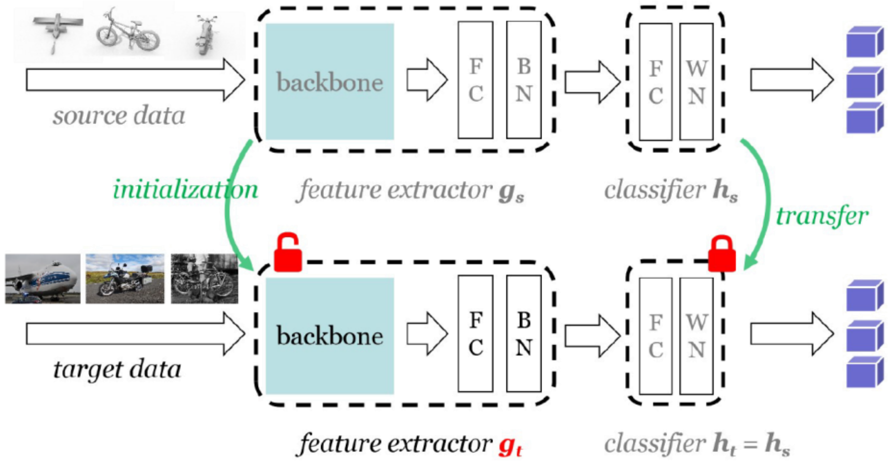
1. **Minimizing the $\mathcal{H}$-divergence requires pairwise calculation of data.**

2. **The communication cost and privacy security.**

3. **The negative transfer problem.**

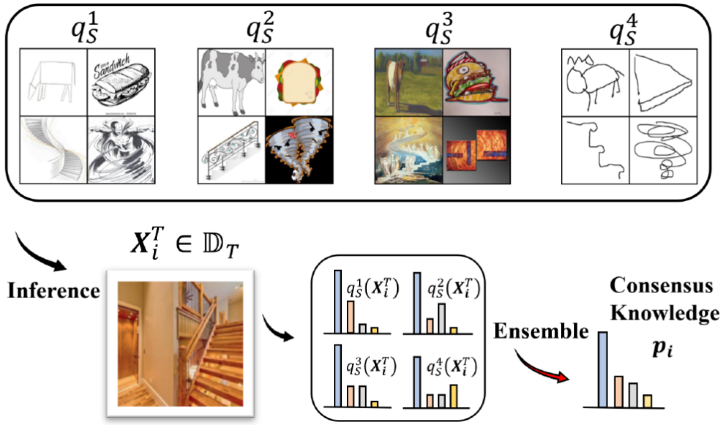


[DomainNet Dataset.]

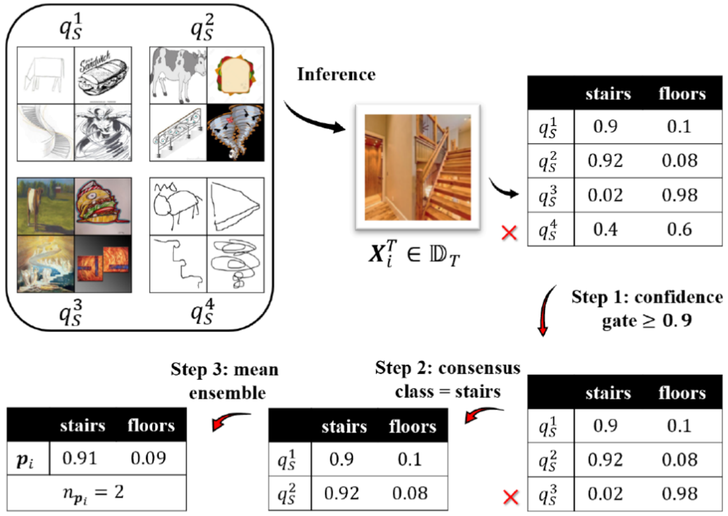


[Liang J, et al. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation (SHOT), ICML 2020.]

Our Decentralized UMDA Paradigm: KD3A



(a) Knowledge distillation process in UMDA.



	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98
q_S^4	0.4	0.6

Step 1: confidence gate ≥ 0.9

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98

Step 2: consensus class = stairs

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08

Step 3: mean ensemble

	stairs	floors
p_i	0.91	0.09
$n_{p_i} = 2$		

(b) Knowledge vote ensemble.

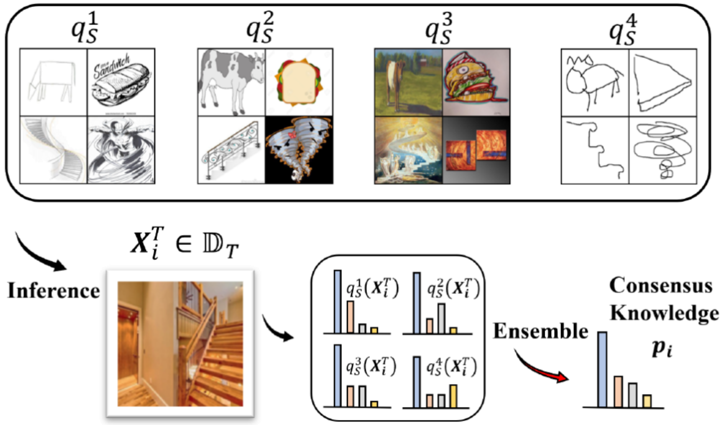
KD3A: Using Three Components in Tandem to Solve the Decentralized UMDA Challenges

1. **Knowledge vote: producing high-quality domain consensus on $\mathbb{D}_T$**
2. **Consensus focus: against negative transfer**
3. **BatchNorm MMD: decentralized optimization of $\mathcal{H}$-divergence**

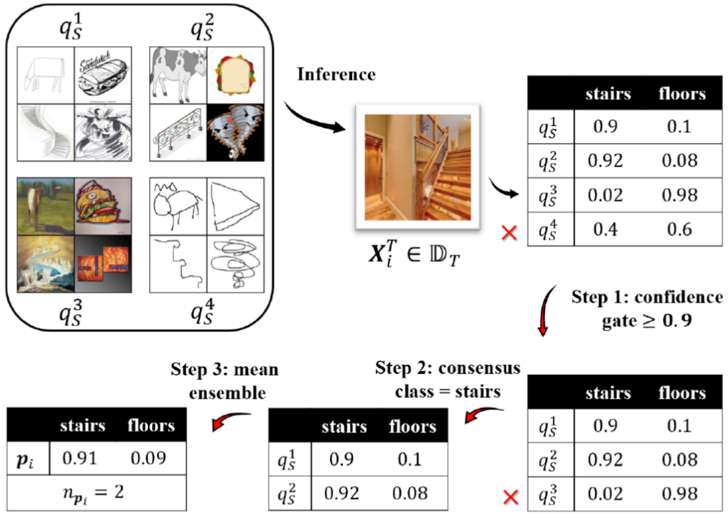
Knowledge Distillation: Extending Source Domains with Consensus Knowledge

1. Get an extended source domain $\mathbb{D}_S^{K+1}$ as $\mathbb{D}_S^{K+1} = \{(X_i^T, p_i)\}_{i=1}^{N_T}$
2. Train the extended source model h_S^{K+1} through knowledge distillation loss as $L^{\mathrm{kd}}(\mathbf{X}_i^T, q_S^{K+1}) = D_{\mathrm{KL}}(\mathbf{p}_i \| q_S^{K+1}(\mathbf{X}_i^T))$.
3. Target model is the aggregation of source models as $h_T := \sum_{k=1}^{K+1} \alpha_k \, h_S^k$
4. The related task risk for $\mathbb{D}_S^{K+1}$: $\epsilon_{\mathbb{D}_S^{K+1}}(h) = \Pr_{(\mathbf{X},\mathbf{p})\sim\mathbb{D}_S^{K+1}} [h(\mathbf{X}) \neq \arg_c \max \mathbf{p}_c]$.

Our Decentralized UMDA Paradigm: KD3A



(a) Knowledge distillation process in UMDA.



	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98
q_S^4	0.4	0.6

Step 1: confidence gate ≥ 0.9

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98

Step 2: consensus class = stairs

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08

Step 3: mean ensemble

	stairs	floors
p_i	0.91	0.09
$n_{p_i} = 2$		

(b) Knowledge vote ensemble.

KD3A: Using Three Components in Tandem to Solve the Decentralized UMDA Challenges

1. Knowledge vote: producing high-quality domain consensus on $\mathbb{D}_T$
2. Consensus focus: against negative transfer
3. BatchNorm MMD: decentralized optimization of $\mathcal{H}$-divergence

Knowledge Distillation: Extending Source Domains with Consensus Knowledge

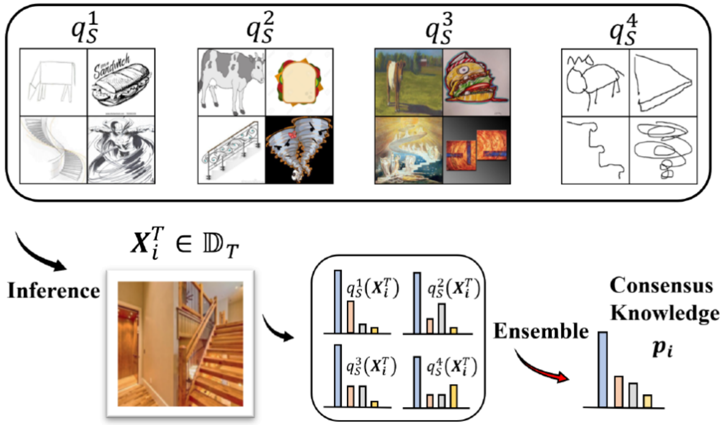
Proposition 1 *(The generalization bound for knowledge distillation). Let $\mathcal{H}$ be the model space and $\epsilon_{\mathbb{D}_S^{K+1}}(h)$ be the task risk of the new source domain $\mathbb{D}_S^{K+1}$ based on knowledge distillation. Then for all $h_T \in \mathcal{H}$, we have:*

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^{K+1}}(h_T) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) + \min\{\lambda_1, \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)|\} \quad (4)$$

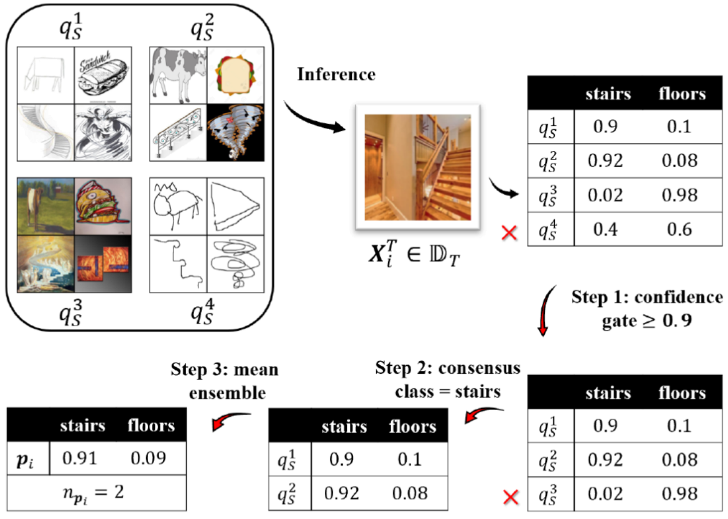
Notice: the new domain $\mathbb{D}_S^{K+1}$ can improve the original bound if the consensus knowledge is good enough to represent the ground-truth label, that is:

$$\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_T}(h) - \epsilon_{\mathbb{D}_S^{K+1}}(h)| \leq \lambda_1.$$

Our Decentralized UMDA Paradigm: KD3A



(a) Knowledge distillation process in UMDA.



	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98
q_S^4	0.4	0.6

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08
q_S^3	0.02	0.98

	stairs	floors
q_S^1	0.9	0.1
q_S^2	0.92	0.08

	stairs	floors
p_i	0.91	0.09
$n_{p_i} = 2$		

(b) Knowledge vote ensemble.

Knowledge Vote: Producing Good Consensus

Main motivation: if a certain consensus knowledge is supported by more source domains with high confidence (e.g., > 0.9), it will be more likely to be the true label.

Three steps:

1. **Confidence gate.** For each $\boldsymbol{X}_i^T \in \mathbb{D}_T$, we use a high-level confidence gate to filter the predictions $\{q_S^k(\boldsymbol{X}_i^T)\}_{k=1}^K$ of teacher models and eliminate the unconfident models.
2. **Consensus class vote.** For the remained models, the predictions are added up to find the consensus class with the maximum accumulated confidence. Then the inconsistent models are dropped.
3. **Mean ensemble** is conducted after the class vote to get the consensus knowledge $\boldsymbol{p}_i$. The number of domains that support $\boldsymbol{p}_i$ is also recorded as $n_{\boldsymbol{p}_i}$. For those $\boldsymbol{X}^T$with all teacher models eliminated by confidence gate, the naïve mean ensemble are conducted on all teachers to get $\boldsymbol{p}$ and a low weight is assigned to $\boldsymbol{X}^T$ as $n_{\boldsymbol{p}} = 0.001$.

Knowledge vote builds $\mathbb{D}_S^{K+1} = \{(\boldsymbol{X}_i^T, \boldsymbol{p}_i, n_{\boldsymbol{p}_i})\}_{i=1}^{N_T}$ with the **loss objective:**

$$L^{\mathrm{kv}}(\mathbf{X}_i^T, q) = n_{\mathbf{p}_i} \cdot D_{\mathrm{KL}}(\mathbf{p}_i \| q(\mathbf{X}_i^T))$$

Our Decentralized UMDA Paradigm: KD3A

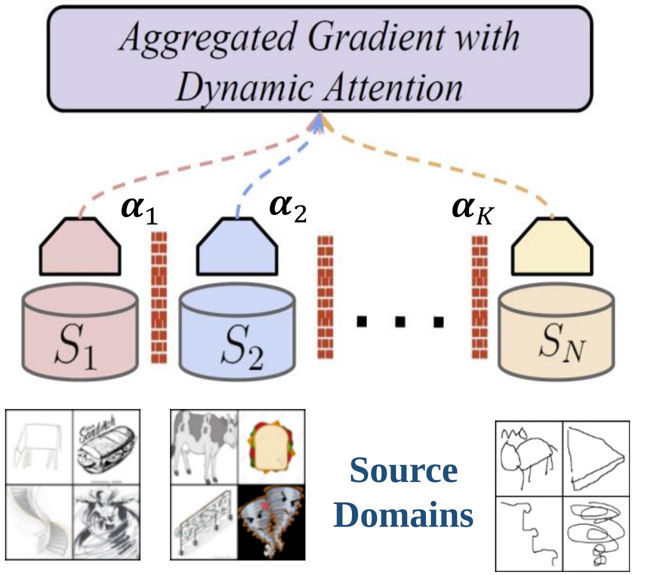
Consensus Focus: Against Negative Transfer

Domain weight $\boldsymbol{\alpha}$ determine the contribution of each source domain. To get better UMDA performance, we should assign the low weights to bad source domains.

- **Previous methods:** utilizing $\mathcal{H}$-divergence to re-weight source domains and identify irrelevant sources.

$$\boldsymbol{\alpha}_k = N_k e^{-d_{\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T)} / \sum_k N_k e^{-d_{\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T)}.$$

Notice: requires pairwise calculation.
Can not identify malicious domain!



[Aggregate Source Models with Domain Weight.]

- **Our motivation:** measure the quality of consensus from knowledge vote, assign high weights to those domains which provide high-quality consensus and penalize those domains which provide bad consensus.

1. **Consensus quality**: if one consensus class is supported by more source domains with higher confidence, then it will be more likely to represent the true label, which leads to higher consensus quality.

$$\mathcal{S} = \{\mathbb{D}_S^k\}_{k=1}^K, \forall\, \mathcal{S}' \subset \mathcal{S}, \quad Q(\mathcal{S}') = \sum_{\mathbf{X}_i^T \in \mathbb{D}_T} n_{\mathbf{p}_i}(\mathcal{S}') \cdot \max \mathbf{p}_i(\mathcal{S}')$$

Total consensus quality. Marginal contribution for $\mathbb{D}_S^k$.

2. **Domain quality:** describing the marginal contribution to the total consensus quality of each source domain as $\mathrm{CF}(\mathbb{D}_S^k) = Q(\mathcal{S}) - Q(\mathcal{S} \setminus \{\mathbb{D}_S^k\})$

3. **Re-weighting strategy from consensus focus:** $\boldsymbol{\alpha}_{K+1} = \frac{N_T}{\sum_{k=1}^K N_k + N_T}$, $\boldsymbol{\alpha}_k^{\mathrm{CF}} = (1 - \boldsymbol{\alpha}_{K+1}) \cdot \frac{N_k \cdot \mathrm{CF}(\mathbb{D}_S^k)}{\sum_{k=1}^K N_k \cdot \mathrm{CF}(\mathbb{D}_S^k)}$

Does not need to access original data.
Can identify malicious domain!

Our Decentralized UMDA Paradigm: KD3A

BatchNorm MMD: Decentralized Optimization of $\mathcal{H}$-divergence

Minimizing the $\boldsymbol{\mathcal{H}}$-divergence can optimize the UMDA upper bound.

- **Previous methods:** optimizing $\mathcal{H}$-divergence with the kernel-based MMD.

$$\min_{h\in\mathcal{H}} \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k d_{\mathrm{MMD}}^{\kappa}(\mathbb{D}_S^k, \mathbb{D}_T)$$

Need to access original data.

- **Our motivation:** utilizes the mean and variance parameters in each BatchNorm layer to optimize the $\mathcal{H}$-divergence without accessing data.

Theorem 1 *Let $\mathcal{H}$ be the model space, $\{\epsilon_{\mathbb{D}_S^k}(h)\}_{k=1}^K$ and $\epsilon_{\mathbb{D}_T}(h)$ be the task risks of source domains $\{\mathbb{D}_S^k\}_{k=1}^K$ and the target domain $\mathbb{D}_T$, and $\boldsymbol{\alpha} \in \mathcal{R}_+^K, \sum_{k=1}^K \boldsymbol{\alpha}_k = 1$ be the domain weights. Then for all $h \in \mathcal{H}$ we have:*

Task loss $\mathcal{H}$-divergence

$$\epsilon_{\mathbb{D}_T}(h) \leq \sum_{k=1}^{K} \boldsymbol{\alpha}_k \left(\epsilon_{\mathbb{D}_S^k}(h) + d_{\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T)\right) + \lambda_0 \qquad (3)$$

[The Generalization Bound of UMDA.]

1. **Build Kernel-MMD** with batch-normalized feature π_l and quartic kernel $\kappa\left(X^S, X^T\right) = \left(\langle \pi_l^S, \pi_l^T\rangle + \frac{1}{2}\right)^2$:

$$d_{\mathrm{MMD}}(\mathbb{D}_S^k, \mathbb{D}_T) = \|\mathbb{E}(\boldsymbol{\pi}_l^k) - \mathbb{E}(\boldsymbol{\pi}_l^T)\|_2^2 + \|\mathbb{E}[\boldsymbol{\pi}_l^k]^2 - \mathbb{E}[\boldsymbol{\pi}_l^T]^2\|_2^2$$

2. **Obtaining** $\{\mathbb{E}(\boldsymbol{\pi}_l^k), \mathrm{Var}(\boldsymbol{\pi}_l^k)\}_{l=1}^L$ from L BatchNorm layers of the $K+1$ source domain models : $\mathbb{E}[\boldsymbol{\pi}]^2 = Var(\boldsymbol{\pi}) + [\mathbb{E}(\boldsymbol{\pi})]^2$

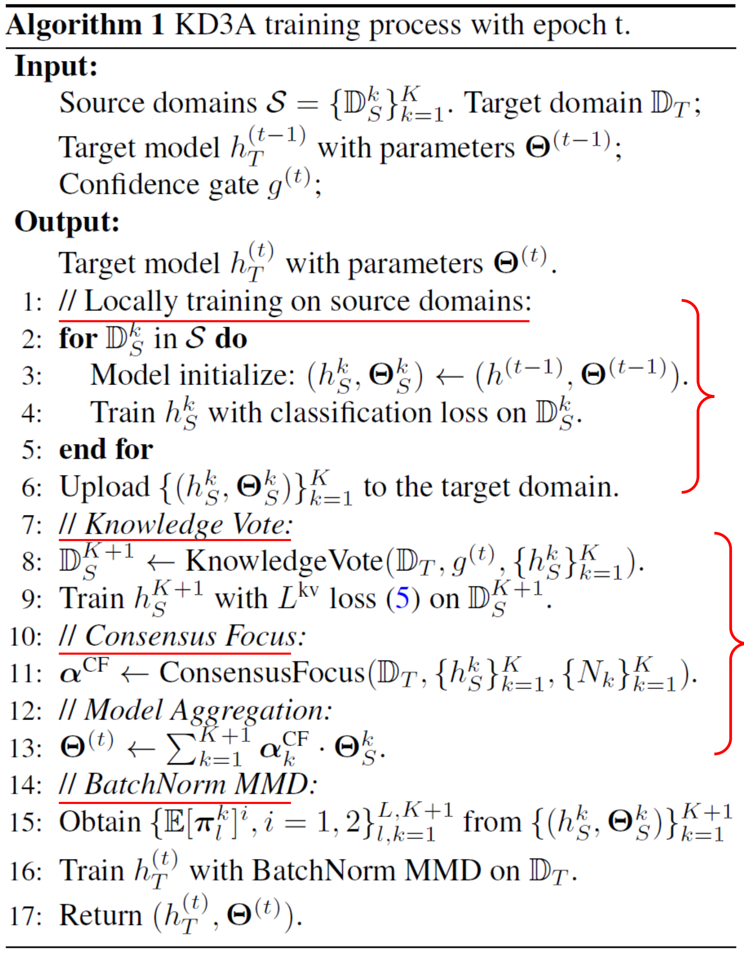
3. **Training** target model h_T:

$$\sum_{l=1}^{L} \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k (\|\mu(\boldsymbol{\pi}_l^T) - \mathbb{E}(\boldsymbol{\pi}_l^k)\|_2^2 + \|\mu[\boldsymbol{\pi}_l^T]^2 - \mathbb{E}[\boldsymbol{\pi}_l^k]^2\|_2^2)$$

Tips: directly optimizing this loss requires to traverse all BatchNorm layers, which is time-consuming. Instead, we use a EM-liked method. First, getting the global optimal solution as $\mu_{op}\left(\pi_l^T\right) = \sum_{k=1}^{K+1} \alpha_k\, \mathbb{E}\left(\pi_l^k\right)$, $\mu_{op}\left[\pi_l^T\right]^2 = \sum_{k=1}^{K+1} \alpha_k\, \mathbb{E}\left[\pi_l^k\right]^2$. Then, directly substituting the solution into the target model. This heuristic method works well in practice.

Our Decentralized UMDA Paradigm: KD3A

Algorithm of KD3A



Algorithm 1 KD3A training process with epoch t.

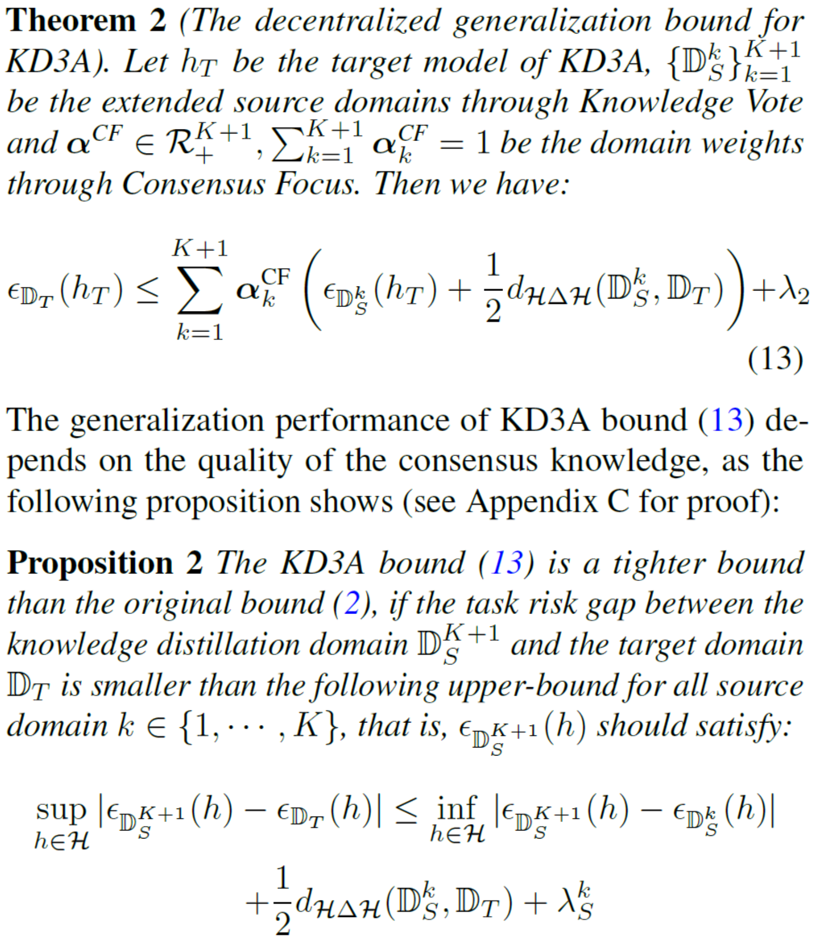
```
Input:
    Source domains $\mathcal{S} = \{\mathbb{D}_S^k\}_{k=1}^K$. Target domain $\mathbb{D}_T$;
    Target model $h_T^{(t-1)}$ with parameters $\boldsymbol{\Theta}^{(t-1)}$;
    Confidence gate $g^{(t)}$;
Output:
    Target model $h_T^{(t)}$ with parameters $\boldsymbol{\Theta}^{(t)}$.
// Locally training on source domains:
for $\mathbb{D}_S^k$ in $\mathcal{S}$ do
    Model initialize: $(h_S^k, \boldsymbol{\Theta}_S^k) \leftarrow (h^{(t-1)}, \boldsymbol{\Theta}^{(t-1)})$.
    Train $h_S^k$ with classification loss on $\mathbb{D}_S^k$.
end for
Upload $\{(h_S^k, \boldsymbol{\Theta}_S^k)\}_{k=1}^K$ to the target domain.
// Knowledge Vote:
$\mathbb{D}_S^{K+1} \leftarrow \text{KnowledgeVote}(\mathbb{D}_T, g^{(t)}, \{h_S^k\}_{k=1}^K)$.
Train $h_S^{K+1}$ with $L^{\text{kv}}$ loss (5) on $\mathbb{D}_S^{K+1}$.
// Consensus Focus:
$\boldsymbol{\alpha}^{\text{CF}} \leftarrow \text{ConsensusFocus}(\mathbb{D}_T, \{h_S^k\}_{k=1}^K, \{N_k\}_{k=1}^K)$.
// Model Aggregation:
$\boldsymbol{\Theta}^{(t)} \leftarrow \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k^{\text{CF}} \cdot \boldsymbol{\Theta}_S^k$.
// BatchNorm MMD:
Obtain $\{\mathbb{E}[\boldsymbol{\pi}_l^k]^i, i = 1, 2\}_{l,k=1}^{L,K+1}$ from $\{(h_S^k, \boldsymbol{\Theta}_S^k)\}_{k=1}^{K+1}$
Train $h_T^{(t)}$ with BatchNorm MMD on $\mathbb{D}_T$.
Return $(h_T^{(t)}, \boldsymbol{\Theta}^{(t)})$.
```

Step 1

Step 2

Step 3

Generalization Bound for KD3A



Theorem 2 *(The decentralized generalization bound for KD3A). Let h_T be the target model of KD3A, $\{\mathbb{D}_S^k\}_{k=1}^{K+1}$ be the extended source domains through Knowledge Vote and $\boldsymbol{\alpha}^{CF} \in \mathcal{R}_+^{K+1}, \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k^{CF} = 1$ be the domain weights through Consensus Focus. Then we have:*

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k^{\text{CF}} \left(\epsilon_{\mathbb{D}_S^k}(h_T) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) \right) + \lambda_2 \tag{13}$$

The generalization performance of KD3A bound (13) depends on the quality of the consensus knowledge, as the following proposition shows (see Appendix C for proof):

Proposition 2 *The KD3A bound (13) is a tighter bound than the original bound (2), if the task risk gap between the knowledge distillation domain $\mathbb{D}_S^{K+1}$ and the target domain $\mathbb{D}_T$ is smaller than the following upper-bound for all source domain $k \in \{1, \cdots, K\}$, that is, $\epsilon_{\mathbb{D}_S^{K+1}}(h)$ should satisfy:*

$$\sup_{h \in \mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)| \leq \inf_{h \in \mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_S^k}(h)| + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k$$

For good domains:

KD3A provides better consensus knowledge with *Knowledge Vote*, making $\epsilon_{\mathbb{D}_S^{K+1}}$ closer to $\epsilon_{\mathbb{D}_T}$

For bad domains:

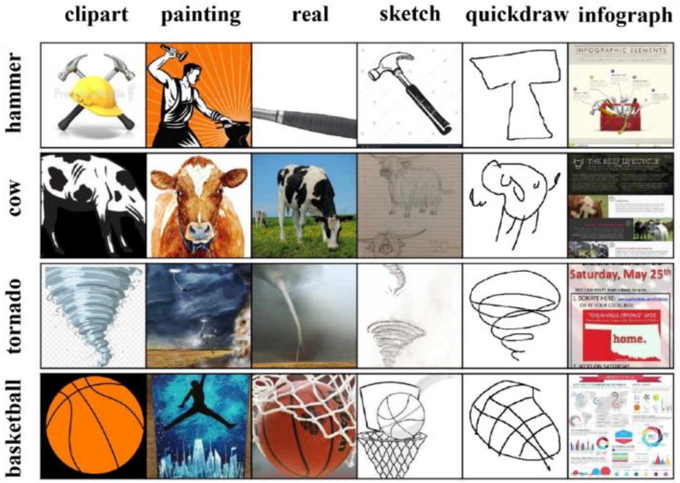
KD3A filters out their knowledge with *Consensus Focus*, making $\epsilon_{\mathbb{D}_S^{K+1}}$ stay away from bad domains.

Experiments

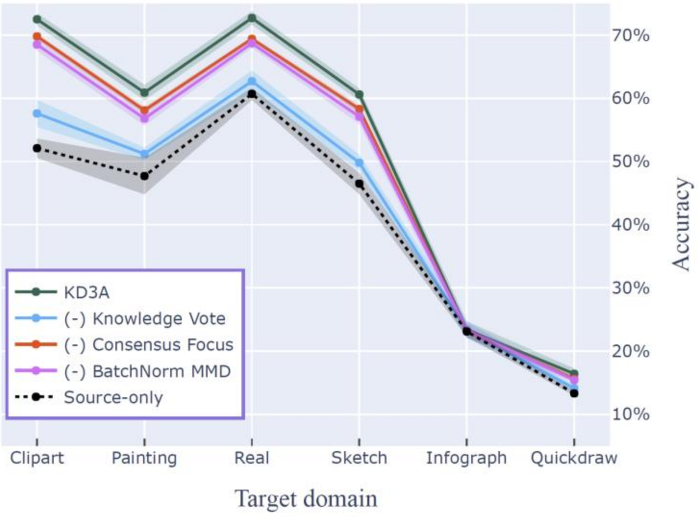
Domain Adaptation Performance on DomainNet

Standards	Methods	*Clipart*	*Infograph*	*Painting*	*Quickdraw*	*Real*	*Sketch*	Avg
W/o DA	Oracle	$69.3_{\pm 0.37}$	$34.5_{\pm 0.42}$	$66.3_{\pm 0.67}$	$66.8_{\pm 0.51}$	$80.1_{\pm 0.59}$	$60.7_{\pm 0.48}$	63.0
	Source-only	$52.1_{\pm 0.51}$	$23.1_{\pm 0.28}$	$47.7_{\pm 0.96}$	$13.3_{\pm 0.72}$	$60.7_{\pm 0.32}$	$46.5_{\pm 0.56}$	40.6
$\mathcal{H}-$div.	MDAN	$60.3_{\pm 0.41}$	$25.0_{\pm 0.43}$	$50.3_{\pm 0.36}$	$8.2_{\pm 1.92}$	$61.5_{\pm 0.46}$	$51.3_{\pm 0.58}$	42.8
	M^3SDA	$58.6_{\pm 0.53}$	$\mathbf{26.0_{\pm 0.89}}$	$52.3_{\pm 0.55}$	$6.3_{\pm 0.58}$	$62.7_{\pm 0.51}$	$49.5_{\pm 0.76}$	42.6
Knowledge Ensemble	DAEL	$70.8_{\pm 0.14}$	$26.5_{\pm 0.13}$	$57.4_{\pm 0.28}$	$12.2_{\pm 0.7}$	$65.0_{\pm 0.23}$	$60.6_{\pm 0.25}$	48.7
Source Selection	CMSS	$64.2_{\pm 0.18}$	$28.0_{\pm 0.2}$	$53.6_{\pm 0.39}$	$16.0_{\pm 0.12}$	$63.4_{\pm 0.21}$	$53.8_{\pm 0.35}$	46.5
Others	DSBN*	60.3	22.6	52.3	9.1	62.7	47.6	42.4
Decentralized UMDA	SHOT*	61.7	22.2	52.6	12.2	67.7	48.6	44.2
	FADA*	59.1	21.7	47.9	8.8	60.8	50.4	41.5
	FADA	$45.3_{\pm 0.7}$	$16.3_{\pm 0.8}$	$38.9_{\pm 0.7}$	$7.9_{\pm 0.4}$	$46.7_{\pm 0.4}$	$26.8_{\pm 0.4}$	30.3
	KD3A	$\mathbf{72.5_{\pm 0.62}}$	$23.4_{\pm 0.43}$	$\mathbf{60.9_{\pm 0.71}}$	$\mathbf{16.4_{\pm 0.28}}$	$\mathbf{72.7_{\pm 0.55}}$	$\mathbf{60.6_{\pm 0.32}}$	**51.1**

Table 1. UMDA accuracy (%) on the DomainNet dataset. Our model KD3A achieves 51.1% accuracy, significantly outperforming all other baselines. Moreover, KD3A achieves the oracle performance on two domains: clipart and sketch. *: The best results recorded in our re-implementation.

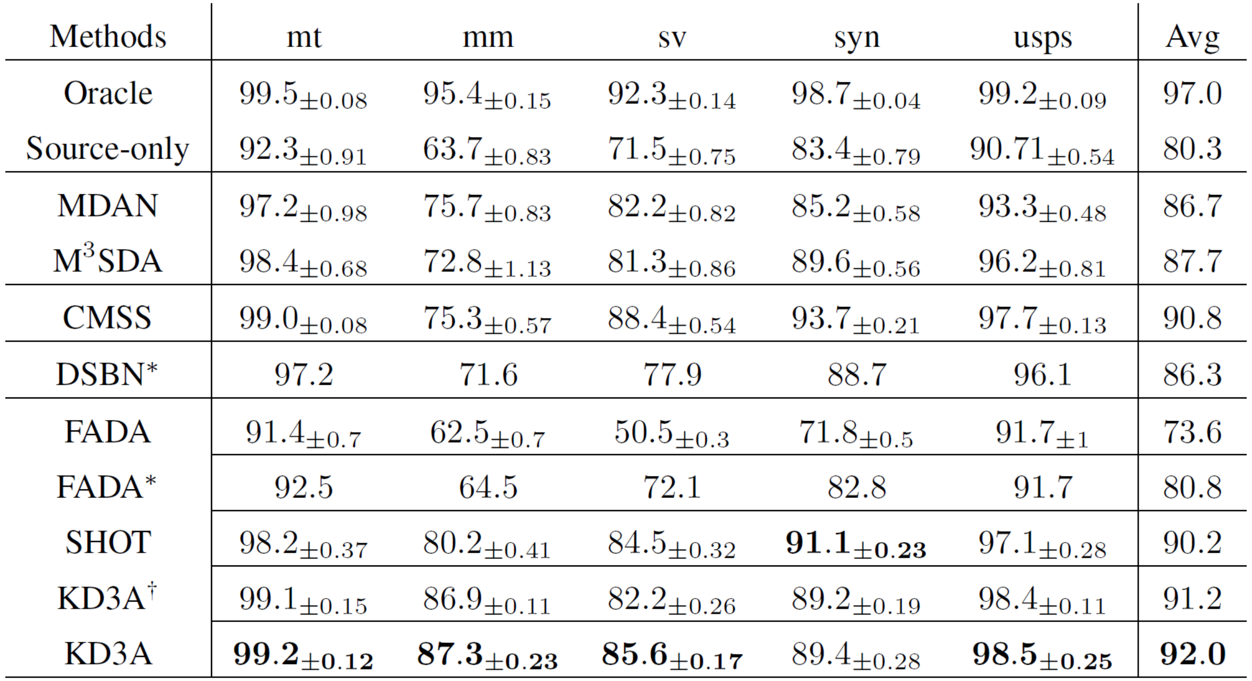


[DomainNet Dataset.]



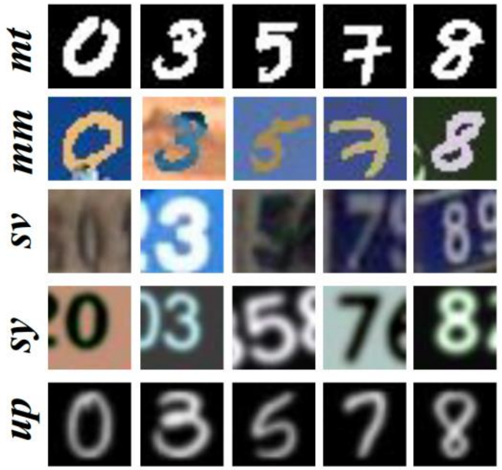
Experiments

Domain Adaptation Performance on DigitFive



Methods	mt	mm	sv	syn	usps	Avg
Oracle	99.5$_{\pm 0.08}$	95.4$_{\pm 0.15}$	92.3$_{\pm 0.14}$	98.7$_{\pm 0.04}$	99.2$_{\pm 0.09}$	97.0
Source-only	92.3$_{\pm 0.91}$	63.7$_{\pm 0.83}$	71.5$_{\pm 0.75}$	83.4$_{\pm 0.79}$	90.71$_{\pm 0.54}$	80.3
MDAN	97.2$_{\pm 0.98}$	75.7$_{\pm 0.83}$	82.2$_{\pm 0.82}$	85.2$_{\pm 0.58}$	93.3$_{\pm 0.48}$	86.7
M^{3}SDA	98.4$_{\pm 0.68}$	72.8$_{\pm 1.13}$	81.3$_{\pm 0.86}$	89.6$_{\pm 0.56}$	96.2$_{\pm 0.81}$	87.7
CMSS	99.0$_{\pm 0.08}$	75.3$_{\pm 0.57}$	88.4$_{\pm 0.54}$	93.7$_{\pm 0.21}$	97.7$_{\pm 0.13}$	90.8
DSBN*	97.2	71.6	77.9	88.7	96.1	86.3
FADA	91.4$_{\pm 0.7}$	62.5$_{\pm 0.7}$	50.5$_{\pm 0.3}$	71.8$_{\pm 0.5}$	91.7$_{\pm 1}$	73.6
FADA*	92.5	64.5	72.1	82.8	91.7	80.8
SHOT	98.2$_{\pm 0.37}$	80.2$_{\pm 0.41}$	84.5$_{\pm 0.32}$	**91.1$_{\pm 0.23}$**	97.1$_{\pm 0.28}$	90.2
KD3A†	99.1$_{\pm 0.15}$	86.9$_{\pm 0.11}$	82.2$_{\pm 0.26}$	89.2$_{\pm 0.19}$	98.4$_{\pm 0.11}$	91.2
KD3A	**99.2$_{\pm 0.12}$**	**87.3$_{\pm 0.23}$**	**85.6$_{\pm 0.17}$**	89.4$_{\pm 0.28}$	**98.5$_{\pm 0.25}$**	**92.0**

Table 6. UMDA accuracy (%) on the **Digit-5**. *: The best results recorded in our re-implementation. †: Methods trained without data-augmentation. Our model KD3A achieves 92.0% accuracy and outperforms all other baselines.



[DigitFive Dataset.]

Experiments

Domain Adaptation Performance on Office-Caltech10

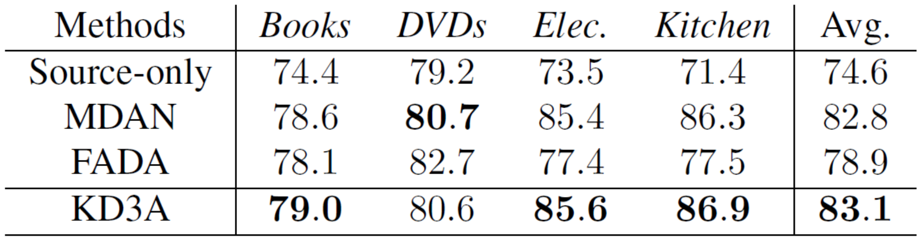
Methods	A	C	D	W	Avg
Oracle	99.7	98.4	99.8	99.7	99.4
Source-only	86.1	87.8	98.3	99.0	92.8
MDAN	98.9	98.6	91.8	95.4	96.1
M^{3}SDA	94.5	92.2	**99.2**	99.5	96.4
CMSS	96.0	93.7	99.3	99.6	97.2
DSBN*	93.2	91.6	98.9	99.3	95.8
FADA	84.2$_{\pm 0.5}$	88.7$_{\pm 0.5}$	87.1$_{\pm 0.6}$	88.1$_{\pm 0.4}$	87.1
SHOT	96.4	96.2	98.5	**99.7**	97.7
KD3A†	96.0$_{\pm 0.07}$	95.2$_{\pm 0.08}$	97.9$_{\pm 0.11}$	99.6$_{\pm 0.03}$	97.2
KD3A	**97.4**$_{\pm 0.08}$	**96.4**$_{\pm 0.11}$	98.4$_{\pm 0.08}$	**99.7**$_{\pm 0.02}$	**97.9**

Table 9. UMDA accuracy (%) on the Office-Caltech10. *: The best results recorded in our re-implementation. †: Methods trained without data-augmentation.

[Office-Caltech10 Dataset.]

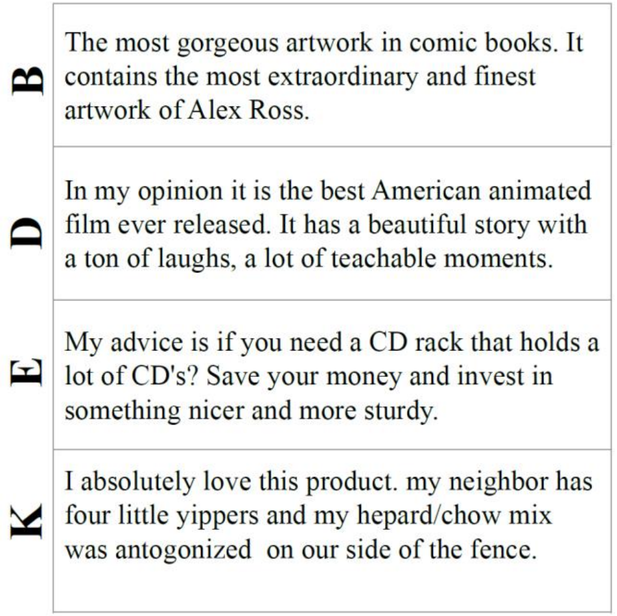
Experiments

Domain Adaptation Performance on AmazonReview



Methods	*Books*	*DVDs*	*Elec.*	*Kitchen*	Avg.
Source-only	74.4	79.2	73.5	71.4	74.6
MDAN	78.6	**80.7**	85.4	86.3	82.8
FADA	78.1	82.7	77.4	77.5	78.9
KD3A	**79.0**	80.6	**85.6**	**86.9**	**83.1**

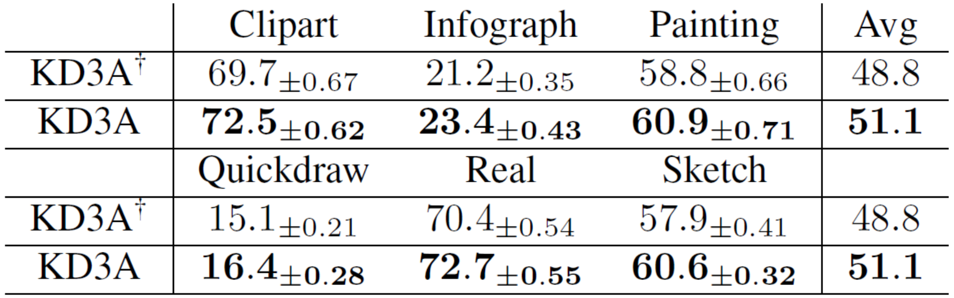
Table 8. The UMDA performance on Amazon Review dataset.



B	The most gorgeous artwork in comic books. It contains the most extraordinary and finest artwork of Alex Ross.
D	In my opinion it is the best American animated film ever released. It has a beautiful story with a ton of laughs, a lot of teachable moments.
E	My advice is if you need a CD rack that holds a lot of CD's? Save your money and invest in something nicer and more sturdy.
K	I absolutely love this product. my neighbor has four little yippers and my hepard/chow mix was antogonized on our side of the fence.

[AmazonReview Dataset.]

Data Augmentation in Domain Adaptation



	Clipart	Infograph	Painting	Avg
KD3A†	$69.7_{\pm0.67}$	$21.2_{\pm0.35}$	$58.8_{\pm0.66}$	48.8
KD3A	$\mathbf{72.5_{\pm0.62}}$	$\mathbf{23.4_{\pm0.43}}$	$\mathbf{60.9_{\pm0.71}}$	**51.1**
	Quickdraw	Real	Sketch	
KD3A†	$15.1_{\pm0.21}$	$70.4_{\pm0.54}$	$57.9_{\pm0.41}$	48.8
KD3A	$\mathbf{16.4_{\pm0.28}}$	$\mathbf{72.7_{\pm0.55}}$	$\mathbf{60.6_{\pm0.32}}$	**51.1**

Table 7. The ablation study for data-augmentation strategies on DomainNet.†: Methods trained without data-augmentation.

Augmentation Strategy:

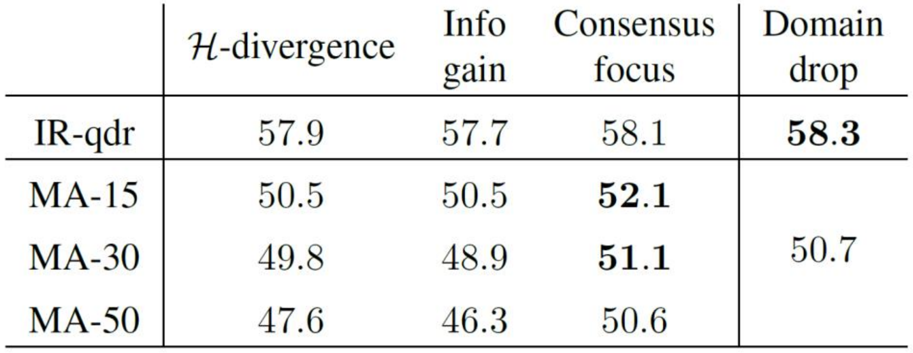
To reduce hyper-parameters, we use mixup as a unified augmentation strategy and simply set the mix-parameter $\alpha = 0.2$ in all experiments.

Experiments

Robustness To Negative Transfer

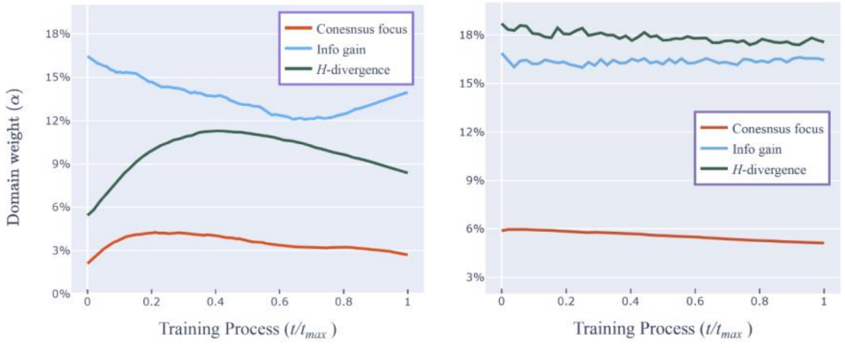
We construct irrelevant and malicious source domains on DomainNet and conduct synthesized experiments to show the robustness of KD3A to negative transfer.

Since ***Quickdraw*** is very different from others, and all models perform bad on it, we take ***Quickdraw*** as the irrelevant domain**,** denoted by IR-qdr. Moreover, we perform poisoning attack on the high-quality domain ***Real*** with m% wrong labels to construct malicious domain, denoted by MA-m.



	$\mathcal{H}$-divergence	Info gain	Consensus focus	Domain drop
IR-qdr	57.9	57.7	58.1	**58.3**
MA-15	50.5	50.5	**52.1**	50.7
MA-30	49.8	48.9	**51.1**	
MA-50	47.6	46.3	50.6	

Table 2. Average UMDA accuracy (%) with irrelevant and malicious domains. IR-qdr means to use the *Quickdraw* as the irrelevant source domain, while MA-m means to construct a malicious source domain with $m\%$ mislabeled data. With consensus focus, our KD3A is robust to negative transfer.

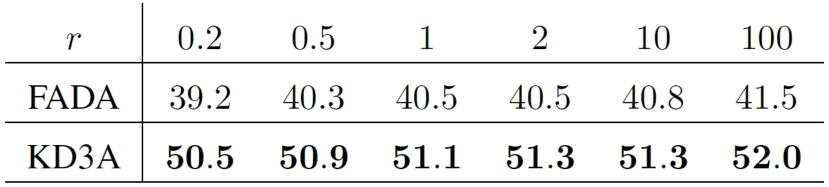


(a) IR-qdr. (b) MA-30.

Figure 4. Weights assigned to the irrelevant and malicious domains in the training process. Our consensus focus can identify these bad domains with the low weights.

Experiments

Communication Efficiency And Privacy Security



r	0.2	0.5	1	2	10	100
FADA	39.2	40.3	40.5	40.5	40.8	41.5
KD3A	**50.5**	**50.9**	**51.1**	**51.3**	**51.3**	**52.0**

Table 3. Average UMDA accuracy (%) with different communication rounds r for our KD3A and FADA. KD3A achieves good performance with low communication cost (e.g., $r \leq 1$).

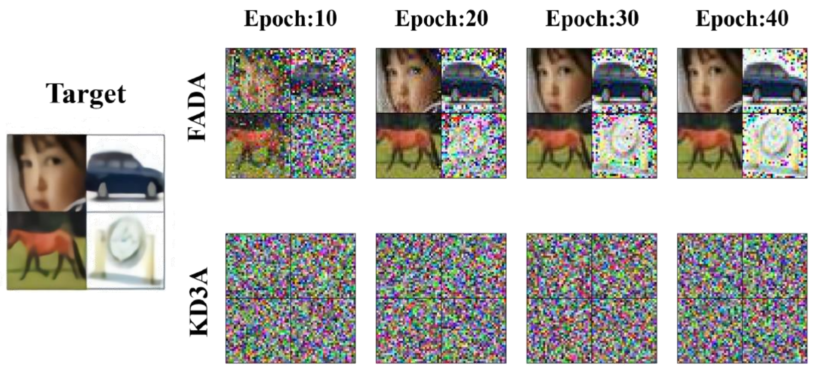


Figure 5. The gradient leakage attack (Zhu et al., 2019) on decentralized training strategy. KD3A is robust to this attack while FADA causes the privacy leakage.

Conclusions:

1. Due to the adversarial training strategy, FADA works under large communication rounds (i.e. ***r*** **= 100**).

2. KD3A works under the low communication cost with ***r = 1***, leading to a ***100*** × communication reduction.

3. KD3A is robust to communication rounds. For example, the accuracy only drops ***0.9%*** when ***r*** decreases from 100 to 1.

4. KD3A works under extremely low communication cost (e.g., ***r=0.2&0.5***).

5. Due to the low communication cost, our KD3A is robust to the advanced gradient leakage attack, which demonstrates high privacy security.

Thanks

Acknowledgement

National Key Research and Development Program (2019YFB1404802)
National Natural Science Foundation of China (U1866602, 61772456)