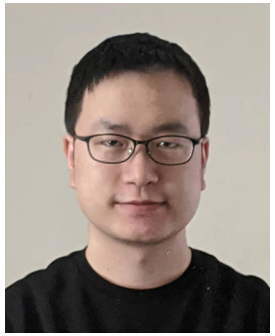
AIFARMS
Artificial Intelligence for Future Agricultural Resilience, Management, and Sustainability

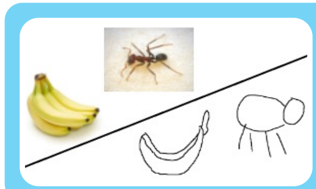
Domain Adaptation with Dynamic Open-Set Targets



Jun Wu
UIUC
junwu3@illinois.edu

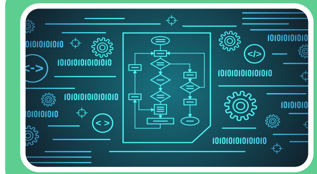
Jingrui He
UIUC
jingrui@illinois.edu

Roadmap

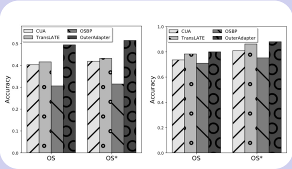


Background
- Standard domain adaptation
- Limitations in real scenarios

Problem Definition
- Dynamic open-set domain adaptation
- Unique challenges



Proposed Model
- Theoretical analysis
- Algorithm: PU learning based adaptation



Experiments
- Effectiveness
- Efficiency

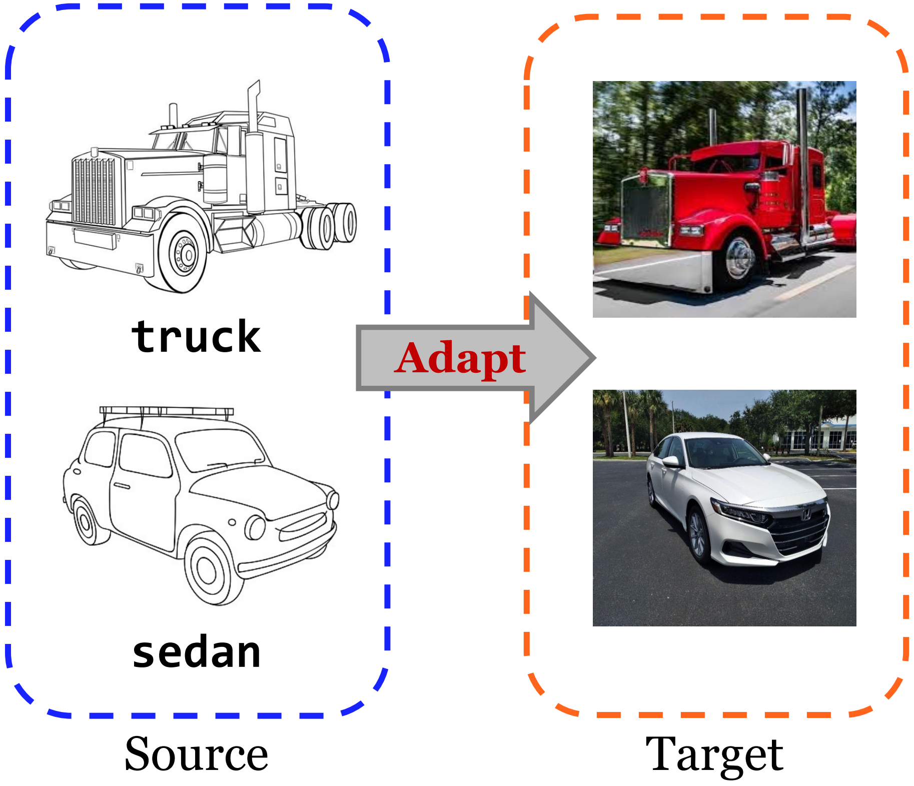
Conclusion
- Problem, analysis, algorithm, result

Unsupervised Domain Adaptation (UDA)

- **Knowledge transfer**
  - Source domain: labeled samples
  - Target domain: unlabeled samples
  - Goal: learn the predictive function on target domain

- **Assumptions:**
  - **Closedness**: share the same group of categories
  - **Relatedness**: share similar data distribution

- **Distribution shift induced by**
  - Background
  - Style
  - Quality
  - …



$$\mathbb{P}^s \neq \mathbb{P}^s$$

Unsupervised Domain Adaptation (UDA)

❑ Knowledge transfer

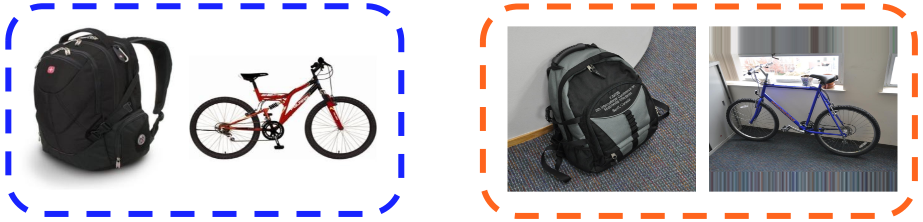
- Source domain: labeled samples
- Target domain: unlabeled samples
- Goal: learn the predictive function on target domain

❑ Assumptions:

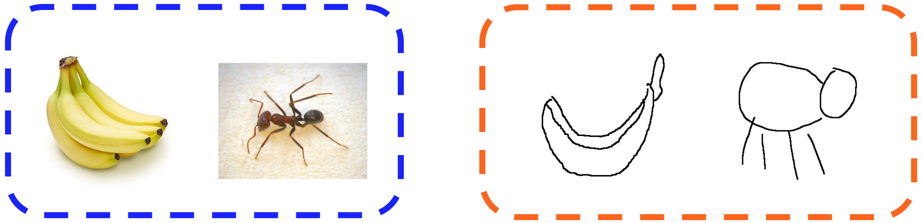
- **Closedness**: share the same group of categories
- **Relatedness**: share similar data distribution

❑ Distribution shift induced by

- Background
- Style
- Quality
- ...



Background shift (Office-31)



Style shift (DomainNet)

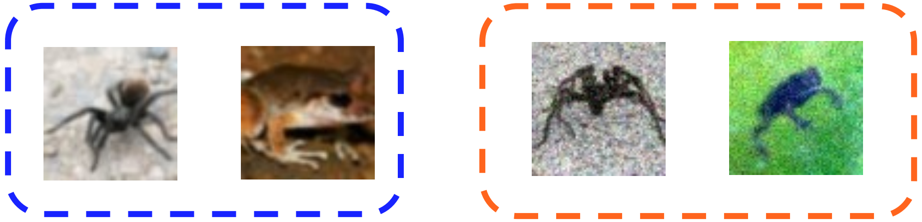


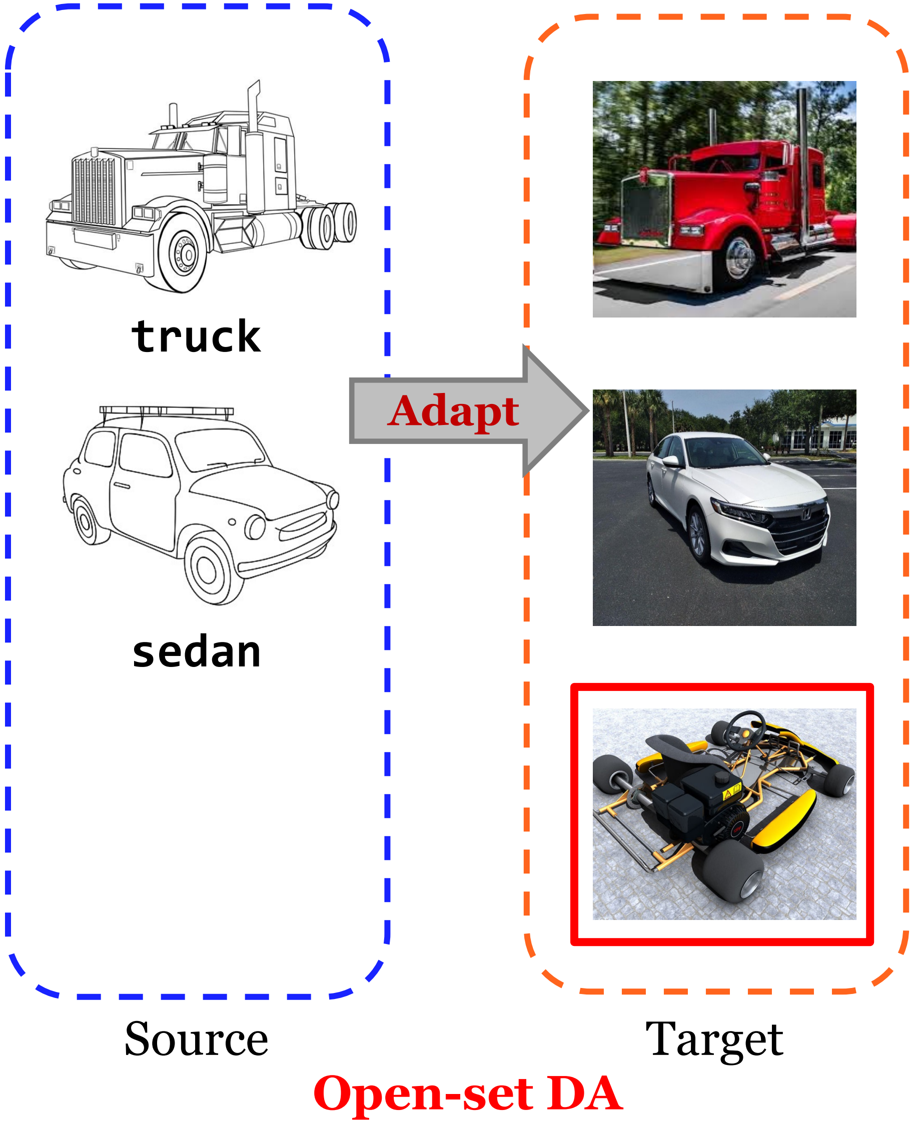
Image quality shift (ImageNet-C)

- Kate Saenko, et al. "**Adapting visual category models to new domains**." ECCV, 2010.
- Xingchao Peng, et al. "**Moment matching for multi-source domain adaptation**." ICCV, 2019.
- Dan Hendrycks, et al. "**Benchmarking Neural Network Robustness to Common Corruptions and Perturbations**." ICLR, 2019.

Limitations: Open-Set Scenario

❑ Open-set domain adaptation

- Source and target have different groups of categories
- $\mathcal{Y}_s \subset \mathcal{Y}_t$



- Judy Hoffman, et al. "**Continuous manifold based adaptation for evolving visual domains**." CVPR, 2014.
- Pau Panareda Busto, et al. "**Open set domain adaptation**." ICCV, 2017.

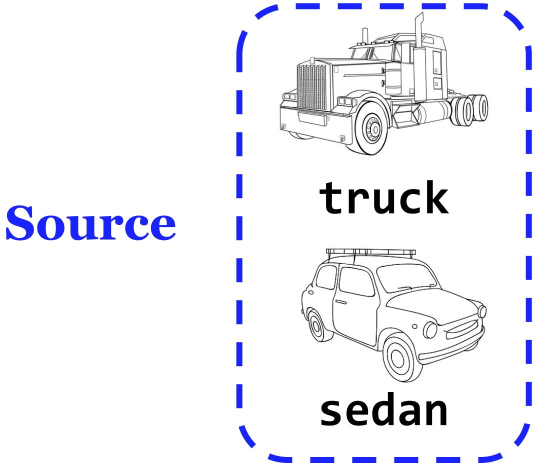
Limitations: Dynamic Adaptation Scenario

❑ Open-set domain adaptation

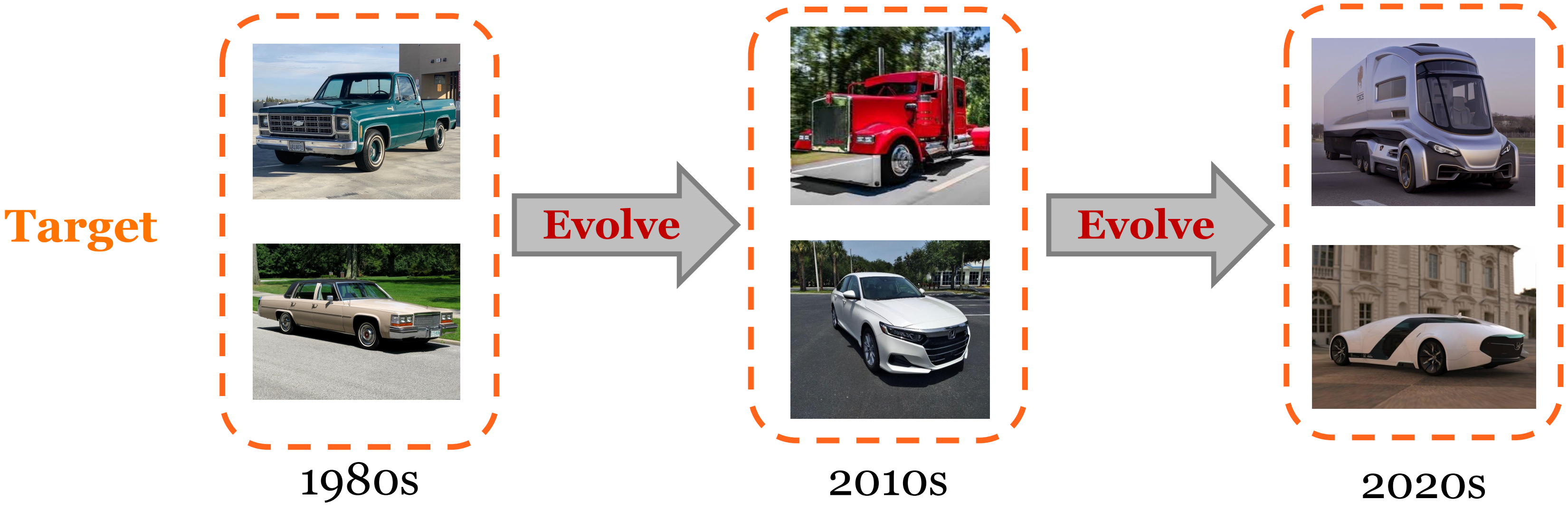
- Source and target have different groups of categories
- $\mathcal{Y}_s \subset \mathcal{Y}_t$

❑ Time-evolving target domain

- The relatedness between source and target domains is changing over time
- $d(\mathcal{D}_s, \mathcal{D}_{t_1}) \leq d(\mathcal{D}_s, \mathcal{D}_{t_2}) \leq \cdots \leq d(\mathcal{D}_s, \mathcal{D}_{t_N})$

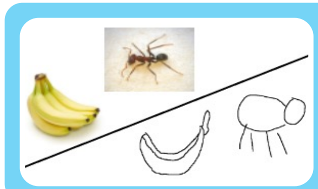


Dynamic DA



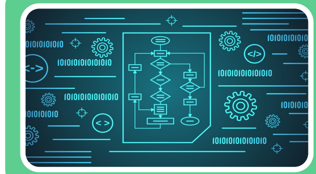
- Judy Hoffman, et al. "**Continuous manifold based adaptation for evolving visual domains**." CVPR, 2014.
- Pau Panareda Busto, et al. "**Open set domain adaptation**." ICCV, 2017.

Roadmap

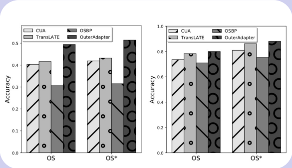


Background
- Standard domain adaptation
- Limitations in real scenarios

Problem Definition
- Dynamic open-set domain adaptation
- Unique challenges



Proposed Model
- Theoretical analysis
- Algorithm: PU learning based adaptation

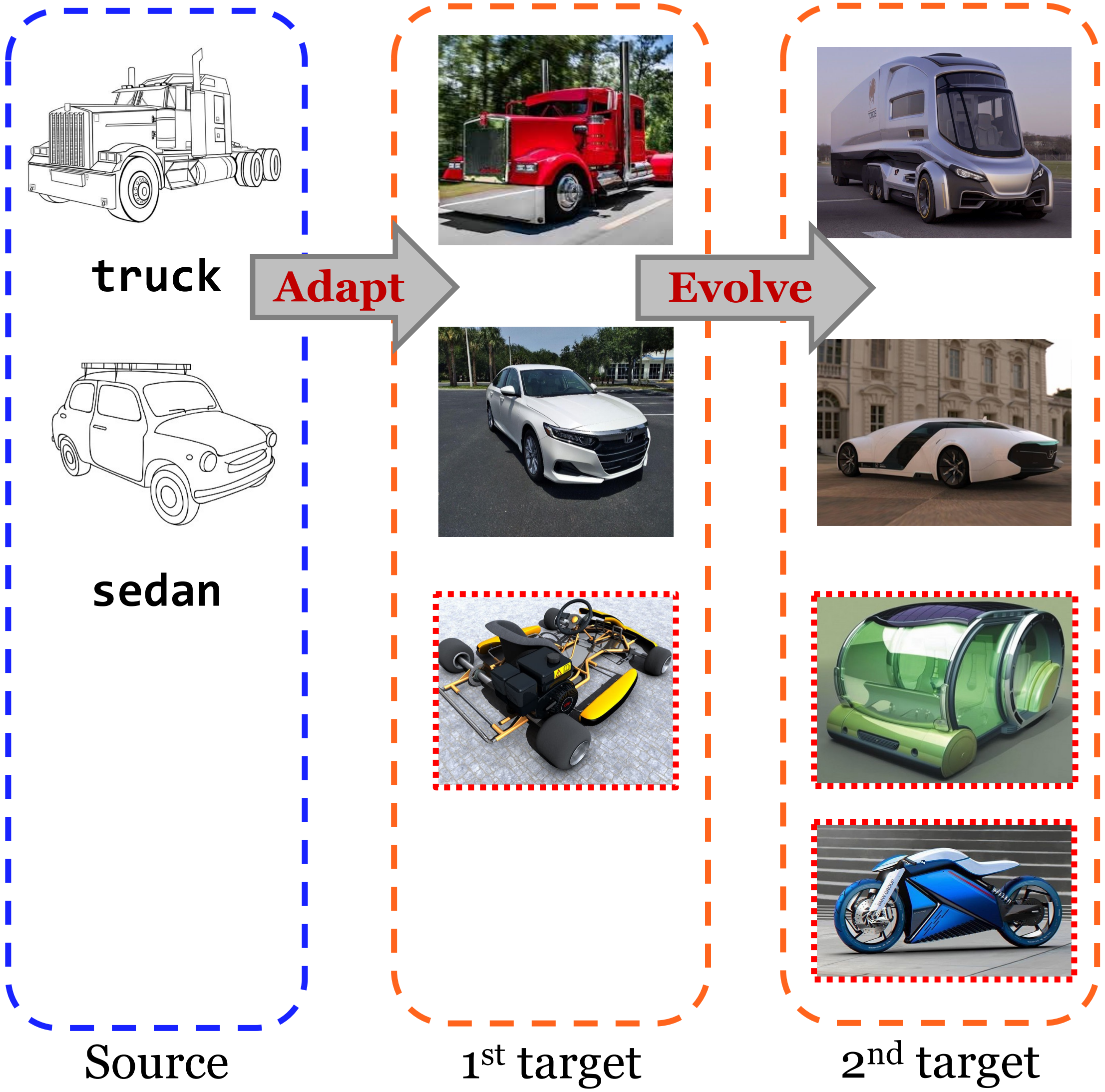


Experiments
- Effectiveness
- Efficiency

Conclusion
- Problem, analysis, algorithm, result

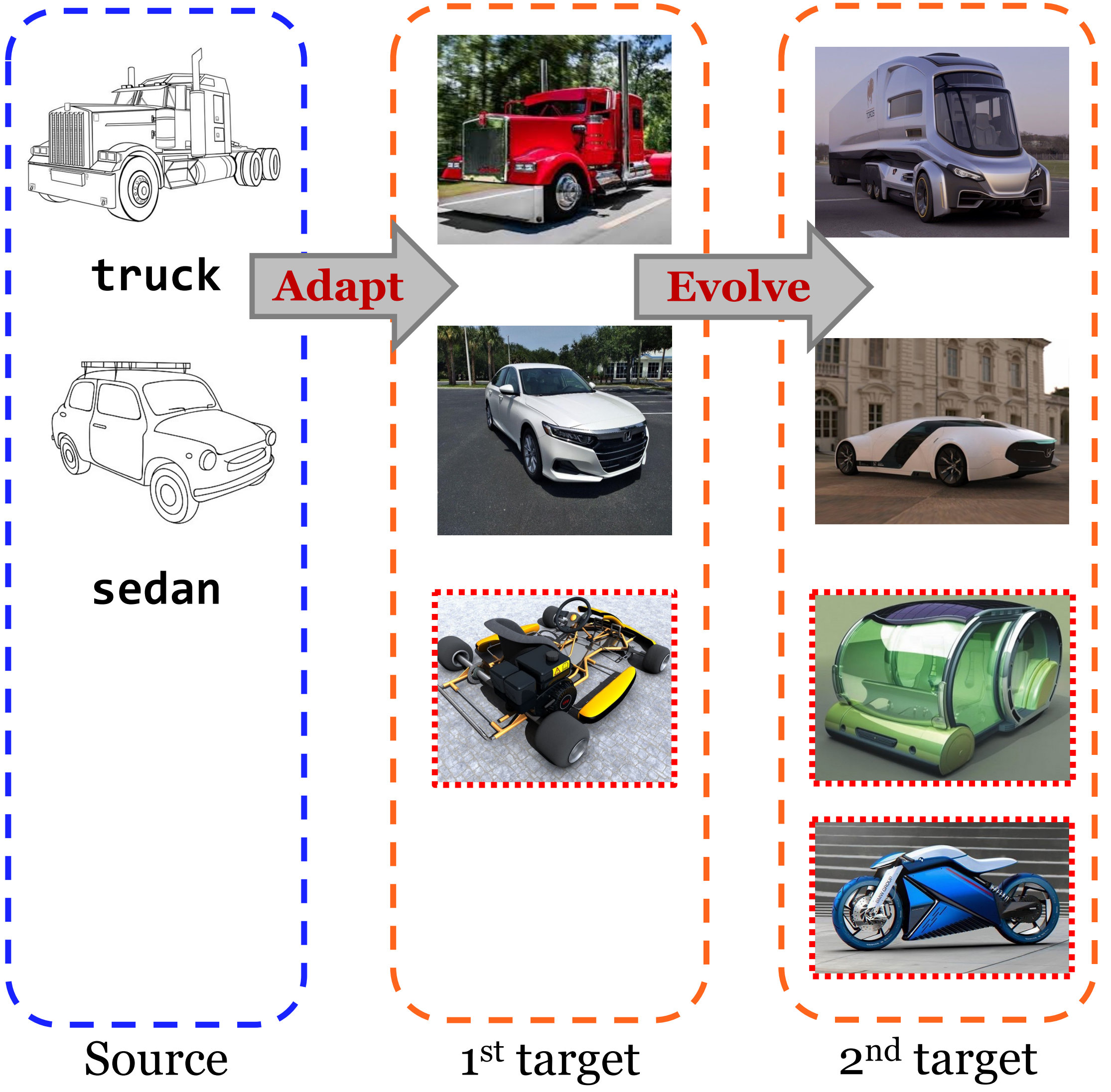
Problem Definition

- **Dynamic open-set domain adaptation**
  - **Given:**
    - A static source domain (fully labeled)
      - $\mathcal{Y}_s = \{1,2,\cdots,C\}$
    - A **time-evolving** target domain (unlabeled) with **novel unseen classes**
      - $\mathcal{Y}_s \subset \mathcal{Y}_{t_j} = \{1,2,\cdots,C,$ "unknown"$\}$
  - **Output:**
    - Classify the data of known classes correctly
    - Identify the data of unseen classes as "unknown"

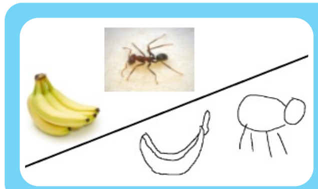


Unique Challenges

- **C1: Evolving distribution**
  - The target distribution is continuously evolving
- **C2: Varying class proportions**
  - The ratio of known target examples changes

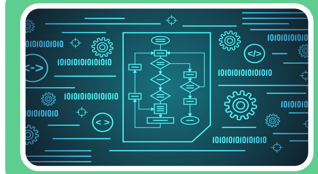


Roadmap

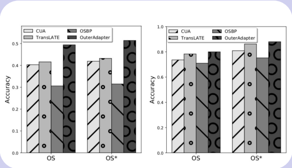


Background
- Standard domain adaptation
- Limitations in real scenarios

Problem Definition
- Dynamic open-set domain adaptation
- Unique challenges



Proposed Model
- Theoretical analysis
- Algorithm: PU learning based adaptation



Experiments
- Effectiveness
- Efficiency

Conclusion
- Problem, analysis, algorithm, result

Distribution Shift under Open-Set Targets

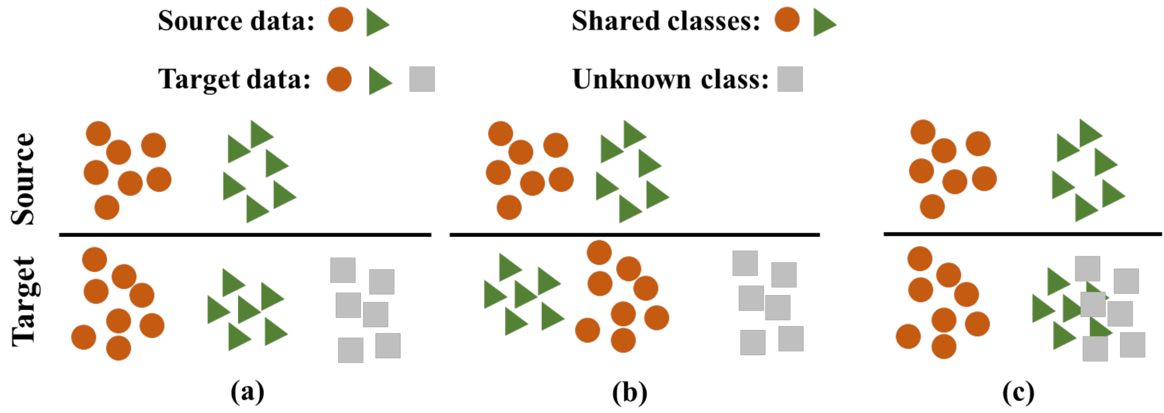
❑ Distribution discrepancy

- Existing discrepancy measure ⇒ (b)(c)

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}_X^s, \mathbb{P}_X^t) = \sup_{h,h' \in \mathcal{H}} \left| \Pr_{\mathbb{Q}_X^s}[B] - \Pr_{\mathbb{P}_X^t}[B] \right|$$

- **Open-set discrepancy measure ⇒ (a)**

$$d_{OS}(\mathbb{Q}^s, \mathbb{P}^t) = d_{\mathcal{C}}(\mathbb{Q}^s, \mathbb{P}^t_{\leq C}) - \rho \cdot d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{Q}_X^s, \mathbb{P}^t_{X|Y=C+1}\right)$$



(b) Feature misalignment

(c) Indistinguishable "unknown" data

Limitations of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}_X^s, \mathbb{P}_X^t)$

- Shai Ben-David, et al. "**A theory of learning from different domains**." Machine learning, 2010.
- Jun Wu, et al. "**Continuous Transfer Learning with Label-informed Distribution Alignment**." arXiv preprint arXiv:2006.03230, 2020.

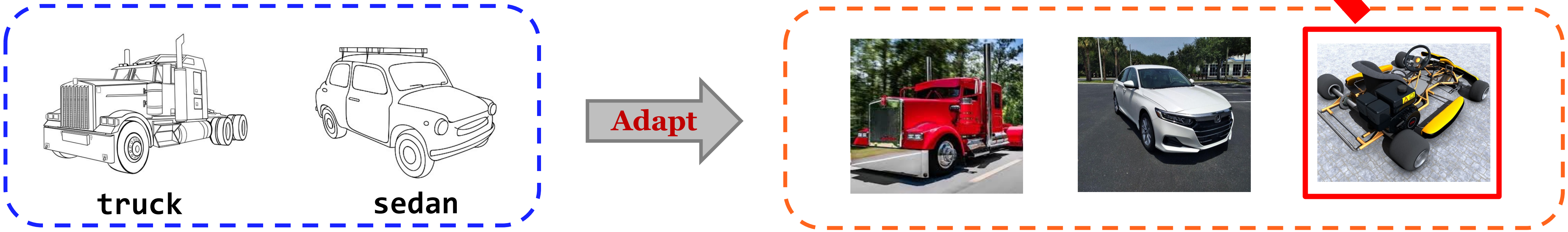
PU-Learning under Open-Set Targets

❑ Open-set DA as positive-unlabeled (PU) learning

- **Positive**: source examples (C classes)
- **Unlabeled**: target examples (C classes + unknown classes)
- Observation: if there is no distribution shift,

Target error ➡ $\epsilon_t(h) = (1 - \pi^t_{C+1}) \cdot \epsilon_s(h) + \underbrace{\mathbb{E}_{x\sim\mathbb{P}^t_X}[L(h(x), y = C+1)] - (1 - \pi^t_{C+1})\mathbb{E}_{x\sim\mathbb{Q}^t_X}[L(h(x), y = C+1)]}_{\text{Positive-unlabeled open-set risk}}$

$\boldsymbol{\pi^t_{C+1} = \mathbb{P}^t(y = C+1)}$



Source

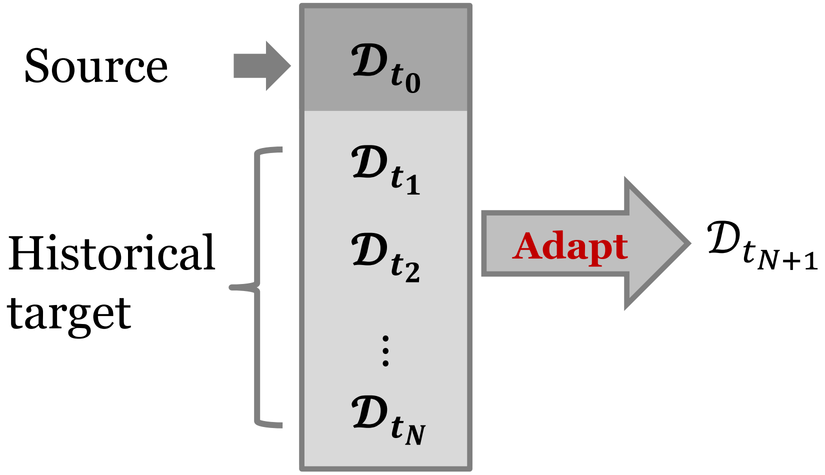
Target

• Mohammad Reza Loghmani, et al. "**Positive-unlabeled learning for open set domain adaptation**." Pattern Recognition Letters, 2020.

Theoretical Analysis

❑ Error upper bound on $\epsilon_{t_{N+1}}(h)$

- Classification error on historical task
  - **- Learn class membership on shared classes**
- Open-set distribution discrepancy $d_{OS}(\cdot,\cdot)$
  - **- Measure distribution shift**
- PU-learning based open-set risk Δ_{PU}
  - **- Identify the "unknown" class in the target domain**



Theorem 1: Assume that the loss function $L(\cdot,\cdot)$ is bounded, i.e., $|L(\cdot,\cdot)| \leq M$. For any hypothesis $h \in \mathcal{H}$ and $\sum_{j=0}^{N} \alpha_j = 1$ where $\alpha_j \geq 0$ $(j = 0,1,\cdots,N)$, the expected error $\epsilon_{t_{N+1}}(h)$ of the target task at the $(N+1)^{\text{th}}$ time stamp is bounded as:

$$\epsilon_{t_{N+1}}(h) \leq \left(1-\pi_{C+1}^{t_{N+1}}\right)\left(\sum_{j=0}^{N} \alpha_j \boxed{\mathbb{E}_{(x,y)\sim\mathbb{P}_{\leq C}^{t_j}}\left[L(h(x),y)\right]} + 4M \sum_{j=0}^{N} \alpha_j \boxed{d_{OS}\left(\mathbb{P}_{\leq C}^{t_j}, \mathbb{P}^{t_{N+1}}\right)}\right) + \boxed{\Delta_{PU}} + CONST$$

where $\Delta_{PU} = \mathbb{E}_{x\sim\mathbb{P}_X^{t_{N+1}}}[L(h(x), y = C+1)] - \left(1-\pi_{C+1}^{t_{N+1}}\right)\sum_{j=0}^{N} \alpha_j \mathbb{E}_{(x,y)\sim\mathbb{P}_{\leq C}^{t_j}}[L(h(x), y = C+1)]$ is the positive-unlabeled open-set risk.

Proposed Algorithm: OuterAdapter

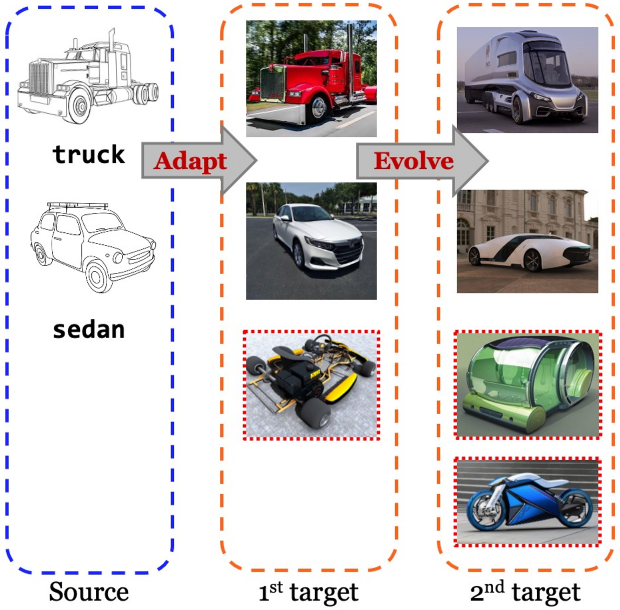
❑ Objective function

- PU loss: **Discriminative feature learning under open-set targets**

$$\min_{\theta} \sum_{j=0}^{N} \frac{(1-\pi_{C+1}^{t_{N+1}})\alpha_j}{n_{t_j}} \sum_{i=1}^{n_{t_j}} \left(L\left(h\left(x_{t_j}^i\right), \hat{y}_{t_j}^i; \theta\right) - L\left(h\left(x_{t_j}^i\right), y = C+1; \theta\right)\right) + \frac{1}{m_{t_{N+1}}} \sum_{i=1}^{m_{t_{N+1}}} L\left(h\left(x_{t_{N+1}}^i\right), y = C+1; \theta\right)$$

$$+\beta \sum_{j=0}^{N} \alpha_j d_{\mathcal{OS}}\left(\mathbb{P}_{\leq C}^{t_j}, \mathbb{P}^{t_{N+1}}; \theta\right)$$

- $\mathcal{OS}$-divergence: **Domain-invariant feature learning**



Proposed Algorithm: OuterAdapter

❑ Objective function

- PU loss: Discriminative feature learning under open-set targets

Importance of j-th target task **Pseudo-label of unlabeled examples in j-th target task**

$$\min_{\theta} \sum_{j=0}^{N} \frac{\left(1-\pi_{C+1}^{t_{N+1}}\right)\alpha_j}{n_{t_j}} \sum_{i=1}^{n_{t_j}} \left(L\left(h\left(x_{t_j}^i\right), \hat{y}_{t_j}^i; \theta\right) - L\left(h\left(x_{t_j}^i\right), y = C+1; \theta\right)\right) + \frac{1}{m_{t_{N+1}}} \sum_{i=1}^{m_{t_{N+1}}} L\left(h\left(x_{t_{N+1}}^i\right), y = C+1; \theta\right)$$

$$+\beta \sum_{j=0}^{N} \alpha_j d_{\mathcal{OS}}\left(\mathbb{P}_{\leq C}^{t_j}, \mathbb{P}^{t_{N+1}}; \theta\right)$$

Open-set discrepancy measure

$\pi_{C+1}^{t_{N+1}}$: Class prior probability

- $\mathcal{OS}$-divergence: Domain-invariant feature learning

Algorithm Details

❑ Key components

- Estimation of pseudo-label $\hat{y}^i_{t_j}$ for historical target task
  - Incrementally update the model h:
  - Generate the pseudo-label $\hat{y} = h(x)$:

$$\begin{array}{ccccccccc} \mathcal{D}_s & \rightarrow & \mathcal{D}_{t_1} & \rightarrow & \mathcal{D}_{t_2} & \rightarrow & \cdots & \rightarrow & \mathcal{D}_{t_N} \\ & & \downarrow & & \downarrow & & & & \downarrow \\ & & \hat{y}^i_{t_1} & & \hat{y}^i_{t_2} & & & & \hat{y}^i_{t_N} \end{array}$$

- Estimation of $d_{\mathcal{OS}}\left(\mathbb{P}^{t_j}_{\leq C}, \mathbb{P}^{t_{N+1}}; \theta\right)$
  - $d_{\mathcal{OS}}\left(\mathbb{P}^{t_j}_{\leq C}, \mathbb{P}^{t_{N+1}}\right) = \underbrace{d_{\mathcal{C}}\left(\mathbb{P}^{t_j}_{\leq C}, \mathbb{P}^{t_{N+1}}_{\leq C}\right)}_{\text{Label-informed } \mathcal{C}\text{-divergence}} - \rho \cdot \underbrace{d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{P}^{t_j}_{X|\leq C}, \mathbb{P}^{t_{N+1}}_{X|Y=C+1}\right)}_{\text{Domain-adversarial learning}}$

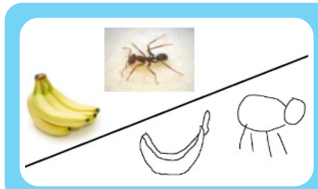
- Estimation of α_j
  - Self-attention scheme
  - $\alpha_j \leftarrow \exp\left(\text{LeakyReLU}\left(\sum_{c=1}^{C=1} a_c^T \cdot \sum_{i=1}^{n_{t_j}} x^i_{t_j} \mathbb{I}\left[\hat{y}^i_{t_j} = c\right]\right)\right)$

OuterAdapter$_\mu$:

$$\alpha_j \leftarrow \frac{\mu^{N-j}}{\sum_{k=0}^{N} \mu^{N-k}} \qquad 0 \leq \mu \leq 1$$

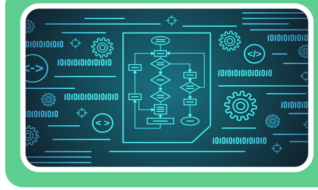
- Yaroslav Ganin, et al. "**Domain-adversarial training of neural networks**." The Journal of Machine Learning Research, 2016.
- Jun Wu, et al. "**Continuous Transfer Learning with Label-informed Distribution Alignment**." arXiv preprint arXiv:2006.03230, 2020.

Roadmap

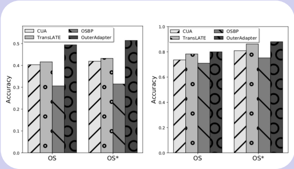


Background
- Standard domain adaptation
- Limitations in real scenarios

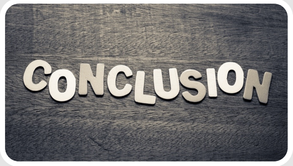
Problem Definition
- Dynamic open-set domain adaptation
- Unique challenges



Proposed Model
- Theoretical analysis
- Algorithm: PU learning based adaptation



Experiments
- Effectiveness
- Efficiency



Conclusion
- Problem, analysis, algorithm, result

Experiments

❑ Experimental settings

- **Data sets:**
  - Office-31
  - Office-Home
  - Syn2Real-O

Data	# domains	# images	# categories
Office-31	3	4, 652	31
Office-Home	4	15, 500	65
Syn2Real-O	2	200, 000	12

- **Baselines:**
  - Static closed-set adaptation: SourceOnly, DANN
  - Multi-source adaptation: MDAN, DARN
  - Dynamic closed-set adaptation: CUA, TransLATE
  - Static open-set adaptation: OSBP, DAMC
  - Dynamic open-set adaptation: OSBP+CUA, DAMC+CUA

- **Evaluation metric :**
  - **OS**: Average classification accuracy over all the classes
  - **OS***: Average classification accuracy over all the known classes

- Kuniaki Saito, et al. "**Open set domain adaptation by backpropagation**." ECCV, 2018.
- Han Zhao, et al. "**Adversarial multiple source domain adaptation**." NeurIPS, 2018.
- Andreea Bobu, et al. "**Adapting to continuously shifting domains**." ICLR Workshop, 2018.

Results

Dynamic open-set domain adaptation on Office-31 (Amazon → DSLR)

	Method	$\mathcal{D}_{t_1}$		$\mathcal{D}_{t_2}$		$\mathcal{D}_{t_3}$		$\mathcal{D}_{t_4}$		$\mathcal{D}_{t_5}$		$\mathcal{D}_{t_6}$	
		OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
Static closed-set	SourceOnly	$0.871_{\pm0.009}$	$0.958_{\pm0.009}$	$0.822_{\pm0.007}$	$0.905_{\pm0.008}$	$0.690_{\pm0.003}$	$0.759_{\pm0.003}$	$0.733_{\pm0.027}$	$0.806_{\pm0.031}$	$0.649_{\pm0.009}$	$0.714_{\pm0.011}$	$0.542_{\pm0.008}$	$0.596_{\pm0.009}$
	DANN	$0.891_{\pm0.006}$	$0.980_{\pm0.007}$	$0.821_{\pm0.003}$	$0.903_{\pm0.003}$	$0.775_{\pm0.009}$	$0.853_{\pm0.010}$	$0.804_{\pm0.020}$	$0.884_{\pm0.023}$	$0.767_{\pm0.019}$	$0.844_{\pm0.020}$	$0.742_{\pm0.014}$	$0.815_{\pm0.016}$
Multi-source	MDAN	$0.900_{\pm0.005}$	$0.990_{\pm0.005}$	$0.866_{\pm0.017}$	$0.953_{\pm0.019}$	$0.819_{\pm0.013}$	$0.901_{\pm0.019}$	$0.834_{\pm0.011}$	$0.918_{\pm0.012}$	$0.797_{\pm0.016}$	$0.877_{\pm0.017}$	$0.792_{\pm0.014}$	$0.872_{\pm0.016}$
	DARN	$0.884_{\pm0.008}$	$0.972_{\pm0.008}$	$0.831_{\pm0.022}$	$0.914_{\pm0.025}$	$0.815_{\pm0.020}$	$0.897_{\pm0.021}$	$0.815_{\pm0.025}$	$0.896_{\pm0.027}$	$0.797_{\pm0.022}$	$0.877_{\pm0.024}$	$0.803_{\pm0.024}$	$0.883_{\pm0.027}$
Dynamic closed-set	CUA	$0.879_{\pm0.002}$	$0.967_{\pm0.003}$	$0.850_{\pm0.013}$	$0.935_{\pm0.015}$	$0.832_{\pm0.015}$	$0.915_{\pm0.017}$	$0.834_{\pm0.007}$	$0.918_{\pm0.008}$	$0.836_{\pm0.006}$	$0.919_{\pm0.007}$	$0.834_{\pm0.010}$	$0.917_{\pm0.011}$
	TransLATE	$0.897_{\pm0.006}$	$0.987_{\pm0.007}$	$0.883_{\pm0.014}$	$0.971_{\pm0.015}$	$0.849_{\pm0.026}$	$0.934_{\pm0.029}$	$0.862_{\pm0.015}$	$0.948_{\pm0.017}$	$0.856_{\pm0.026}$	$0.942_{\pm0.028}$	$0.846_{\pm0.021}$	$0.930_{\pm0.023}$
Static open-set	OSBP	**$0.907_{\pm0.003}$**	**$0.993_{\pm0.001}$**	$0.848_{\pm0.013}$	$0.929_{\pm0.010}$	$0.792_{\pm0.033}$	$0.868_{\pm0.034}$	$0.788_{\pm0.003}$	$0.862_{\pm0.001}$	$0.813_{\pm0.007}$	$0.892_{\pm0.005}$	$0.777_{\pm0.033}$	$0.853_{\pm0.035}$
	DAMC	$0.894_{\pm0.002}$	$0.980_{\pm0.000}$	$0.878_{\pm0.011}$	$0.962_{\pm0.010}$	$0.828_{\pm0.006}$	$0.901_{\pm0.007}$	$0.792_{\pm0.025}$	$0.858_{\pm0.025}$	$0.770_{\pm0.028}$	$0.838_{\pm0.031}$	$0.749_{\pm0.017}$	$0.814_{\pm0.026}$
Dynamic open-set	OSBP+CUA	**$0.907_{\pm0.003}$**	**$0.993_{\pm0.001}$**	$0.855_{\pm0.001}$	$0.940_{\pm0.000}$	$0.853_{\pm0.003}$	$0.938_{\pm0.004}$	$0.852_{\pm0.003}$	$0.936_{\pm0.004}$	$0.855_{\pm0.001}$	$0.940_{\pm0.001}$	$0.846_{\pm0.003}$	$0.931_{\pm0.003}$
	DAMC+CUA	$0.894_{\pm0.002}$	$0.980_{\pm0.000}$	$0.868_{\pm0.015}$	$0.953_{\pm0.017}$	$0.847_{\pm0.018}$	$0.931_{\pm0.019}$	$0.839_{\pm0.010}$	$0.923_{\pm0.011}$	$0.840_{\pm0.031}$	$0.924_{\pm0.034}$	$0.823_{\pm0.016}$	$0.905_{\pm0.018}$
	OuterAdapter$_\mu$	$0.901_{\pm0.007}$	$0.991_{\pm0.008}$	$0.883_{\pm0.007}$	$0.971_{\pm0.008}$	**$0.877_{\pm0.019}$**	**$0.964_{\pm0.021}$**	**$0.872_{\pm0.008}$**	**$0.959_{\pm0.009}$**	$0.862_{\pm0.021}$	$0.948_{\pm0.023}$	$0.865_{\pm0.007}$	$0.951_{\pm0.008}$
	OuterAdapter	$0.901_{\pm0.007}$	$0.991_{\pm0.008}$	**$0.895_{\pm0.003}$**	**$0.985_{\pm0.004}$**	$0.875_{\pm0.009}$	$0.962_{\pm0.010}$	$0.868_{\pm0.012}$	$0.954_{\pm0.013}$	**$0.869_{\pm0.010}$**	**$0.955_{\pm0.011}$**	**$0.874_{\pm0.010}$**	**$0.962_{\pm0.010}$**

Observations:

- ❑ Open-set approaches can better classify the data within the shared classes
- ❑ OuterAdapter outperforms the baselines by a large margin

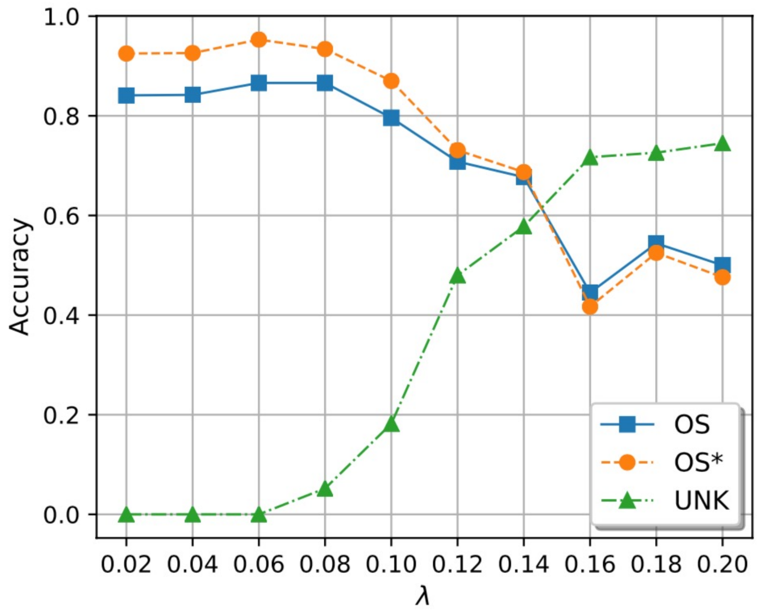
Model Analysis

(a) Ablation study

Method	OS	OS*
OuterAdapter w/o PU-Learning	0.545	0.600
OuterAdapter w/o historical data	0.730	0.803
OuterAdapter w/o $\mathcal{OS}$-divergence	0.782	0.860
OuterAdapter w $\mathcal{H}$-divergence	0.829	0.912
OuterAdapter w $\mathcal{C}$-divergence	0.839	0.922
OuterAdapter	**0.848**	**0.933**

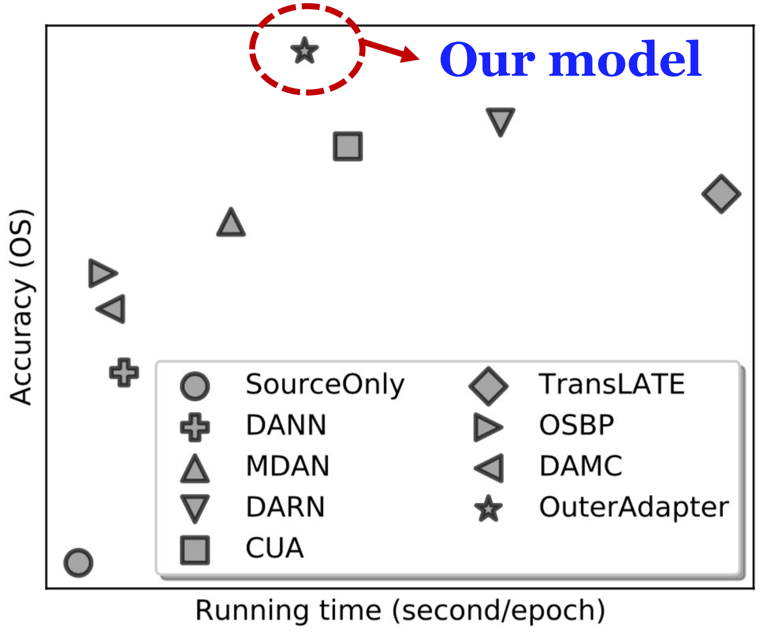
Both **$\mathcal{OS}$-divergence** and **historical target data** improve knowledge transfer in the dynamic setting

(b) Hyper-parameter sensitivity



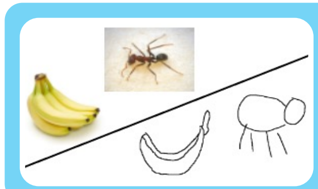
$\lambda = 1 - \pi_{C+1}^{t_{N+1}}$ affects the trade-off of correctly classifying the data within shared classes and identifying "unknown" data

(c) Efficiency



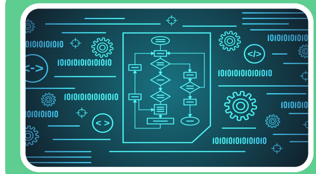
OuterAdapter achieves better model performance with **less computational complexity**

Roadmap

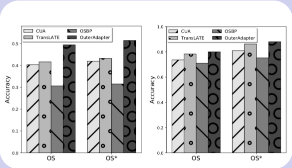


Background
- Standard domain adaptation
- Limitations in real scenarios

Problem Definition
- Dynamic open-set domain adaptation
- Unique challenges



Proposed Model
- Theoretical analysis
- Algorithm: PU learning based adaptation

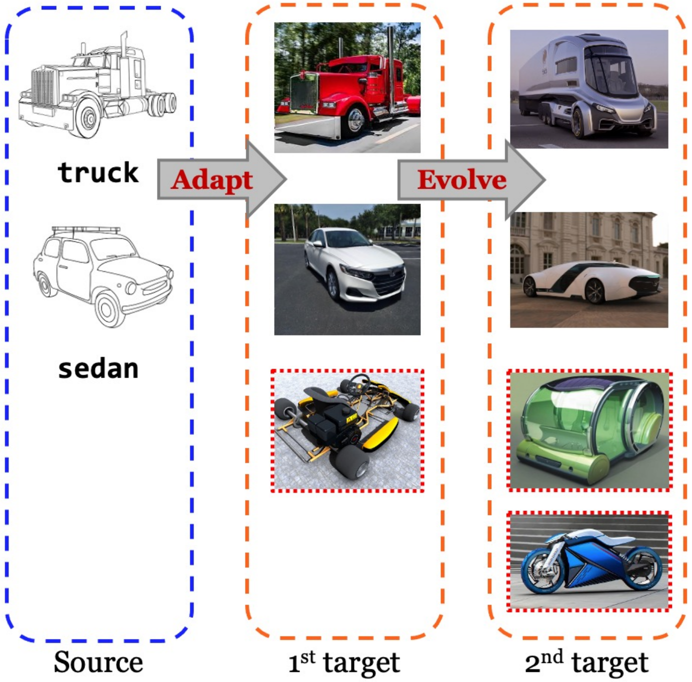


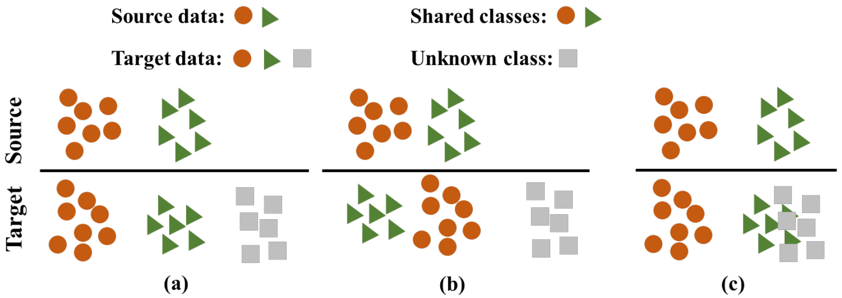
Experiments
- Effectiveness
- Efficiency

Conclusion
- Problem, analysis, algorithm, result

Conclusion

- **Problem**: Dynamic open-set domain adaptation
  - Evolving distribution and varying class proportion
- **Analysis**: Generalization error bound
  - Error upper bound based on PU learning
- **Algorithm**: PU learning based adaptation
  - $\mathcal{OS}$-divergence
  - Open-set discriminative feature learning
- **Evaluation**: Effectiveness and efficiency
  - Trade-off of correctly classifying the data within shared classes and identifying "unknown" data





THANK YOU!
&
QUESTIONS?

Contact: Jun Wu (Email: junwu3@illinois.edu)