DINE: Domain Adaptation from Single and Multiple Black-box Predictors

Jian Liang[1], Dapeng Hu[2], Jiashi Feng[3], and Ran He[1]

[1] Institute of Automation, CAS
[2] National University of Singapore
[3] ByteDance

June 2022

Outline

① *Background*

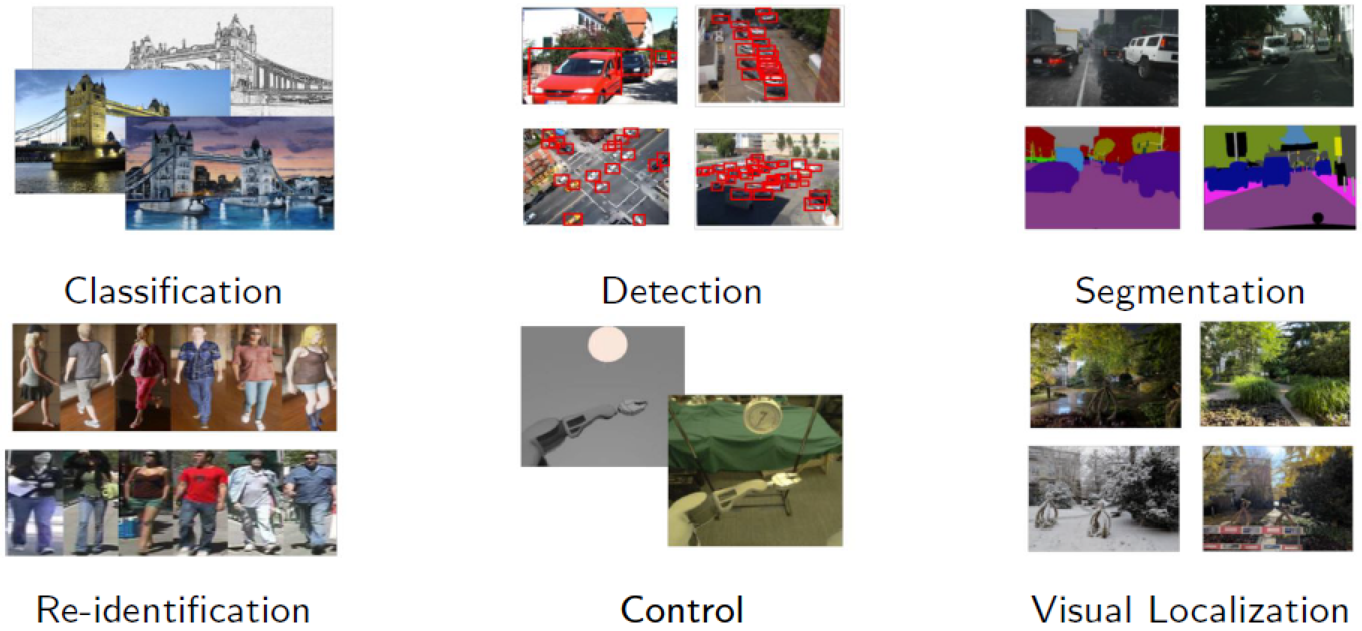
② *Problem Setting*

③ *Proposed Method*

④ *Experiments*

Background

- Unsupervised Domain Adaptation (DA)
  - **Source Domain** $\mathcal{D}_s$: n_s labeled samples $\{x_s^i, y_s^i\}_{i=1}^{n_s}$ from $P_S(X,Y)$;
  - **Target Domain** $\mathcal{D}_t$: n_t unlabeled samples $\{x_t^i, ?\}_{i=1}^{n_t}$ from $P_T(X,Y)$;
  - **Goal**: Use $\{x_t^i\}_{i=1}^{n_t}$ during training (transductive) and learn a good classifier to get the values of ? under **domain shift** (i.e., $P_S \neq P_T$).

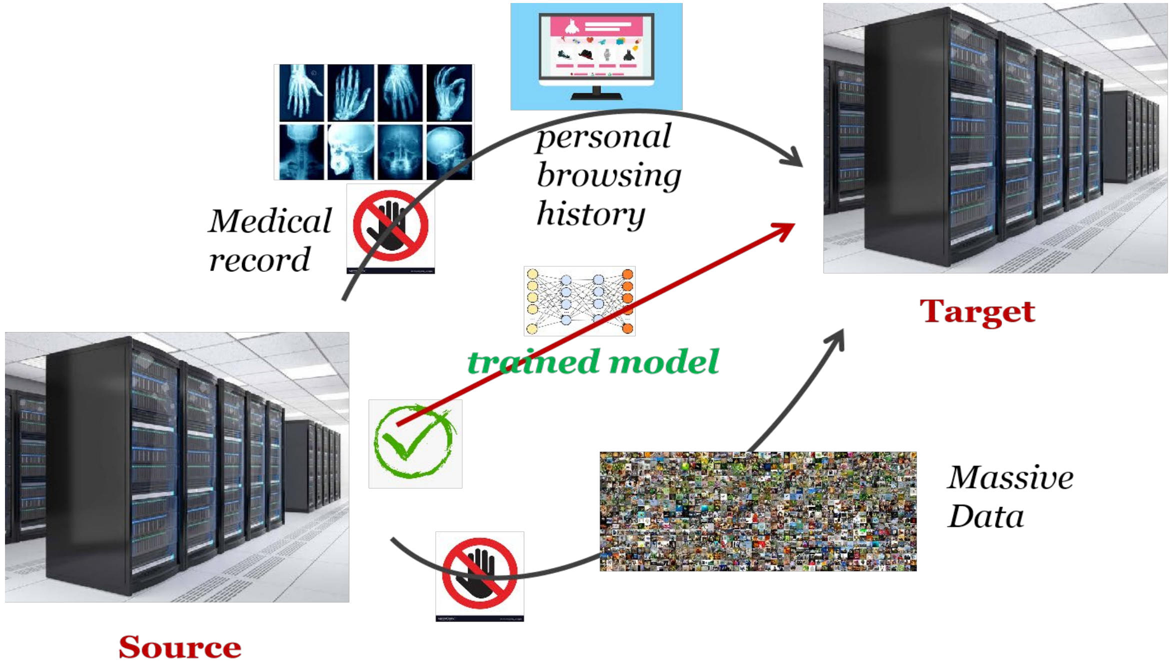


Credit to Gabriela Csurka, TaskCV-2019 talk.

weather/ light/ place …

Background

- Limitations of DA methods
  - **Not Secure**: the full access to source data is required.
  - **Concentrated**: processing different domains in the same machine.



Credit to SHOT [ICML-2020].

- Source data-Free Domain Adaptation (SFDA)

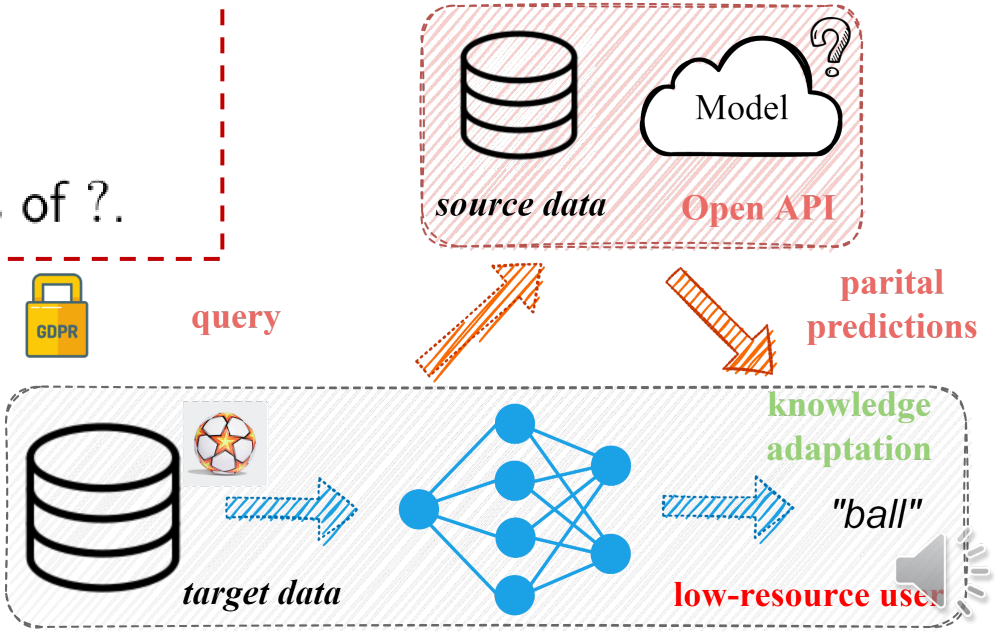
- **Source Model** $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ trained on $\mathcal{D}_s$;
- **Target Domain** $\mathcal{D}_t$: n_t unlabeled samples $\{x_t^i, ?\}_{i=1}^{n_t}$;
- **Goal**: learn a good classifier $f_t : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ to get the values of $?$.

Problem Setting

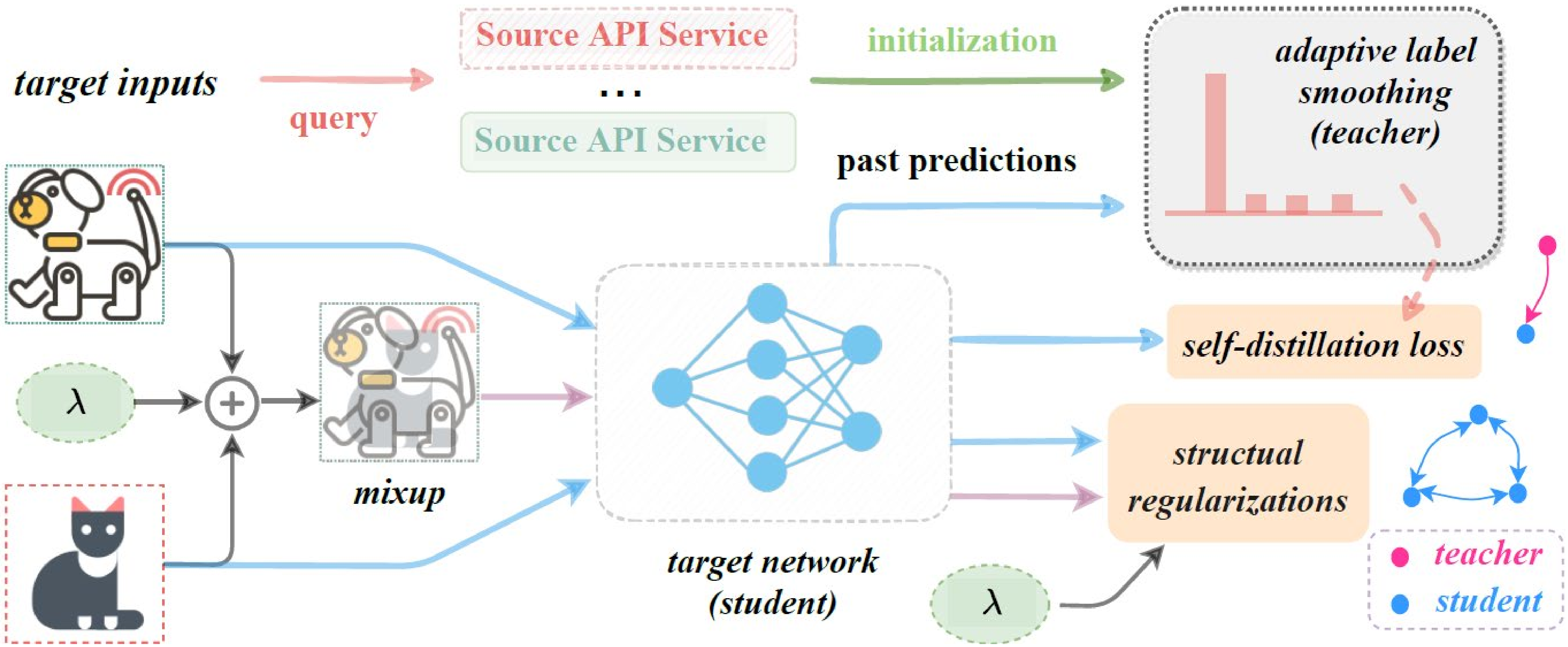
As a **white-box** model may still leak the raw source data by **model inversion attacks**, we assume the source model to be **black-box** (*i.e.*, only its predictions for target data are available).

- Black-Box Source data-Free Domain Adaptation (BB-SFDA)

- **Black-box Source Model** $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ trained on $\mathcal{D}_s$, where only its predictions for target data are available;
- **Target Domain** $\mathcal{D}_t$: n_t unlabeled samples $\{x_t^i, ?\}_{i=1}^{n_t}$;
- **Goal**: learn a good classifier $f_t : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ to get the values of ?.



Proposed Method (DIstill and fine-tuNE, DINE)



An overview of the proposed DINE framework.

Proposed Method (Distill and Fine-tune, DINE)

A. Adaptive Self-Knowledge Distillation

$$AdaLS(p,r) = \begin{cases} p_i, & i \in \mathrm{T}_p^r \\ \dfrac{1 - \sum_{j \in \mathrm{T}_p^r} p_j}{K - r}, & otherwise. \end{cases}$$

T_p^r denotes the index set of top-r classes in source predictions, and K denotes the size of p.

Initialization

$$P^T(x_t) \leftarrow \gamma P^T(x_t) + (1-\gamma) f_t(x_t) \longrightarrow \mathcal{D}_{kl}(P^T(x_t) || f_t(x_t))$$

B. Distillation with Structural Regularizations

$$\mathcal{L}_{mix} = l_{ce}(Mix_\lambda(f_t'(x_i^t), f_t'(x_j^t)), f_t(Mix_\lambda(x_i^t, x_j^t)))$$

← pairwise structural distillation

$$Mix_\lambda(a,b) = \lambda a + (1-\lambda) b$$

$$\mathcal{L}_{im} = \hbar(\mathbb{E}_{x_t \in \mathcal{X}_t} f_t(x_t)) - \mathbb{E}_{x_t \in \mathcal{X}_t} \hbar(f_t(x_t))$$

← global (batch-level) structural distillation

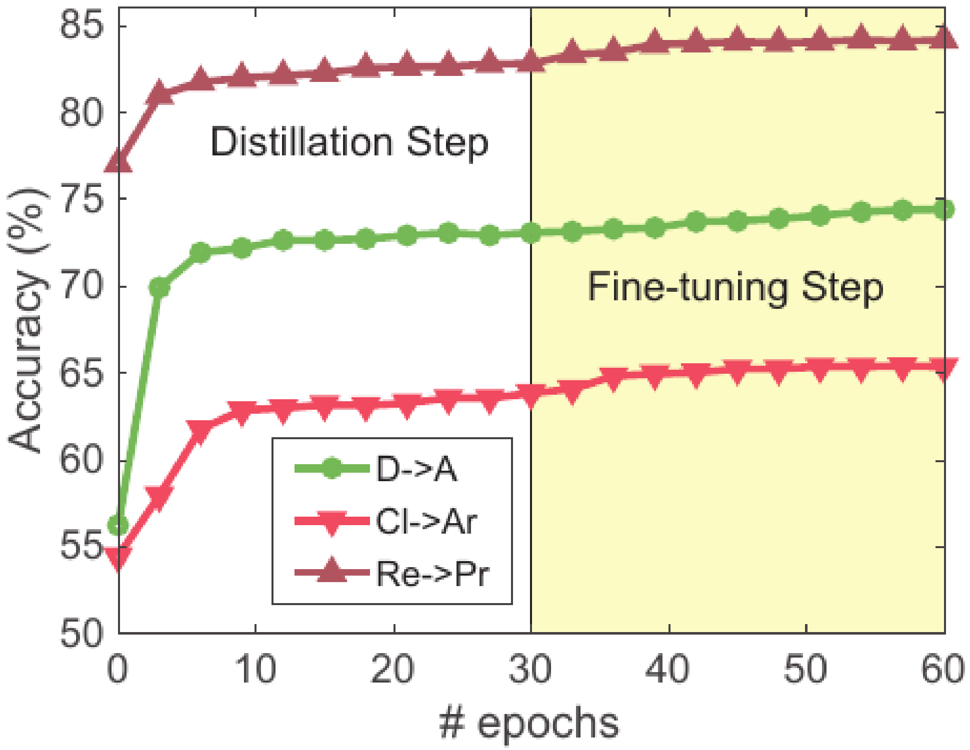
$$\hbar(p) = -\Sigma_i\, p_i log p_i$$

Experiments

Accuracies (%) on **Office-Home** [76] for single-source closed-set UDA.

Method	Type	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
No Adapt.	Pred.	44.1	66.9	74.2	54.5	63.3	66.1	52.8	41.2	73.2	66.1	46.7	77.5	60.6
NLL-OT [2]	Pred.	49.1	71.7	77.3	60.2	68.7	73.1	57.0	46.5	76.8	67.1	52.3	79.5	64.9
NLL-KL [85]	Pred.	49.0	71.5	77.1	59.0	68.7	72.9	56.4	46.9	76.6	66.2	52.3	79.1	64.6
HD-SHOT [44]	Pred.	48.6	72.8	77.0	60.7	70.0	73.2	56.6	47.0	76.7	67.5	52.6	80.2	65.3
SD-SHOT [44]	Pred.	50.1	75.0	78.8	63.2	72.9	76.4	60.0	48.0	79.4	69.2	54.2	81.6	67.4
DINE	Pred.	52.2	78.4	81.3	65.3	76.6	78.7	62.7	49.6	82.2	69.8	55.8	84.2	69.7
DINE (full)	Pred.	54.2	77.9	81.6	65.9	77.7	79.9	64.1	50.5	82.1	71.1	58.0	84.3	70.6
ResNet-50↑, **ViT**↓ (source backbone) → **ResNet-50** (target backbone)														
No Adapt.	Pred.	54.5	83.2	87.2	78.0	83.8	86.1	74.5	49.7	87.4	78.6	52.6	86.2	75.1
NLL-OT [2]	Pred.	58.8	84.4	87.6	78.2	84.7	86.7	76.0	54.0	88.0	79.7	57.2	87.2	76.9
NLL-KL [85]	Pred.	59.5	84.3	87.6	77.4	84.8	86.8	75.1	54.9	88.0	79.0	57.9	87.2	76.9
HD-SHOT [44]	Pred.	57.2	84.2	87.3	78.4	84.9	86.4	74.8	56.0	87.6	78.9	57.5	87.0	76.7
SD-SHOT [44]	Pred.	59.4	85.2	87.8	79.6	86.6	87.1	76.4	58.3	87.8	80.0	59.5	87.9	78.0
DINE	Pred.	**64.9**	87.4	88.8	80.5	**89.6**	87.8	79.0	**62.9**	89.1	81.5	64.6	**90.0**	80.5
DINE (full)	Pred.	64.4	**87.9**	**89.0**	**80.9**	**89.6**	**88.7**	**79.6**	62.5	**89.4**	**81.7**	**65.2**	89.7	**80.7**
SHOT [46]	Mod.	57.7	79.1	81.5	67.6	77.9	77.8	68.1	55.8	82.0	72.8	59.7	84.4	72.0
A^2Net [78]	Mod.	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
SHOT++ [46]	Mod.	58.1	79.5	82.4	68.6	79.9	79.3	68.6	57.2	83.0	74.3	60.4	85.1	73.0
TransDA* [80]	Mod.	67.5	83.3	85.9	74.0	83.8	84.4	77.0	68.0	87.0	80.5	69.9	90.0	79.3
$\text{RADA}_{\text{CDAN}}$ [29]	Data	56.5	76.5	79.5	68.8	76.9	78.1	66.7	54.1	81.0	75.1	58.2	85.1	71.4
ATDOC-NA [45]	Data	58.3	78.8	82.3	69.4	78.2	78.2	67.1	56.0	82.7	72.0	58.2	85.5	72.2
$\text{SCDA}_{\text{DCAN}}$ [40]	Data	60.7	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1

Experiments



accuracy convergence

Method	FT	D → A	Cl → Ar	Re → Pr	Avg.
DINE (w/ Hard)	×	63.6±2.1	60.9±0.4	80.2±0.6	68.2
	✓	67.3±1.7	62.3±0.2	81.7±0.4	70.4
DINE (w/ LS)	×	64.8±1.9	61.7±0.4	81.0±0.5	69.2
	✓	68.4±1.8	63.0±0.5	82.6±0.3	71.3
DINE ($r=1$)	×	70.6±2.0	64.1±0.3	83.7±0.4	72.8
	✓	72.2±1.8	65.3±0.2	84.7±0.1	74.1
DINE ($r=3$)	×	71.4±2.5	64.4±0.5	83.7±0.3	73.1
	✓	**72.9**±2.0	65.6±0.5	**84.7**±0.3	**74.4**
DINE ($r=K$)	×	71.3±2.7	64.6±0.4	83.3±0.4	73.1
	✓	**72.9**±2.5	**65.9**±0.7	84.3±0.3	74.3

study on the effectiveness of AdaLS.

Summary

1. We study **a realistic and challenging UDA problem** and propose a new adaptation framework (DINE) with only black-box predictors provided from source domains.

2. We propose **an adaptive label smoothing strategy** and a structural distillation method by first introducing **structural regularizations** into unsupervised distillation.

3. Empirical results on various benchmarks validate the superiority of the DINE framework over baselines. Provided with large source predictors like ViT, DINE even yields **state-of-the-art performance** for single-source, multi-source, and partial-set UDA.

Thanks for listening!

- If you require any further information, feel free to contact me.

  Email: liangjian92@gmail.com

- Code is available at https://github.com/tim-learn/DINE/.

paper

code