Time Series Domain Adaptation

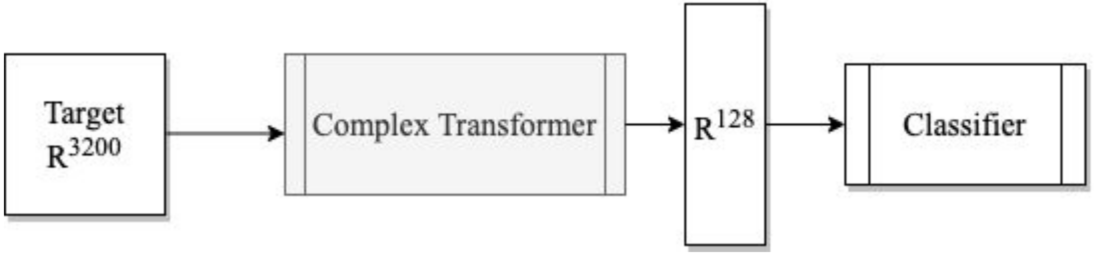
May 15, 2020

Baseline

$$L_{class\_target} = -\sum_{(x'_i, y'_i) \in D_t^{label}} y'_i log(C(T(x'_i)))$$

Only use 70% target-domain data for training, and evaluate at the rest 30%

Accuracy: 32%



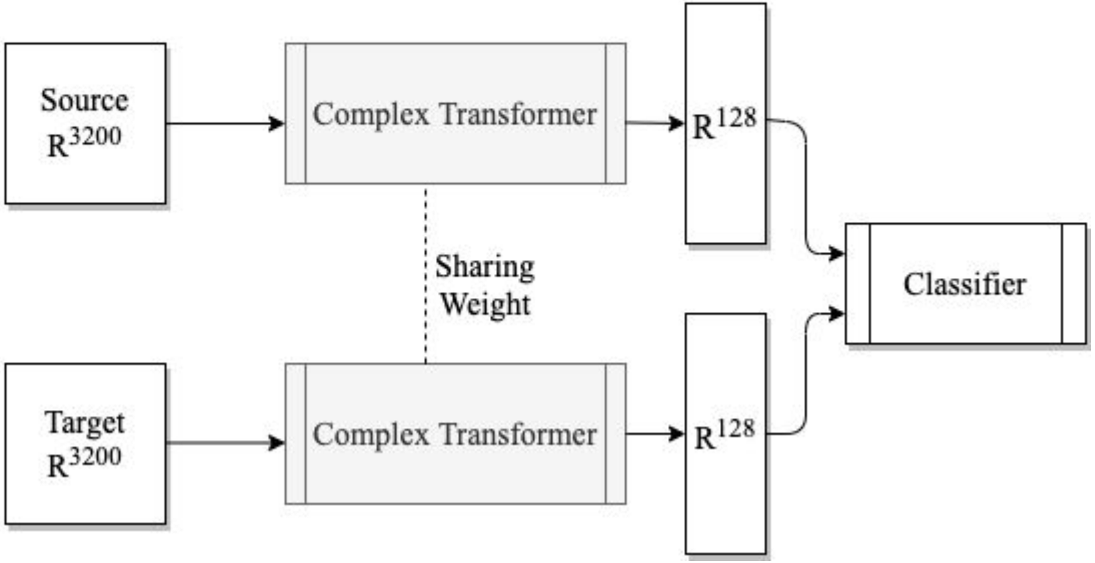
$$L_{class\_target} = - \sum_{(x'_i, y'_i) \in D_t^{label}} y'_i log(C(T(x'_i)))$$

Naive Adaptation

$$L_{class\_source} = - \sum_{(x_i, y_i) \in D_s} y_i log(C(T(x_i)))$$

Use both target-domain (70% training and 30% testing) and source-domain

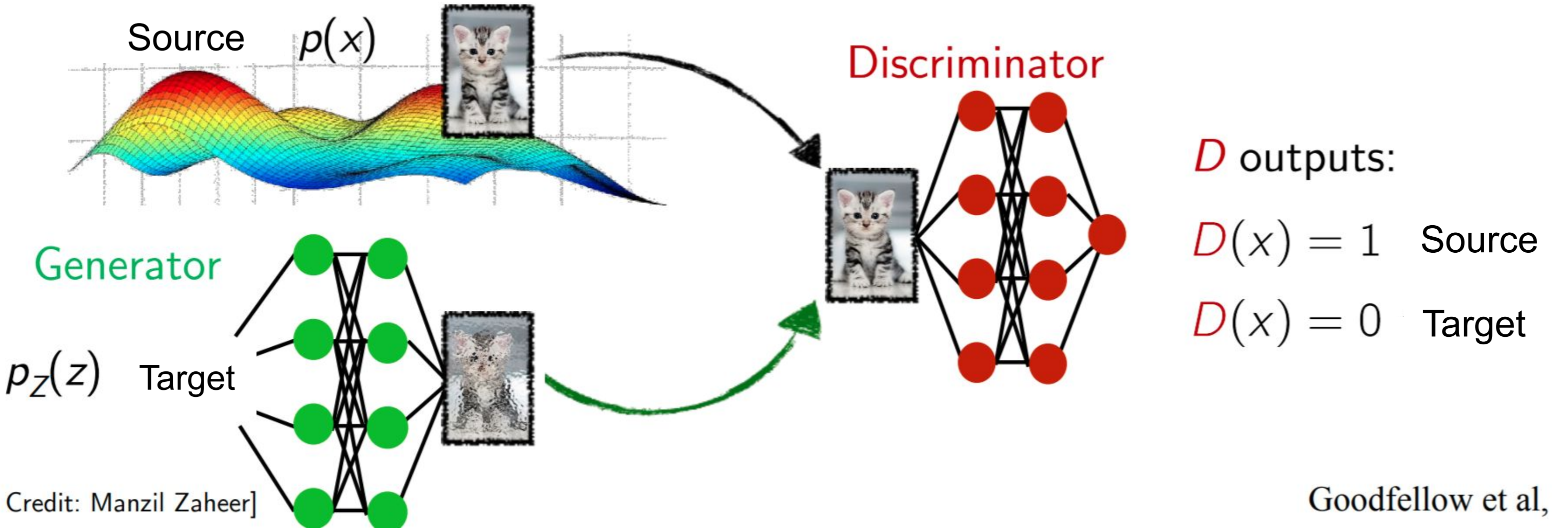
Accuracy on 30% target-domain: 37%



GAN Background

- Min-Max game for adapting target distribution to source distribution

$$\min_{G} \max_{D} \mathbb{E}_{x\sim p}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$



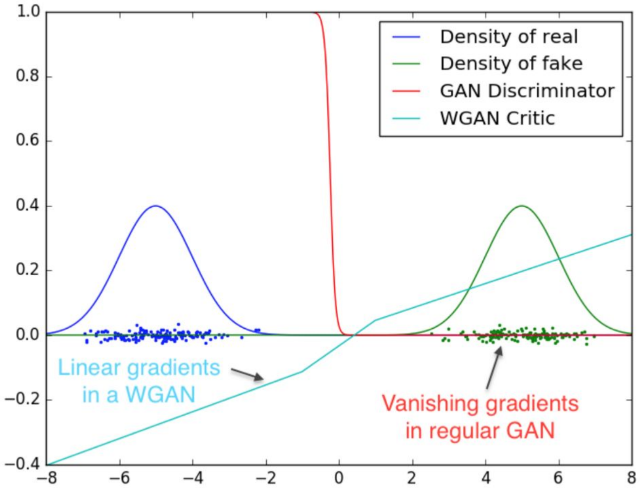
[Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Background

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

- Wasserstein GAN
  - Wasserstein Distance is a measure of the distance between two probability distributions
  - Avoid vanishing gradients in normal GAN
  - State-of-the-art training for GAN

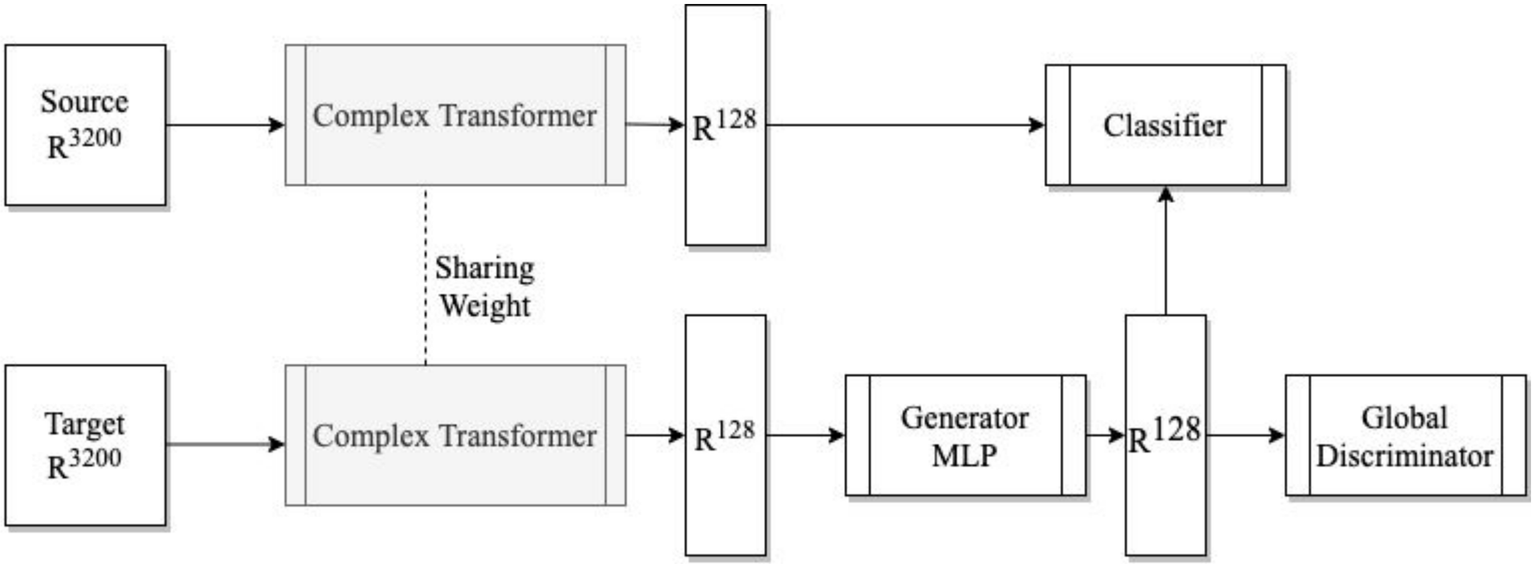


Martin Arjovsky et. al (2017)

Global GAN

$$L_{global\_GAN} = \mathbb{E}_{x' \sim P_t}[D(G(T(x')))] - \mathbb{E}_{x \sim P_s}[D(T(x))] + \lambda_{gp} * \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$$

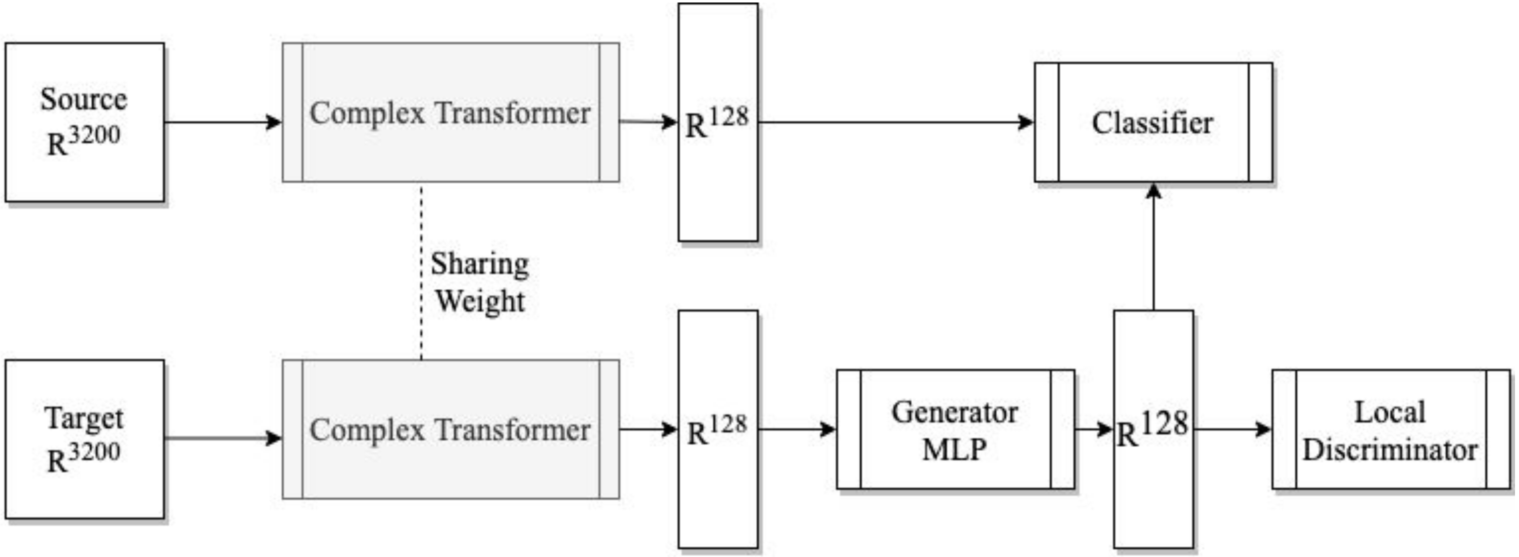
Minimize marginal distribution of source-domain and target-domain
Acc on rest 30% unlabeled target data: 40%



Conditional GAN

$$L_{conditional\_GAN} = \mathbb{E}_{y \sim P_y}[\mathbb{E}_{x' \sim P_t(x|y)}[D(G(T(x'), y))] - \mathbb{E}_{x \sim P_s(x|y)}[D(T(x))] + \lambda_{gp} * \mathbb{E}_{\hat{x} \sim P_{\hat{x}}(x|y)}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]]$$

Minimize conditional distribution of source-domain and target-domain
Acc on rest 30% unlabeled target data: 44%



KL-divergence analysis

- Use MLP to approximate g.
- Train g by using the dual form of Jensen-shannon divergence
- Report on Kullback–Leibler divergence using the trained g

$$JS(P_s||P_t) = \sup_g \mathbb{E}_{P_s}[-softplus(-g(x))] - \mathbb{E}_{P_t}[softplus(g(x))]$$

$$g^* = log\frac{p_s(x)}{p_t(x)}$$

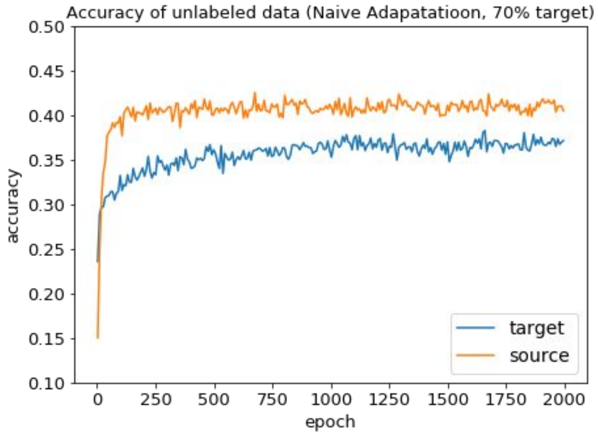
$$KL(P_s||P_t) = \mathbb{E}_{P_s} log\frac{p_s(x)}{p_t(x)}$$

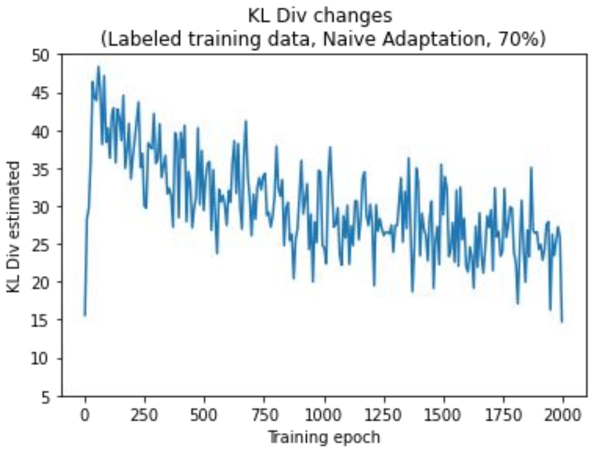
KL-divergence analysis

$$KL(P_s||P_t) = \mathbb{E}_{P_s} log \frac{p_s(x)}{p_t(x)}$$

Naive Adaptation

- Target unlabeled accuracy: 32% -> 37%
- Shared-weight transformer and classifier
- Labeled KL divergence decrease to about 20.



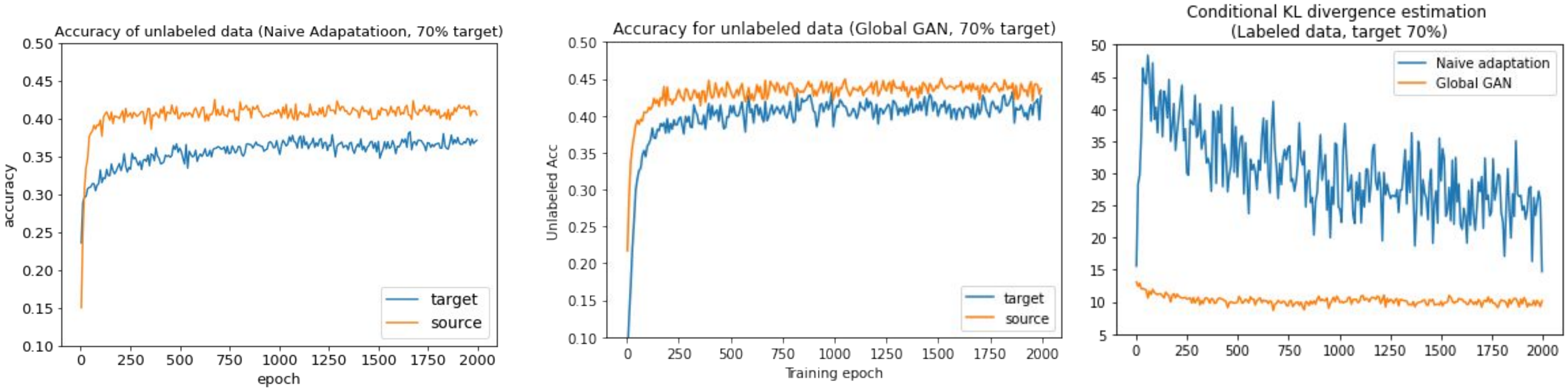


KL-divergence analysis

$$KL(P_s||P_t) = \mathbb{E}_{P_s} log \frac{p_s(x)}{p_t(x)}$$

Global GAN

- Slightly improve unlabeled target accuracy (37% -> 41%)
- Slightly push target data hidden representation to source data hidden representation
- Push target and source testing accuracy together
- Labeled KL divergence decrease to about 10.

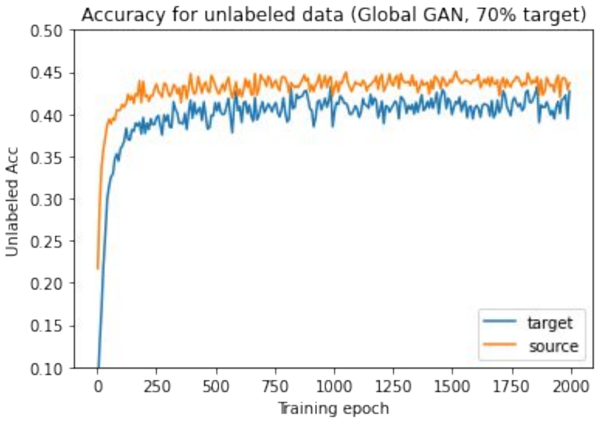


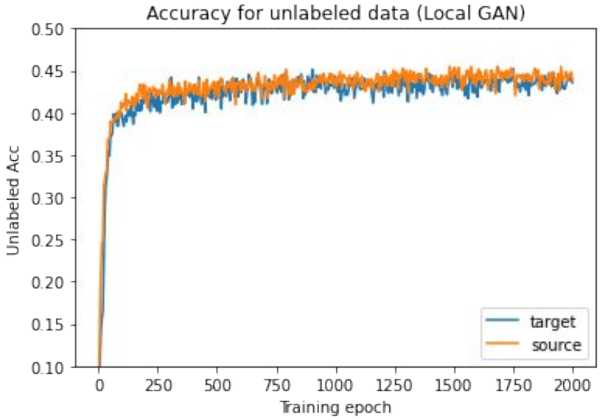
KL-divergence analysis

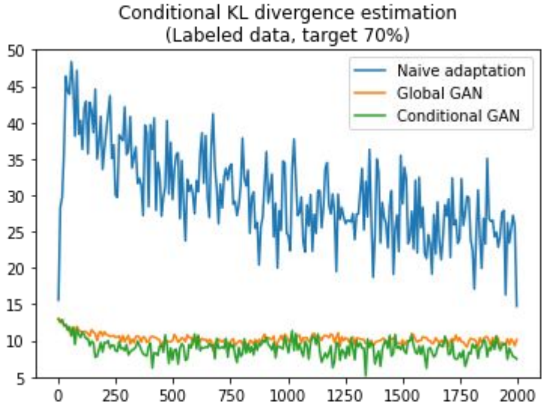
$$KL(P_s||P_t) = \mathbb{E}_{P_s} log \frac{p_s(x)}{p_t(x)}$$

Conditional GAN

- Improve unlabeled target accuracy (41% -> 44%)
- Push target data hidden representation to source data hidden representation
- Push target and source testing accuracy together
- Labeled KL divergence decreases to about 9.

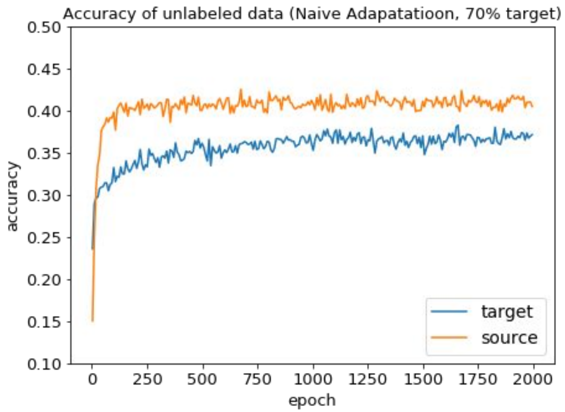


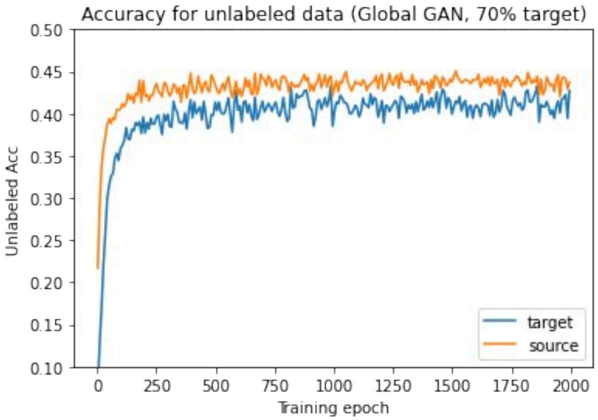


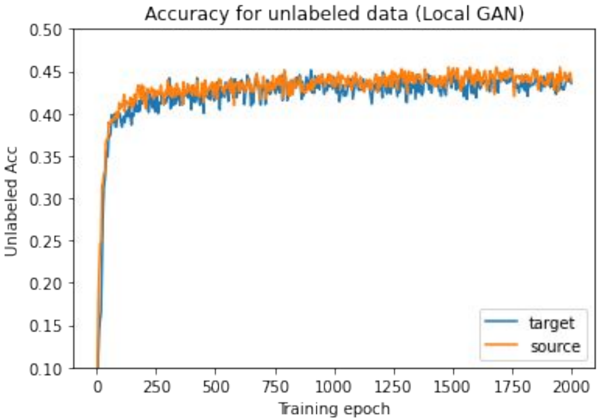


Testing accuracy analysis

- Native adaptation
  - Gap between target-domain and source-domain, because source-domain has more data than target-domain
- Global GAN
  - Shrinks the accuracy gap between domains
- Conditional (Local) GAN
  - Shrinks the accuracy gap even further







Testing accuracy summary

Target label seen during training	Baseline Test Accuracy	Naive adaptation Test Accuracy	Global GAN Test Accuracy	Conditional GAN Test Accuracy
10%	15%	13%	**22%**	21%
30%	22%	22%	28%	**33%**
50%	28%	28%	35%	**38%**
70%	32%	37%	41%	**44%**