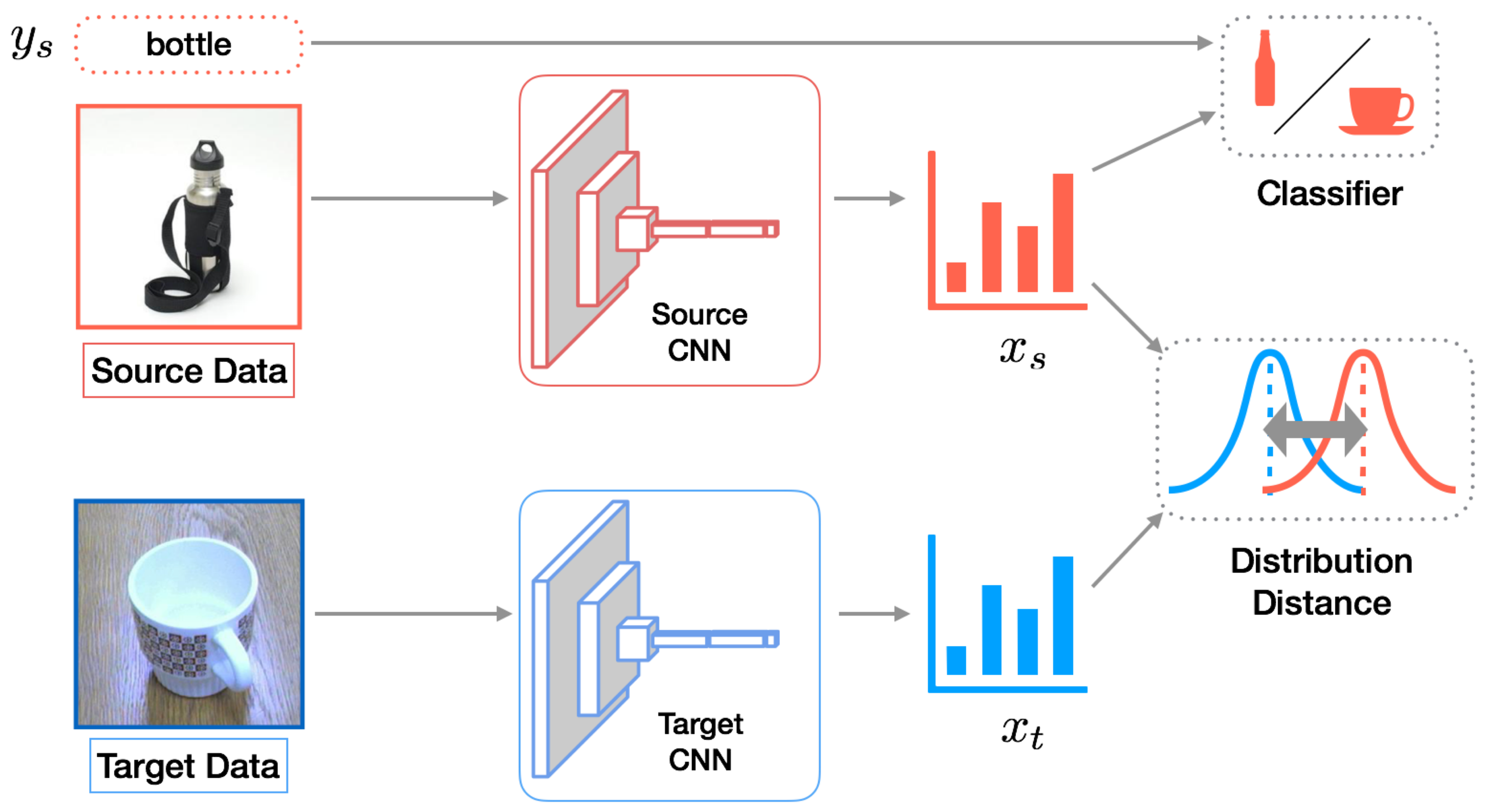
Adapting to Continuously Shifting Domains

Andreea Bobu[1], Eric Tzeng[1], Judy Hoffman[1], Trevor Darrell[1]
[1]Berkeley Artificial Intelligence Research, University of California Berkeley

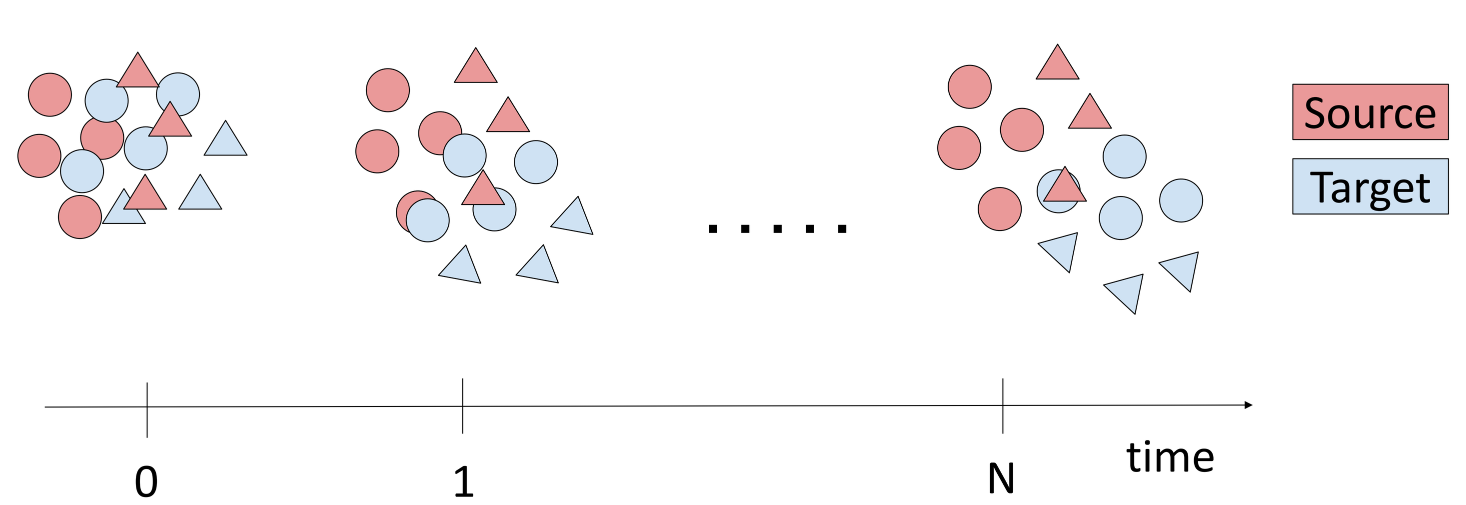
Background: Domain Adaptation



Standard domain adaptation
- Source domain S, with images X_s and labels Y_s drawn from $p_S(x,y)$
- Target domain T, with images X_T unlabeled drawn from $p_T(x,y)$
- Minimize discrepancy between distributions by learning source and target representations, M_s and M_t, and classifier C

$$M_s, C \leftarrow \arg\min_{M_s, C} \overbrace{L_{cls}(C(M_s(X_s)), Y_s)}^{\text{Cross-Entropy loss}}$$
$$M_t \leftarrow \arg\min_{M_t} \underbrace{d(M_s(X_s), M_t(X_t))}_{\text{Distribution distance}}$$

Problem Statement



Problem
- Adapting from one single source to one single target is limiting
- Incremental adaptation alone cannot recover good performance on past domains
- **Goal**: adapt between continuously shifting domains, while avoiding catastrophic forgetting

Continuous Unsupervised Adaptation

Continuous domain adaptation
- Source domain S, with images X_s and labels Y_s drawn from $p_s(x,y)$
- Target domains T_i, with images X_{ti} unlabeled drawn from $p_{ti}(x,y)$
- Source domain is similar to the target domain at time t_0
- Target domain is smoothly varying
- p_{t0} is more similar to p_s than p_{t1} is to p_s

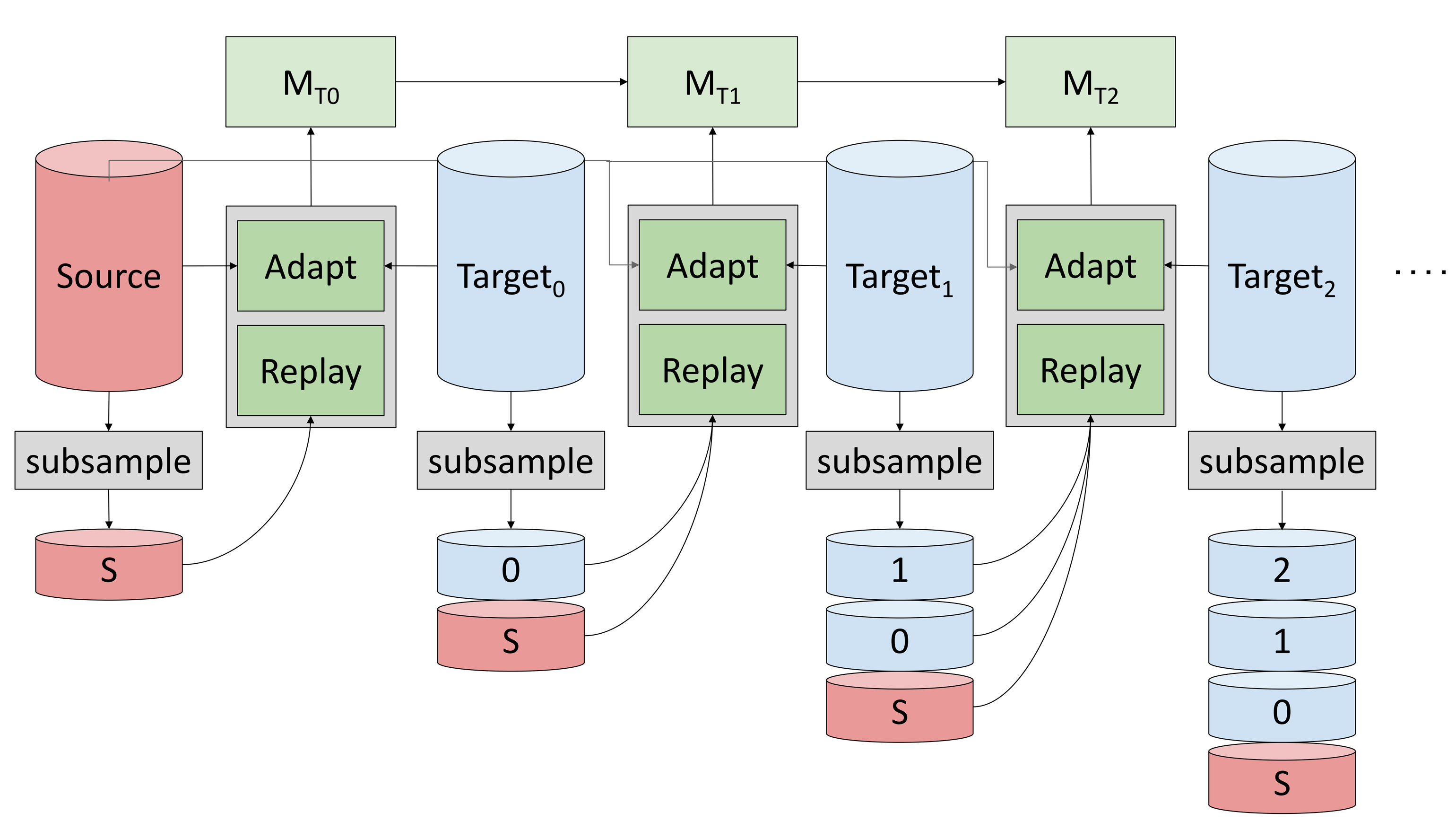
Solution: incrementally adapt to new domains with a replay loss to maintain performance on past domains

CUA Framework

Iterative adaptation with subsampling
- Initialize M_s like in the standard domain adaptation setting
- Subsample α-fraction dataset $\{X_p, Y_p\}$ at every stage
- Optimize the sequential unsupervised adaptation update while matching sampled past data with the replay loss

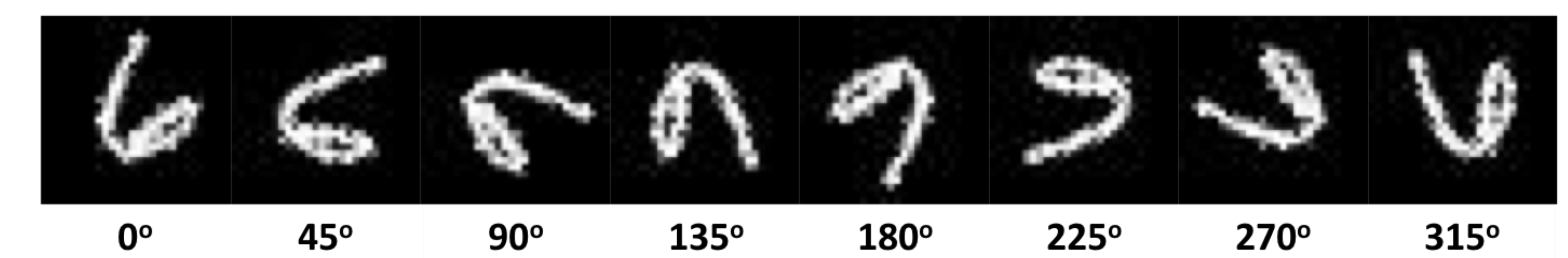
$$M \leftarrow M_{t_{i-1}}; \{X_p, Y_p\} \leftarrow subsample(\{X_s, Y_s\}, \alpha)$$

$$M_t \leftarrow \arg\min_M \{d\left(M_s(X_s), M(X_{t_i})\right) + \lambda L_{replay}\left(C\left(M(X_p)\right), Y_p\right)\}$$



MNIST Results

- 60000 training images of handwritten digits, 10000 test images
- Continuous shift represented by 45° rotations
- Source domain 0°, Target domains every 45° after

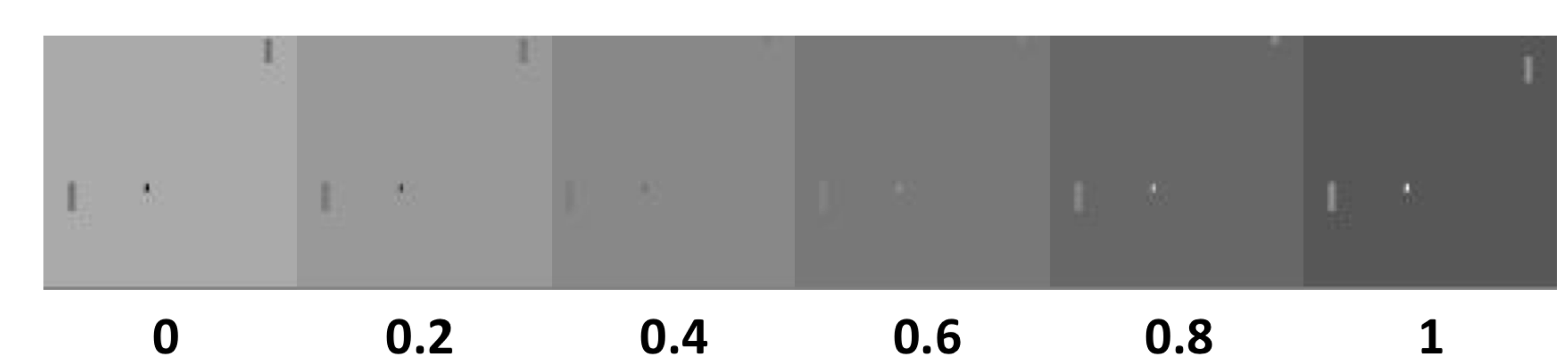


Method	0°	45°	90°	135°	180°	225°	270°	315°	Average (%)
Source	99.2	61.7	17.2	29.1	39.4	29.8	15.8	51.7	43.0 ± 0.8
ADDA	80.8	70.4	20.8	28.6	42.1	40.2	23.8	41.2	43.5 ± 1.2
DANN	98.6	64.7	19.9	28.4	41.4	32.9	24.2	67.3	47.2 ± 1.6
CUA - no replay (Ours)	51.6	15.1	32.7	38.7	30.4	27.1	73.6	96.0	45.7 ± 1.4
CUA (Ours)	**90.4**	**84.4**	**82.0**	**77.3**	**85.8**	**88.2**	**92.7**	**96.5**	**90.4 ± 1.6**
Target Supervised (Oracle)	96.9	96.7	96.8	97.4	96.6	96.5	96.8	96.4	97.0

Atari Results

- Atari game pong with base model ACKTR
- Continuous shift represented by color inversion $\theta \in [0,1]$, where every pixel x_{orig} is inverted into x_{inv}

$$x_{inv} = (1-\theta) * x_{orig} + \theta * (1 - x_{orig})$$



Method	Inversion factor θ 0.0	0.1	0.2	0.3
Source only	21.0	21.0	17.6	−2.28
MMD (Long & Wang, 2015)	21.0	21.0	17.0	15.9
CUA (Ours)	21.0	21.0	21.0	21.0
Target with reward (Oracle)	21.0	21.0	21.0	21.0