DISGUISE ADVERSARIAL NETWORKS

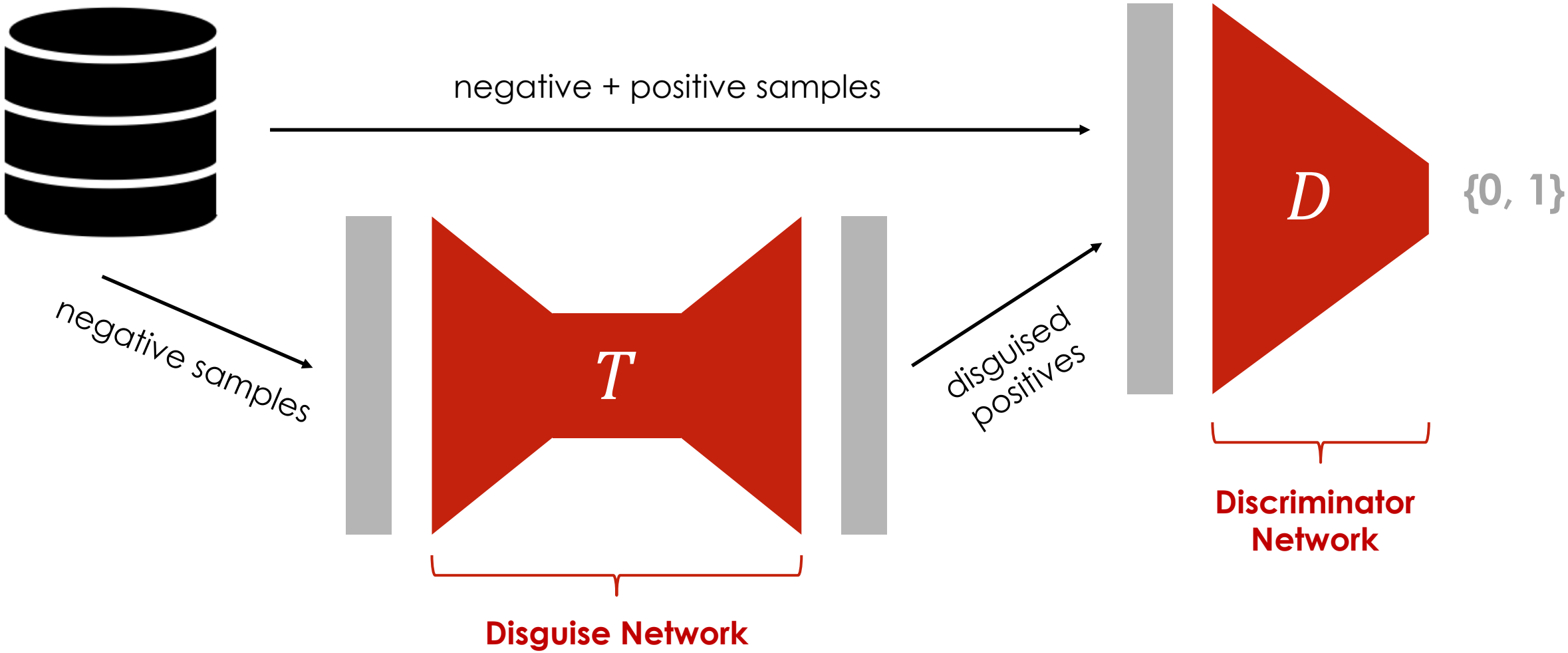
PAPER SUMMARY

10/2/2018

MOTIVATION

- **Problem**: Classifiers perform poorly on severely imbalanced datasets
  - Too many negative samples
  - Very few positive samples
- **Solution**: Disguise negative samples as positive samples
- Conceptually like a **fancy non-linear SMOTE**, but sampling from and transforming negative examples instead

GENERAL ARCHITECTURE



TRAINING THE DISGUISE NETWORK

Want $D(T(x^-)) = 1$ Want $T(x^-) \approx x^-$

$$\text{minimize } \mathcal{L}_1(T) = -\mathbb{E}_{x^- \sim P_{\Omega^-}}\left[\log D\left(T(x^-)\right)\right] + \lambda \|T(x^-) - x^-\|_1$$

Disguise Quality
assesses how well T transforms negative samples to look like positive samples

Regularizer
restricts T from making too drastic of transformations to negative samples

TRAINING THE DISCRIMINATOR NETWORK
(FIRST ATTEMPT)

Let $T(x^-) \in \Omega^+$

$$\text{minimize } \mathcal{L}_2(D) = -\mathbb{E}_{x^- \sim P_{\Omega^-}}\left[\log\left(1 - D(x^-)\right)\right] \quad \text{Want } D(x^-) = 0$$
$$-\mathbb{E}_{x^+ \sim P_{\Omega^+}}\left[\log\left(D(x^+)\right)\right] \quad \text{Want } D(x^+) = 1$$

- **Problem**: T may not always successfully disguise x^-
- **Problem**: Forcing a badly disguised x^- to be positive will confuse D

TRAINING THE DISCRIMINATOR NETWORK
(SECOND ATTEMPT)

- **Solution**: Don't try to enforce labels on $T(x^-)$
- Just ask D to be confident about its classifications for $T(x^-)$

$$\text{minimize } \mathcal{L}_2(D) = -\mathbb{E}_{x^- \sim P_{\Omega^-}}\left[\log\left(1 - D(x^-)\right)\right] \quad \text{Want } D(x^-) = 0$$

$$-\mathbb{E}_{x^+ \sim P_{\Omega^+}}\left[\log\left(D(x^+)\right)\right] \quad \text{Want } D(x^+) = 1$$

$$+\eta\mathbb{E}_{x^- \sim P_{\Omega^-}}\left[H\left(D\left(T(x^-)\right)\right)\right] \quad \text{Want } D(T(x^-)) \approx 0 \text{ or } 1 \text{ but not } 0.5$$

where $H(\hat{y}) = \underbrace{-[\hat{y}\log(\hat{y}) + (1-\hat{y})\log(1-\hat{y})]}$

Entropy of Predictive Distribution
measures the degree of uncertainty in D's predictions on disguised samples

EVALUATING THE NETWORK

Data: 2 CTR datasets

Datasets	Total logs	Period	CTR	Dim
Display	46 million	7 days	0.26	228
Mobile	40 million	10 days	0.17	100

← **After one-hot encoding**

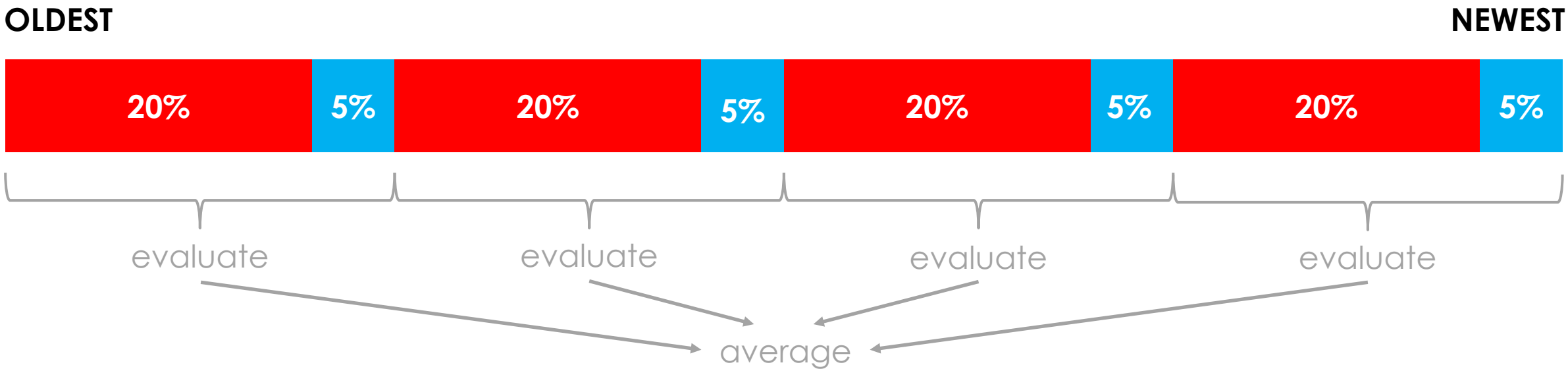
In comparison, MediaMath data is:

- **Smaller**: fewer logs
- **Longer**: over a wider period of time
- **More imbalanced**: smaller proportion of positive examples

EVALUATING THE NETWORK

Problem: Some ads are relevant for only a small window of time

Solution: Sort data chronologically and divide into four batches of **train** and **test** splits



DID IT WORK?

proportion of samples predicted as positive

precision

		Display Ads			Mobile Ads		
		Rd-frequency	Rd-CTR	AUC	Rd-frequency	Rd-CTR	AUC
Supervised	DNN	0.07	0.53	0.69	0.02	0.51	0.67
	SVM	0.07	0.50	0.62	0.03	0.47	0.63
Imbalance	Centroid	0.08	0.49	0.66	0.06	0.51	0.64
	ADASYN	0.08	0.62	0.70	0.04	0.55	0.66
	SMOTE	0.09	0.57	0.67	0.05	0.53	0.65
GAN	Modified-GAN	0.09	0.57	0.69	0.04	0.52	0.63
	SGAN	0.09	0.61	0.68	0.04	0.51	0.67
DAN	TL-DAN	0.04	0.68	0.69	0.01	0.58	0.64
	Noise-DAN	0.08	0.62	0.71	0.02	0.53	0.69
	DAN	0.11	0.66	0.75	0.05	0.57	0.73

QUESTIONS?