Generative Adversarial Network

俞鼎耀

ICLR 2018

SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
koyama.masanori@gmail.com
yyoshida@nii.ac.jp
[1]Preferred Networks, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

Motivation

One of the challenges in the study of generative adversarial networks is the instability of its training. In high dimensional spaces, the density ratio estimation by the discriminator is often inaccurate and unstable during the training, and generator networks fail to learn the multimodal structure of the target distribution.

Advantages

- Lipschitz constant is the only hyper-parameter to be tuned, and the algorithm does not require intensive tuning of the only hyper-parameter for satisfactory performance.
- Implementation is simple and the additional computational cost is small

GAN

$$f(\boldsymbol{x},\theta) = W^{L+1}a_L(W^L(a_{L-1}(W^{L-1}(\ldots a_1(W^1\boldsymbol{x})\ldots)))), \tag{1}$$

$$W^l \in \mathbb{R}^{d_l \times d_{l-1}},\ W^{L+1} \in \mathbb{R}^{1\times d_L}$$

$$D(\boldsymbol{x},\theta) = \mathcal{A}(f(\boldsymbol{x},\theta)), \tag{2}$$

$$\min_G \max_D V(G,D) \qquad \mathrm{E}_{\boldsymbol{x}\sim q_{\text{data}}}[\log D(\boldsymbol{x})] + \mathrm{E}_{\boldsymbol{x}'\sim p_G}[\log(1 - D(\boldsymbol{x}'))]$$

If f(x) takes the form

$$f^*(\boldsymbol{x}) = \log q_{\text{data}}(\boldsymbol{x}) - \log p_G(\boldsymbol{x})$$

$$\nabla_{\boldsymbol{x}} f^*(x) = \frac{1}{q_{\text{data}}(\boldsymbol{x})}\nabla_{\boldsymbol{x}} q_{\text{data}}(\boldsymbol{x}) - \frac{1}{p_G(\boldsymbol{x})}\nabla_{\boldsymbol{x}} p_G(\boldsymbol{x})$$

Unbounded gradient

GAN

A number of works (Uehara et al., 2016; Qi, 2017; Gulrajani et al., 2017) advocate the importance of Lipschitz continuity in assuring the boundedness of statistics.

A particularly successful works in this array are (Qi, 2017; Arjovsky et al., 2017; Gulrajani et al., 2017), which proposed methods to control the Lipschitz constant of the discriminator by adding regularization terms defined on input examples ***x***.

Method

$$\underset{\|f\|_{\mathrm{Lip}}\leq K}{\arg\max}\, V(G,D)$$

$$\|f(\boldsymbol{x})-f(\boldsymbol{x}')\|/\|\boldsymbol{x}-\boldsymbol{x}'\|\leq M$$

$$\sigma(A):=\max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}}\frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2}=\max_{\|\boldsymbol{h}\|_2\leq 1}\|A\boldsymbol{h}\|_2, \qquad (6)$$

$$\begin{aligned}\|f\|_{\mathrm{Lip}}\leq&\|(\boldsymbol{h}_L\mapsto W^{L+1}\boldsymbol{h}_L)\|_{\mathrm{Lip}}\cdot\|a_L\|_{\mathrm{Lip}}\cdot\|(\boldsymbol{h}_{L-1}\mapsto W^{L}\boldsymbol{h}_{L-1})\|_{\mathrm{Lip}}\\ &\cdots\|a_1\|_{\mathrm{Lip}}\cdot\|(\boldsymbol{h}_0\mapsto W^{1}\boldsymbol{h}_0)\|_{\mathrm{Lip}}=\prod_{l=1}^{L+1}\|(\boldsymbol{h}_{l-1}\mapsto W^{l}\boldsymbol{h}_{l-1})\|_{\mathrm{Lip}}=\prod_{l=1}^{L+1}\sigma(W^l). \qquad (7)\end{aligned}$$

$$\|g_1\circ g_2\|_{\mathrm{Lip}}\leq\|g_1\|_{\mathrm{Lip}}\cdot\|g_2\|_{\mathrm{Lip}}$$

spectral normalization

$$\bar{W}_{\mathrm{SN}}(W):=W/\sigma(W).$$

If we normalize each *W l* using (8), we can appeal to the inequality (7) and the fact that $\sigma\ (\overline{W_{SN}}(W))$ = 1 to see that *f* Lip is bounded from above by 1.

Calculate $\sigma(W)$

$\sigma(W)$ is the largest single value of W

$power\ iteration\ method$

Gradient

$$\frac{\partial \bar{W}_{\mathrm{SN}}(W)}{\partial W_{ij}} = \frac{1}{\sigma(W)} E_{ij} - \frac{1}{\sigma(W)^2} \frac{\partial \sigma(W)}{\partial W_{ij}} W = \frac{1}{\sigma(W)} E_{ij} - \frac{[\boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}]_{ij}}{\sigma(W)^2} W \tag{9}$$

$$= \frac{1}{\sigma(W)} \left(E_{ij} - [\boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}]_{ij} \bar{W}_{\mathrm{SN}} \right), \tag{10}$$

where *Eij* is the matrix whose (*i; j*)-th entry is 1 and zero everywhere else, and ***u***1 and ***v***1 are respectively the first left and right singular vectors of *W*

$$\frac{\partial V(G,D)}{\partial W} = \frac{1}{\sigma(W)} \left(\hat{\mathrm{E}} \left[\boldsymbol{\delta} \boldsymbol{h}^{\mathrm{T}} \right] - \left(\hat{\mathrm{E}} \left[\boldsymbol{\delta}^{\mathrm{T}} \bar{W}_{\mathrm{SN}} \boldsymbol{h} \right] \right) \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} \right) \tag{11}$$

$$= \frac{1}{\sigma(W)} \left(\hat{\mathrm{E}} \left[\boldsymbol{\delta} \boldsymbol{h}^{\mathrm{T}} \right] - \lambda \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} \right) \tag{12}$$

where $\boldsymbol{\delta} := \left(\partial V(G,D) / \partial \left(\bar{W}_{\mathrm{SN}} \boldsymbol{h} \right) \right)^{\mathrm{T}}, \lambda := \hat{\mathrm{E}} \left[\boldsymbol{\delta}^{\mathrm{T}} \left(\bar{W}_{\mathrm{SN}} \boldsymbol{h} \right) \right]$

Other method

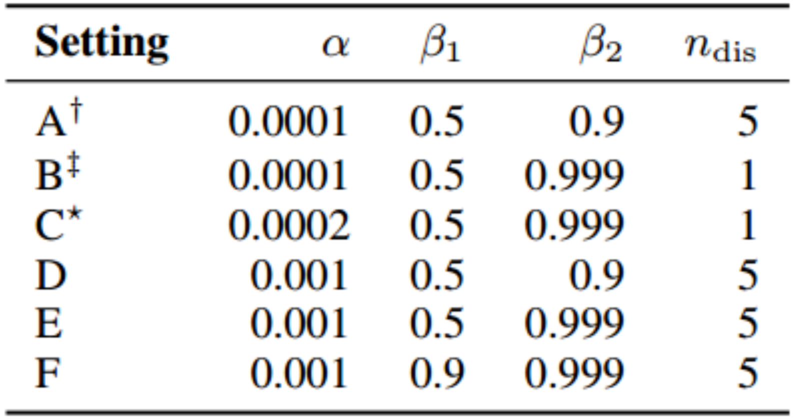
Salimans & Kingma (2016)

$$\sigma_1(\bar{W}_{\mathrm{WN}})^2 + \sigma_2(\bar{W}_{\mathrm{WN}})^2 + \cdots + \sigma_T(\bar{W}_{\mathrm{WN}})^2 = d_o, \text{ where } T = \min(d_i, d_o)$$

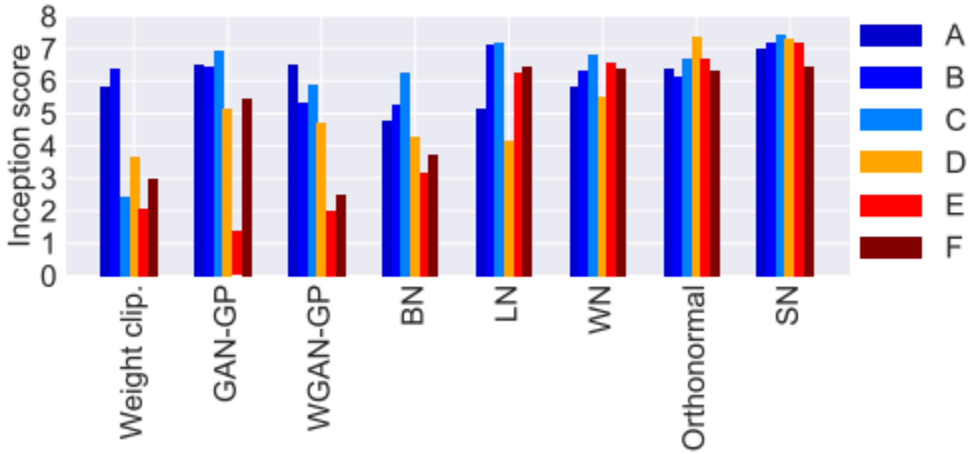
$\sigma_t(A)$ is a t-th singular value of matrix A

Default: Using such W_{WN} corresponds to using only one feature to discriminate the model probability distribution from the target.
Weight clipping (Arjovsky et al., 2017) also suffers from same pitfall.

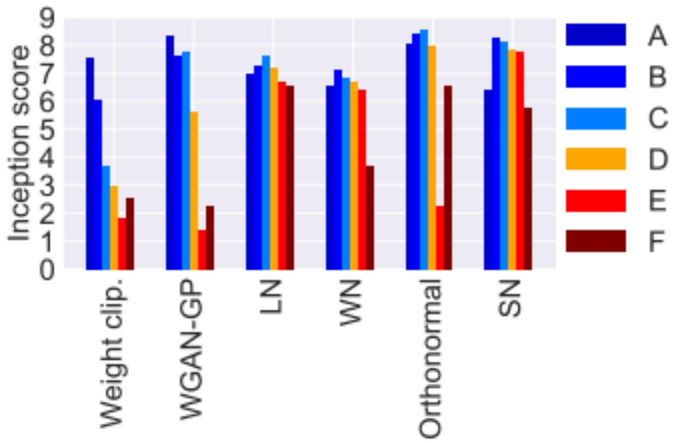
Experiment



Setting	α	β_1	β_2	n_{dis}
A[†]	0.0001	0.5	0.9	5
B[‡]	0.0001	0.5	0.999	1
C[⋆]	0.0002	0.5	0.999	1
D	0.001	0.5	0.9	5
E	0.001	0.5	0.999	5
F	0.001	0.9	0.999	5



(a) CIFAR-10

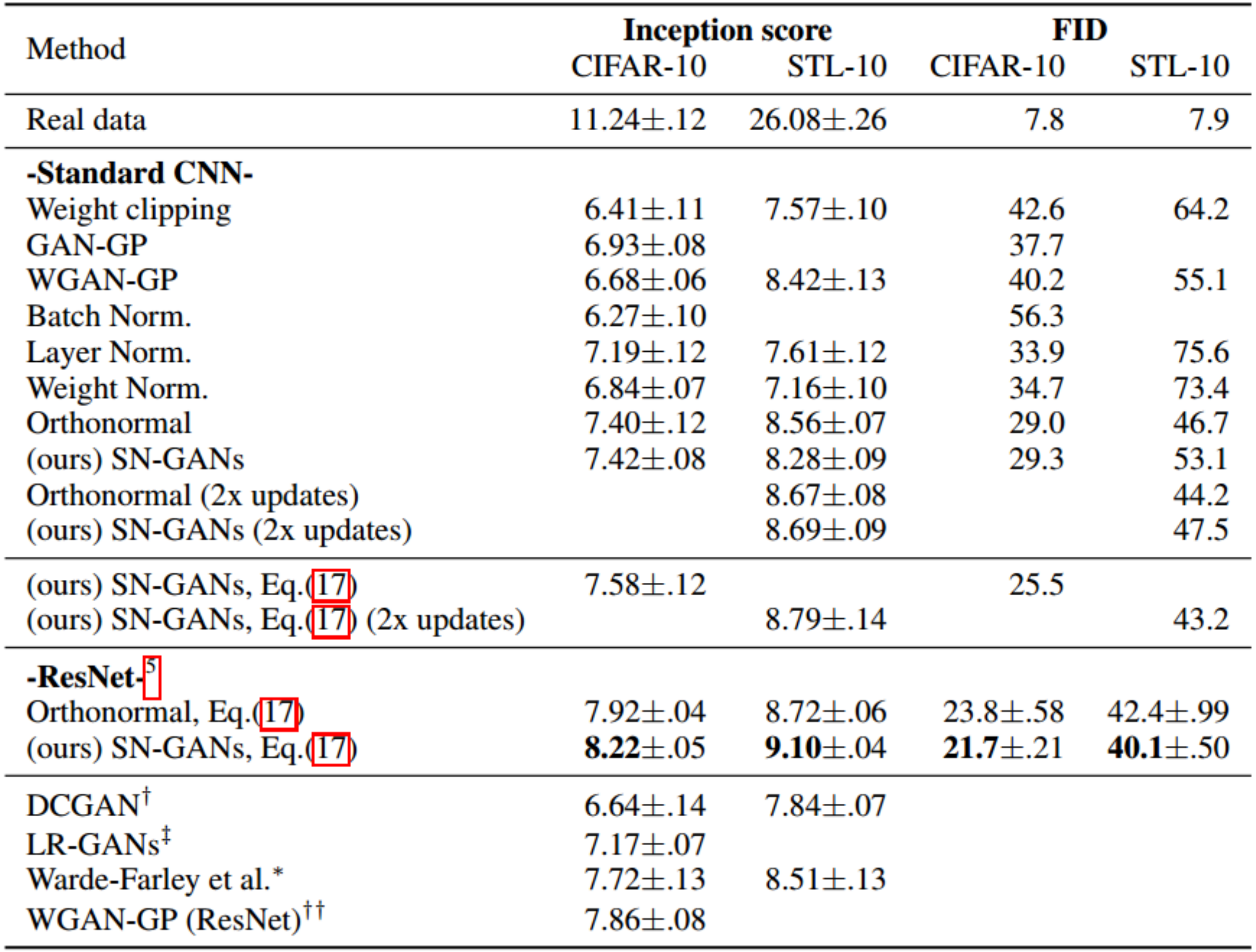


(b) STL-10

α: learning rate
(β1, β2):
momentum parameters

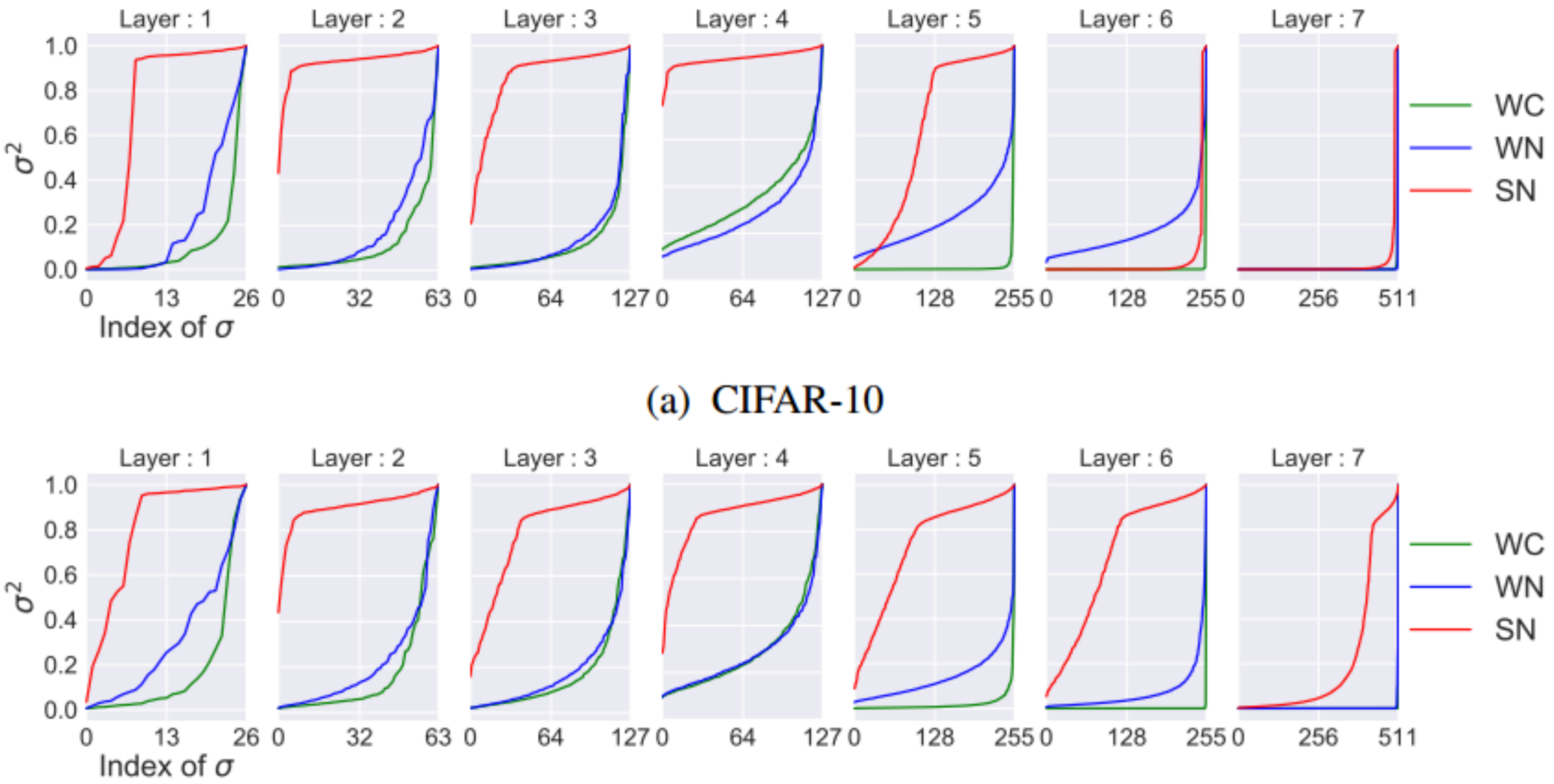
unsupervised image generation on CIFAR-10 (Torralba
et al., 2008) and STL-10 (Coates et al., 2011) ,
ILSVRC2012 dataset (ImageNet) (Russakovsky et al., 2015)

Experiment



Method	Inception score		FID	
	CIFAR-10	STL-10	CIFAR-10	STL-10
Real data	11.24±.12	26.08±.26	7.8	7.9
-Standard CNN-				
Weight clipping	6.41±.11	7.57±.10	42.6	64.2
GAN-GP	6.93±.08		37.7	
WGAN-GP	6.68±.06	8.42±.13	40.2	55.1
Batch Norm.	6.27±.10		56.3	
Layer Norm.	7.19±.12	7.61±.12	33.9	75.6
Weight Norm.	6.84±.07	7.16±.10	34.7	73.4
Orthonormal	7.40±.12	8.56±.07	29.0	46.7
(ours) SN-GANs	7.42±.08	8.28±.09	29.3	53.1
Orthonormal (2x updates)		8.67±.08		44.2
(ours) SN-GANs (2x updates)		8.69±.09		47.5
(ours) SN-GANs, Eq.(17)	7.58±.12		25.5	
(ours) SN-GANs, Eq.(17) (2x updates)		8.79±.14		43.2
-ResNet-[5]				
Orthonormal, Eq.(17)	7.92±.04	8.72±.06	23.8±.58	42.4±.99
(ours) SN-GANs, Eq.(17)	**8.22**±.05	**9.10**±.04	**21.7**±.21	**40.1**±.50
DCGAN[†]	6.64±.14	7.84±.07		
LR-GANs[‡]	7.17±.07			
Warde-Farley et al.[*]	7.72±.13	8.51±.13		
WGAN-GP (ResNet)[††]	7.86±.08			

Experiment

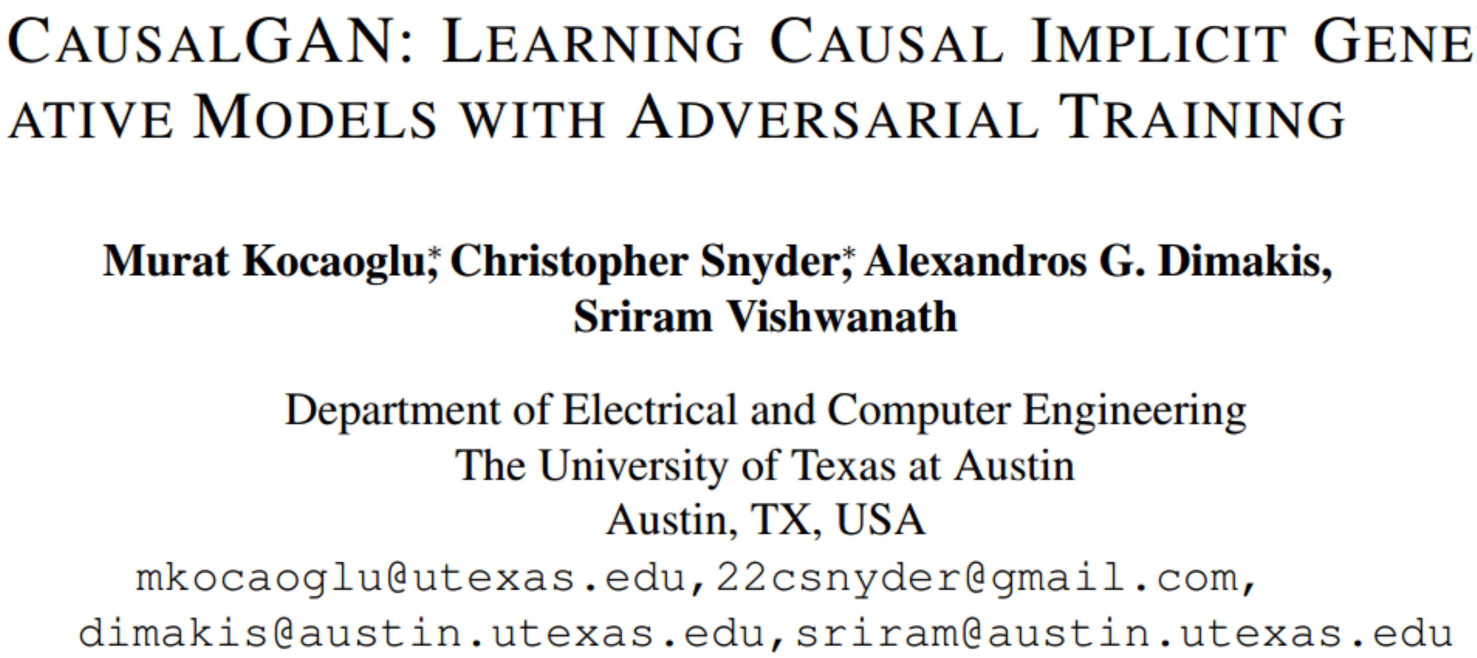


(a) CIFAR-10

(b) STL-10

Weight Clipping, Weight Normalization and Spectrum Normalization

ICLR 2018



CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training

Murat Kocaoglu*, Christopher Snyder*, Alexandros G. Dimakis,
Sriram Vishwanath

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX, USA
mkocaoglu@utexas.edu, 22csnyder@gmail.com,
dimakis@austin.utexas.edu, sriram@austin.utexas.edu

Introduction

causal implicit generative models (CiGMs)

models that allow sampling from not only the true observational but also the true interventional distributions.
a two-stage procedure for learning a CiGM over the labels and the image

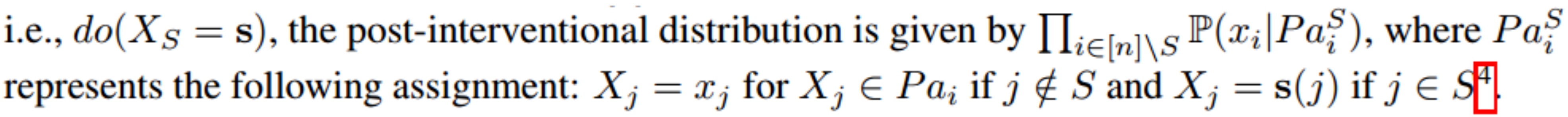
Causal Graph

A structural causal model is a tuple $M = \left(V, \epsilon, F, P_\epsilon(.)\right)$ that contains a set of functions $F = \{f_1, f_2, \dots, f_n\}$, a set of random variables $V = \{X_1, X_2, \dots, X_n\}$, a set of exogenous variables $\epsilon = \{E_1, E_2, \dots, E_n\}$, and a probability distribution over the exogenous variables P_ϵ

The causal graph D is the directed acyclic graph on nodes V, a node X_j is a parent of node X_i if and only if X_j is in the domain f_i, i.e. $X_i = f_i(X_j, S, E_i)$ for some $S \subset V$

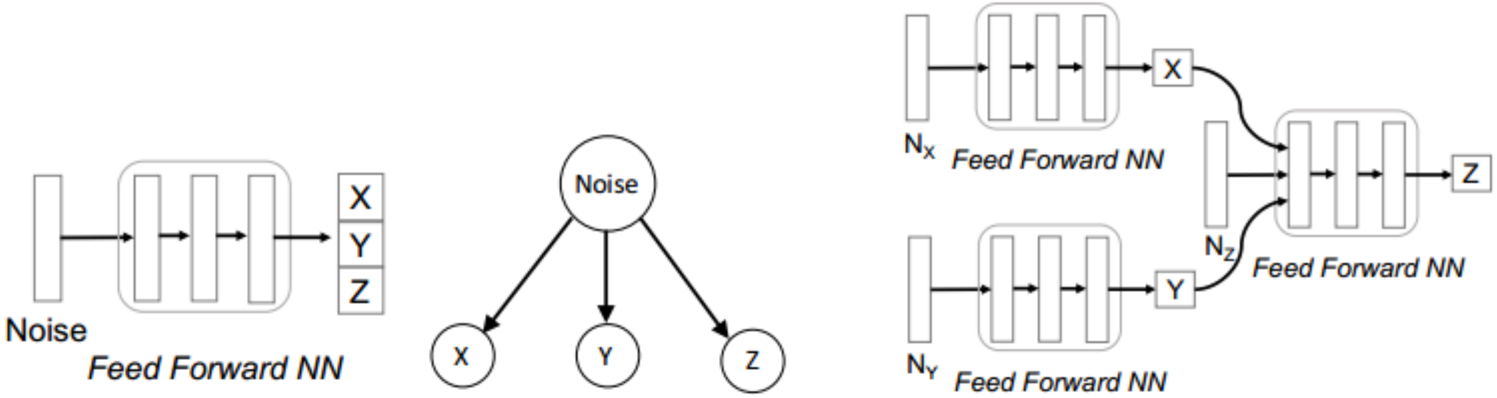
Intervention

An intervention removes the connections of node X_i to it parents, whereas conditioning does not change the causal graph from which data is sampled.



i.e., $do(X_S = \mathrm{s})$, the post-interventional distribution is given by $\prod_{i\in[n]\setminus S}\mathbb{P}(x_i|Pa_i^S)$, where Pa_i^S represents the following assignment: $X_j = x_j$ for $X_j \in Pa_i$ if $j \notin S$ and $X_j = \mathrm{s}(j)$ if $j \in S$[4].

Example

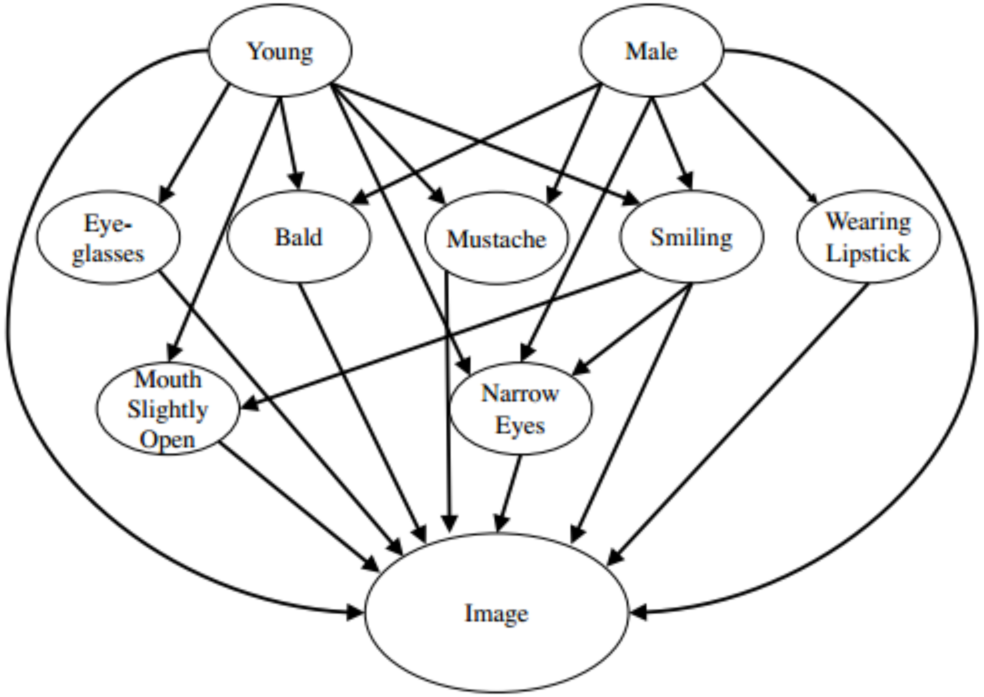


(a) Naive feedforward generator architecture and the causal graph it represents.

(b) Generator neural network architecture that represent the causal graph $X \to Z \leftarrow Y$.

Figure 2: (a) The causal graph implied by the naive feedforward generator architecture. (b) A neural network implementation of the causal graph $X \to Z \leftarrow Y$: Each feed forward neural net captures the function f in the structural equation model $V = f(Pa_V, E)$.

Example


Young
Male
Eye-glasses
Bald
Mustache
Smiling
Wearing Lipstick
Mouth Slightly Open
Narrow Eyes
Image

Model

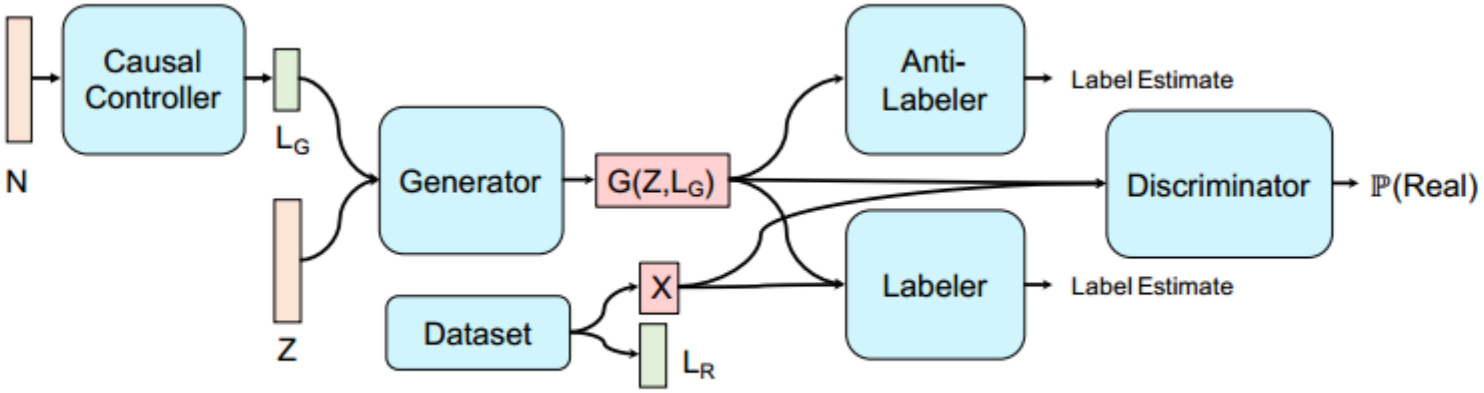


Figure 3: CausalGAN architecture: Causal controller is a pretrained causal implicit generative model for the image labels. Labeler is trained on the real data, Anti-Labeler is trained on generated data. Generator minimizes Labeler loss and maximizes Anti-Labeler loss.

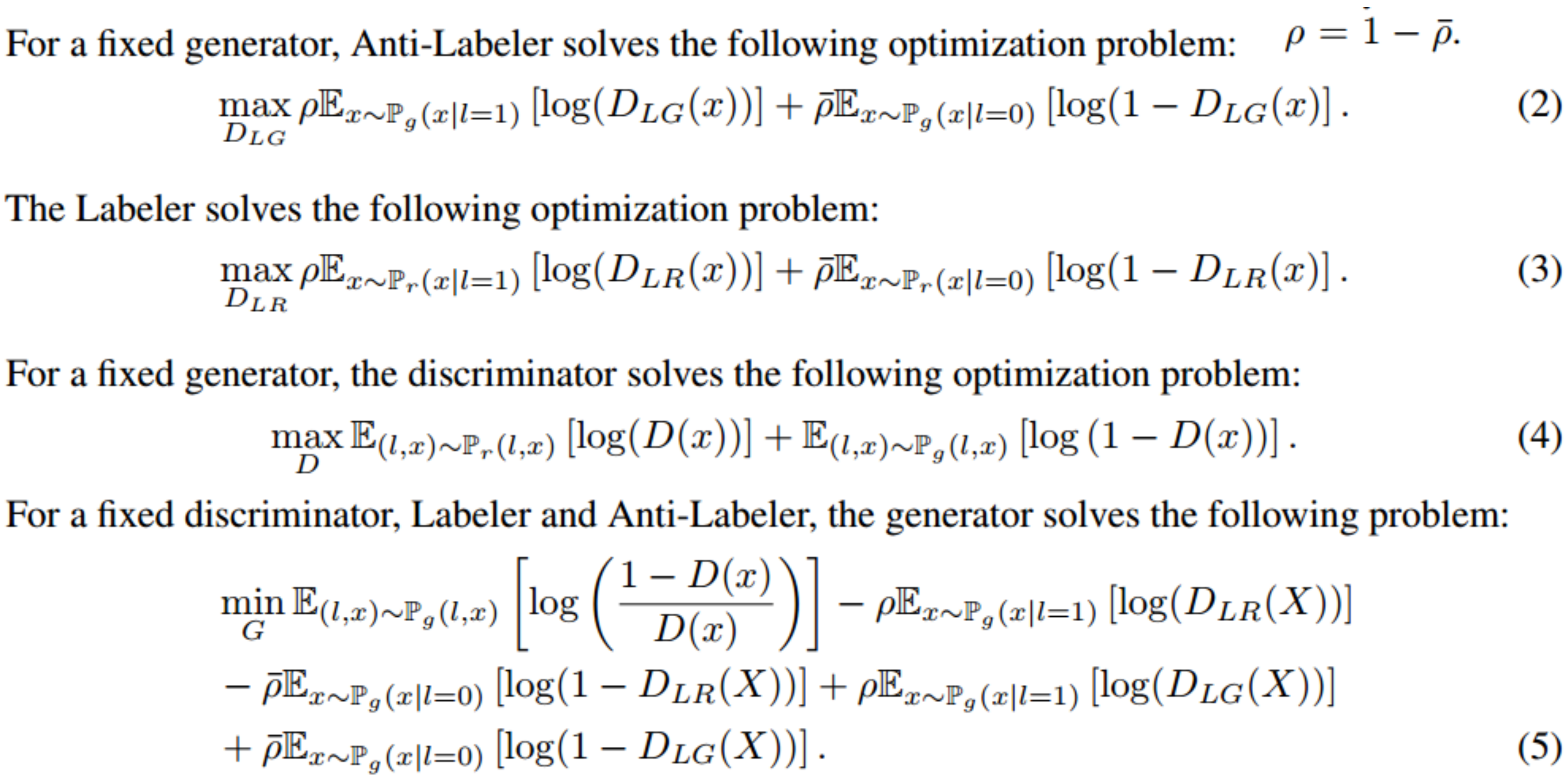
Architecture

Causal Controller: WGAN Arjovsky et al. (2017) , used for controlling which distribution the images will be sampled from when intervened or conditioned on a set of labels. (binary)

Laberer and Anti-Laberer: *The Labeler* is trained to estimate the labels of images in the dataset. The Anti-Labeler is trained to estimate the labels of the images sampled from the generator

Generator: The objective of the generator is 3-fold: producing realistic images by competing with the discriminator, producing images consistent with the labels by minimizing the Labeler loss and avoiding unrealistic image distributions that are easy to label by maximizing the Anti-Labeler loss.

Loss Function



For a fixed generator, Anti-Labeler solves the following optimization problem: $\rho = 1 - \bar{\rho}$.

$$\max_{D_{LG}} \rho \mathbb{E}_{x \sim \mathbb{P}_g(x|l=1)} \left[\log(D_{LG}(x))\right] + \bar{\rho} \mathbb{E}_{x \sim \mathbb{P}_g(x|l=0)} \left[\log(1 - D_{LG}(x)\right]. \tag{2}$$

The Labeler solves the following optimization problem:

$$\max_{D_{LR}} \rho \mathbb{E}_{x \sim \mathbb{P}_r(x|l=1)} \left[\log(D_{LR}(x))\right] + \bar{\rho} \mathbb{E}_{x \sim \mathbb{P}_r(x|l=0)} \left[\log(1 - D_{LR}(x)\right]. \tag{3}$$

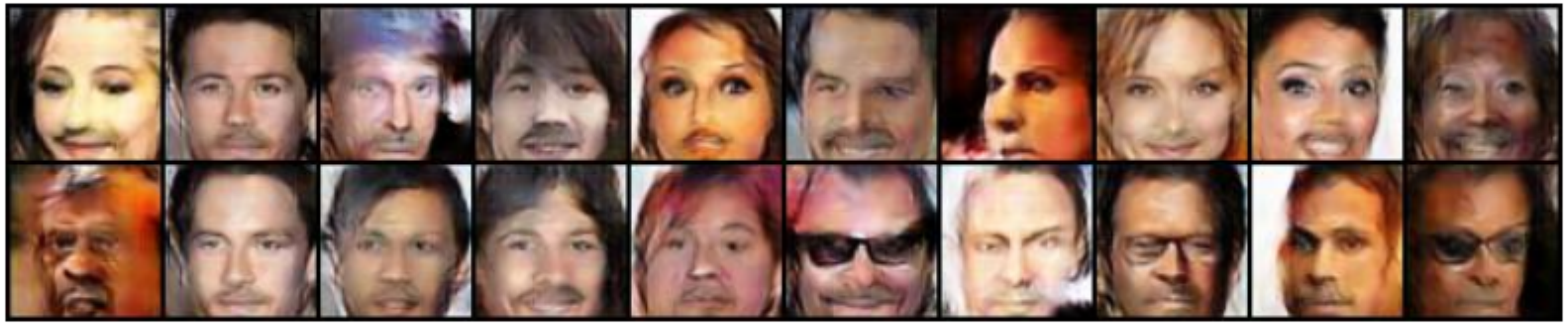
For a fixed generator, the discriminator solves the following optimization problem:

$$\max_{D} \mathbb{E}_{(l,x) \sim \mathbb{P}_r(l,x)} \left[\log(D(x))\right] + \mathbb{E}_{(l,x) \sim \mathbb{P}_g(l,x)} \left[\log\left(1 - D(x)\right)\right]. \tag{4}$$

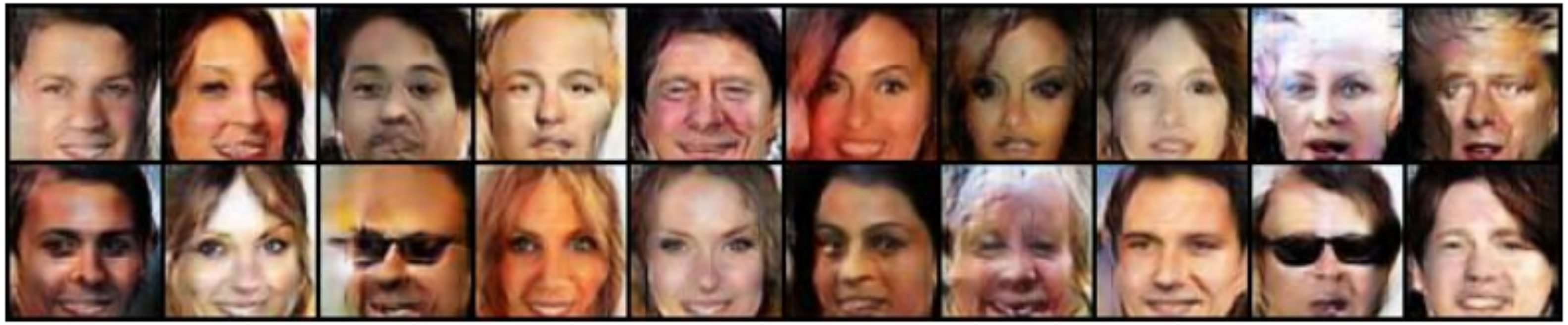
For a fixed discriminator, Labeler and Anti-Labeler, the generator solves the following problem:

$$\begin{aligned} &\min_{G} \mathbb{E}_{(l,x) \sim \mathbb{P}_g(l,x)} \left[\log\left(\frac{1 - D(x)}{D(x)}\right)\right] - \rho \mathbb{E}_{x \sim \mathbb{P}_g(x|l=1)} \left[\log(D_{LR}(X))\right] \\ &- \bar{\rho} \mathbb{E}_{x \sim \mathbb{P}_g(x|l=0)} \left[\log(1 - D_{LR}(X))\right] + \rho \mathbb{E}_{x \sim \mathbb{P}_g(x|l=1)} \left[\log(D_{LG}(X))\right] \\ &+ \bar{\rho} \mathbb{E}_{x \sim \mathbb{P}_g(x|l=0)} \left[\log(1 - D_{LG}(X))\right]. \end{aligned} \tag{5}$$

Result

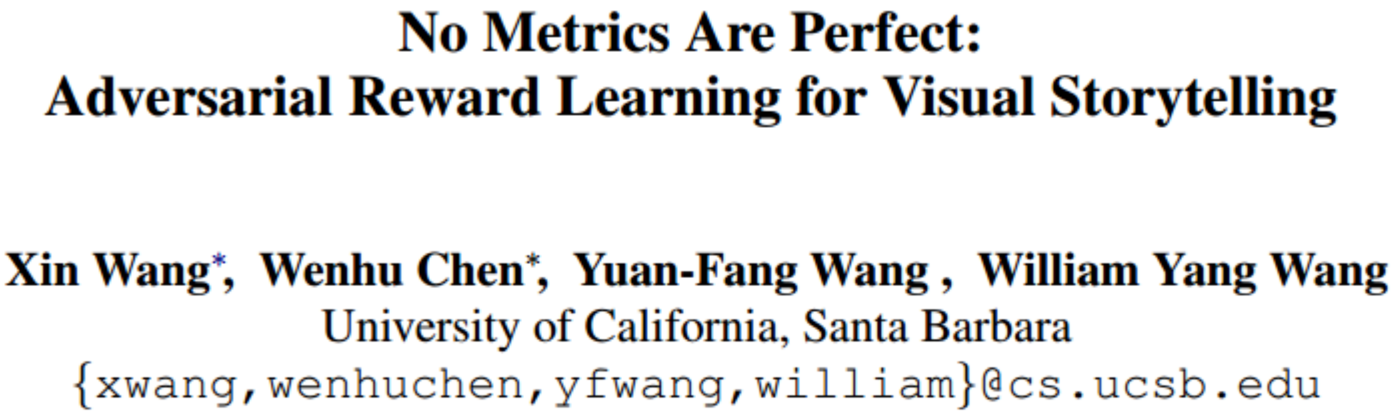


Top: Intervene Mustache=1, Bottom: Condition Mustache=1



Top: Intervene Mouth Slightly Open=1, Bottom: Condition Mouth Slightly Open=1

ACL 2018



No Metrics Are Perfect: Adversarial Reward Learning for Visual Storytelling

Xin Wang*, Wenhu Chen*, Yuan-Fang Wang , William Yang Wang
University of California, Santa Barbara
{xwang,wenhuchen,yfwang,william}@cs.ucsb.edu

Motivation

the limitations of automatic metrics on evaluating story quality, reinforcement learning methods with hand-crafted rewards also face difficulties in gaining an overall performance boost.

learn an implicit reward function from human demonstrations, and then optimize policy search with the learned reward function.

Contribution

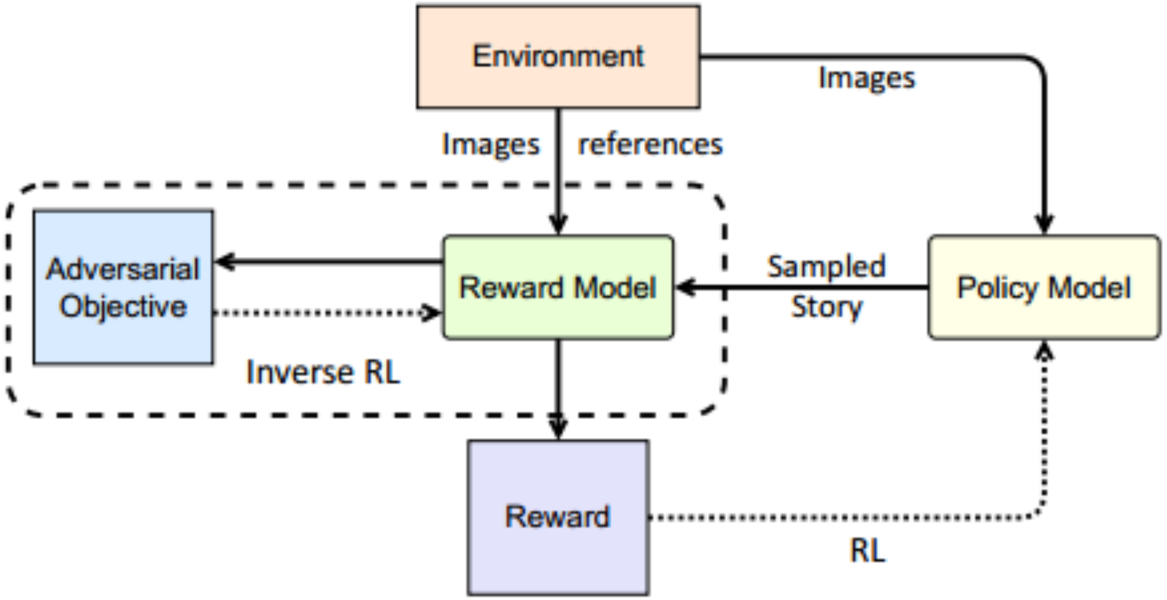
propose an adversarial reward learning framework and apply it to boost visual story generation.

evaluate our approach on the Visual Storytelling (VIST) dataset and achieve the state-of-the-art results on automatic metrics.

empirically demonstrate that automatic metrics are not perfect for either training or evaluation.

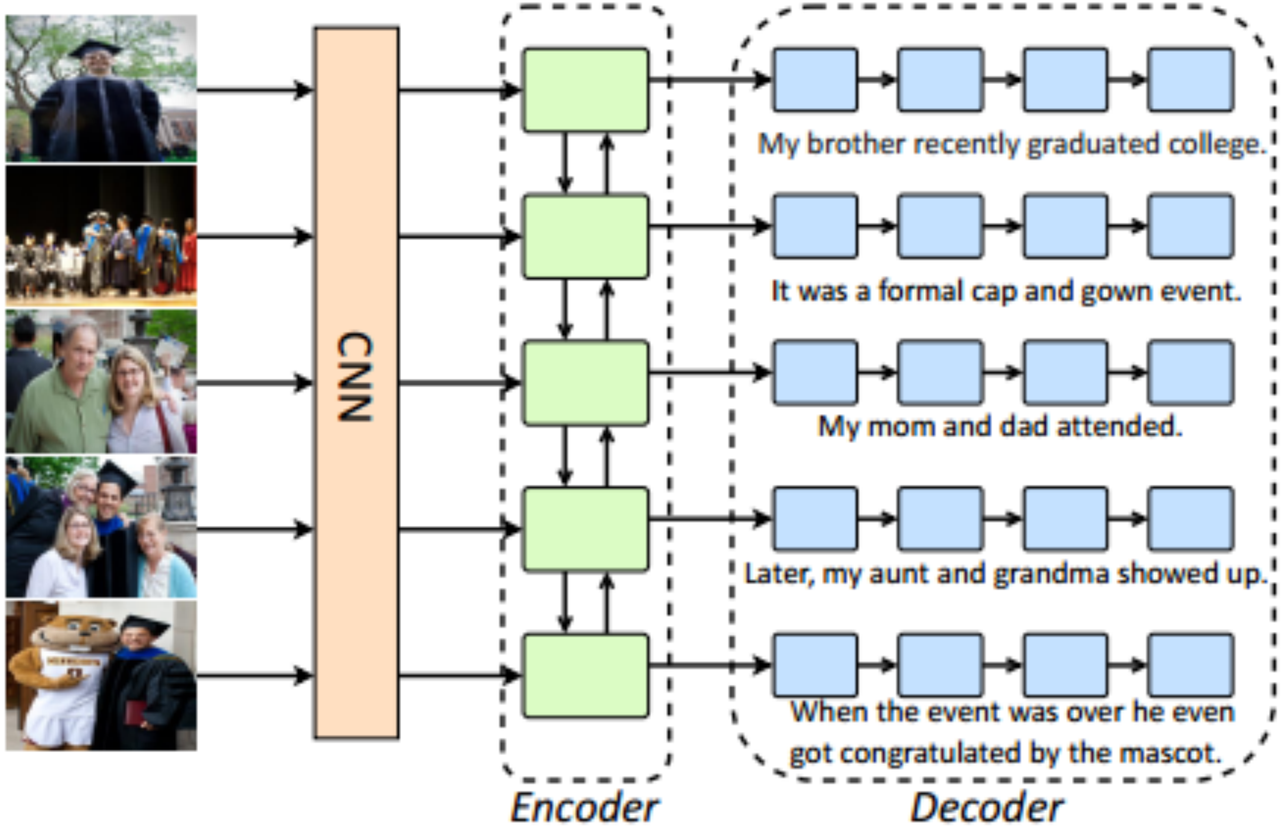
design and perform a comprehensive human evaluation via Amazon Mechanical Turk, which demonstrates the superiority of the generated stories of our method on relevance, expressiveness, and concreteness.

Framework



a **policy model** $\pi_\beta(W)$ and a **reward model** $R_\theta(W)$. The policy model takes an image sequence I as the input and performs sequential actions (choosing words *w* from the vocabulary V) to form a narrative story *W* . The reward model is optimized by the adversarial objective (see Section 3.3) and aims at deriving a human-like reward from both human-annotated stories and sampled predictions

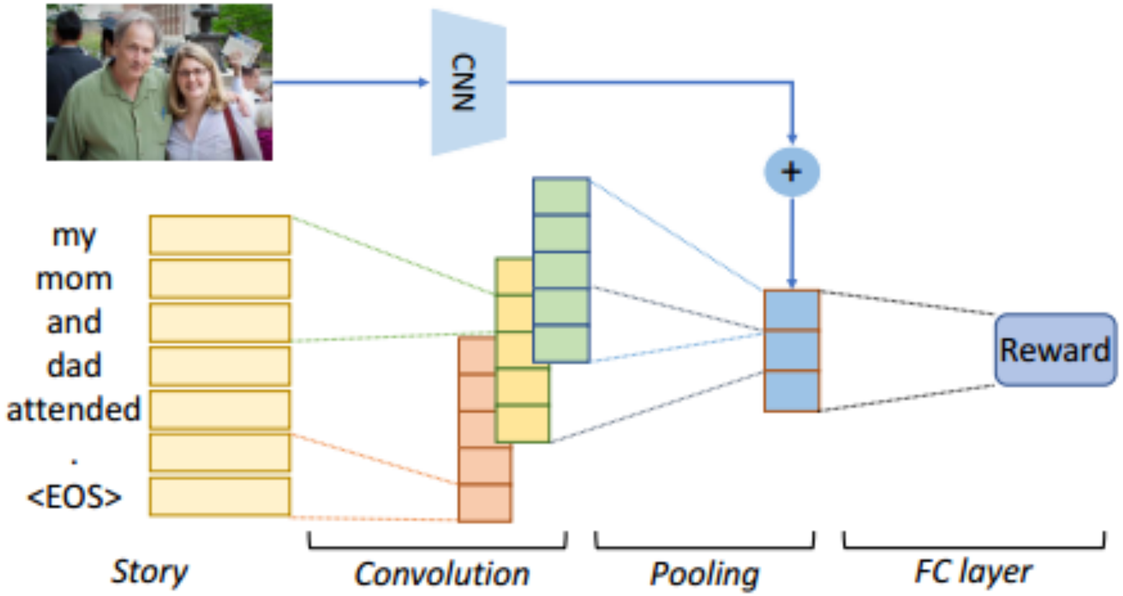
Model


CNN
My brother recently graduated college.
It was a formal cap and gown event.
My mom and dad attended.
Later, my aunt and grandma showed up.
When the event was over he even got congratulated by the mascot.
Encoder
Decoder

$$s_t^i = \mathrm{GRU}(s_{t-1}^i, [w_{t-1}^i, h_i]) \; , \quad (1)$$

$$\pi_\beta(w_t^i | w_{1:t-1}^i) = softmax(W_s s_t^i + b_s) \; , \quad (2)$$

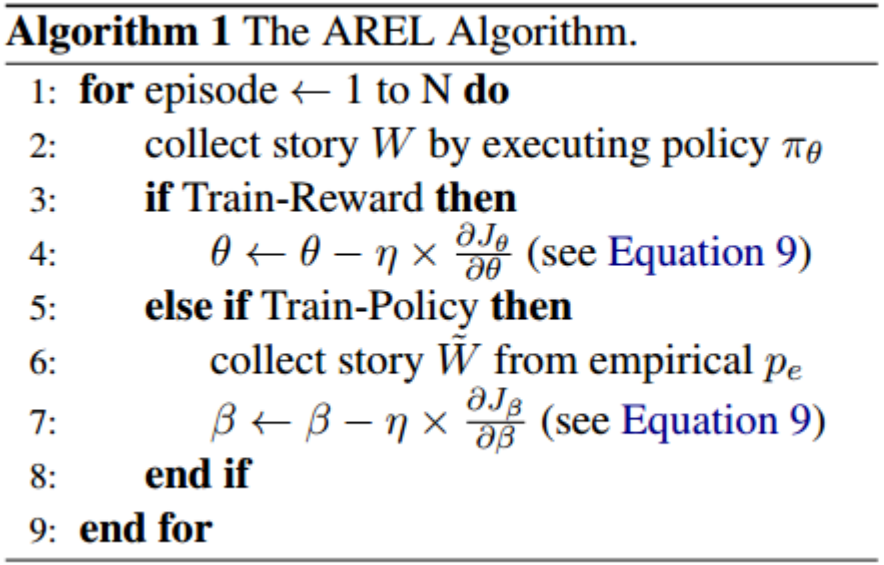
Model



$$R_\theta(W) = W_r(f_{conv}(W) + W_i I_{CNN}) + b_r$$

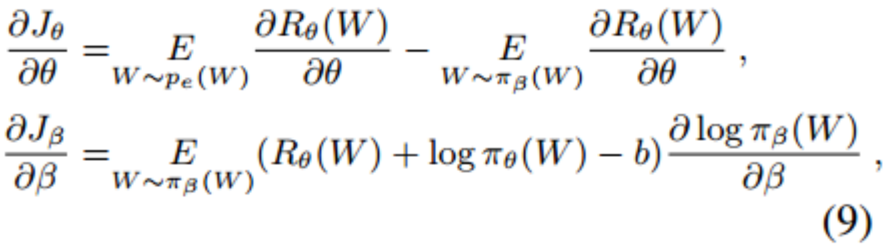
W_r, b_r denotes the weights in the output layer, and f_{conv} denotes the operations in CNN. I_{CNN} is the high-level visual feature extracted from the image, and W_i projects it into the sentence representation space.

Algorithm



Algorithm 1 The AREL Algorithm.

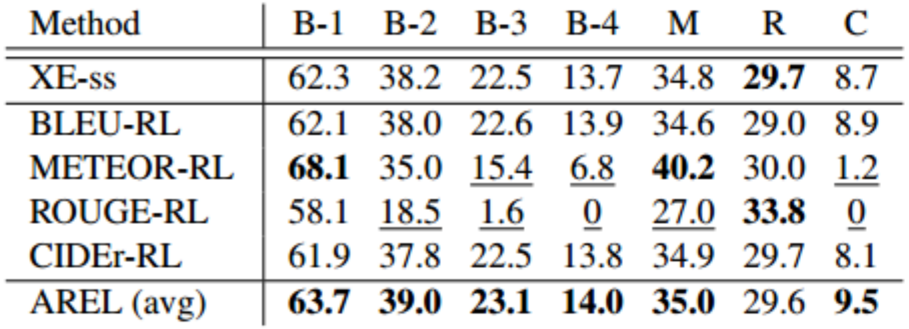
```
for episode ← 1 to N do
    collect story $W$ by executing policy $\pi_\theta$
    if Train-Reward then
        $\theta \leftarrow \theta - \eta \times \frac{\partial J_\theta}{\partial \theta}$ (see Equation 9)
    else if Train-Policy then
        collect story $\tilde{W}$ from empirical $p_e$
        $\beta \leftarrow \beta - \eta \times \frac{\partial J_\beta}{\partial \beta}$ (see Equation 9)
    end if
end for
```



$$\begin{aligned} \frac{\partial J_\theta}{\partial \theta} &= \underset{W \sim p_e(W)}{E} \frac{\partial R_\theta(W)}{\partial \theta} - \underset{W \sim \pi_\beta(W)}{E} \frac{\partial R_\theta(W)}{\partial \theta}, \\ \frac{\partial J_\beta}{\partial \beta} &= \underset{W \sim \pi_\beta(W)}{E} (R_\theta(W) + \log \pi_\theta(W) - b) \frac{\partial \log \pi_\beta(W)}{\partial \beta}, \end{aligned} \tag{9}$$

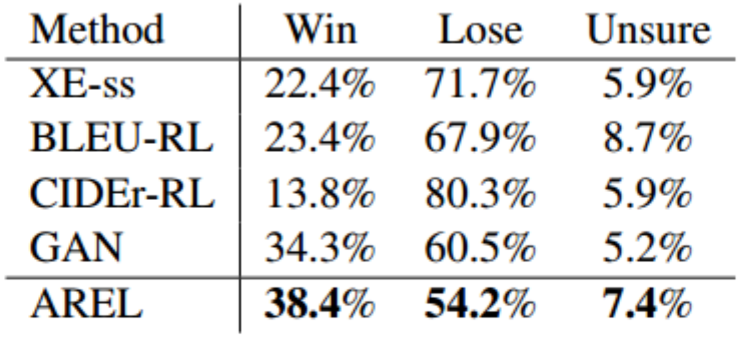
Experiment

VIST dataset (Huang et al., 2016) 10,117 Flickr albums with 210,819 unique photos.



Method	B-1	B-2	B-3	B-4	M	R	C
XE-ss	62.3	38.2	22.5	13.7	34.8	**29.7**	8.7
BLEU-RL	62.1	38.0	22.6	13.9	34.6	29.0	8.9
METEOR-RL	**68.1**	35.0	15.4	6.8	**40.2**	30.0	1.2
ROUGE-RL	58.1	18.5	1.6	0	27.0	**33.8**	0
CIDEr-RL	61.9	37.8	22.5	13.8	34.9	29.7	8.1
AREL (avg)	**63.7**	**39.0**	**23.1**	**14.0**	**35.0**	29.6	**9.5**

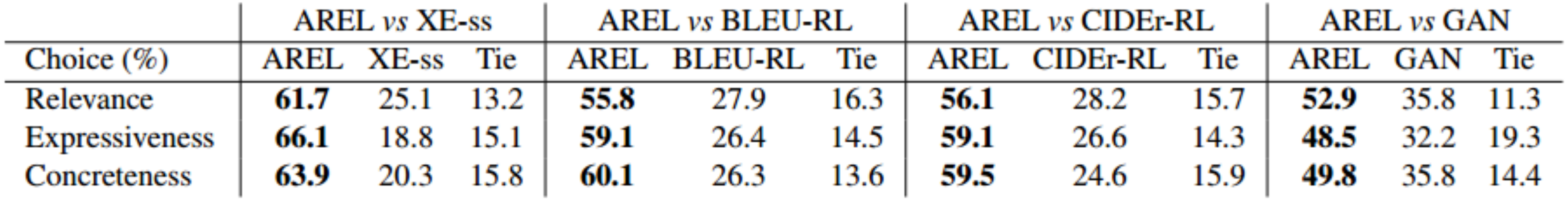
Table 2: Comparison with different RL models with different metric scores as the rewards. We report the average scores of the AREL models as AREL (avg). Although METEOR-RL and ROUGE-RL models achieve very high scores on their own metrics, the underlined scores are severely damaged. Actually, they are gaming their own metrics with nonsense sentences.



Method	Win	Lose	Unsure
XE-ss	22.4%	71.7%	5.9%
BLEU-RL	23.4%	67.9%	8.7%
CIDEr-RL	13.8%	80.3%	5.9%
GAN	34.3%	60.5%	5.2%
AREL	**38.4%**	**54.2%**	**7.4%**

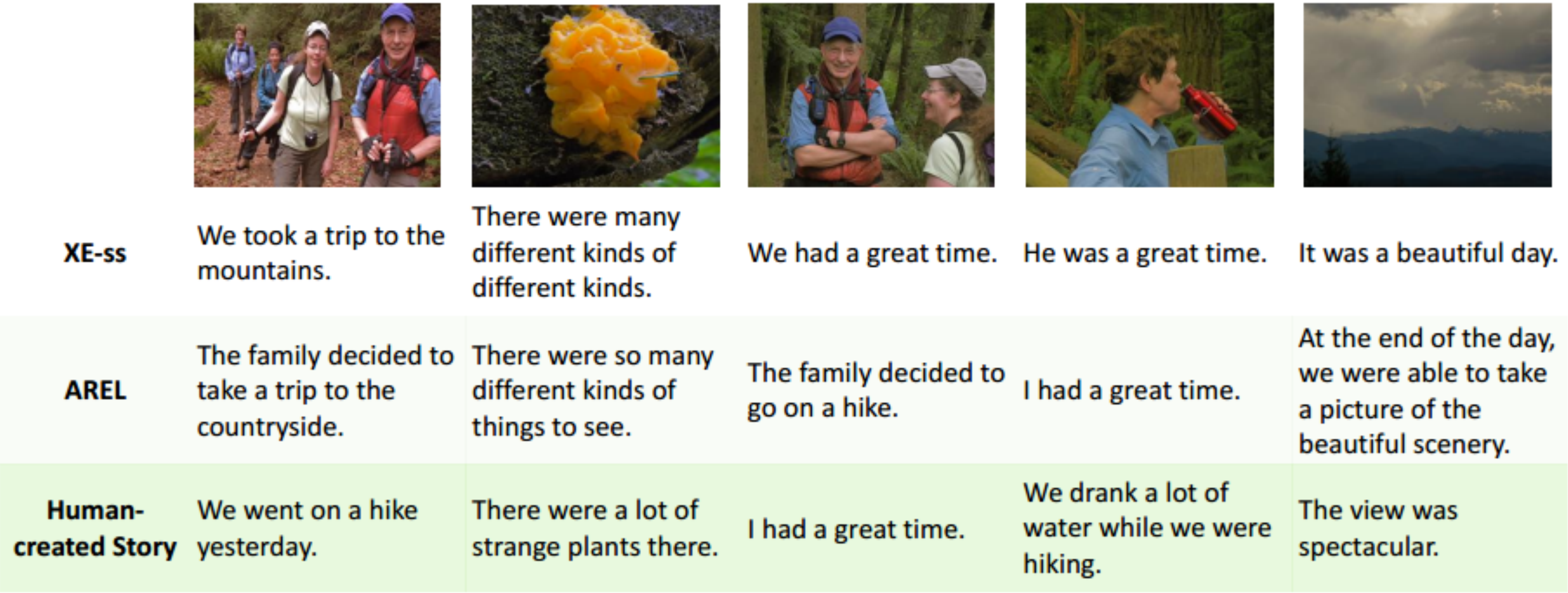
Table 3: Turing test results.

Experiment



	AREL *vs* XE-ss			AREL *vs* BLEU-RL			AREL *vs* CIDEr-RL			AREL *vs* GAN		
Choice (%)	AREL	XE-ss	Tie	AREL	BLEU-RL	Tie	AREL	CIDEr-RL	Tie	AREL	GAN	Tie
Relevance	**61.7**	25.1	13.2	**55.8**	27.9	16.3	**56.1**	28.2	15.7	**52.9**	35.8	11.3
Expressiveness	**66.1**	18.8	15.1	**59.1**	26.4	14.5	**59.1**	26.6	14.3	**48.5**	32.2	19.3
Concreteness	**63.9**	20.3	15.8	**60.1**	26.3	13.6	**59.5**	24.6	15.9	**49.8**	35.8	14.4

Table 4: Pairwise human comparisons. The results indicate the consistent superiority of our AREL model in generating more human-like stories than the SOTA methods.



XE-ss	We took a trip to the mountains.	There were many different kinds of different kinds.	We had a great time.	He was a great time.	It was a beautiful day.
AREL	The family decided to take a trip to the countryside.	There were so many different kinds of things to see.	The family decided to go on a hike.	I had a great time.	At the end of the day, we were able to take a picture of the beautiful scenery.
Human-created Story	We went on a hike yesterday.	There were a lot of strange plants there.	I had a great time.	We drank a lot of water while we were hiking.	The view was spectacular.