数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Generative Adversarial Nets Introduction

Lantao Yu
2016 / 9 / 21

Success of Deep Learning

- **Discriminative Models**
  - **backpropagation**
  - **dropout algorithms**
- Generative Models
  - Less of an impact
  - Intractable probabilistic computations
  - ...

Two-player minimax game

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Train Generator

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

Train Discriminator

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

Optimal Discriminator

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

Reformulate the minimax game

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

$$
= -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)
$$

Natural Image Generation

- CGAN
- LAPGAN
- DCGAN
- GRAN
- VAEGAN
- …

Exposure bias in Sequence Generation

- In the inference stage, the model generates a sequence iteratively and predicts next token conditioned on its previously predicted ones that may be never observed in the training data.

Training

$$-\mathbb{E}_{\mathbf{x}\sim P}\log Q(x)$$

Inference

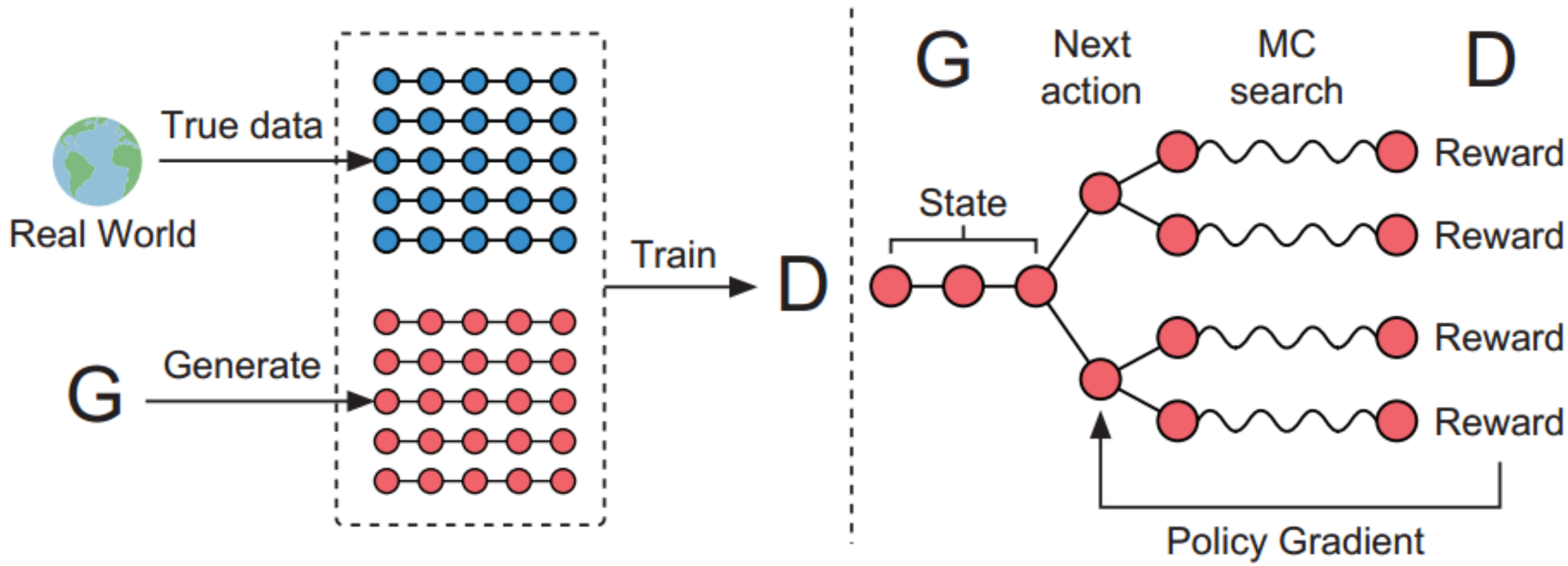
$$-\mathbb{E}_{\mathbf{x}\sim Q}\log P(x)$$

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

KL[P||Q] & KL[Q||P]

$$JSD[P\|Q] = JSD[P\|Q] = \frac{1}{2}KL\left[P\middle\|\frac{P+Q}{2}\right] + \frac{1}{2}KL\left[Q\middle\|\frac{P+Q}{2}\right]$$

SeqGAN with Policy Gradient



Objective Function

$$J(\theta) = \mathbb{E}[R_T | s_0, \theta] = \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1 | s_0) \cdot Q_{D_\phi}^{G_\theta}(s_0, y_1)$$

$$Q^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \mathcal{R}_s^a + \sum_{s' \in S} \delta_{ss'}^a V^{G_\theta}(s') = V^{G_\theta}(Y_{1:t})$$

$$V^{G_\theta}(s = Y_{1:t-1}) = \sum_{y_t \in \mathcal{Y}} G_\theta(y_t | Y_{1:t-1}) \cdot Q^{G_\theta}(Y_{1:t-1}, y_t)$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{Y_{1:t-1} \sim G_\theta}\left[\sum_{y_t \in \mathcal{Y}} \nabla_\theta G_\theta(y_t | Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) \right]$$

$$
\begin{aligned}
&\nabla_\theta J(\theta) \\
&= \nabla_\theta V^{G_\theta}(s_0) = \nabla_\theta [\sum_{y_1 \in \mathcal{Y}} G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1)] \\
&= \sum_{y_1 \in \mathcal{Y}} [\nabla_\theta G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1) + G_\theta(y_1|s_0) \cdot \nabla_\theta Q^{G_\theta}(s_0, y_1)] \\
&= \sum_{y_1 \in \mathcal{Y}} [\nabla_\theta G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1) + G_\theta(y_1|s_0) \cdot \nabla_\theta V^{G_\theta}(Y_{1:1})] \\
&= \sum_{y_1 \in \mathcal{Y}} \nabla_\theta G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1) + \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1|s_0) \nabla_\theta [\sum_{y_2 \in \mathcal{Y}} G_\theta(y_2|Y_{1:1}) Q^{G_\theta}(Y_{1:1}, y_2)] \\
&= \sum_{y_1 \in \mathcal{Y}} \nabla_\theta G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1) + \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1|s_0) \sum_{y_2 \in \mathcal{Y}} [\nabla_\theta G_\theta(y_2|Y_{1:1}) \cdot Q^{G_\theta}(Y_{1:1}, y_2) \\
&\qquad + G_\theta(y_2|Y_{1:1}) \nabla_\theta Q^{G_\theta}(Y_{1:1}, y_2)] \\
&= \sum_{y_1 \in \mathcal{Y}} \nabla_\theta G_\theta(y_1|s_0) \cdot Q^{G_\theta}(s_0, y_1) + \sum_{Y_{1:1}} P(Y_{1:1}|s_0; G_\theta) \sum_{y_2 \in \mathcal{Y}} \nabla_\theta G_\theta(y_2|Y_{1:1}) \cdot Q^{G_\theta}(Y_{1:1}, y_2) \\
&\qquad + \sum_{Y_{1:2}} P(Y_{1:2}|s_0; G_\theta) \nabla_\theta V^{G_\theta}(Y_{1:2}) \\
&= \sum_{t=1}^{T} \sum_{Y_{1:t-1}} P(Y_{1:t-1}|s_0; G_\theta) \sum_{y_t \in \mathcal{Y}} \nabla_\theta G_\theta(y_t|Y_{1:t-1}) \cdot Q^{G_\theta}(Y_{1:t-1}, y_t) \\
&= \mathbb{E}_{Y_{1:t-1} \sim G_\theta} [\sum_{y_t \in \mathcal{Y}} \nabla_\theta G_\theta(y_t|Y_{1:t-1}) \cdot Q^{G_\theta}(Y_{1:t-1}, y_t)],
\end{aligned}
$$

Unbiased Estimation of the gradient

$$\nabla_\theta J(\theta) \simeq \frac{1}{T}\sum_{t=1}^{T}\sum_{y_t\in\mathcal{Y}} \nabla_\theta G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) \quad (7)$$

$$= \frac{1}{T}\sum_{t=1}^{T}\sum_{y_t\in\mathcal{Y}} G_\theta(y_t|Y_{1:t-1})\nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t)$$

$$= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{y_t\sim G_\theta(y_t|Y_{1:t-1})}[\nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t)],$$

Action Value Approximation

$$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \tag{4}$$
$$\begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n),\ Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t}; N) & \text{for} \quad t < T \\ D_\phi(Y_{1:t}) & \text{for} \quad t = T, \end{cases}$$

where $Y_{1:t}^n = (y_1, \ldots, y_t)$

$Y_{t+1:T}^n$ is sampled based on a roll-out policy and current state

Train Discriminator

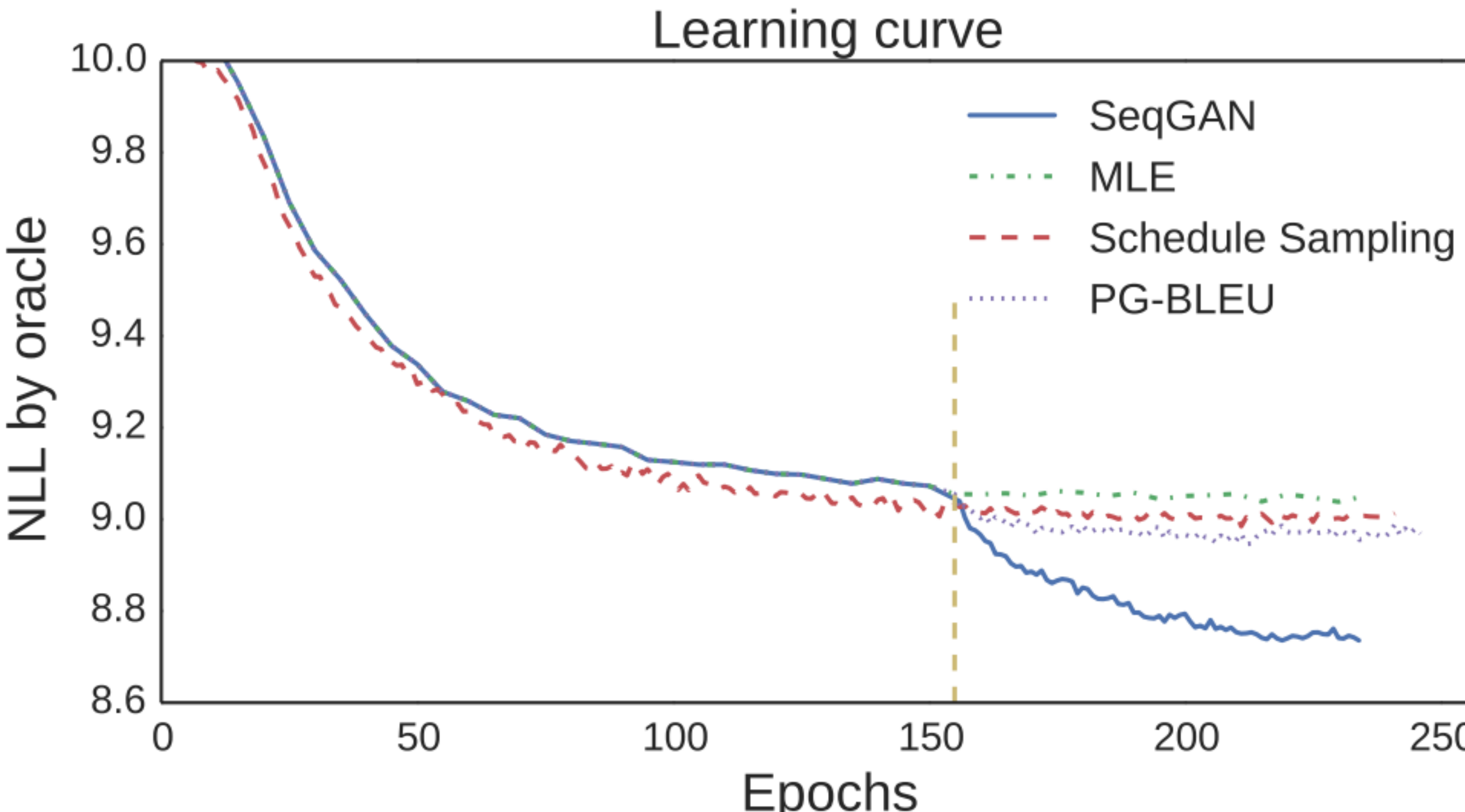
$$\min_{\phi} -\mathbb{E}_{Y\sim p_{\text{data}}}[\log D_\phi(Y)] - \mathbb{E}_{Y\sim G_\theta}[\log(1 - D_\phi(Y))].$$

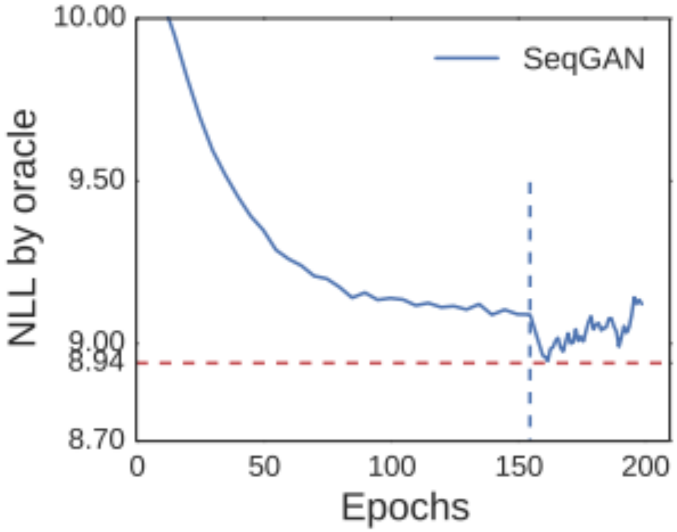
Synthetic data experiments

An oracle evaluation mechanism:

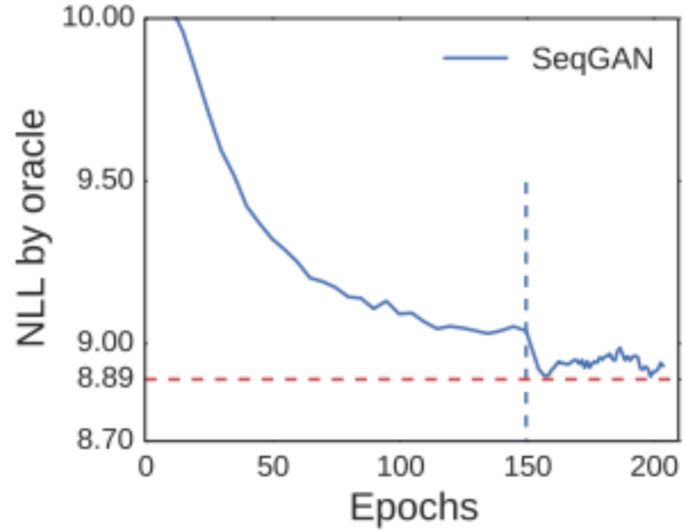
$$\mathrm{NLL}_{\mathrm{oracle}} = -\mathbb{E}_{Y_{1:T} \sim G_\theta} \left[\sum_{t=1}^{T} \log G_{\mathrm{oracle}}(y_t | Y_{1:t-1}) \right]$$

Results

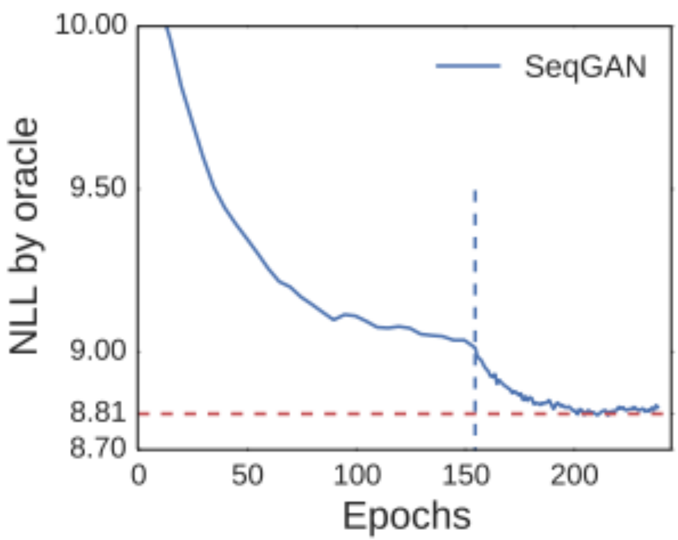

Learning curve
NLL by oracle
10.0
9.8
9.6
9.4
9.2
9.0
8.8
8.6
0
50
100
150
200
250
Epochs
SeqGAN
MLE
Schedule Sampling
PG-BLEU



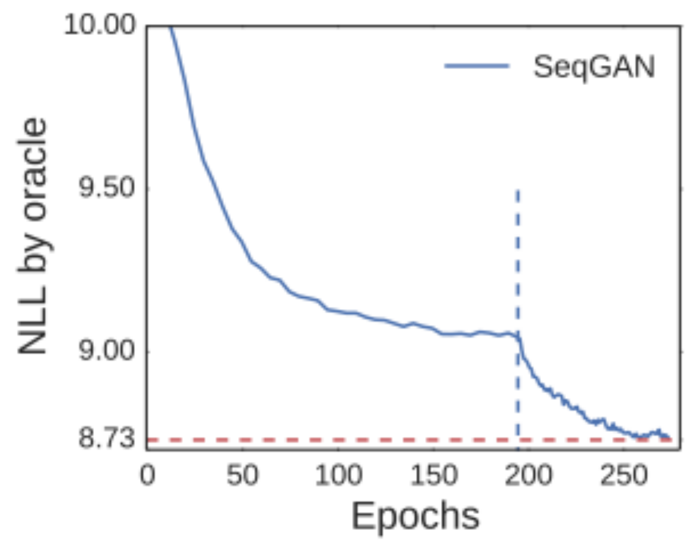
(a) g-steps=100, d-steps=1, *k*=10



(b) g-steps=30, d-steps=1, *k*=30



(c) g-steps=1, d-steps=1, *k*=10



(d) g-steps=1, d-steps=5, *k*=3

Table 2: Chinese poem generation performance comparison.

Algorithm	Human score	p-value	BLEU-2	p-value
MLE	0.4165	0.0034	0.6670	$< 10^{-6}$
SeqGAN	**0.5356**		**0.7389**	
Real data	0.6011		0.746	

Table 3: Obama political speech generation performance.

Algorithm	BLEU-3	p-value	BLEU-4	p-value
MLE	0.519	$< 10^{-6}$	0.416	0.00014
SeqGAN	**0.556**		**0.427**	

Table 4: Music generation performance comparison.

Algorithm	BLEU-4	p-value	MSE	p-value
MLE	0.9210	$< 10^{-6}$	22.38	0.00034
SeqGAN	**0.9406**		**20.62**	

Future work

- Value network
- Monte Carlo Tree Search
- Mini-batch discrimination
- ...