# GANs专题报告（1）

Introduction of GANs and Application in Speech Processing

# Background

**2014年，Goodfellow在NIPS发表了Generative Adversarial Networks** （https://arxiv.org/abs/1406.2661）

Discriminative model  VS  Generative model

1. Success in discriminative model

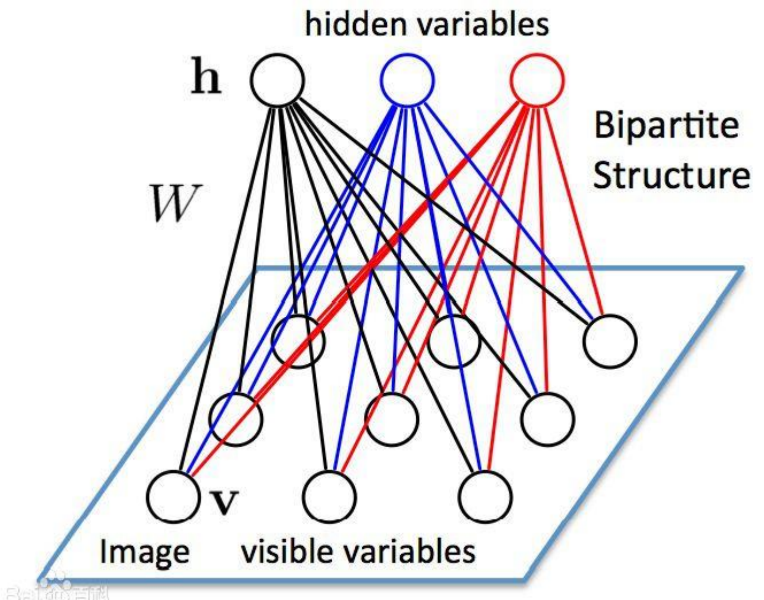
   Using Backpropagation and dropout algorithms

2. Little impact of generative model

# Background —— 之前的做法

Parametric specification + Maximize log likelihood

最经典： 受限玻尔兹曼机（Restricted Boltzmann Machine, RBM）

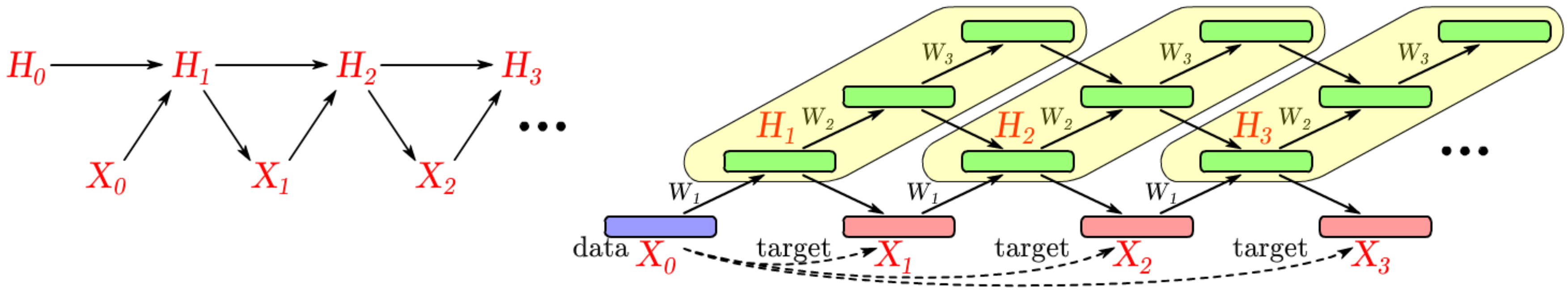



How to use BP rather than likelihood function ?

# Background —— 之前的做法

Markov chains + BP

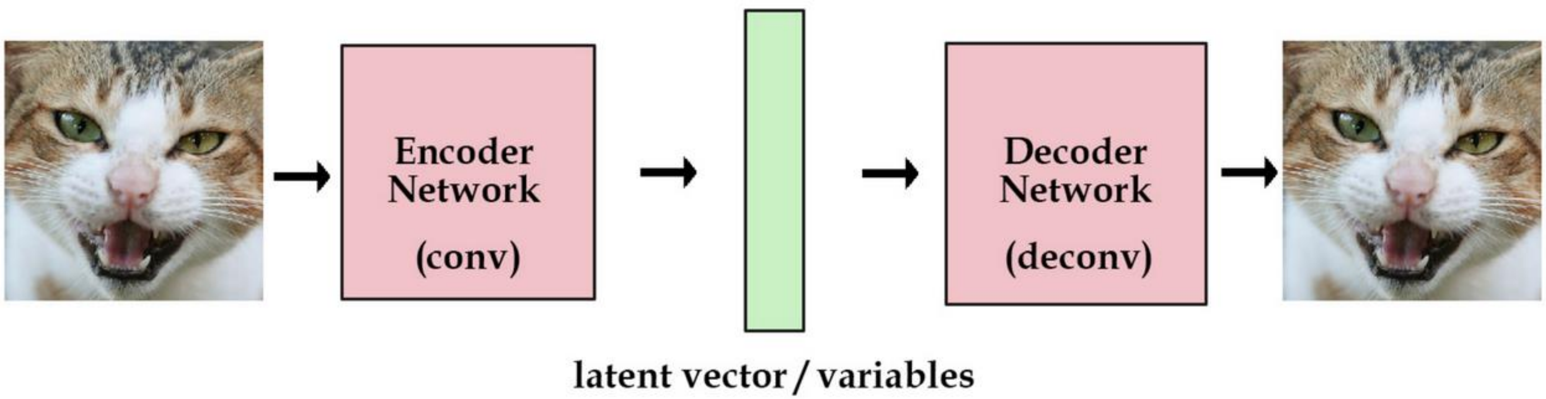
生成式随机网络（Generative stochastic networks， GSN）




How to eliminate Markov chains ?

# Background —— VAEs简介

同时出现了两个方法:
- VAEs : Variational Auto-Encoder
- GANs: Generative Adversarial Networks

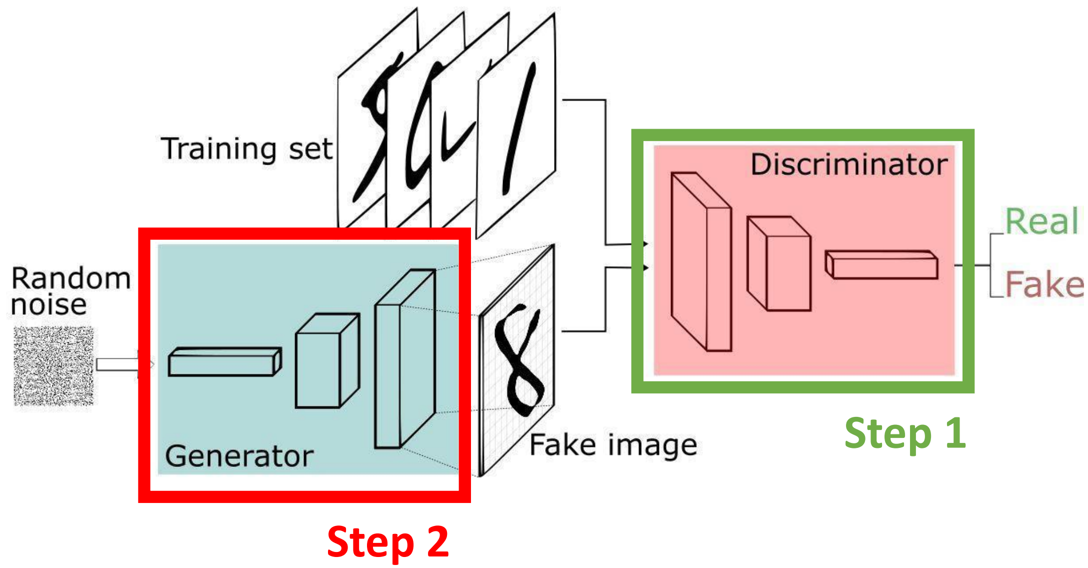



VAE 结构

强迫其服从某已知分布，如高斯分布

事成之后，每从高斯分布中抽取一个样本，我们就得到了一个新的猫的图片！

# GANs简介



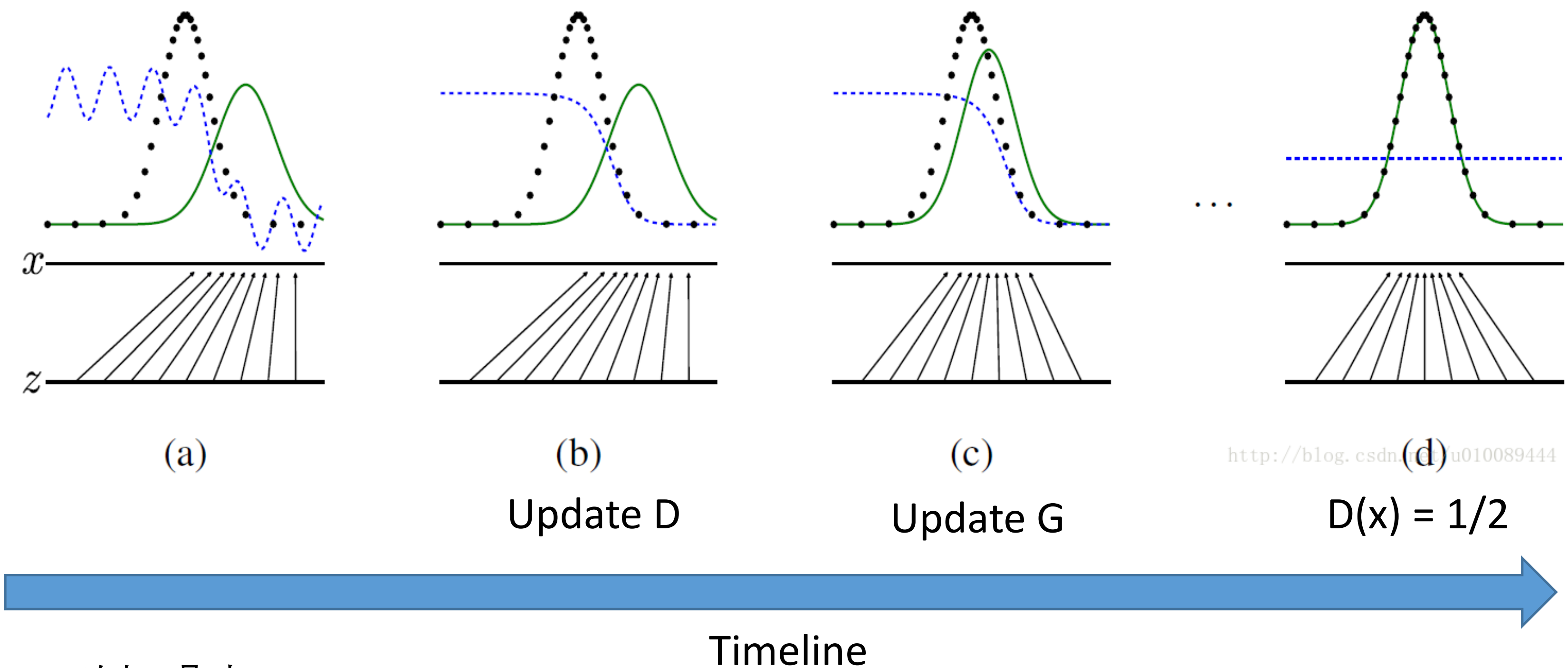

Update D: $\max_D V(D,G) = E_{x \sim p_{data}(x)}[log(D(x))] + E_{z \sim p_z(z)}[log(1 - D(G(z)))]$

Update G: $\min_G V(D,G) = E_{z \sim p_z(z)}[log(1 - D(G(z)))]$



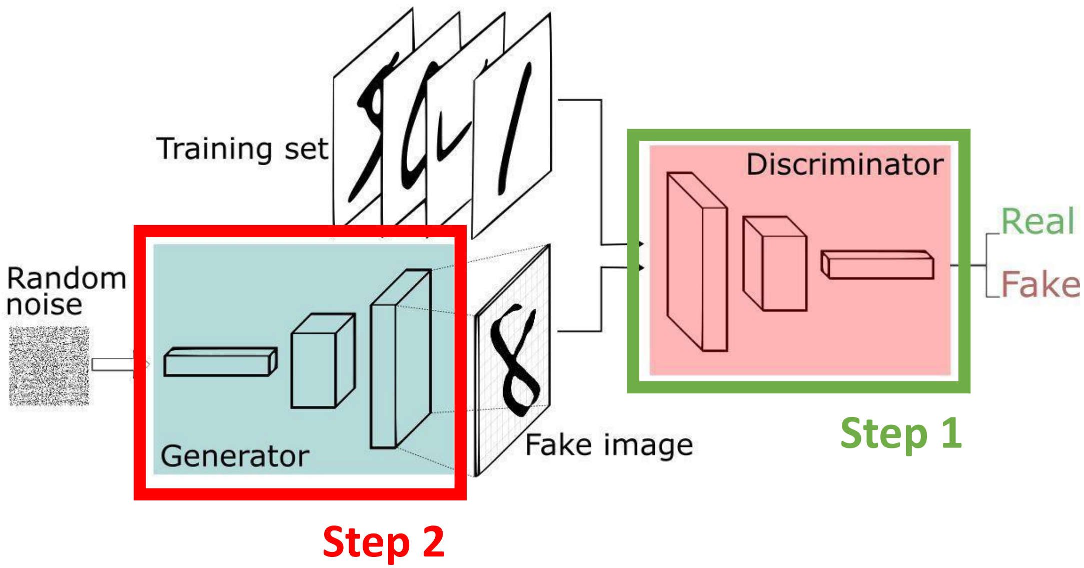
原文中合并形式: $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$

# GANs简介




$z$: 随机噪声
箭头： $x = G(z)$
蓝色： Discriminator
黑色： 真实分布
绿色： 生成分布

# GANs简介

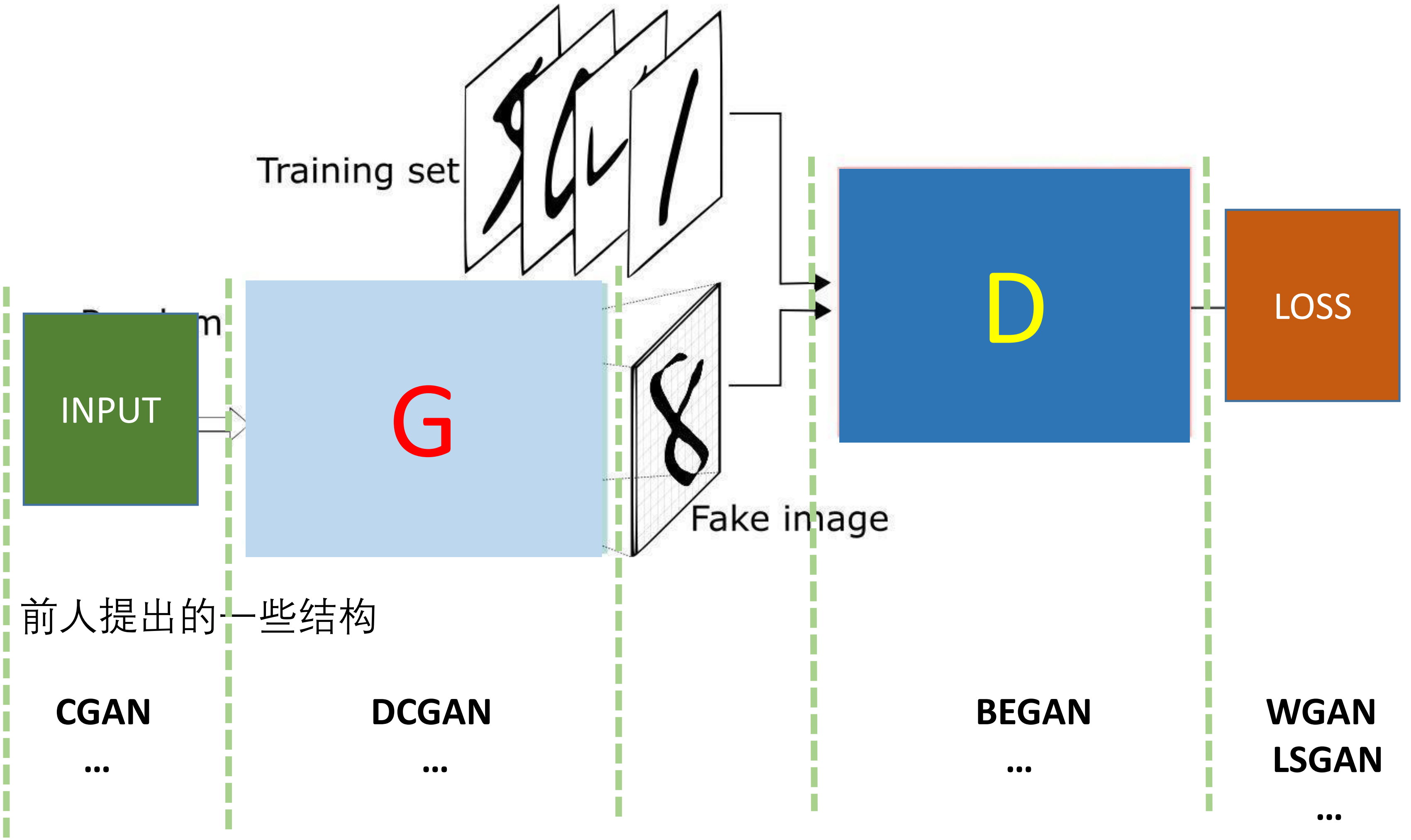



Update D: $$\max_D V(D,G) = E_{x\sim p_{data}(x)}[log(D(x))] + E_{z\sim p_z(z)}[log(1-D(G(z)))]$$

Update G: $$\min_G V(D,G) = E_{z\sim p_z(z)}[log(1-D(G(z)))]$$

原文中合并形式: $$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

# 如果我们来设计GAN，哪些部分是可以调整的呢？



# GANs优缺点

- Advantage:

不再需要马尔可夫链

数据并不直接更新模型，仅使用D的梯度信号更新G的网络参数

表示一些很尖锐或者衰减型的分布

- Disadvantages：

G和D的训练必须同步(训练困难)

对于不同的z，可能都生成同样的x，多样性下降（WGAN解决）

# GAN应用一：语音合成

- Speech Synthesis:

Automatically synthesize speech waveform

- Application

1. TTS ( Text to Speech )

2. VC ( Voice Conversation )

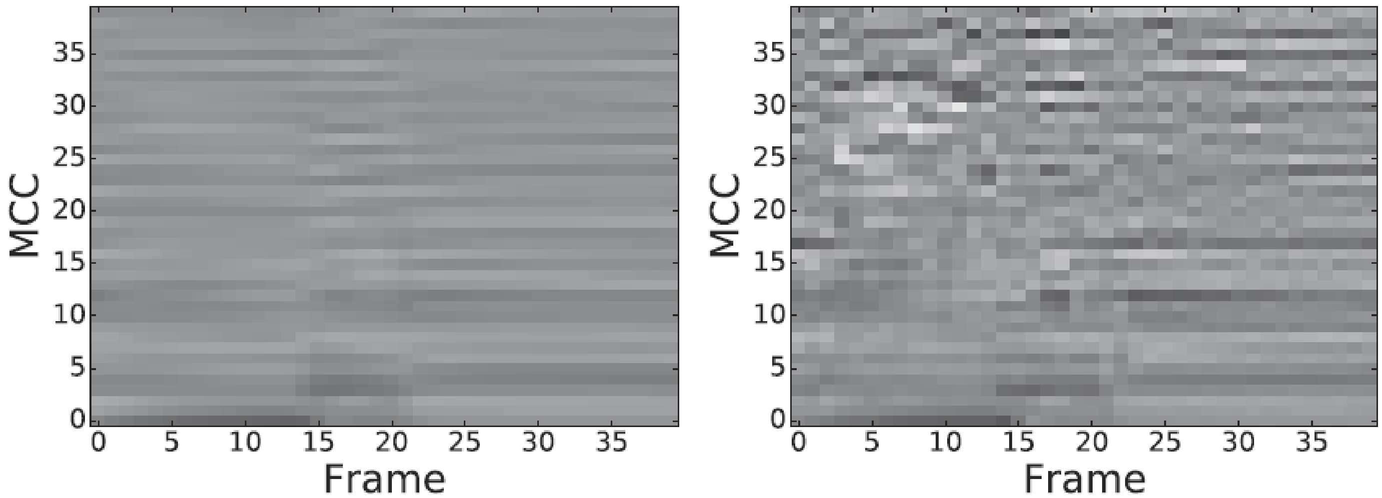
*modifying the speech signal of one speaker (source speaker) so that it sounds as if it had been pronounced by a different speaker (target speaker)

# 传统的做法

## HMM & DNN

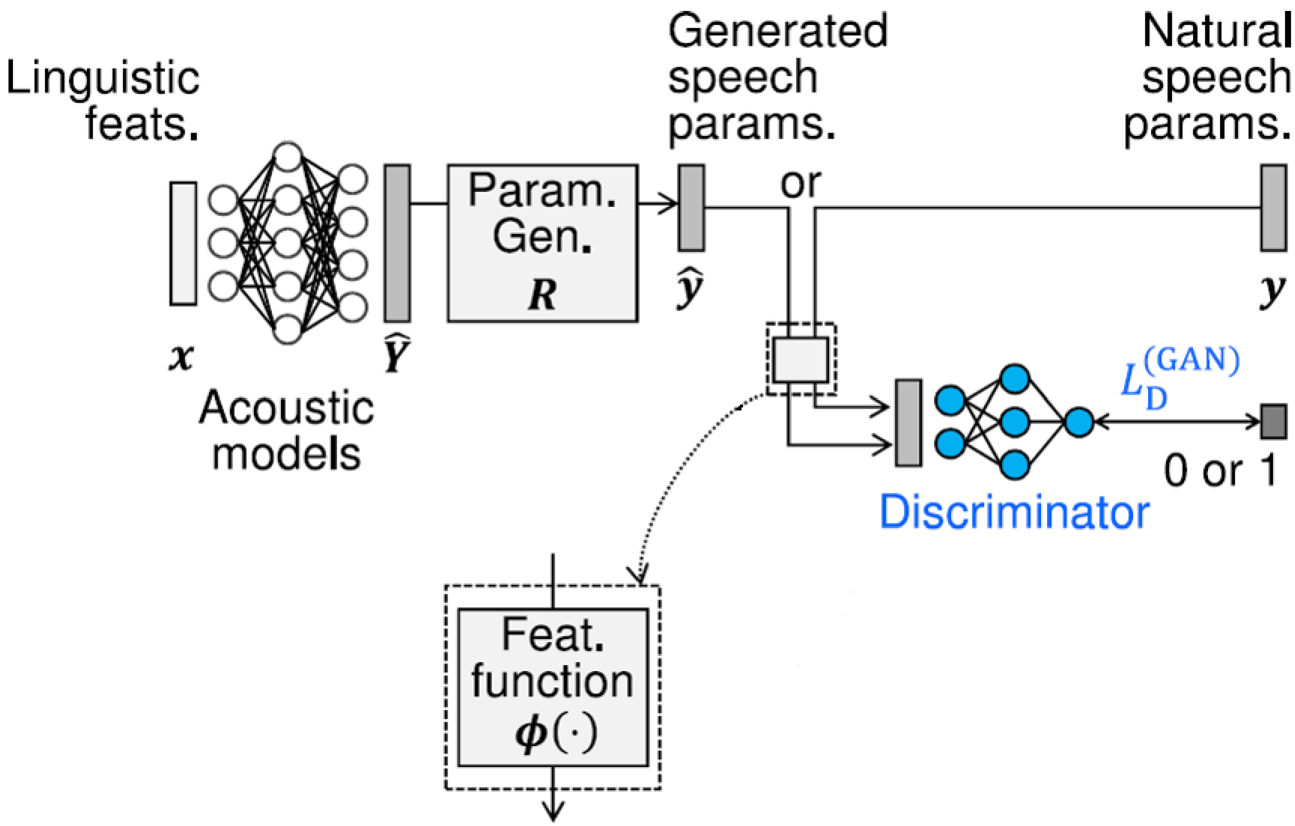
Problem：

- Vocoding
- Accuracy of acoustic models
- over-smoothing

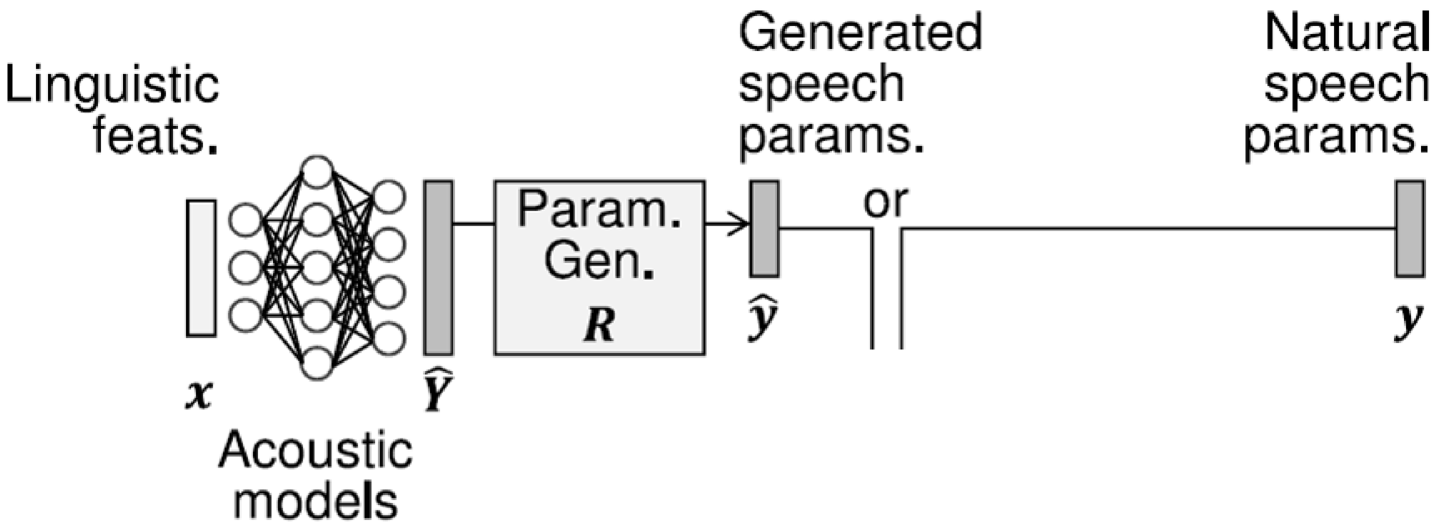



(a) SYN (b) NAT

# 工作1:
# Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks (TASLP 2017)



# Convolutional DNN-Based




# MSE (Mean Square Error)

**Natural:**

$$y = \left[y_1^T, y_2^T, \dots y_t^T, \dots y_T^T\right]^T$$

$$Y_t = [y_t^T, \Delta y_t^T, \Delta\Delta y_t^T]^T$$

$$Y = \left[Y_1^T, Y_2^T, \dots Y_t^T, \dots Y_T^T\right]^T$$

**Generative:**

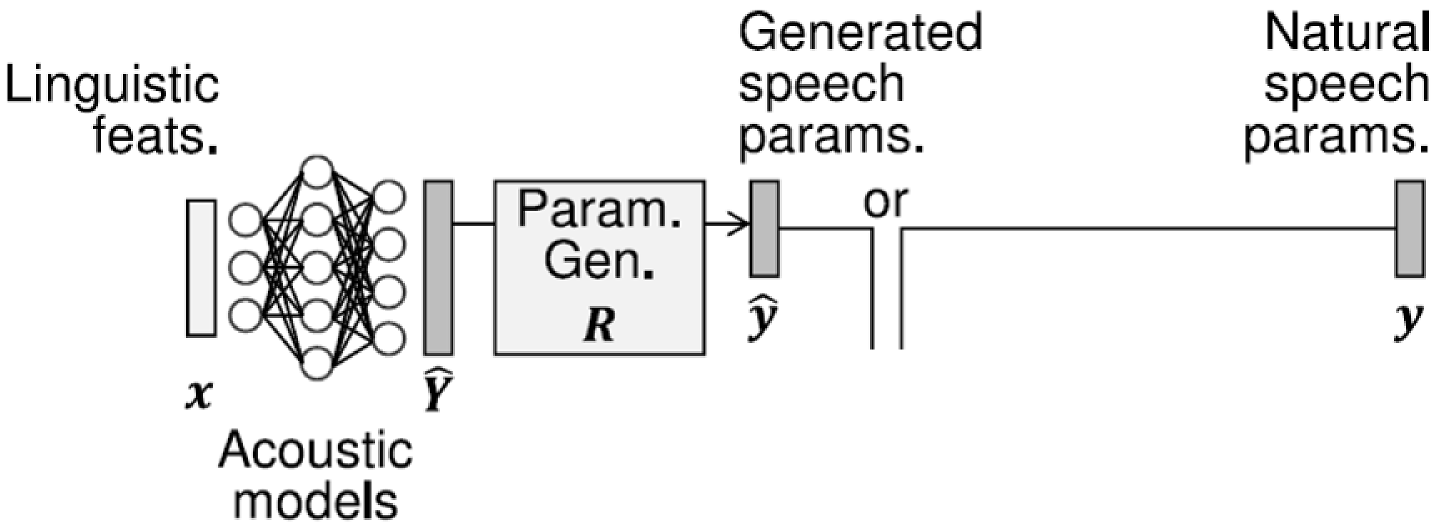
$$\hat{y} = \left[\hat{y}_1^T, \hat{y}_2^T, \dots \hat{y}_t^T, \dots \hat{y}_T^T\right]^T$$

$$\hat{Y} = \left[\hat{Y}_1^T, \hat{Y}_2^T, \dots \hat{Y}_t^T, \dots \hat{Y}_T^T\right]^T$$

**Loss:**

$$L_{MSE}\left(Y, \hat{Y}\right) = \frac{1}{T}(\hat{Y} - Y)^T(\hat{Y} - Y)$$

# Convolutional DNN-Based




# MGE (Minimum Generation Error)

**Natural:**

$$y = \left[y_1^T, y_2^T, \dots y_t^T, \dots y_T^T\right]^T$$

**Generative:**

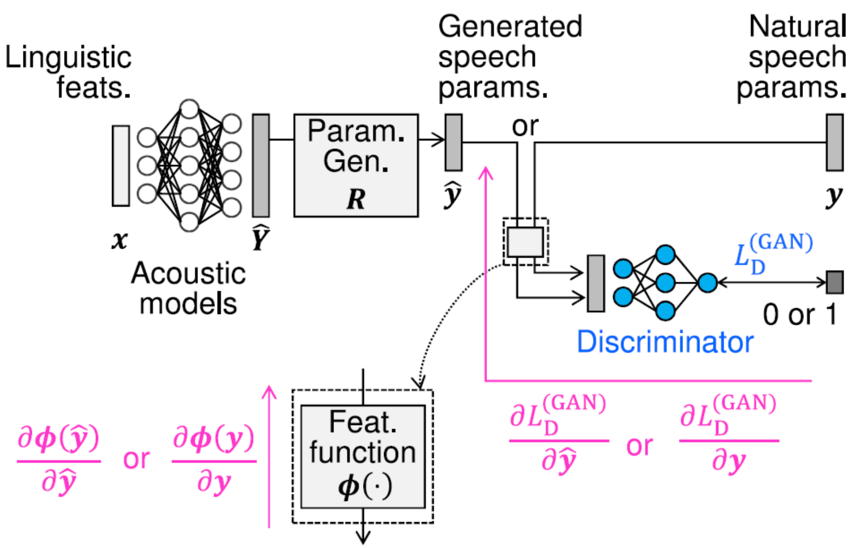
$$\hat{Y} = \left[\hat{Y}_1^T, \hat{Y}_2^T, \dots \hat{Y}_t^T, \dots \hat{Y}_T^T\right]^T$$

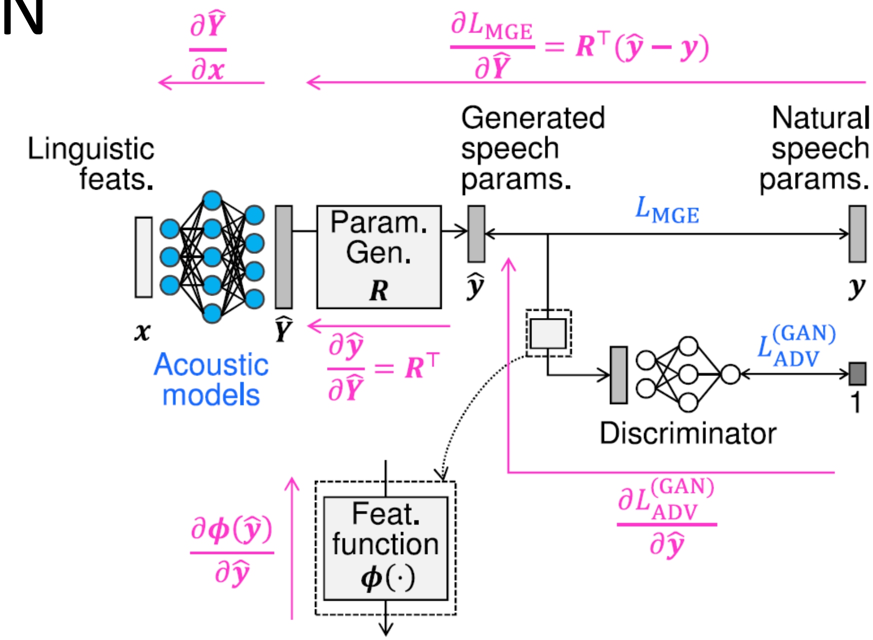
$$R = (W^T \Sigma^{-1} W)^{-1}\, W^T \Sigma^{-1}$$

**Loss:**

$$L_{MGE}(y, \hat{y}) = \frac{1}{T}(R\hat{Y} - y)^T (R\hat{Y} - y)$$

# DNN-Based incorporating GAN




Update D:

$$L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T}\log\frac{1}{1+\exp\left(-D\left(\boldsymbol{y}_t\right)\right)} - \frac{1}{T}\sum_{t=1}^{T}\log\left(1-\frac{1}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}\right)$$

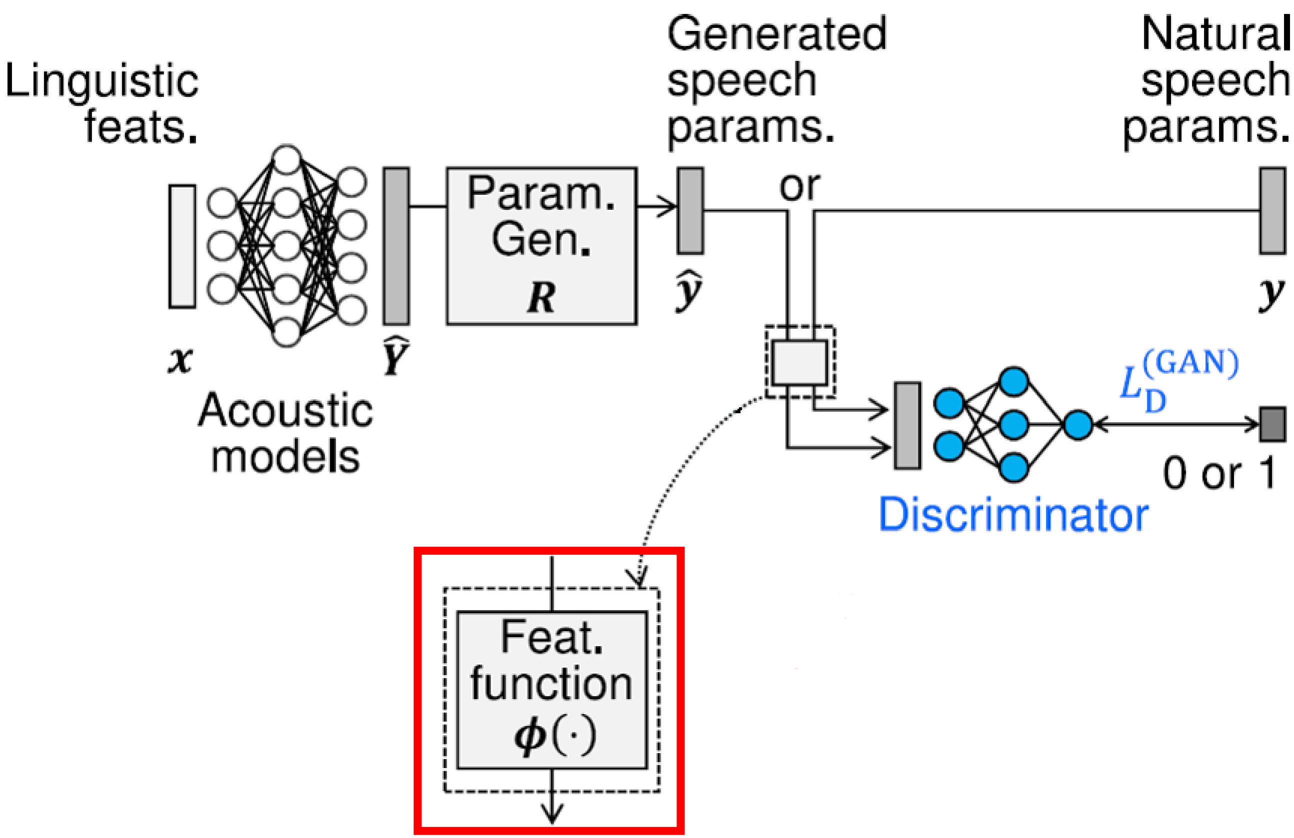
$$L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T}\log\frac{1}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}.$$

Update G:

$$L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \boxed{L_{\mathrm{MGE}}(\boldsymbol{y}, \hat{\boldsymbol{y}})} + \omega_{\mathrm{D}}\frac{E_{L_{\mathrm{MGE}}}}{E_{L_{\mathrm{ADV}}}} L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$$

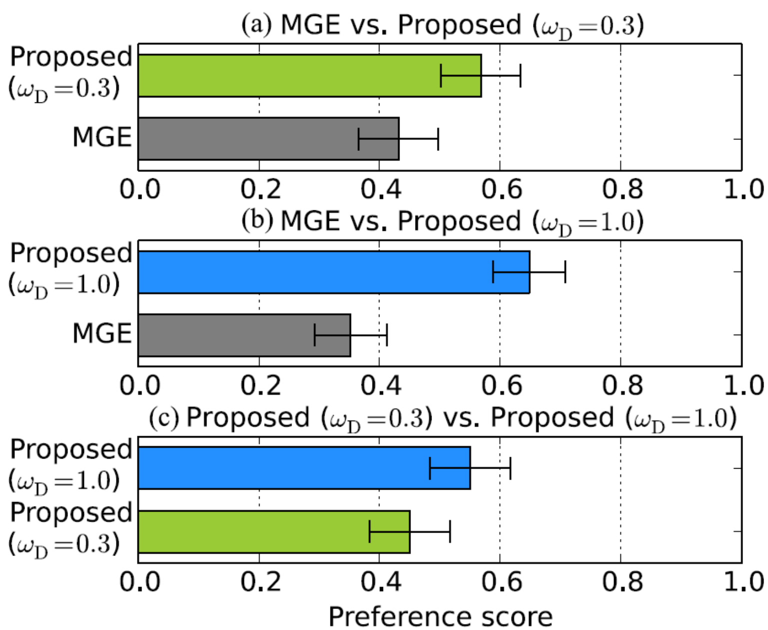
# Feature Function

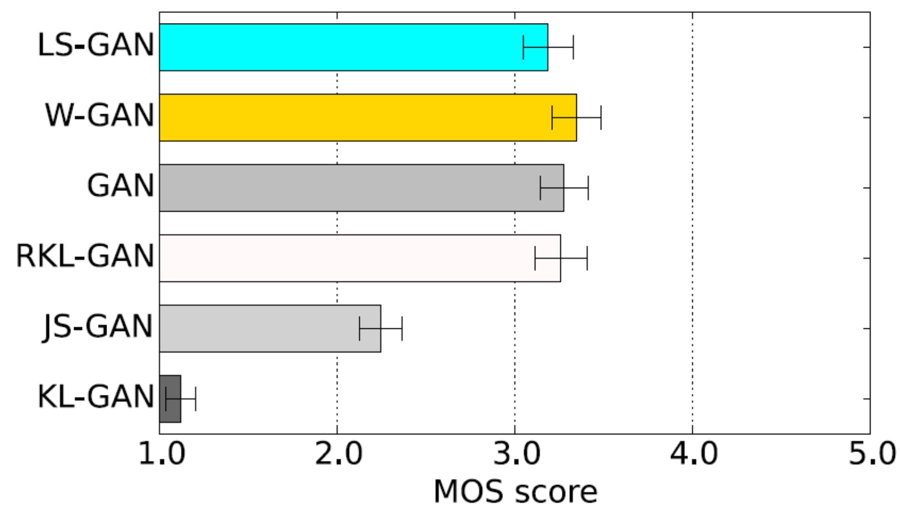
在TASLP 2015&2016发表的文章中，有学者针对声音反诈骗（anti-spoofing）的任务提出了一些方法，作者认为可以加入到discriminator之前来提升discriminator的性能，从而最终提高Generator的性能



# Result

文中作者做了非常多的对比实验，以下抽取其中两个

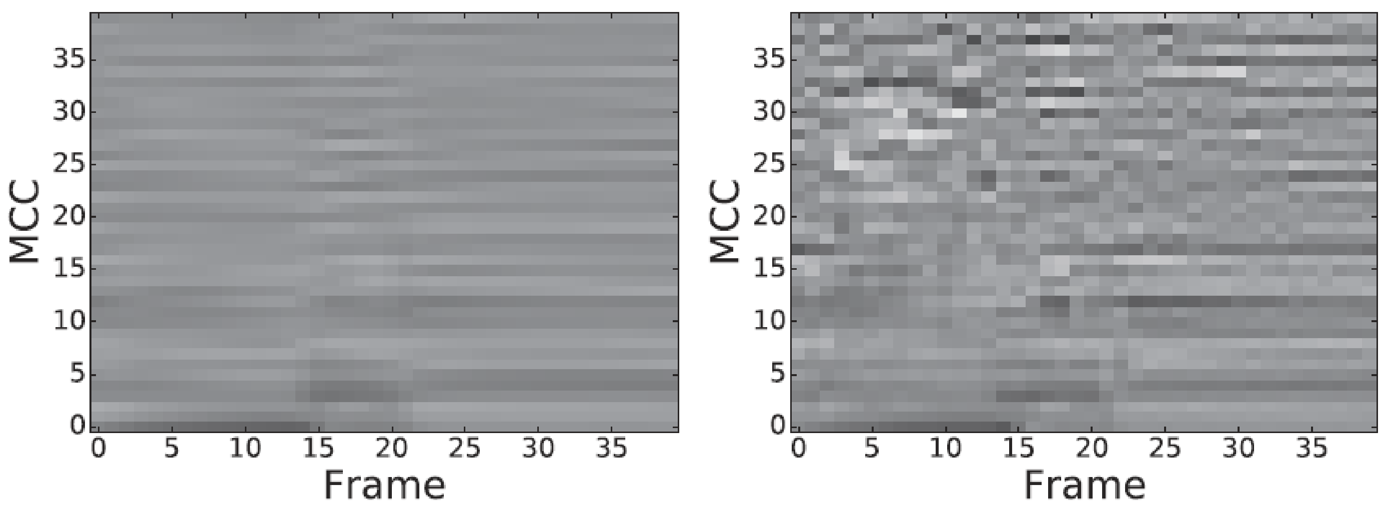



比较GAN与baseline以及超参数$\omega_D$的影响

比较不同的散度(Divergence)

思考，实际上两个语谱图已经比较接近了，只是(a)是over-smoothing，那么是不是可以直接在(a)上面做优化?



(a) SYN (b) NAT

# 工作2:
# Generative adversarial network-based postfilter for statistical parametric speech synthesis (ICASSP 2017)

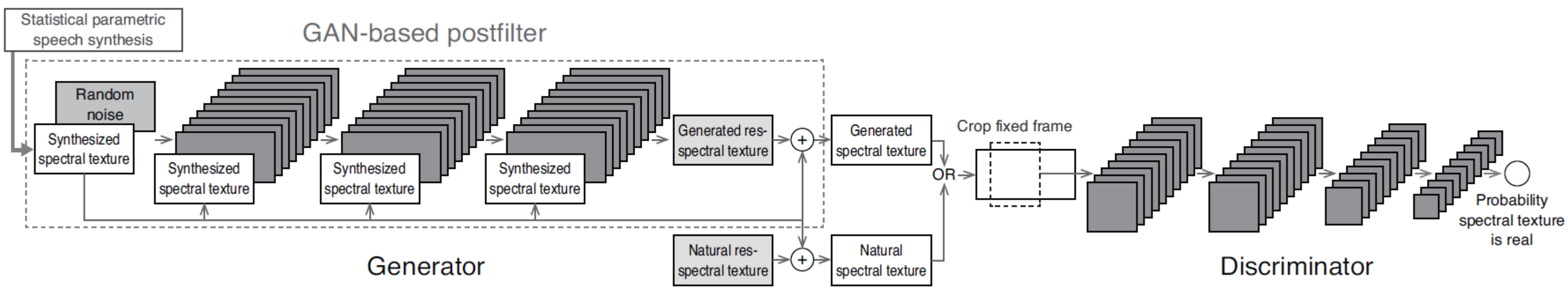



**Fig. 2**. System overview of proposed GAN-based postfilter.

Reconstructing detailed spectral structures in both the time and frequency directions simultaneously

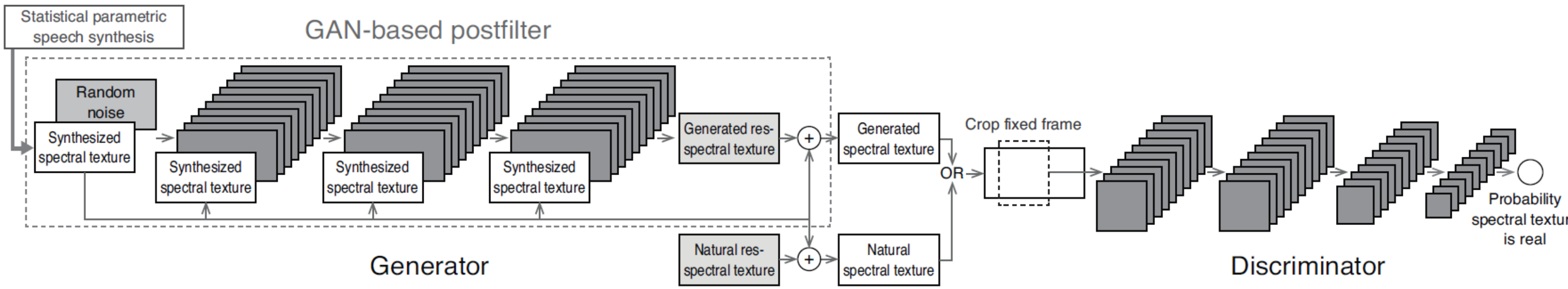



**Fig. 2**. System overview of proposed GAN-based postfilter.

$$\min_G \max_D \mathbb{E}_{x,y\sim P_{\mathrm{Data}}(x,y)}[\log D(x,y)] + \mathbb{E}_{z\sim P_{\mathrm{Noise}}(z),y\sim P_y(y)}[\log(1-D(G(z,y),y))].$$

We use $y$ as a synthesized spectral texture.

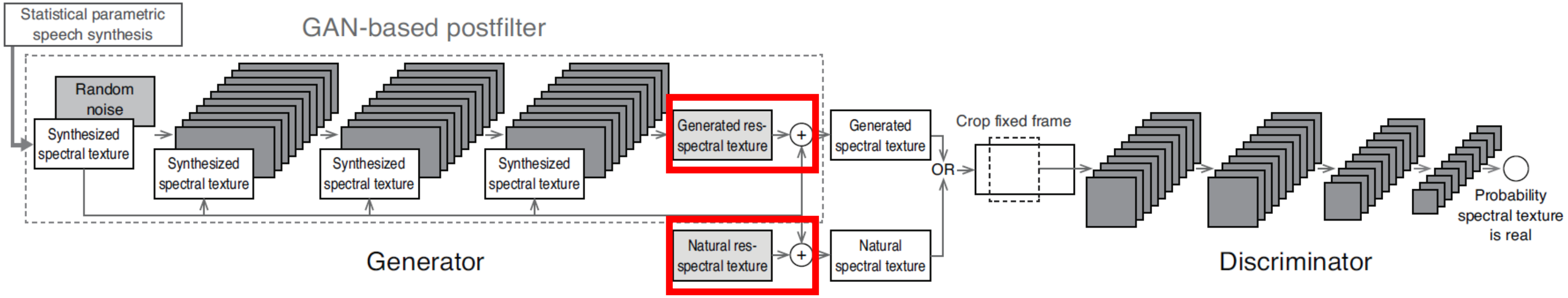



**Fig. 2**. System overview of proposed GAN-based postfilter.

- Residual representation
- Convolutional architecture （FCN）

**Table 1**. Network architectures for GAN-based postfilter.

| **Generator (Input:** $D \times T$ Mel-cepstrum + $D \times T$ noise) |
|---|
| $5 \times 5$ 128 conv., ReLU + input Mel-cepstrum |
| $5 \times 5$ 256 conv., ReLU + input Mel-cepstrum |
| $5 \times 5$ 128 conv., ReLU + input Mel-cepstrum |
| $5 \times 5$ 1 conv. |

| **Discriminator (Input:** $D \times \boxed{T_c}$ Mel-cepstrum) |
|---|
| $5 \times 5$ 64 conv., LReLU |
| $5 \times 5$ 128 conv. $\downarrow$, LReLU |
| $3 \times 3$ 256 conv. $\downarrow$, LReLU |
| $3 \times 3$ 128 conv. $\downarrow$, LReLU |
| 1 fully connected, sigmoid |

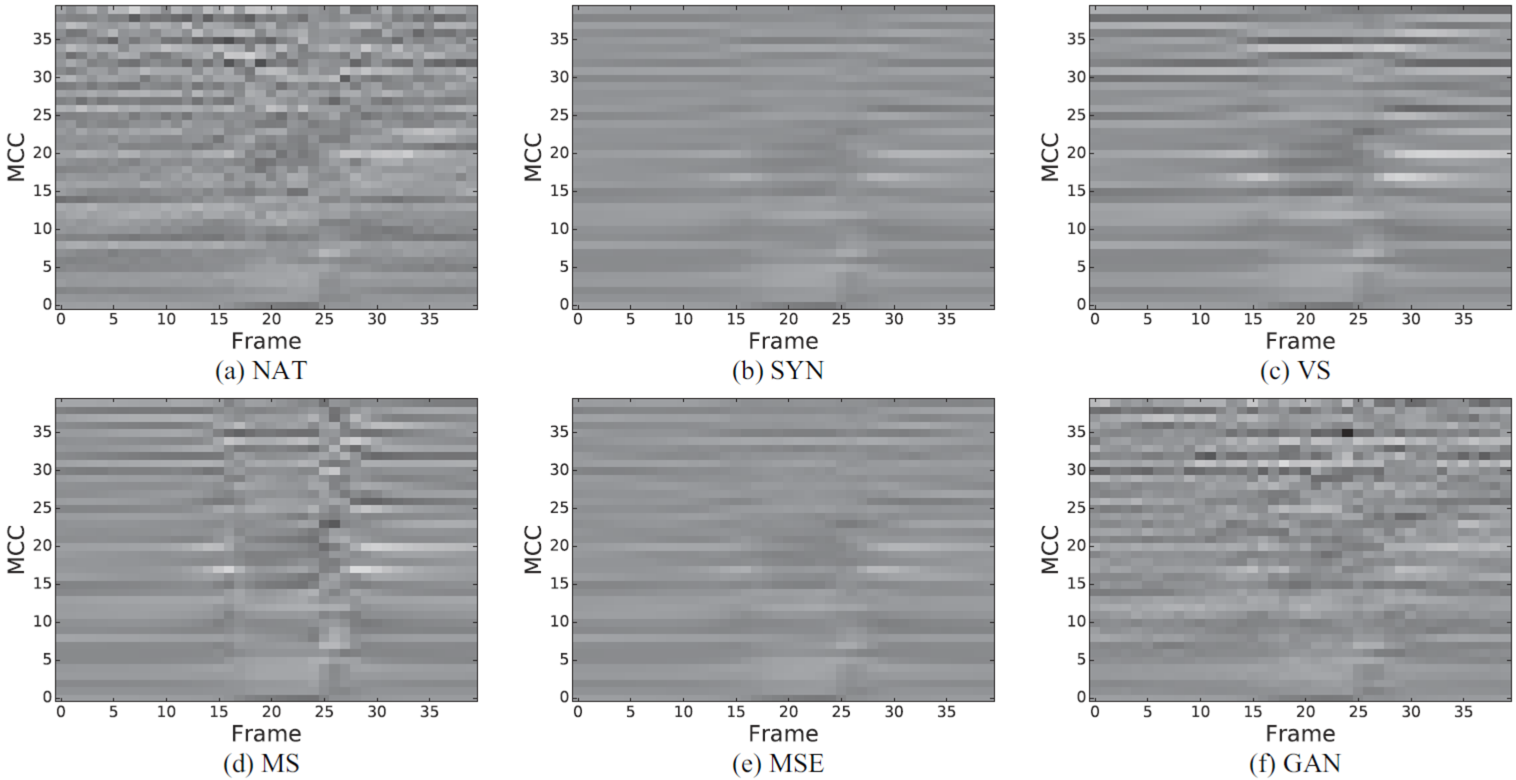
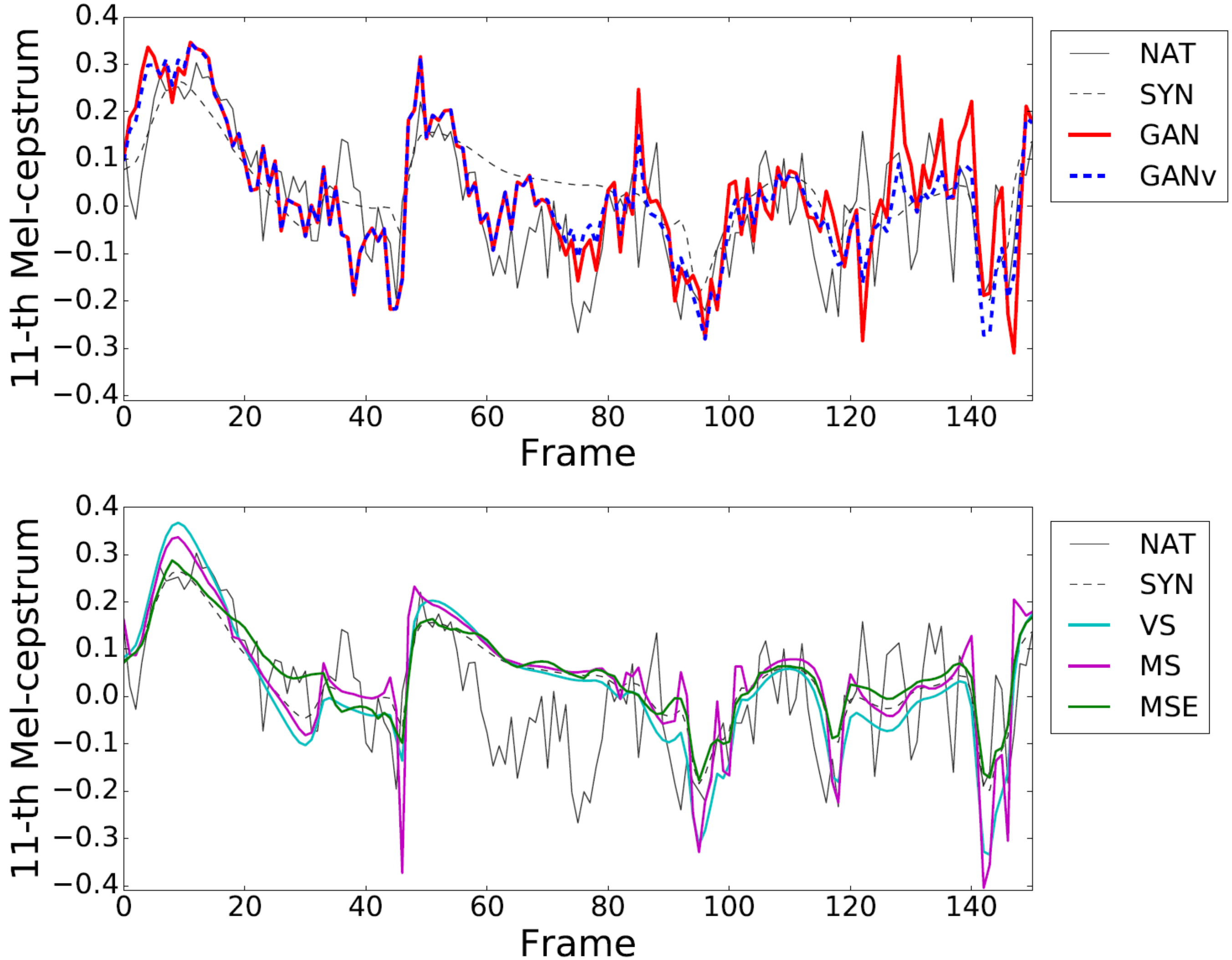



**Fig. 3**. Comparison of spectral textures generated by different methods.[1]

**NAT:** extracted from a natural speech by STRAIGHT
**SYN:** generated by DNN-based statistical parametric speech synthesis
**VS:** variance scaling-based postfilter
**MS:** modulation spectrum-based postfilter
**MSE:** DNN-based postfilter with mean squared error as the loss function
**GANv:** applying the GAN-based postfilter only in voiced




As shown, the trajectories of SYN, VS, MS, and MSE are too smooth, but GAN and GANv can predict the trajectory that has a similar complexity to the natural one.

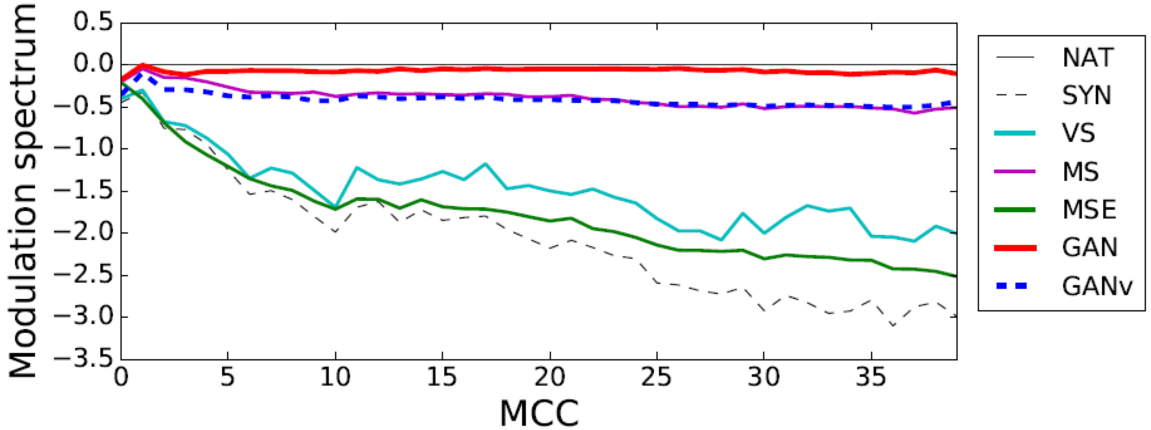



**Fig. 5**. Averaging difference in modulation spectrum per Mel-cepstral coefficient for different methods compared to natural speech.

**Table 2**. Average preference score (%) with 95% confidence intervals. Bold font indicates the number is over 30%.

| | Former | Latter | Neutral |
|---|---|---|---|
| GAN vs. SYN | **56.5** ± 4.9 | 22.0 ± 4.1 | 21.5 ± 4.0 |
| GAN vs. GANv | 11.3 ± 3.1 | **37.3** ± 4.8 | **51.5** ± 4.9 |
| GAN vs. NAT | 16.8 ± 3.7 | **53.5** ± 4.9 | 29.8 ± 4.5 |
| GANv vs. NAT | **30.3** ± 4.5 | **34.5** ± 4.7 | **35.3** ± 4.7 |

Human evaluation

# GAN应用二：语音增强

- Speech Enhancement:

  Noise reduction for speech

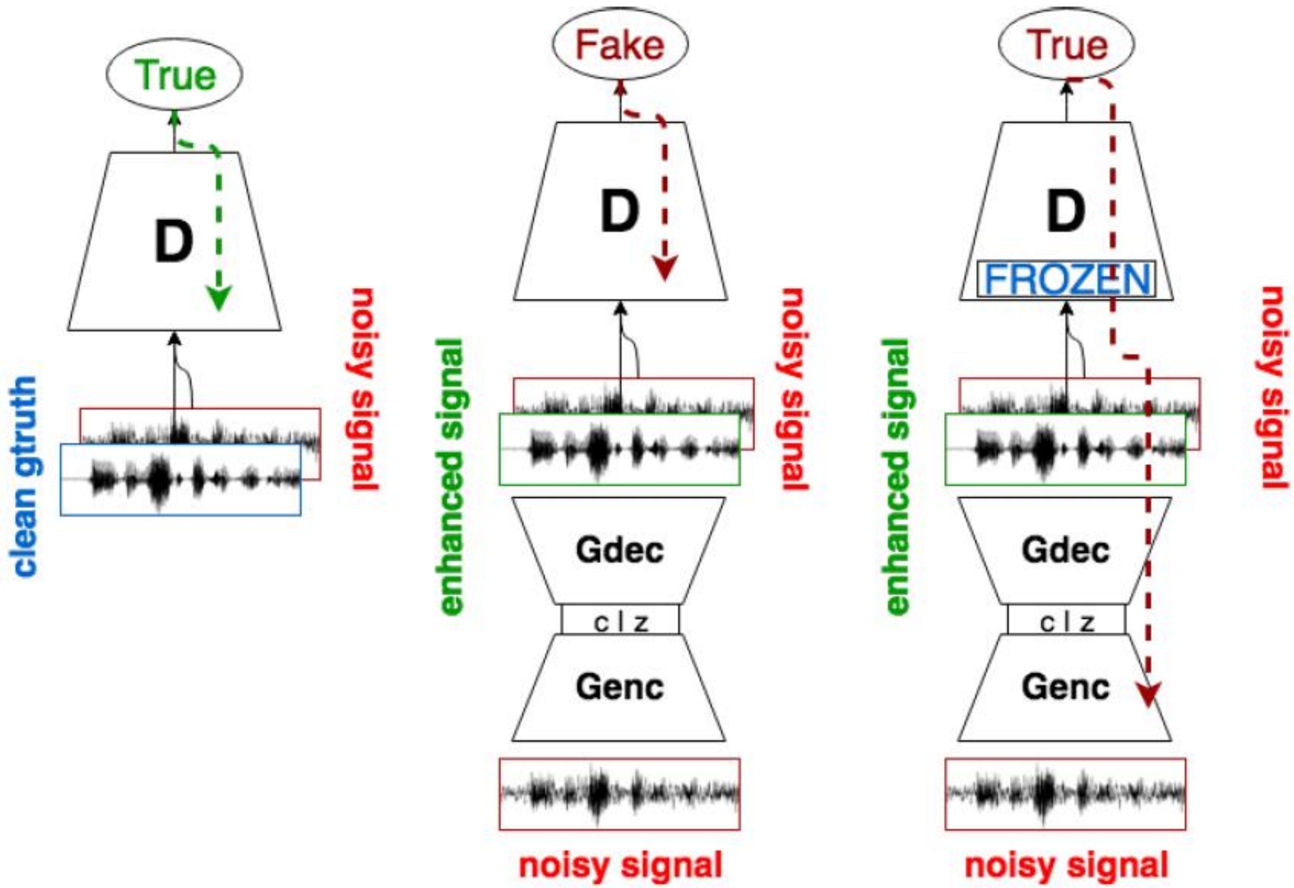
传统的做法：

- spectral subtraction
- Wiener filtering
- statistical model-based method
- subspace algorithms

## 工作3：

**SEGAN: Speech Enhancement Generative Adversarial Network**

(Interspeech2017)



Motivation：

以前的做法有许多限制:

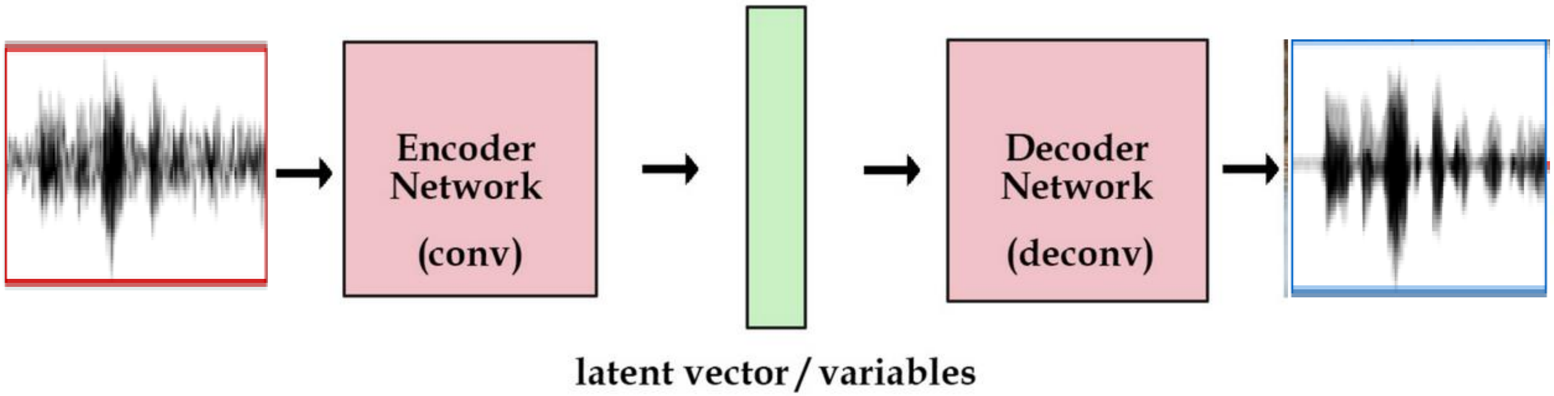
- Use of fixed-length analysis window
- Linear filter
- Gaussian process assumption

https://arxiv.org/pdf/1609.03499.pdf page14

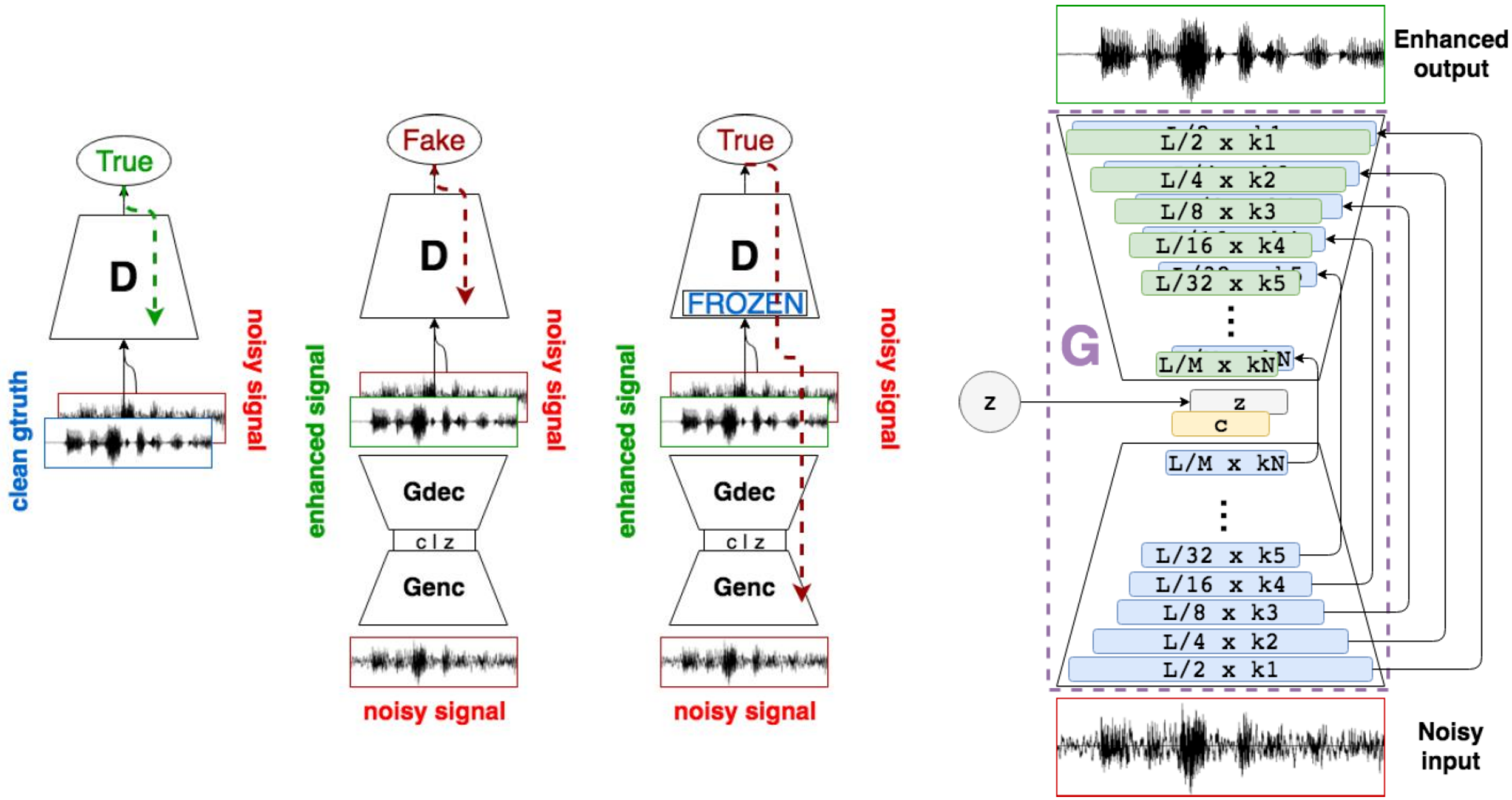
SEGAN：

- It works end-to-end, with the raw audio and no hand-crafted features are extracted
- same shared parametrization and more generalizable
- No recursive operation like in RNNs ,no causality

# Auto-Encoder




Add adversarial component?




Condition

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}[(D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})-1)^2]+$$

$$+\boxed{\lambda \left\| G(\mathbf{z},\tilde{\mathbf{x}}) - \mathbf{x} \right\|_1}.$$

和语音合成的工作一样

# Result

| Metric | Noisy | Wiener | SEGAN |
|---|---|---|---|
| PESQ | 1.97 | 2.22 | 2.16 |
| CSIG | 3.35 | 3.23 | 3.48 |
| CBAK | 2.44 | 2.68 | 2.94 |
| COVL | 2.63 | 2.67 | 2.80 |
| SSNR | 1.68 | 5.07 | 7.73 |

| Perference | Former | Latter | Neutral |
|---|---|---|---|
| Noisy VS SEGAN | 8% | **67%** | 25% |
| Wiener VS SEGAN | 23% | **53%** | 24% |

在今年发表的论文中，作者还在不同的文化和不同的噪声类型上做了实验

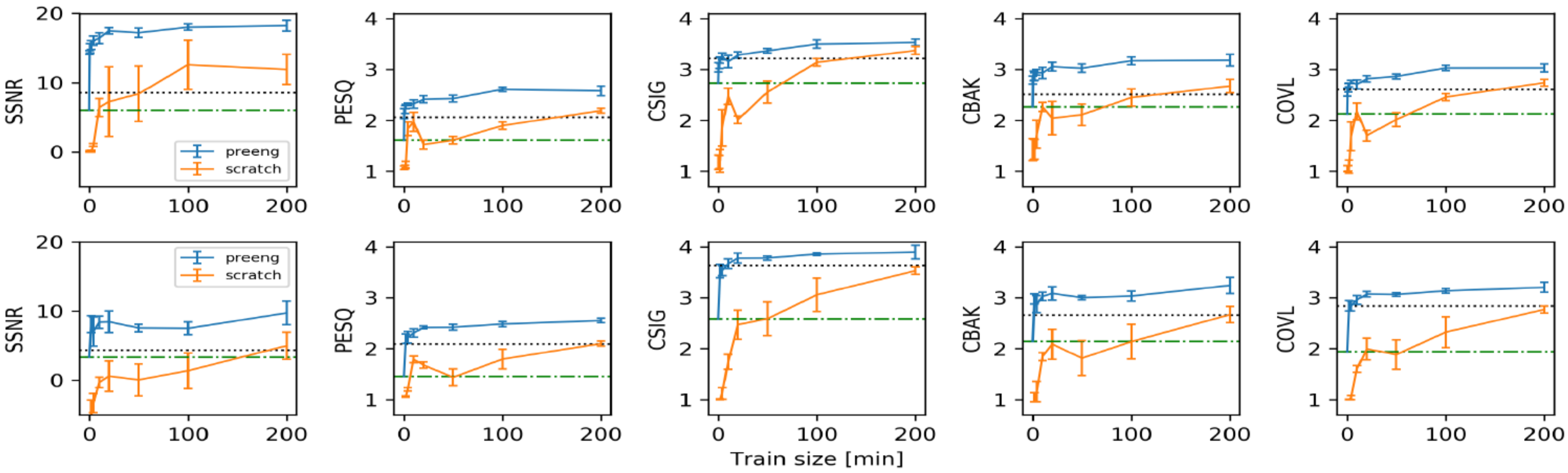
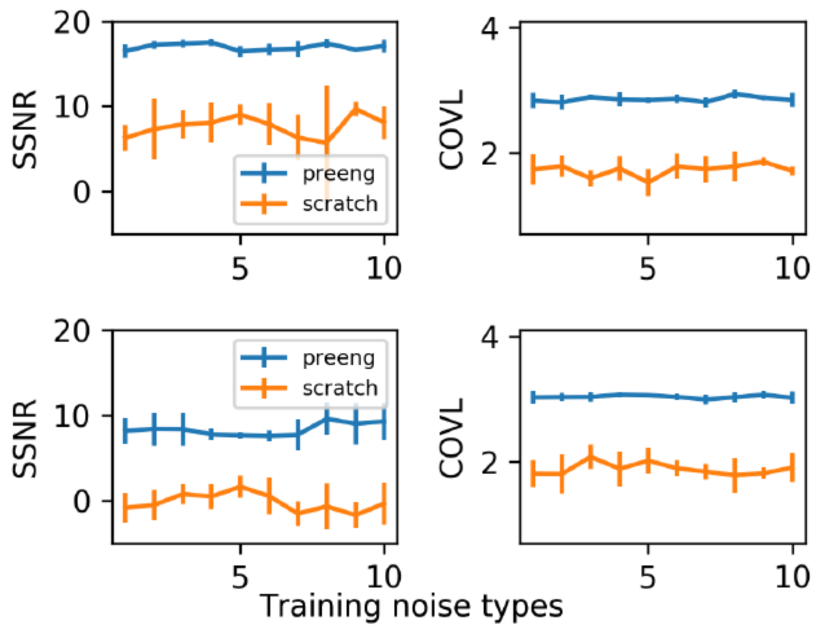



**Fig. 2.** Objective metrics for Catalan (top row) and Korean (bottom row). Blue line (preeng): Pre-trained with English. Orange line (scratch): trained from scratch. Green dashed line: SEGAN level without fine tunning. Black dash-dotted line: Noisy level.

**先在英语上训练，然后用另外一个语言finetune效果最佳（蓝色）**

**如果不finetune，直接训练的话需要较大的数据量（橙色）**

**噪声实验：**




Calatan(加泰罗尼亚语)

Korean

**有趣的现象，在干净的音频上叠加不同种类的噪声，噪声数量的增加似乎没有影响降噪的效果。**

# GAN应用三：情感分析

- Emotion Recognition:

  discrete：  happy，  angry，  ……..

  continuous :  Arousal ,  Valence

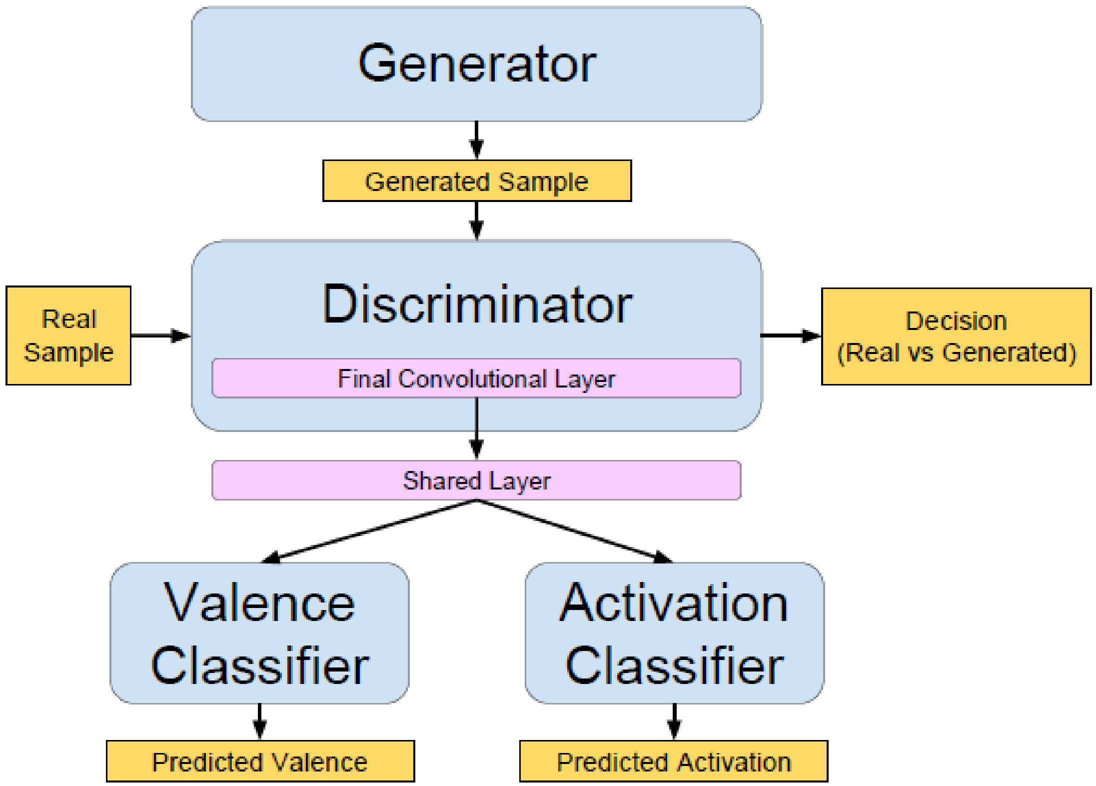
- Limitation:

Little scale of labeled data

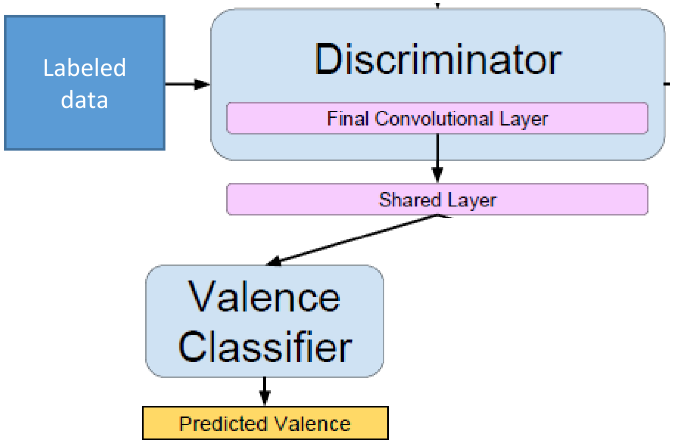
Can we use GANs to implement semi-supervised training ?

## 工作4：

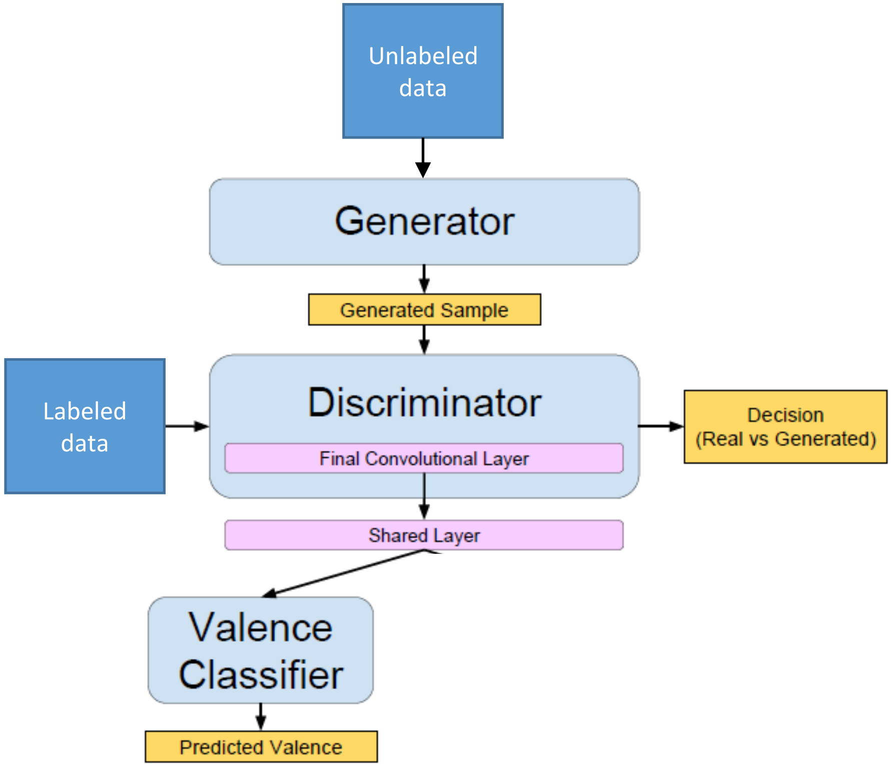
**Learning representations of emotional speech with deep convolutional generative adversarial networks（ICASSP 2017）**



Simple supervised training for valence recognition



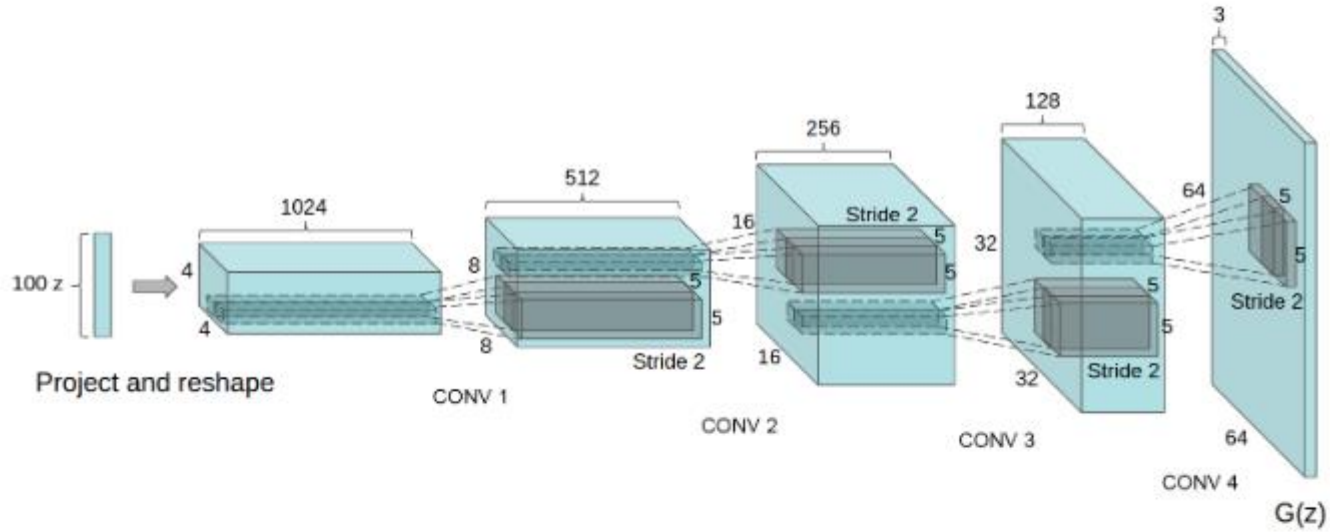

Semi-supervised training for valence recognition




$$\mathcal{L}_r(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\log \hat{y}_{r,n}$$

$$\mathcal{L}_f(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\log(1-\hat{y}_{g,n})$$

$$\mathcal{L}_g(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\log(\hat{y}_{g,n})$$




**本文的结构：DCGAN （Deep Convolution GANs）**

DCGAN作者想弥补CNN在supervised 和 unsupervised之间的gap。作者提出了将CNN和GAN相结合的DCGAN,并展示了它在unsupervised learning所取得的不俗的成绩

**本文直接套用了DCGAN的框架，区别：**

emotion classification a fully connected layer is attached to the final convolutional layer of the DCGAN’s discriminator

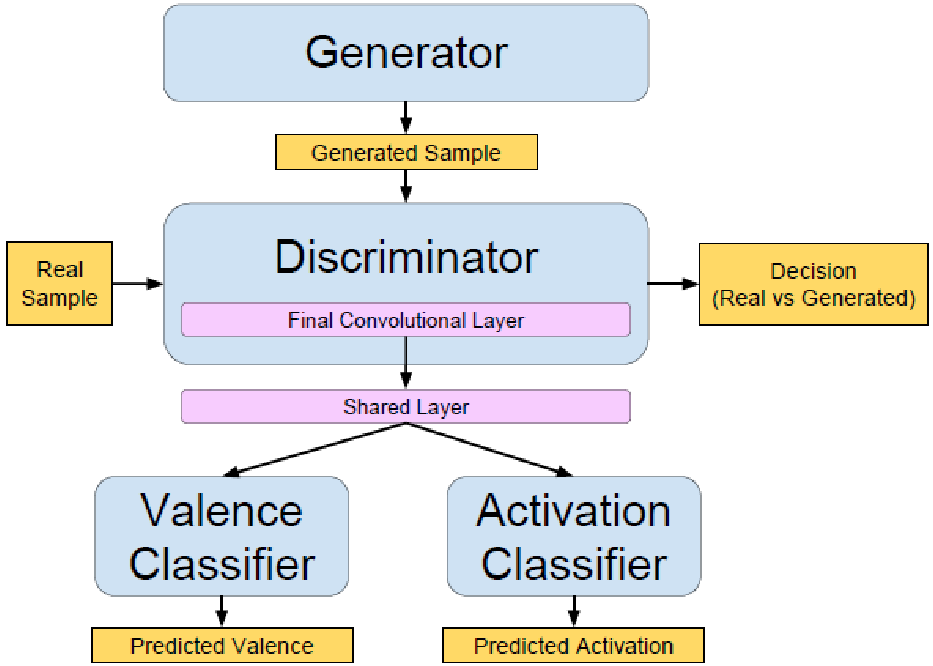
## 数据集

| | | | |
|---|---|---|---|
| IEMOCAP | 有标注 | 12hours | 交互式情绪捕捉 |
| AMI | 无标注 | 100hours | 大量会议中的声音和录像 |

## 预处理:

- 每个音频的时长不一样，所以需要从中抽取一个固定长度的区域。为了避免抽取的区域不是无声的，先从文本中抽一个词，然后以这个词的音频位置为中心确定这个区域
- 通过oversampling，迫使每类label的数据量差不多

# Result




| Model | Accuracy (5 class) | Accuracy (3 class) | Pearson Correlation ($\rho$ value) |
|---|---|---|---|
| BasicCNN | 38.52% | 46.59% | 0.1639 |
| MultitaskCNN | 36.78% | 40.57% | 0.0737 |
| **BasicDCGAN** | **43.88%** | **49.80%** | **0.2618** |
| MultitaskDCGAN | 43.69% | 48.88% | 0.2434 |

# Thanks

由于时间限制，很多细节不一一详述，有兴趣的同学可以看原文