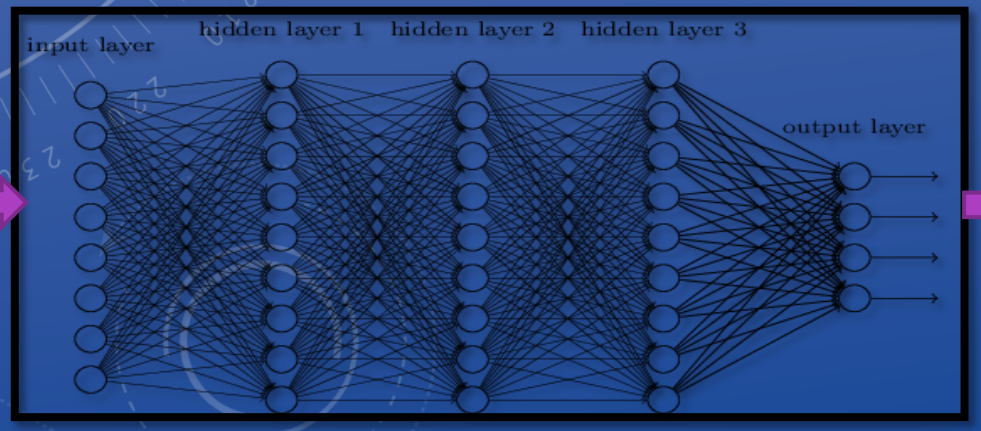
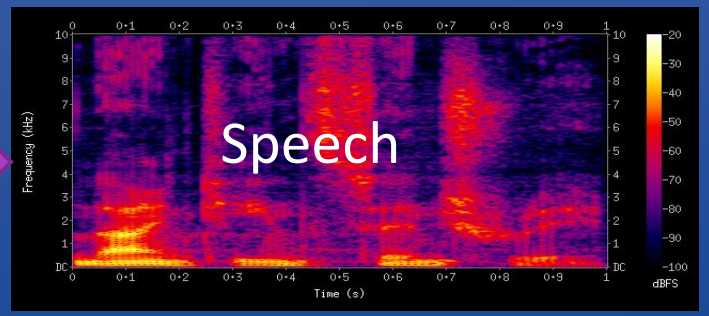
Deep Generative Modeling for Speech Synthesis and Sensor Data Augmentation

Deep Generative Neural Network

Text





Praveen Narayanan
Ford Motor Company

FORD RESEARCH AND INNOVATION CENTER
Palo Alto

deepDSP
Vehicle & Enterprise Science

PROJECT DESCRIPTION

- Use of DNNs increasingly prevalent as a solution for many data intensive applications
- Key bottleneck – requires large amounts of data with rich feature sets

  => Can we produce synthetic, realistic data?
- **This work aims to leverage state of the art DNN approaches to produce synthetic data that are real world representative**
  - Deep Generative modeling:
    - A new research approach using DNN, came into vogue in the last three years
    - Examples: VAE, GAN, PixelRNN, Wavenet
  - VAE – Variational Autoencoder: Maximizing a variational objective bound + reparametrization
  - GAN – Generative Adversarial Nets: Adversarial learning with a discriminator

- Some application areas of generative models:
  - Data augmentation in missing data problems – e.g. when labels are missing or bad
  - Generating samples from high dimensional pdfs – e.g. producing rich feature sets
  - Synthetic data generation for simulation – e.g. reinforced learning in a simulated environment

TECHNICAL SCOPE

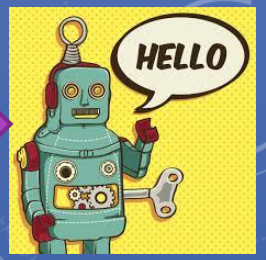
- Text to speech problem
  - Given text, convert to speech
  - Use to train ASR

- Produce speech from text with custom attributes

  Examples:
  Male vs female speech (voice conversion)
  Accented speech: English in different accents
  Multilanguage

- Sensor data augmentation
  - Effecting transformations on data
    - Rotations on point clouds
    - Generating data in adverse weather conditions

Hello



Nǐ huì shuō pǔtōnghuà ma

Do you speak Mandarin ?

Mandarin

Parrot Accent

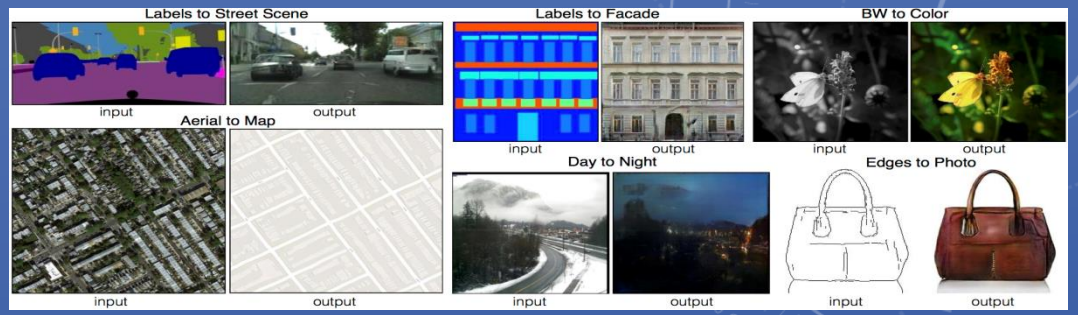
SCOPE OF THIS TALK

- Very brief introduction to generative models (GANs, VAEs, autoregressive models)
- Describe the text to speech problem (TTS)
- High level overview of “Tacotron” – a quasi end to end TTS system from google
- Speech feature processing
  - Different types of features used in speech signal processing
- Describe the CBHG network and our implementation
  - Originally proposed in the context of NMT
  - Used in Tacotron
- Voice conversion using VAEs
- Conditional variational autoencoders to transform images

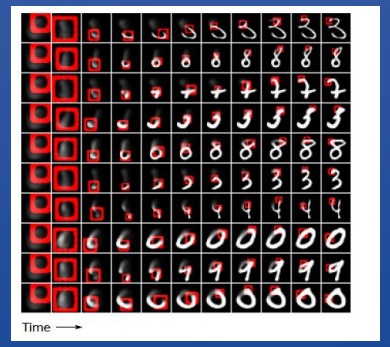
GENERATIVE MODELING "TOOLS"

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Autoregressive models
  - RNNs
    - Vanilla RNNs
    - Gated: LSTM, GRU, possibly bidirectional
    - Seq2seq + attention
  - Dilated convolutions
    - Wavenet, Bytenet, PixelRNN, PixelCNN

[Goodfellow; Kingma and Welling; Rezende and Mohamed; Van den Oord et al]



pix2pix



DRAW

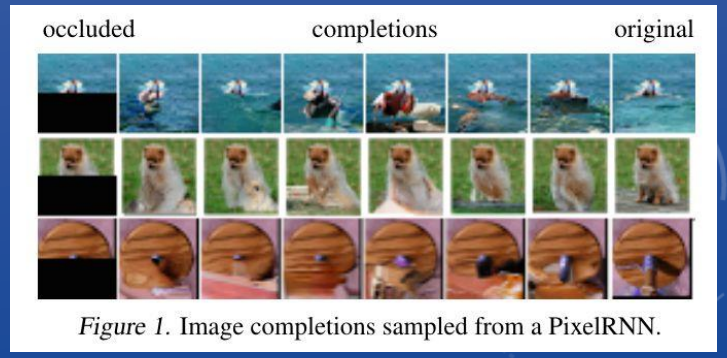


Figure 1. Image completions sampled from a PixelRNN.

PixelRNN

VARIATIONAL AUTOENCODER RESOURCES

Vanilla VAE

- Kingma and Welling
- Rezende and Mohamed

Semi Supervised VAE (SSL+conditioning, etc.)

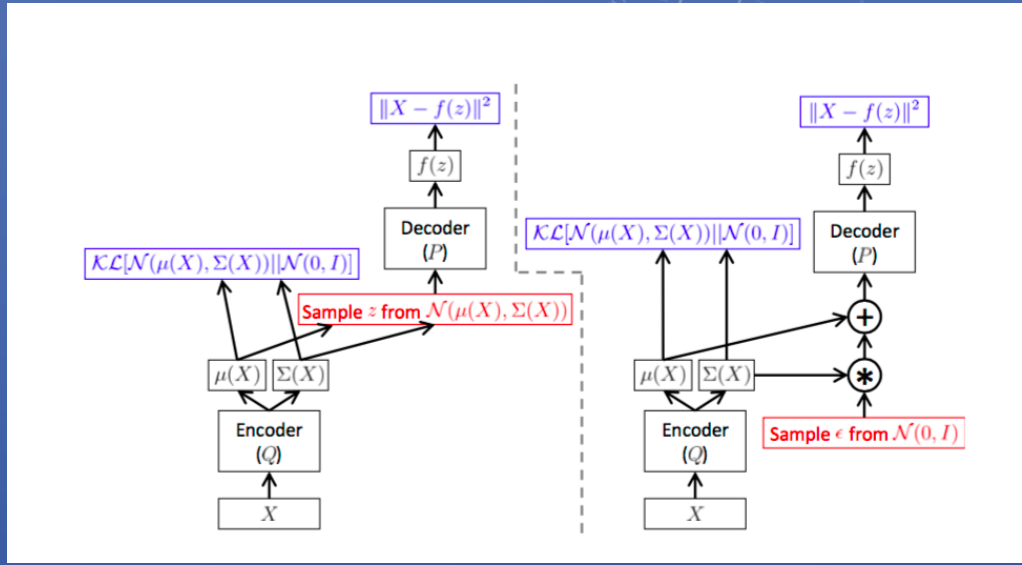
- Kingma et. al

Related

- DRAW (Gregor et al)
- IAF/Variational Normalizing flows (Kingma, Mohamed)

Blogs and helpers

- Tutorial on VAEs (Doersch)
- Brian Keng's blog (http://bjlkeng.github.io/)
- Shakir Mohamed's blog (http://blog.shakirm.com/)
- Ian Goodfellow's book (http://www.deeplearningbook.org/)



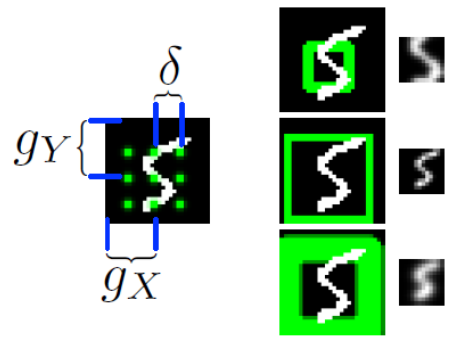


Figure 3. **Left:** A 3×3 grid of filters superimposed on an image. The stride (δ) and centre location (g_X, g_Y) are indicated. **Right:** Three $N \times N$ patches extracted from the image ($N = 12$). The green rectangles on the left indicate the boundary and precision (σ) of the patches, while the patches themselves are shown to the right. The top patch has a small δ and high σ, giving a zoomed-in but blurry view of the centre of the digit; the middle patch has large δ and low σ, effectively downsampling the whole image; and the bottom patch has high δ and σ.

TEXT TO SPEECH

- Given a text sequence, produce a speech sequence using DNNs
- Historical approach:
  - Concatenative TTS (concatenate speech segments)
  - Parametric TTS (Zen et al)
    - HMMs
    - DNNs
- Recent developments
  - Treat as seq2seq problem a la NMT
- Two current approaches
  - RNNs
  - Autoregressive CNNs (Wavenet/Bytenet/PixelRNN)

CURRENT BLEEDING EDGE LANDSCAPE

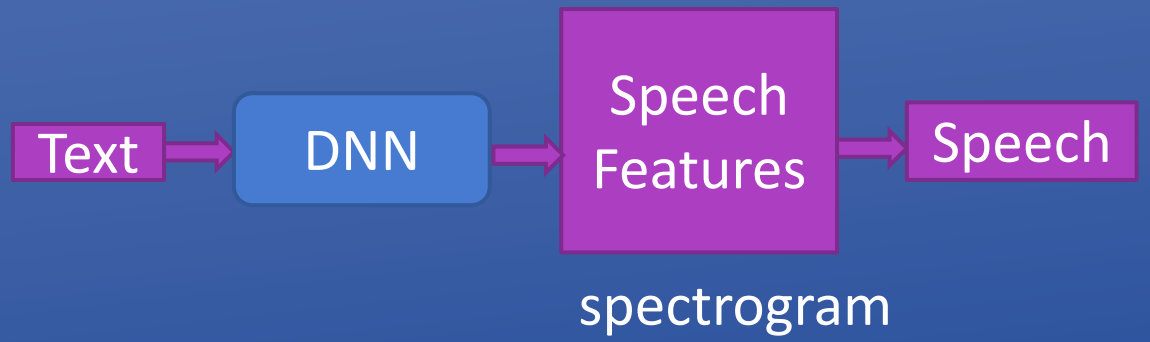
- Last 2 years (!)
  - Baidu DeepVoice series (2016, 2017, 2018)
  - Tacotron series (2017+)
  - DeepVoice, Tacotron are seq2seq models with text in => waveform out
    - Seq2seq + attention (Bahdanau style)
  - Wavenet series [not relevant, but very instructive]
    - Wavenet 1: fast training, slow generation
    - Wavenet 2: (a brilliancy) – two developments (100X over wavenet)
      1) Inverse Autoregressive Flow – fast inference
      2) Probability Density "Distillation" (as against estimation)
         - cooperative training during inference to match PDF of trained wavenet

DNN WORKFLOW

Deep Generative Neural Network

- Tacotron, Baidu Deepvoice
  - Google (2017), Tacotron (2016, 2017)
  - Seq2seq+attention RNN trained end to end

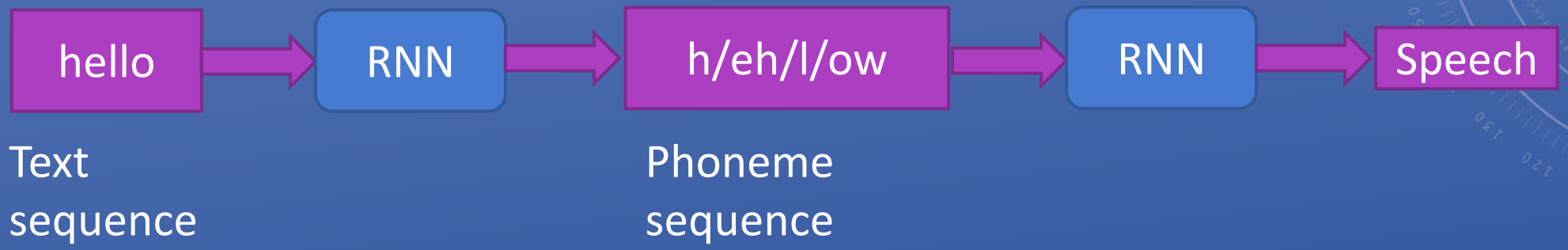
Speech



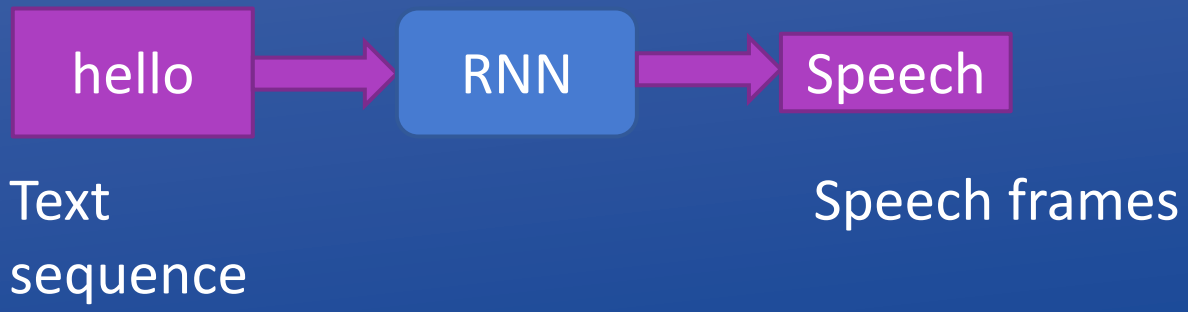
Text to Speech

TEXT VS PHONEME FEATURES

Earlier models



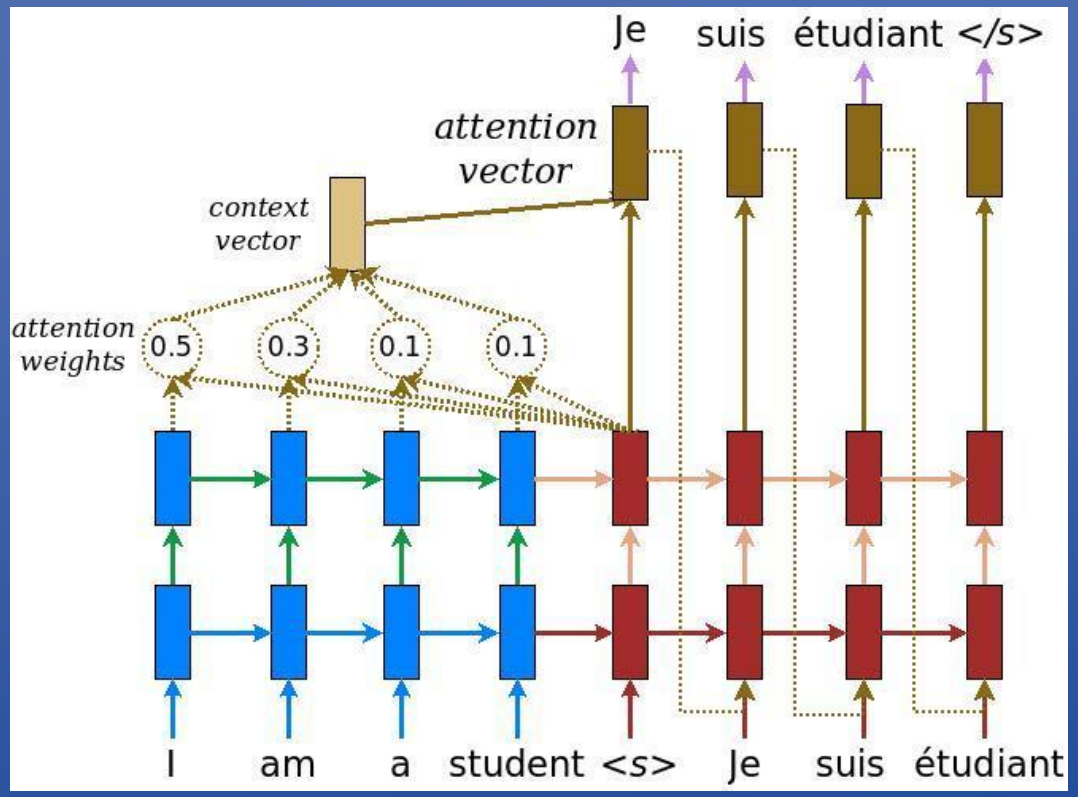
Tacotron



Phoneme (‘token’/segment) > text
Text=>phoneme needs another DNN
Not totally “end-to-end”

SEQ2SEQ+ATTENTION

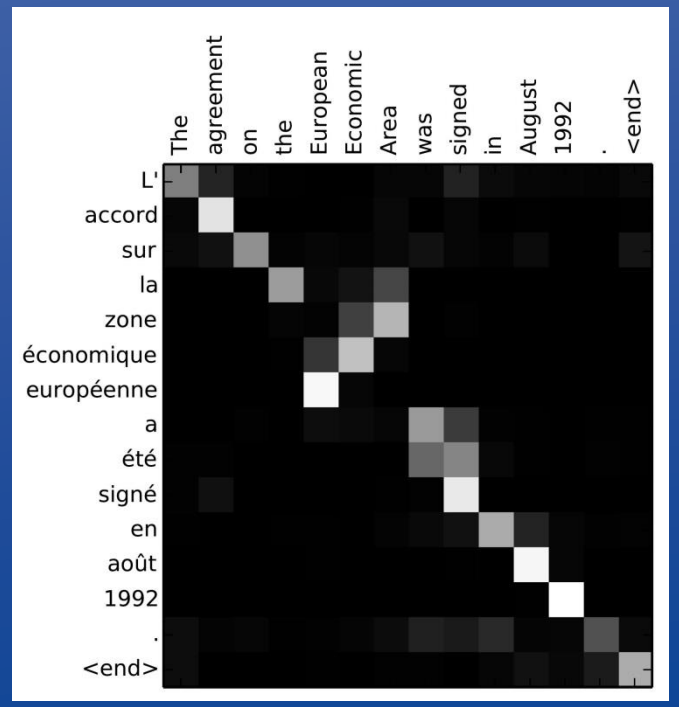
- Originally proposed in NMT context (Bahdanau, Cho et. al.)



I am not a small black cat

je ne suis pas un petit chat noir

Variable word length
Word ordering different



Attention weights
Input and output words

TACOTRON: SEQ2SEQ+ATTENTION

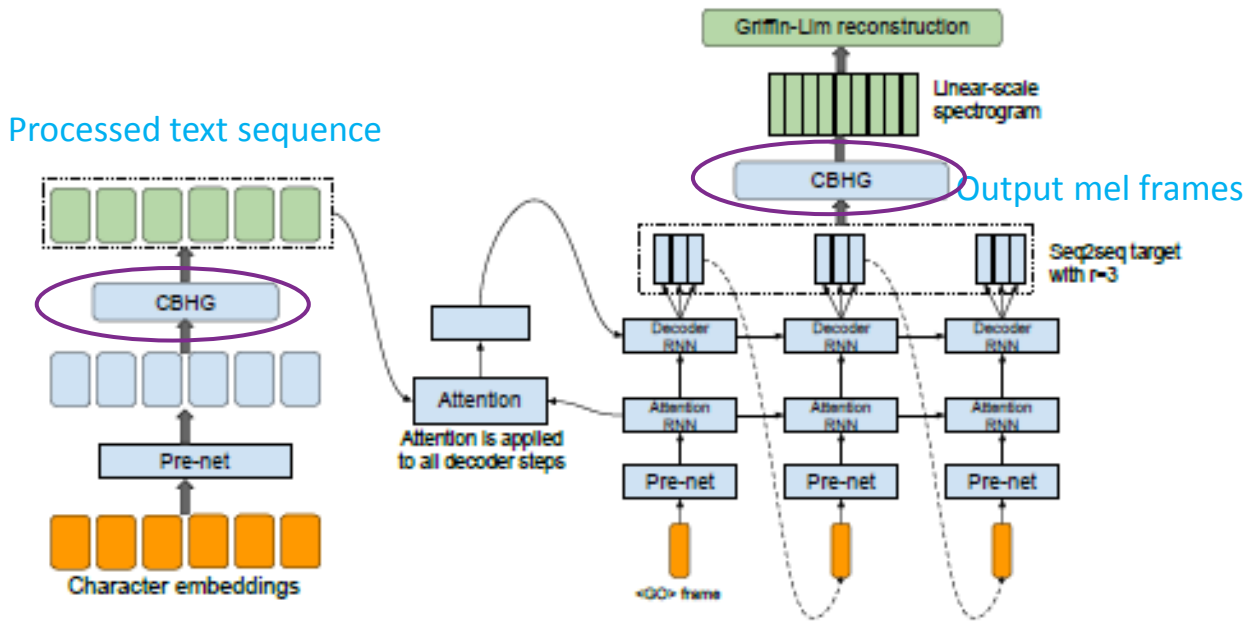
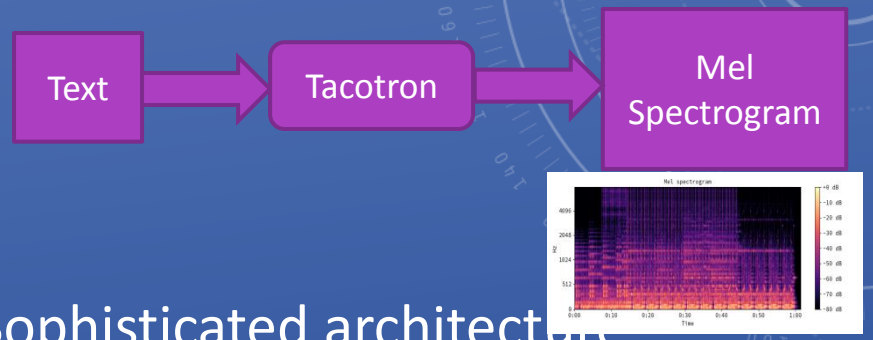


Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*



Sophisticated architecture
Built on top of Bahdanau
Preprocessing of text
Postprocessing of output 'mel' frames

Training:
<text/mel> pairs

AUDIO FEATURES FOR SPEECH DNNS

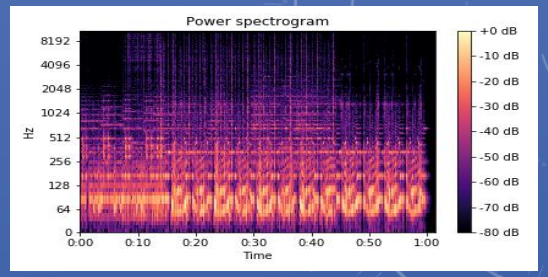
- Main theme: Synthesize voice using generative modeling (VAEs/GANs)
- Sub-theme: Feature generation for audio critical for audio processing
- Audio representations:
  - Raw waveforms: uncompressed , **1D**, amplitude vs time – 16 kHz
  - Linear spectrograms: **2D**, frequency bins vs time (1025 bins)
  - Mel spectrograms: 2D, **compressed** log-scale representation (80 bins)
- Compressed (mel) representations
  - Easier to train neural network
  - Lossy
  - Need compression but also need to keep *sufficient* number of features

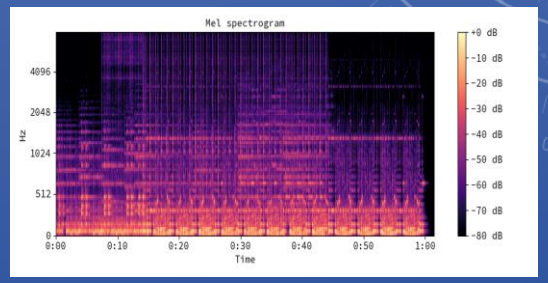
MOTIVATION

Text → DNN → Speech Features → ? → Speech

Text to Speech

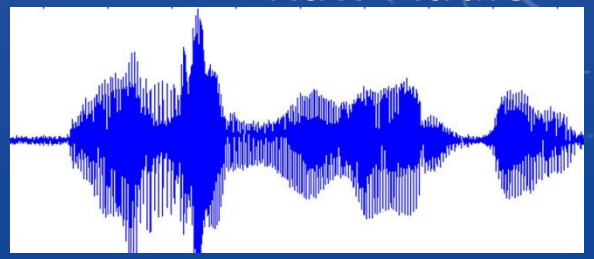
Power & mel spectrogram





Speech → STFT → Speech Features → DNN → Speech Features → ? → Speech

Raw Audio



Speech to transformed speech

MEL FEATURES

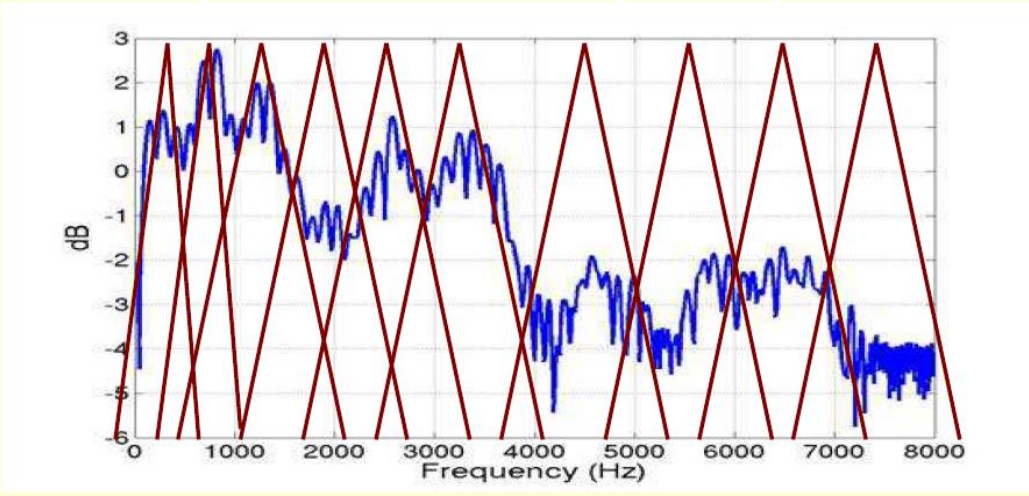
- Order of magnitude compression beneficial to train DNNs
  - Linear spectrograms: 1025 bins
  - Mel: 80 bins
- Energy is mostly contained in a smaller set of bins in linear spectrogram
- Creating mel features
  - Low frequencies matter – closely spaced filters
  - Higher frequencies less important – larger spacing

$$M = 1125 \ln(1 + \frac{f}{700})$$

Linearly spaced bins in mel scale

Bins closely spaced at lower frequencies

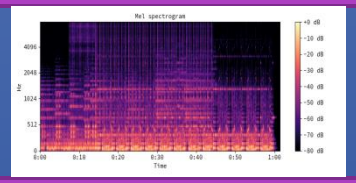
(Kishore Prahllad, CMU)

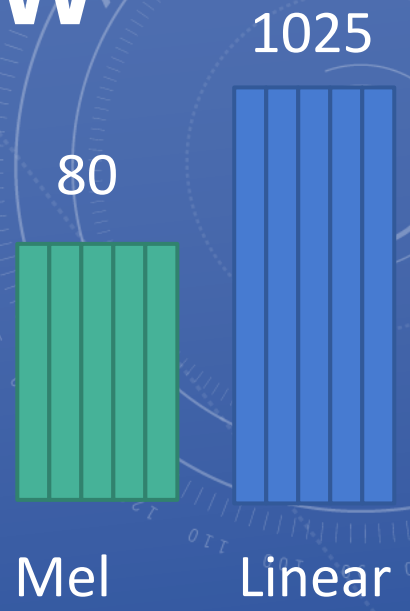


AUDIO PROCESSING WORKFLOW

Feature Generation

Audio → Linear Spectrogram → Mel Spectrogram



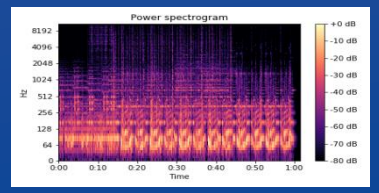


Training Speech data

Mel Spectrogram → VAE Network → Mel Spectrogram

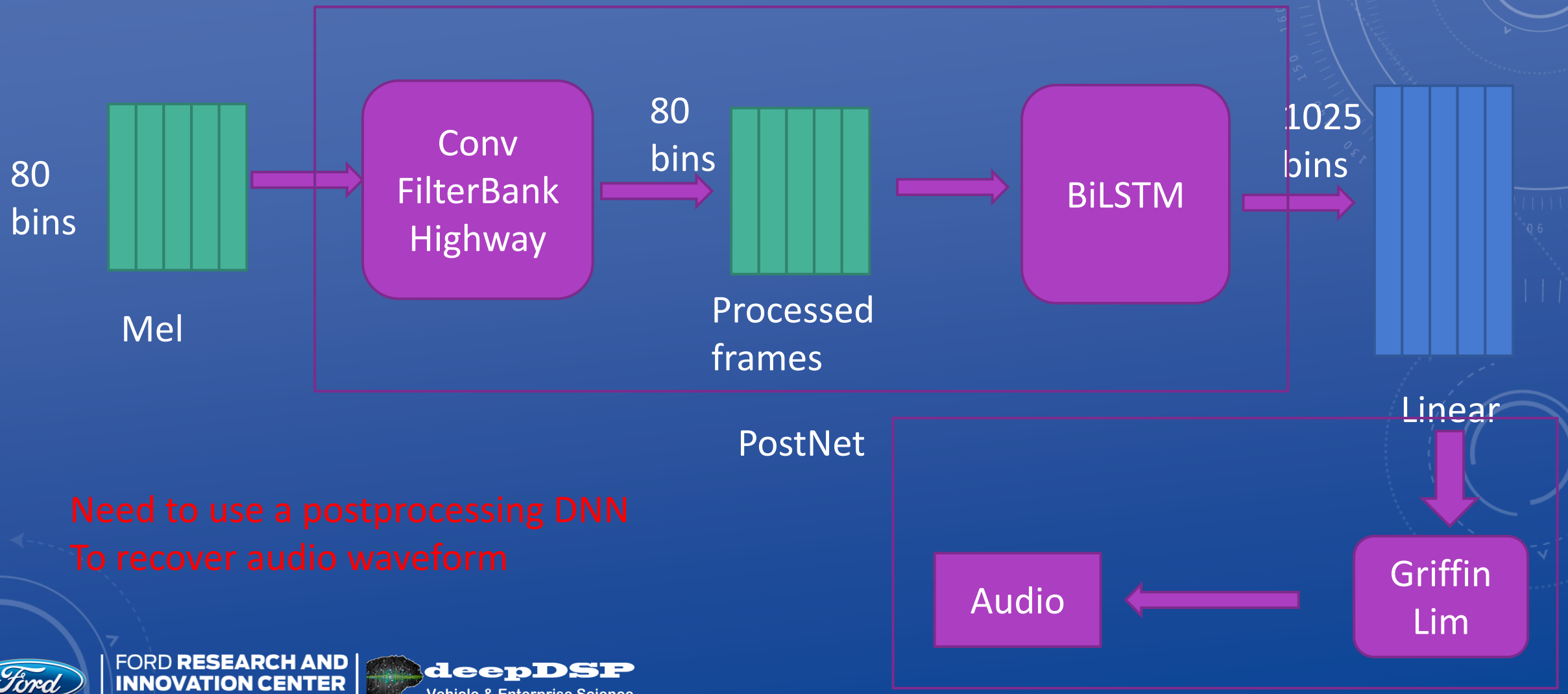
Postprocessing To recover audio

Mel Spectrogram → PostNet → Linear Spectrogram → Audio



POST PROCESSING TO RECOVER AUDIO

- Use of Griffin-Lim procedure to convert from linear spectrogram to waveform



Need to use a postprocessing DNN
To recover audio waveform

CBHG/POSTNET

- Originally, in Tacotron (adapted from Lee et. al.)
- "Fully Character-Level Neural Machine Translation without Explicit Segmentation"
- Tacotron: text=>phoneme bypassed to allow text=>speech
- Used in 2 places:
  - Encoder: Text=>text features
  - Postprocessor net
  - Mel spectrogram => linear spectrogram (=>audio)

(Tacotron)

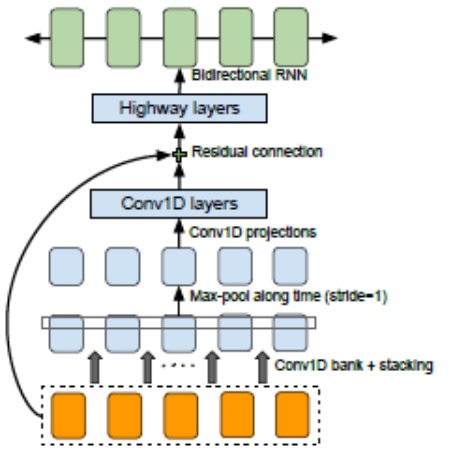


Figure 2: The CBHG (1-D convolution bank + highway network + bidirectional GRU) module adapted from Lee et al. (2016).

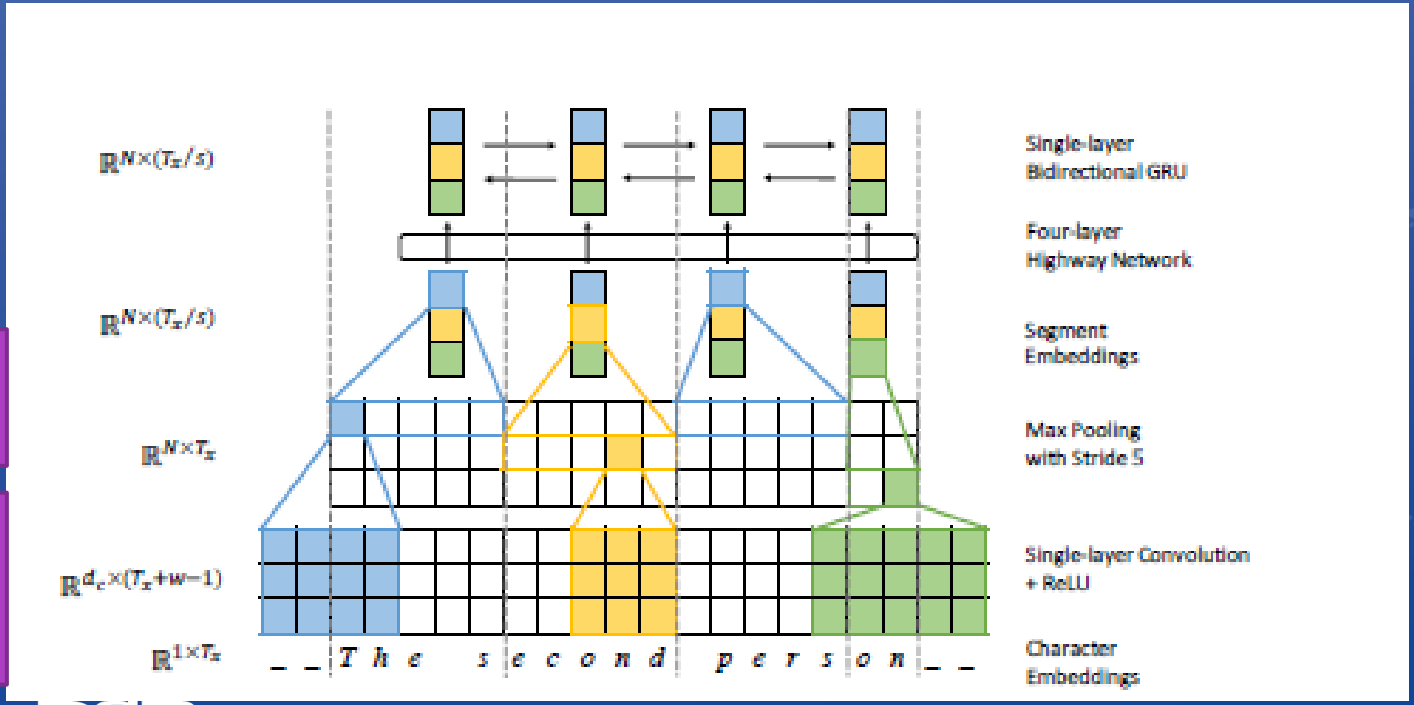
CBHG DESCRIPTION

- Conv+FilterBank+Highway+GRU
- Take convolutions of sizes (1,3,5,7, etc.) to account for words of varying size
- Pad accordingly to create stacks of equal length
- Max pool to create segment embeddings

(Lee et al)

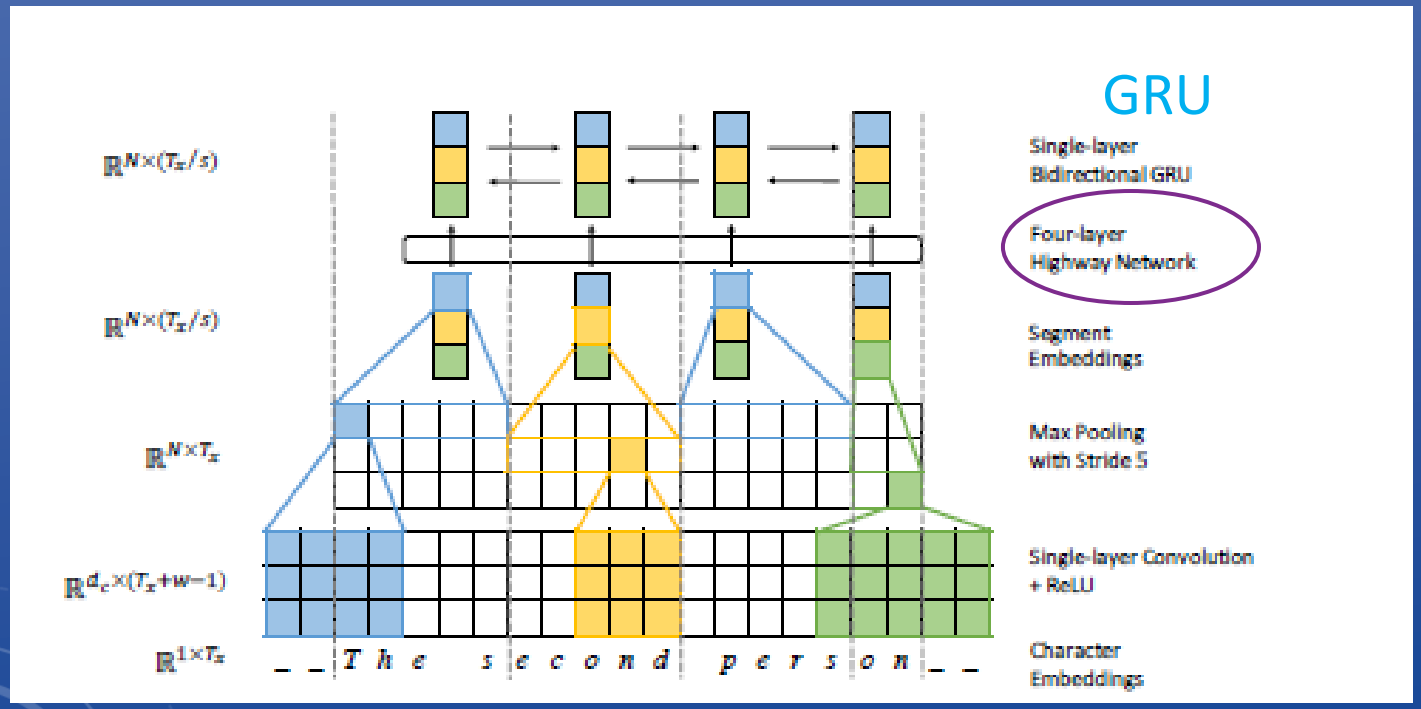
Pool (stride=1)

convolutions



CBHG DESCRIPTION

- Send to highway layers (improves training deep nets – Srivastava)
- Bi-directional GRU or LSTM



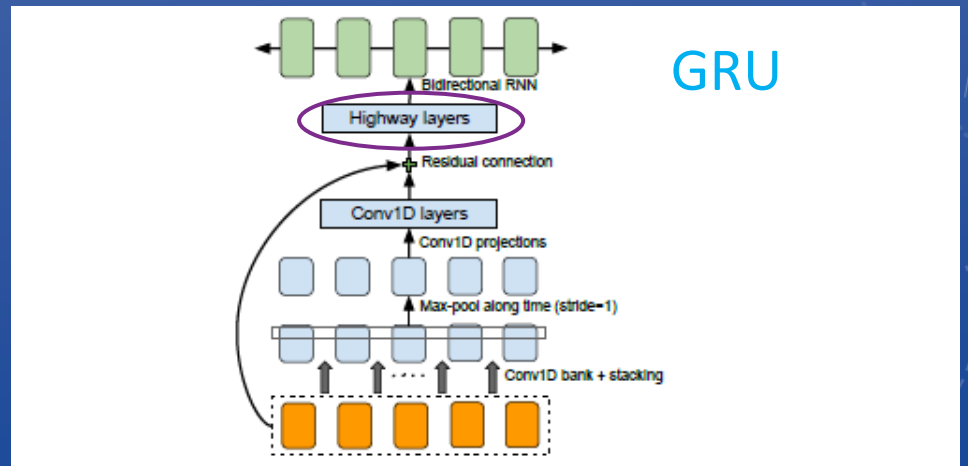
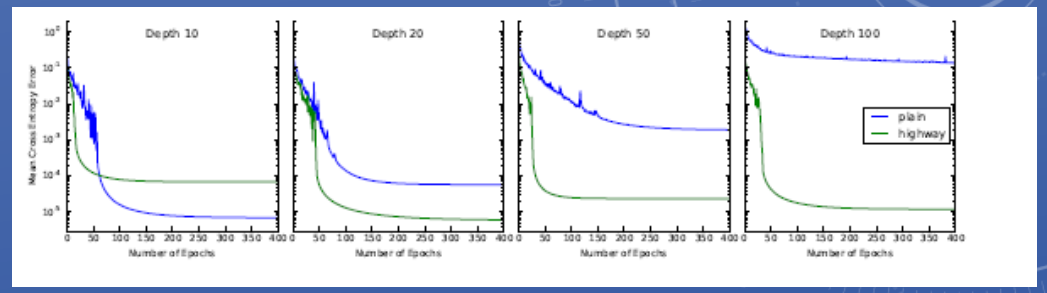


Figure 2: The CBHG (1-D convolution bank + highway network + bidirectional GRU) module adapted from Lee et al. (2016).

HIGHWAY LAYERS OVERVIEW

- Improves upon residual connections
- Residual:
  - $y = f(x) + x$
- Highway motivation: use fraction of input
  - $y = c.f(x) + (1 - c).x$
- Now make ‘c’ a learned metric
  - $y = c(x)f(x) + \big(1 - c(x)\big).x$
- Make c(x) lie between 0 and 1 by passing through sigmoid unit
- Finally, use a stack of highway layers. E.g. y1(x), y2(y1), y3(y2), y4(y3)



Srivastava et al

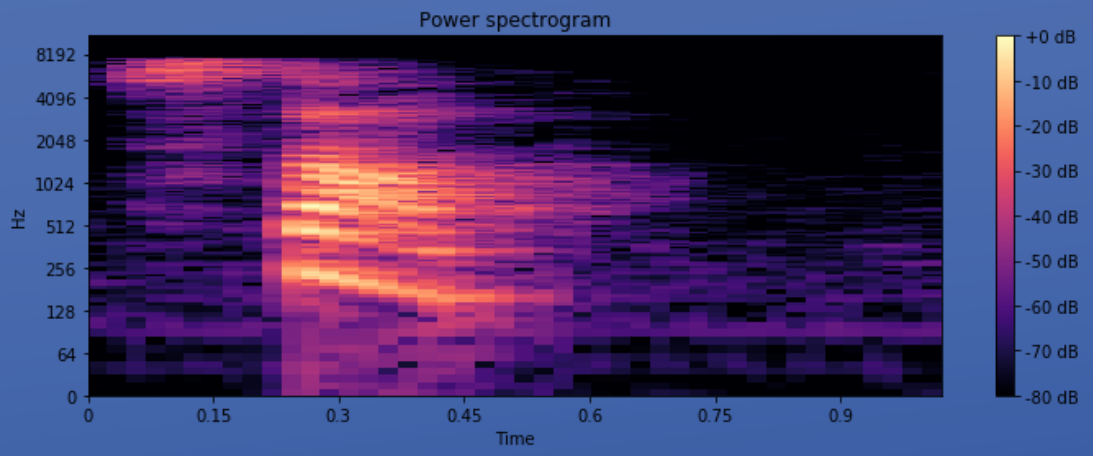
SPECTROGRAM RECONSTRUCTIONS

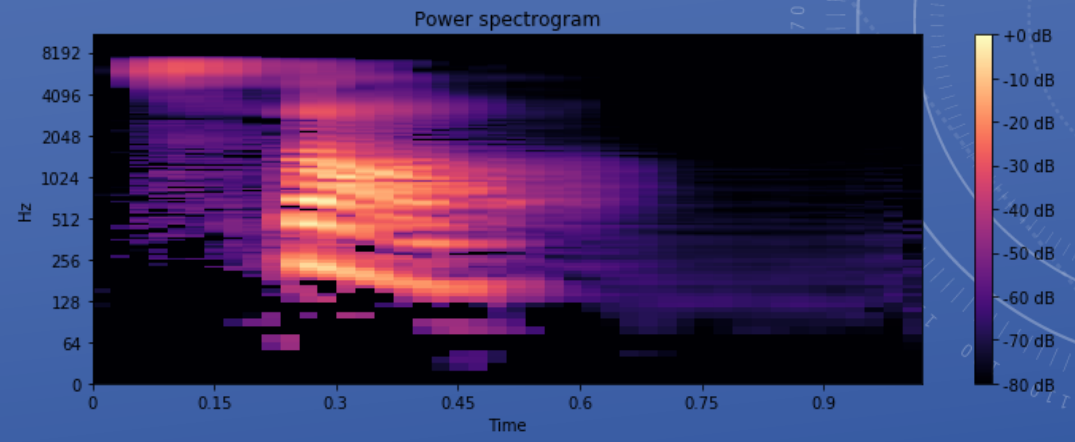
- Use filter sizes of 1, 3, 5 in CBHG
- Use bi-LSTM
- Highway layer stack of 4
- Input: 80 bin mel frames with seq length 44
- Output: 1025 bin linear frames with seq length 44
- PyTorch
- Librosa

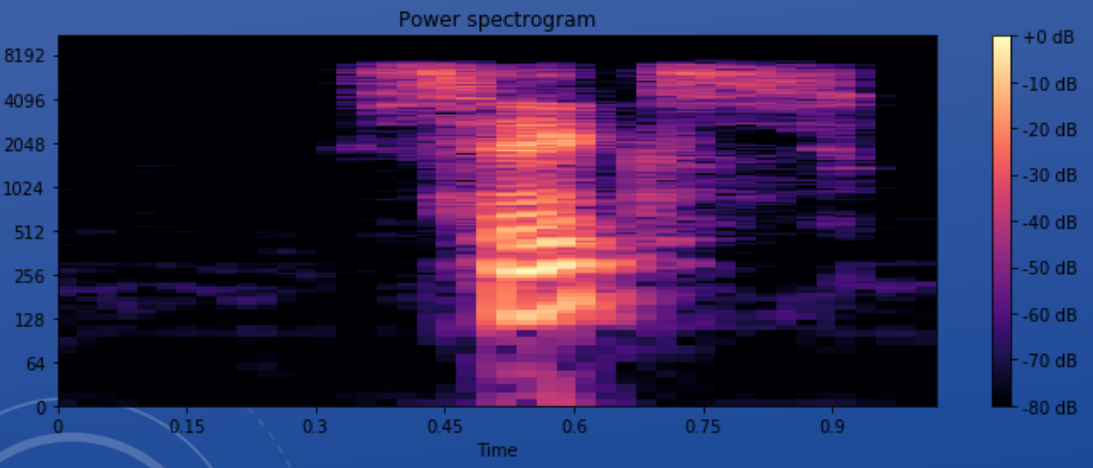
SAMPLES

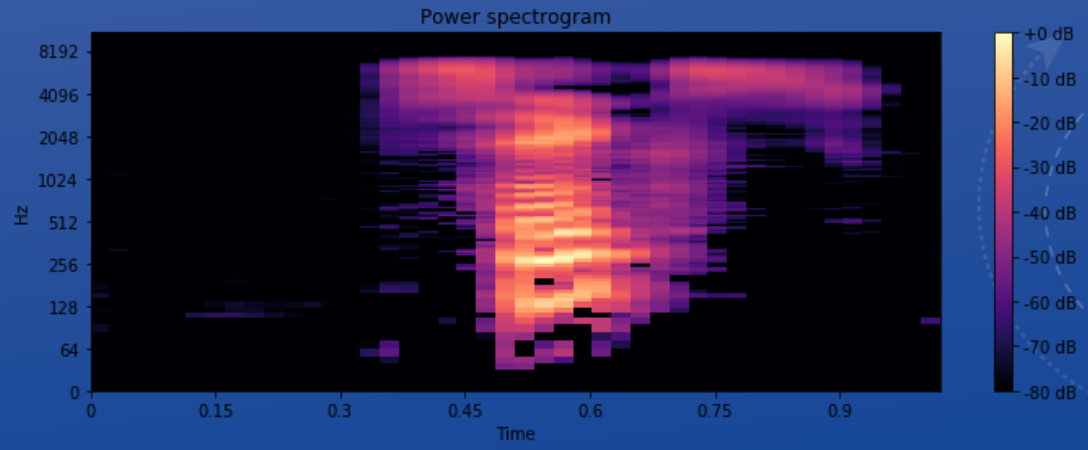
“ground truth”

“reconstructed”









SAMPLES

Ground truth

Reconstruction

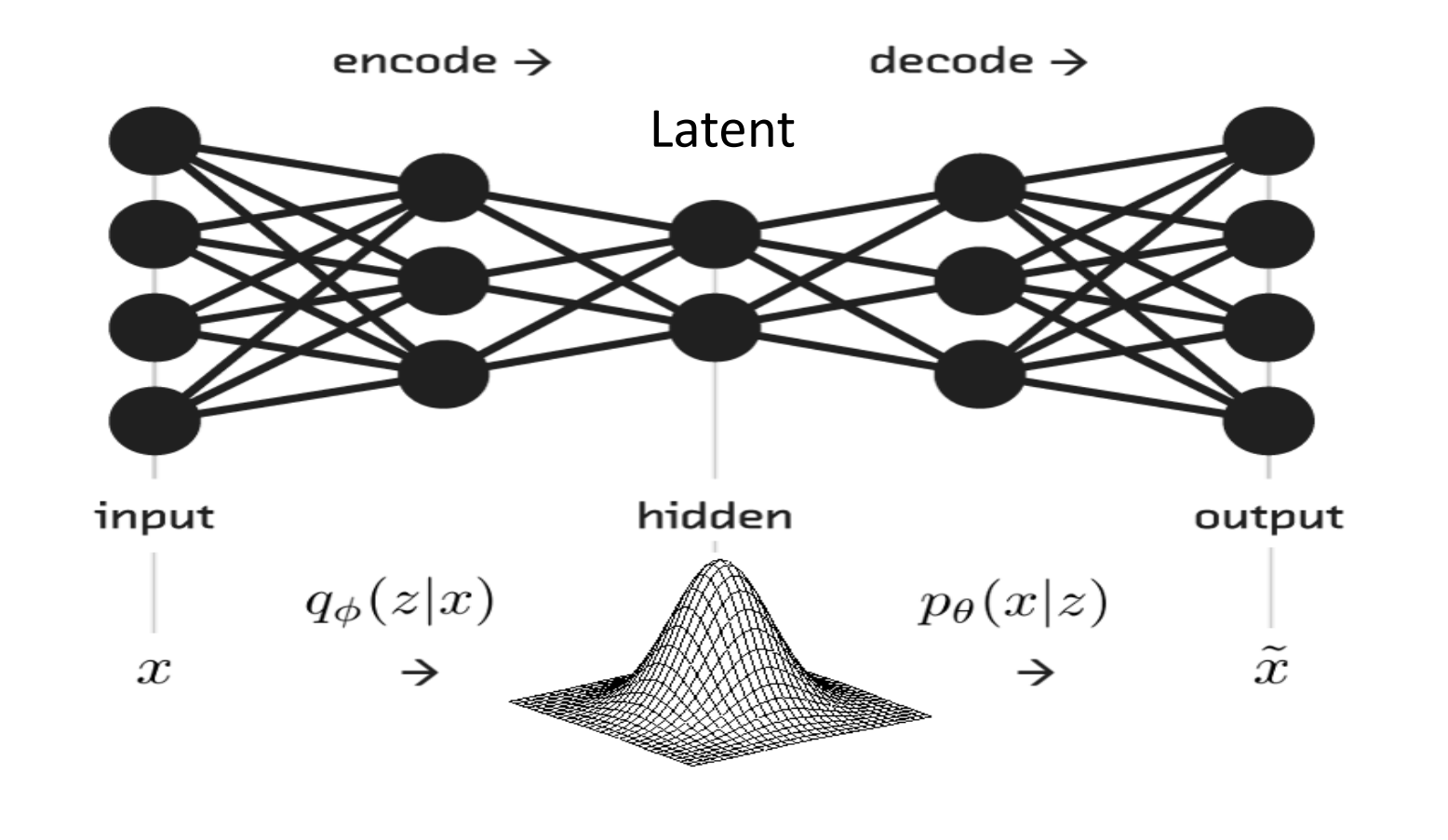
GENERATIVE MODELING WITH VARIATIONAL AUTOENCODERS

DESIDERATA

GENERATIVE MODELING WITH VARIATIONAL AUTOENCODERS

- Variational Inference fashioned into DNN (Kingma and Welling; Rezende and Mohamed)

Input



Reconstruction

PROPERTIES OF VAE

- Feed input data and encode representations in reduced dimensional space
- **Reconstruct** input data from reduced dimensional representation
  - Compression
- **Generate** new data by sampling from latent space

Input

Training

Encoder → Latent Layer → Decoder

Reconstruction

$$\log P(X) - D_{KL}[Q(z|X)\|P(z|X)] = E[\log P(X|z)] - D_{KL}[Q(z|X)\|P(z)]$$

Inference

Sample N(0,I) → Latent Layer → Decoder

Generation

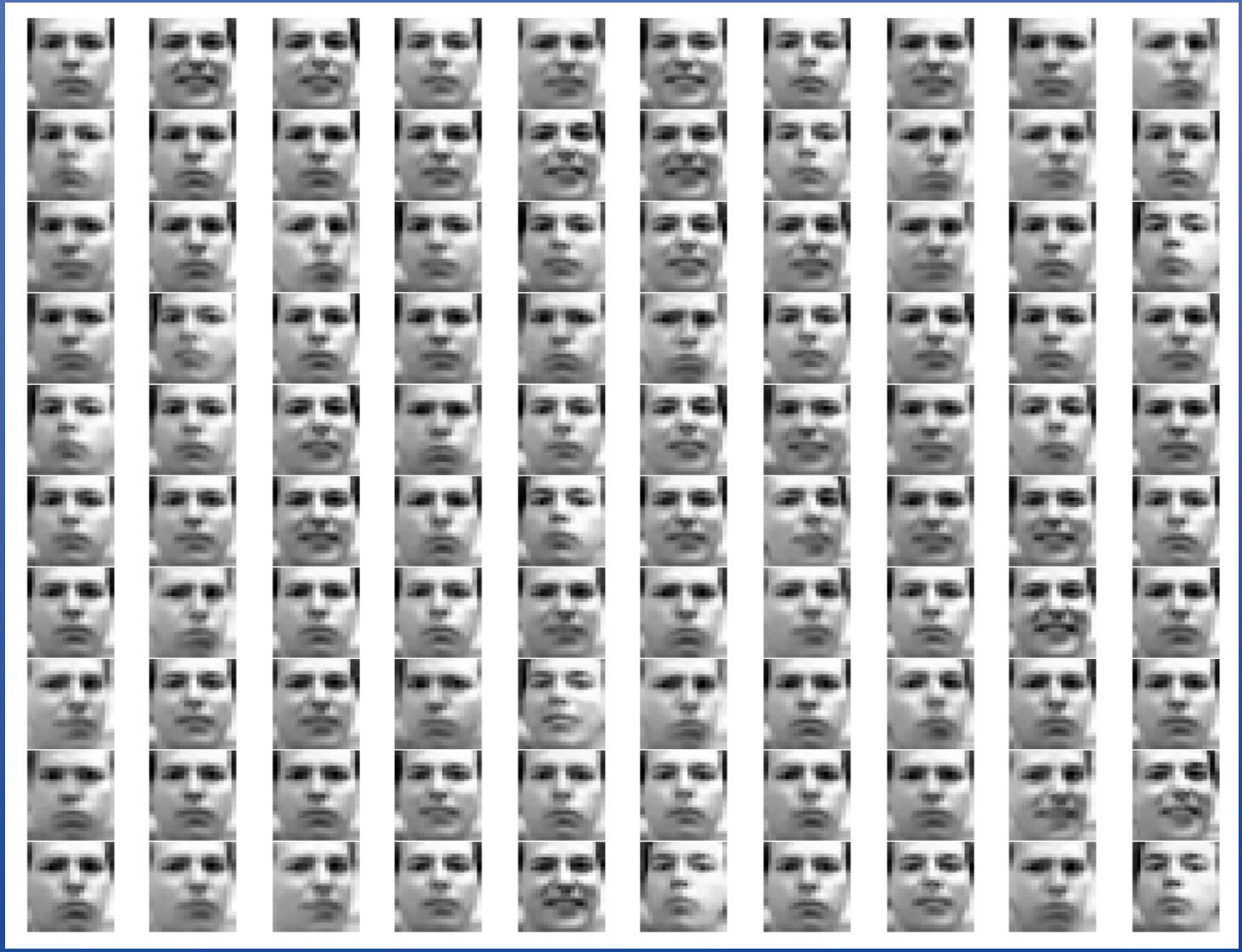
RECONSTRUCTIONS

Original Image: 560 pixels
Reconstructed from 20 latent variables
28X image compression advantage

Ground Truth

Reconstruction

GENERATION



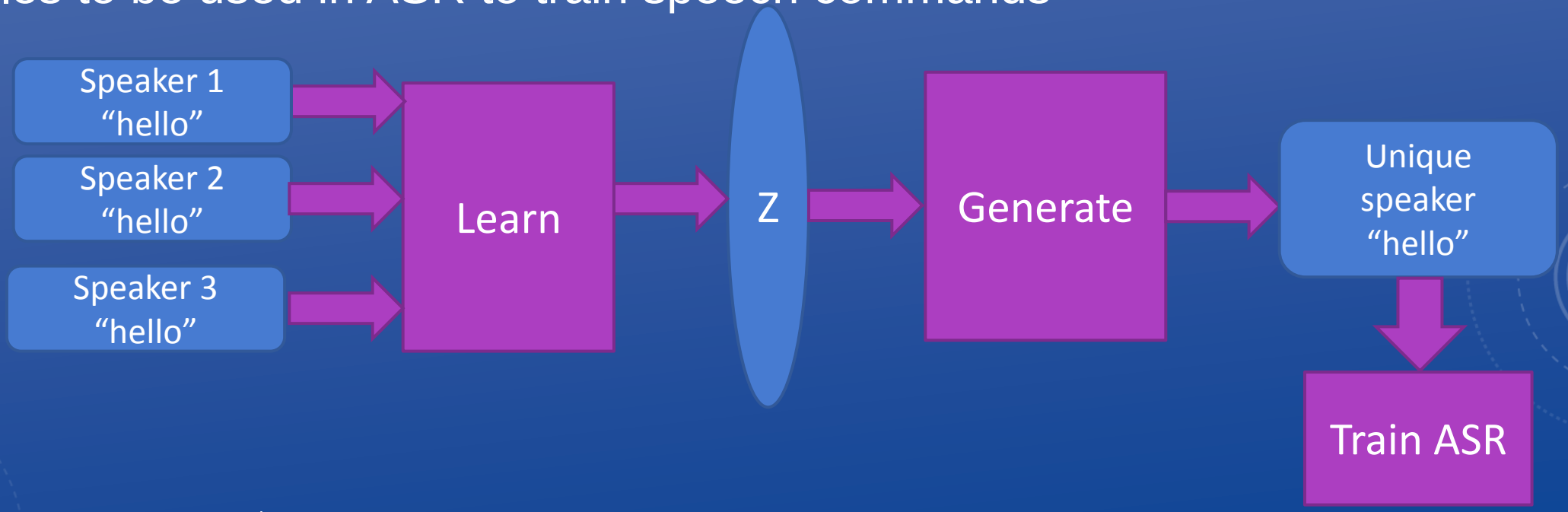
Faces and poses that did not exist!

APPLICATIONS

SPEECH ENCODINGS

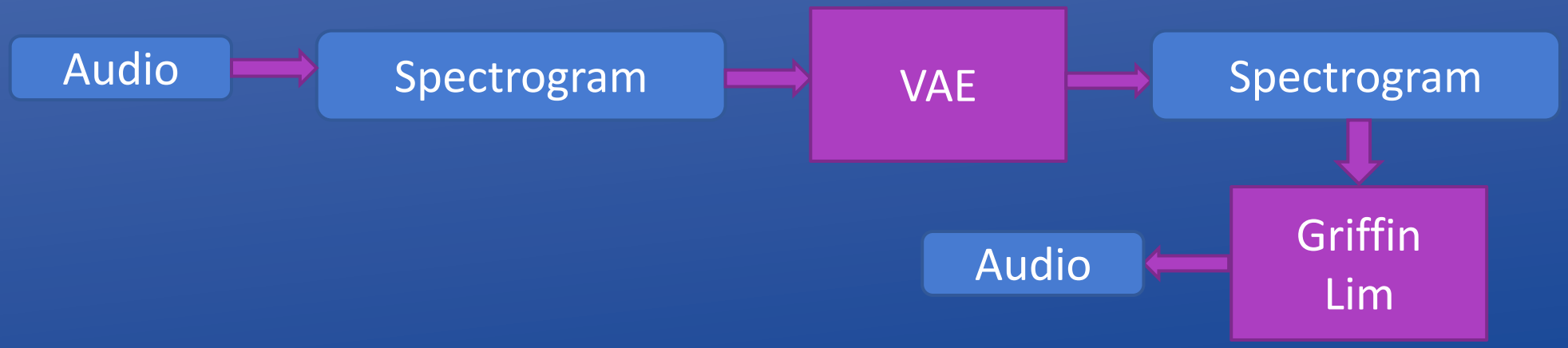
SPEECH ENCODINGS USING VAES

- Encode utterances from several speakers
- Store encodings
- Generate synthetic speech by sampling from VAE
- Tried with simple utterances “hello”, “cat”, “stop” from speech commands dataset
- Samples to be used in ASR to train speech commands



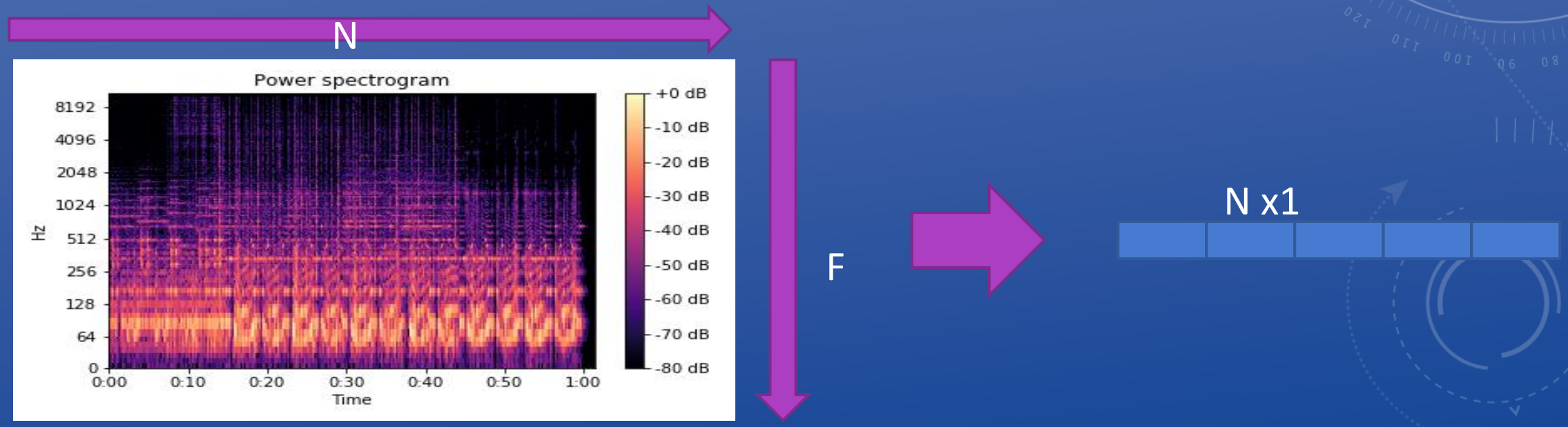
WORKFLOW

- Input: Spectrogram
- Convert audio signals to spectrogram
- Output: spectrogram, converted to audio by Griffin Lim reconstruction
- Librosa used to manipulate audio



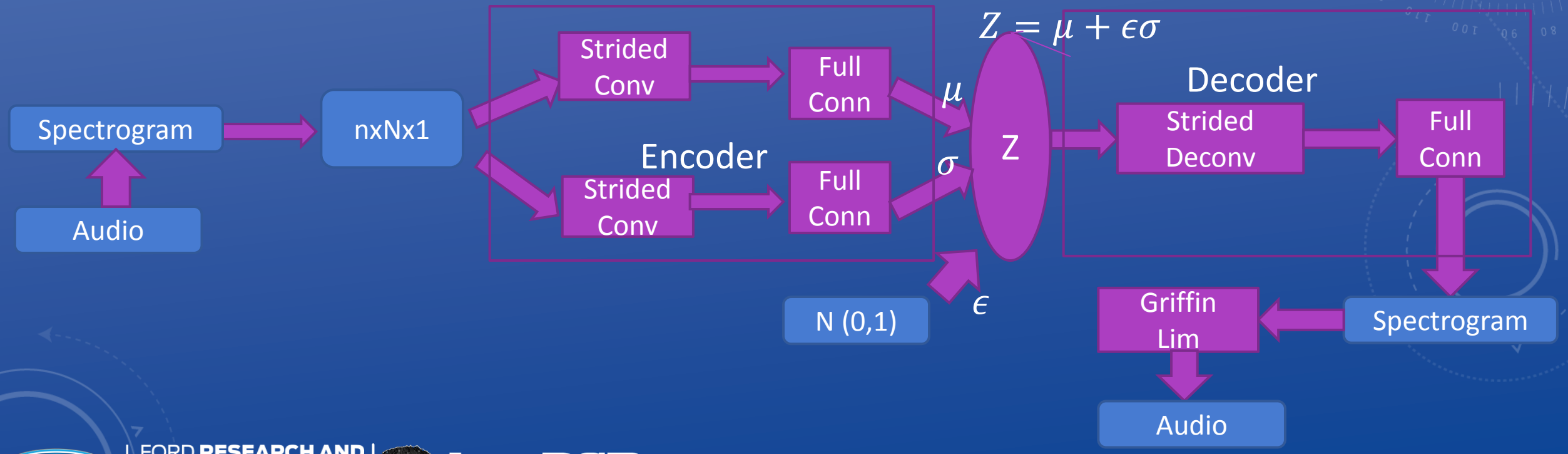
NETWORK ARCHITECTURE

- Input spectrogram: 1025x44 image (NxF)
- Audio signals invariant only in time axis
  - Take full connections in y axis
  - First layer: take 1xF convolutions



NETWORK ARCHITECTURE

- Encoder/Decoder: Convolution/Deconvolution
- Use strides for downsampling
- Multiple channels for filterbanks
- Filter sizes dx1 operating on Nx1 inputs (times #filter banks=n)



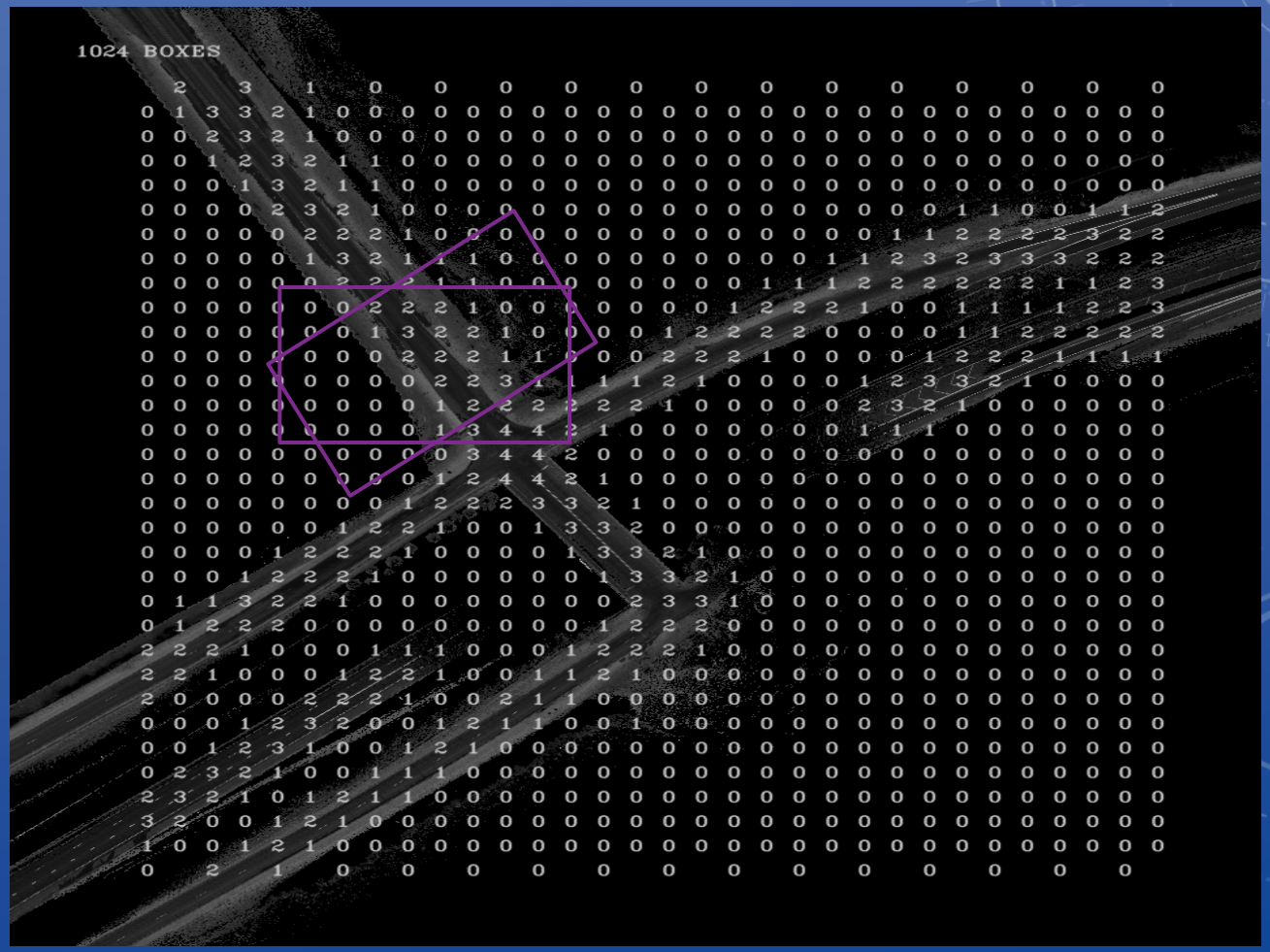
AUDIO SAMPLES

Utterance: “Cat”

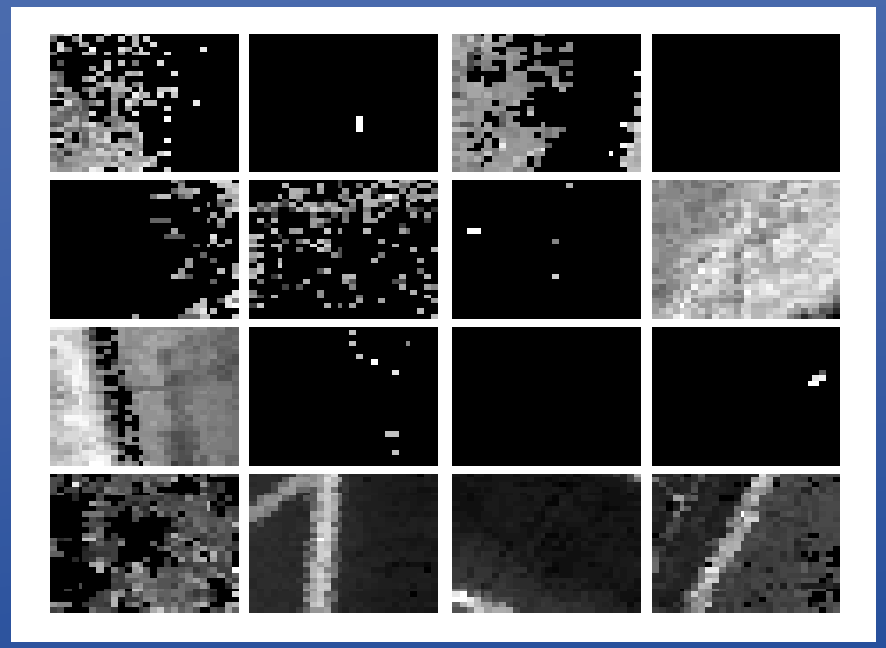
CVAE – ROTATED IMAGES WITH POSE AS LABEL

$$\log P(X|c) - D_{KL}[Q(z|X,c)\|P(z|X,c)] = E[\log P(X|z,c)] - D_{KL}[Q(z|X,c)\|P(z|c)]$$

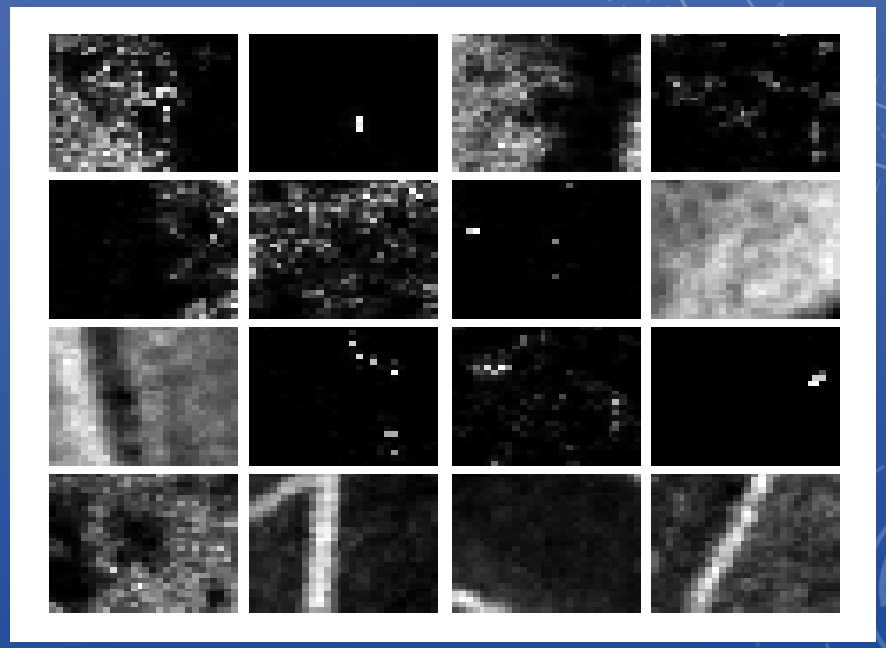
- Give input image + label
- Produce rotated image
- Label==angle
  - One hot encoding



RECONSTRUCTIONS

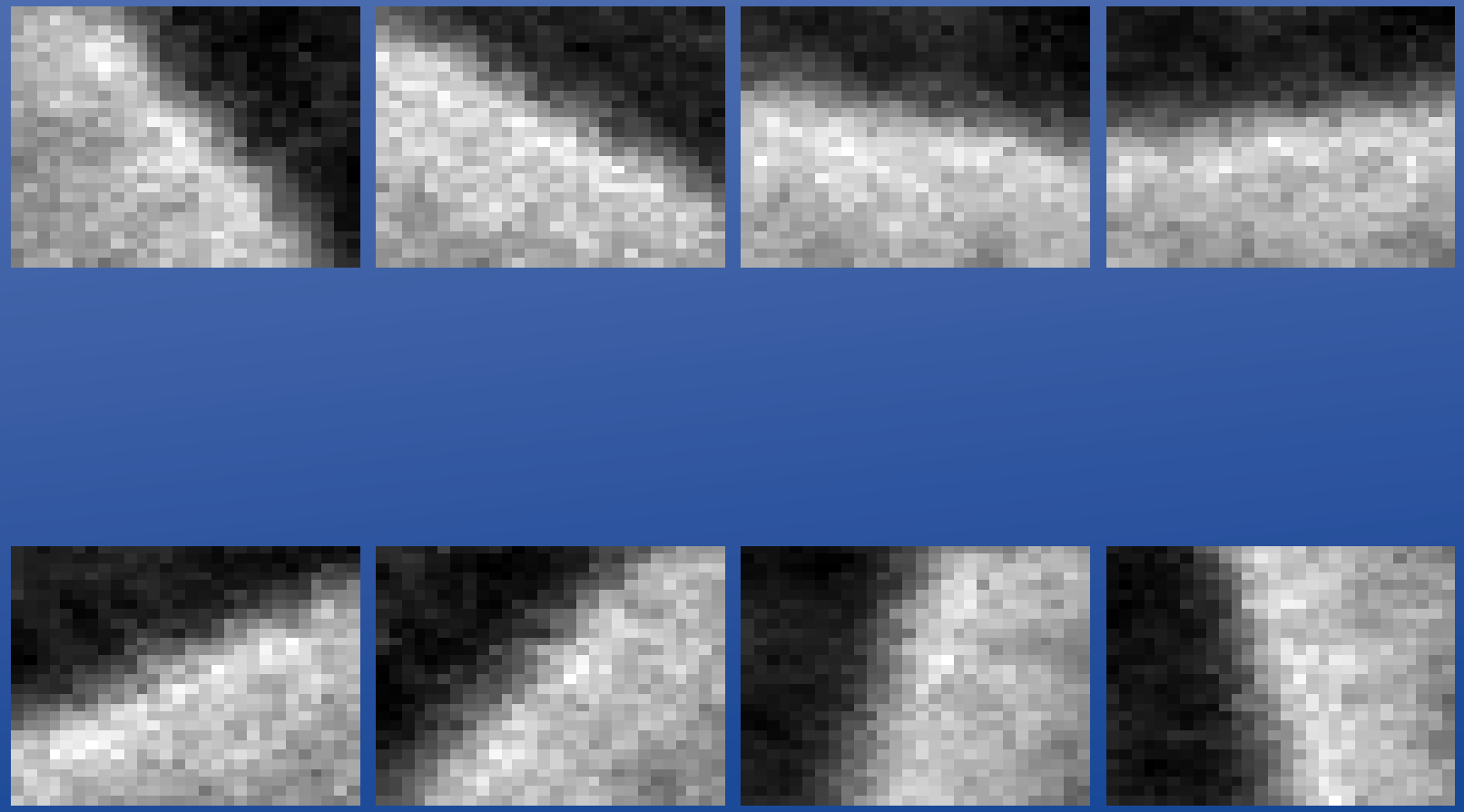


Ground Truth Rotations



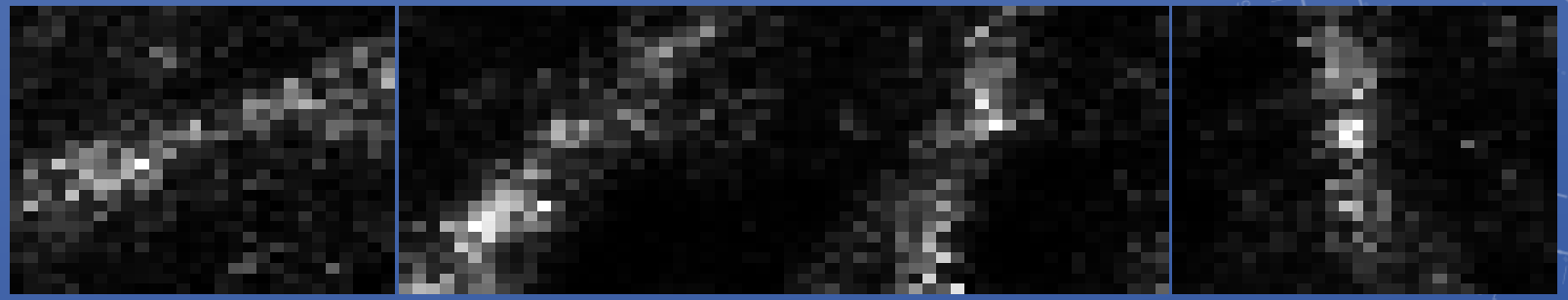
Reconstructed Rotations

RECONSTRUCTIONS

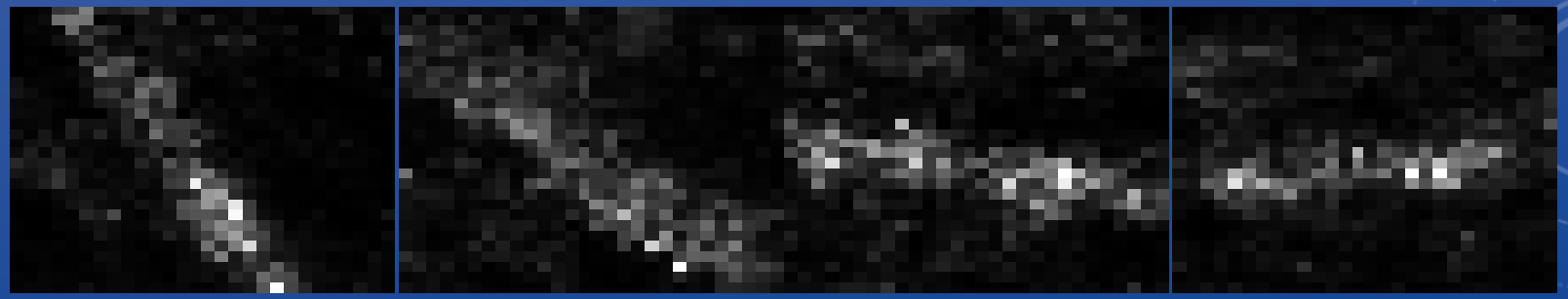


TEST ROTATIONS

Need larger
Training set



Images produced
By data not in
Training set



CONCLUSIONS

- Speech features:
  - Tacotron's CBHG network is a necessary prelude to other operations
- Speech transformations
  - Custom convolutional VAE architecture
  - Improving samples and architecture ongoing
- LIDAR
  - Conditional VAEs/GANs for point cloud transformations

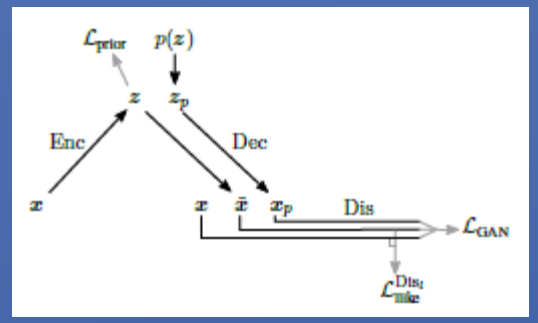
UNFINISHED WORK

- VAE: seq2seq implementations
  - Replace encoder/decoder with recurrent forms
  - Conditioning to produce custom attributes
- Can we do speech transformations with real TTS applications
- What about voice to voice conversion applications?
- GAN formulations for losses
  - Investigate scenarios where GAN losses would be beneficial

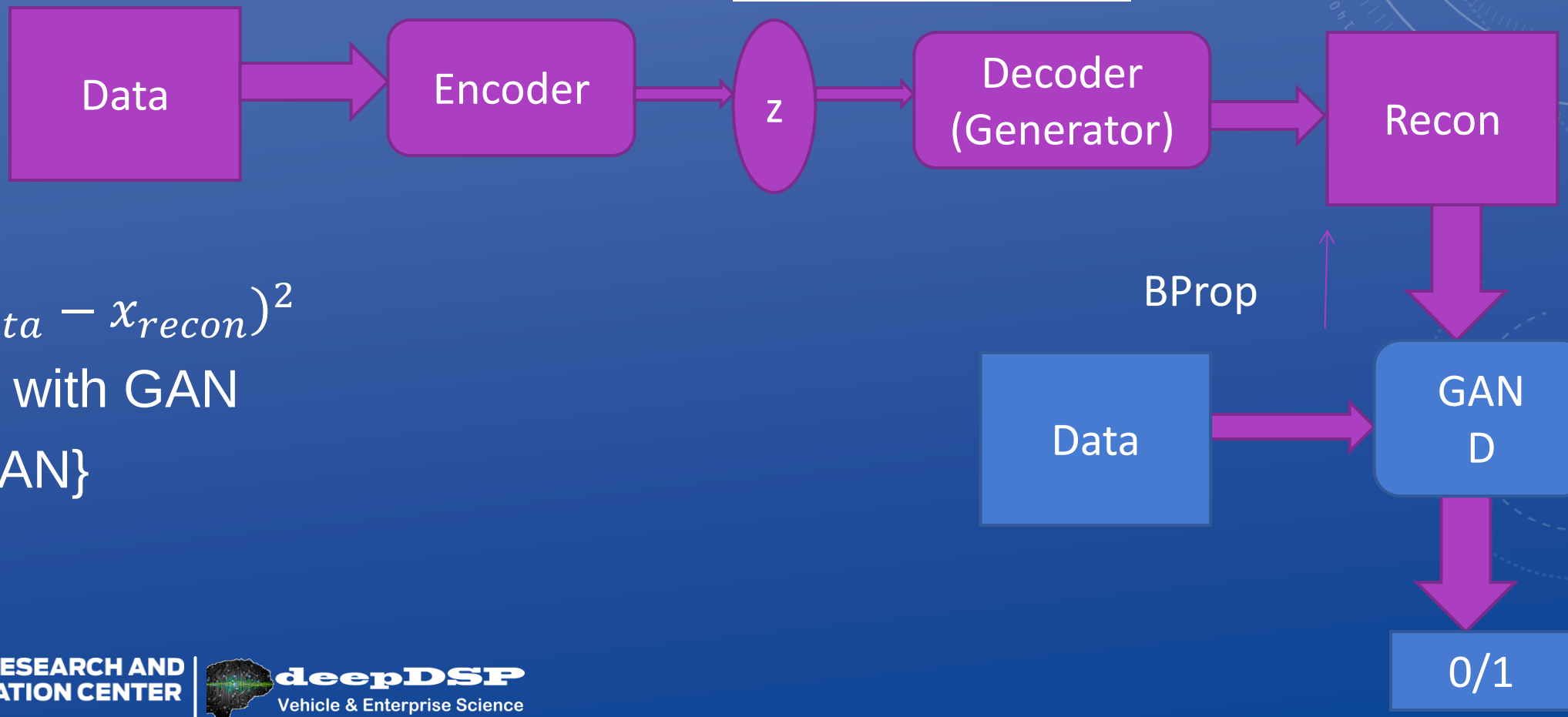
MOTIVATION – VAEGAN

- MSE loss used in VAE

Autoencoding Beyond Pixels



$$\mathcal{L}_{\text{GAN}} = \log(\text{Dis}(\boldsymbol{x})) + \log(1 - \text{Dis}(\text{Dec}(\boldsymbol{z}))) + \log(1 - \text{Dis}(\text{Dec}(\text{Enc}(\boldsymbol{x}))))$$



- $L = (x_{data} - x_{recon})^2$
- Replace with GAN
- L= L_{GAN}

EXAMPLES

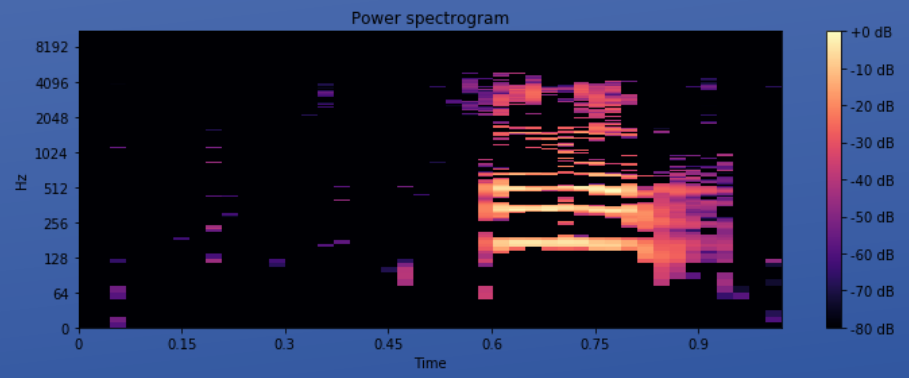
Truth

GAN loss recon

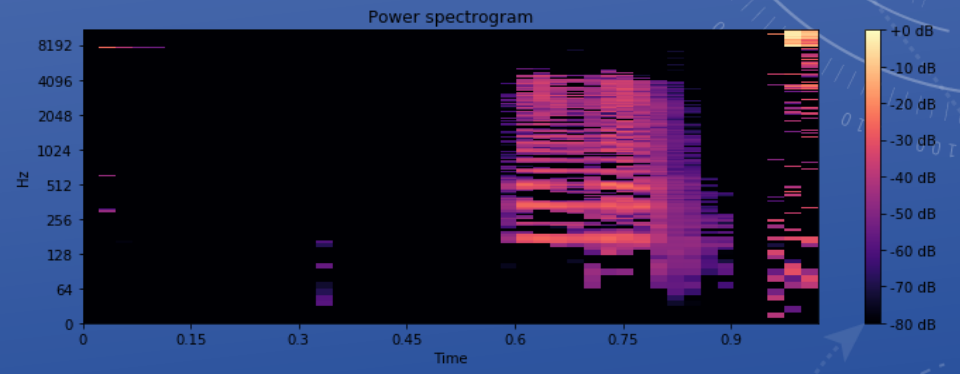
VAE (MSE) loss recon
(Ground truth not shown)

USING GANS AS LOSS FUNCTIONS

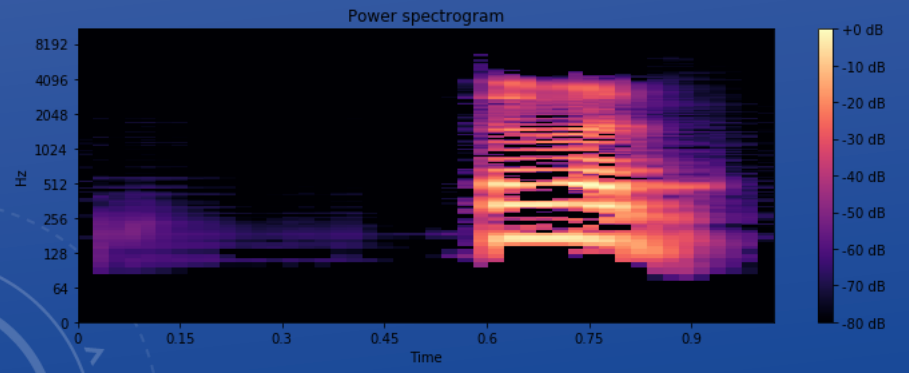
- So far, we have used L1 or L2 losses
- We find that L2 losses reconstruct poorly
- Can we use GANs as loss functions?



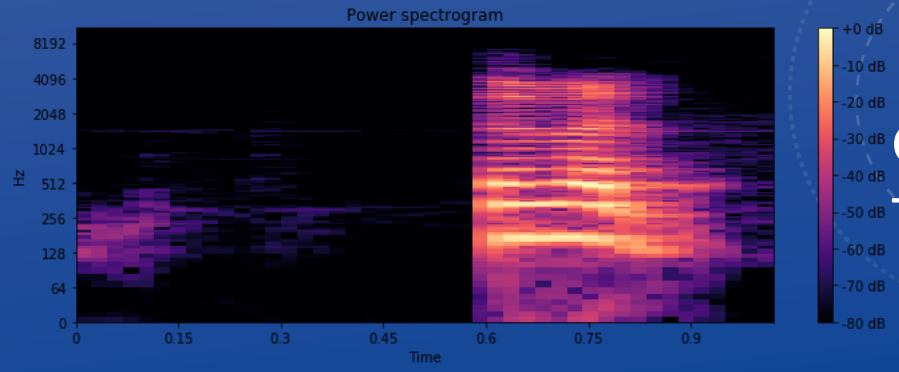
L2



GAN



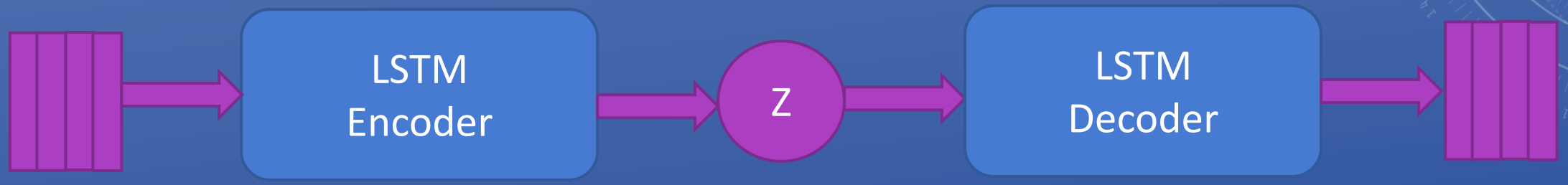
L1



Ground Truth

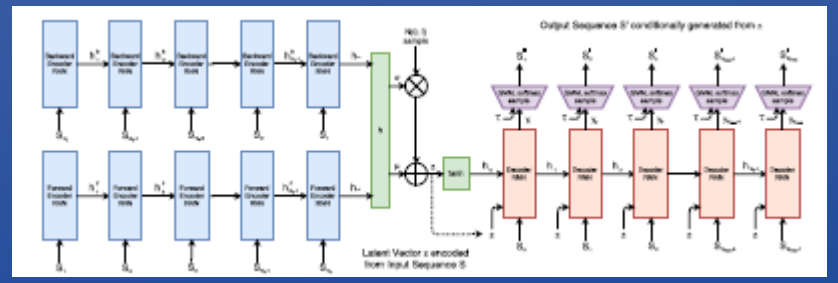
SEQ2SEQ VAE

- Improving upon vanilla vae with recurrent model



Mel in

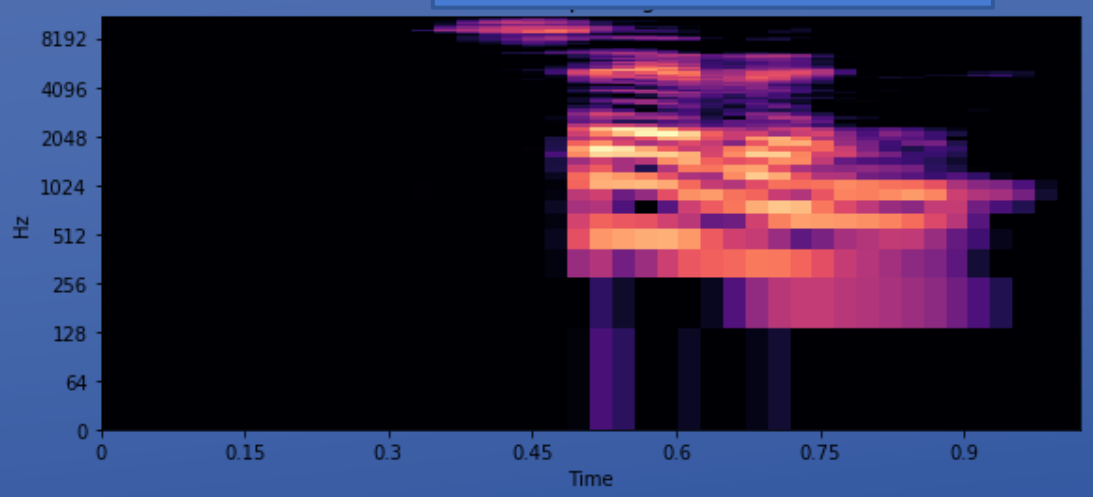
Reconstruction
Mel out



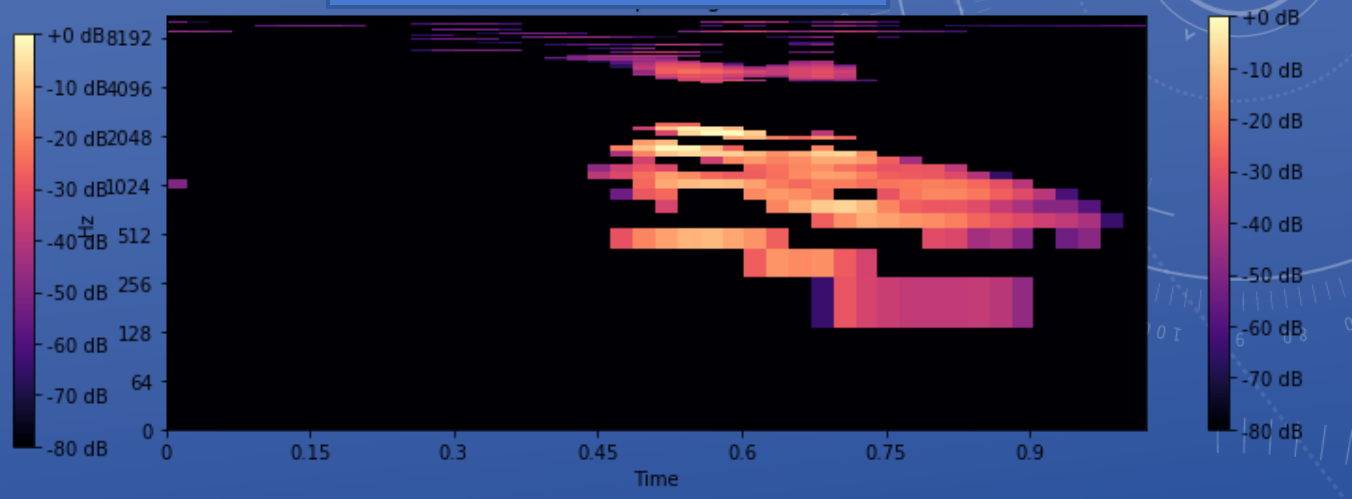
Sketch-RNN

SEQ2SEQ VAE

Ground Truth



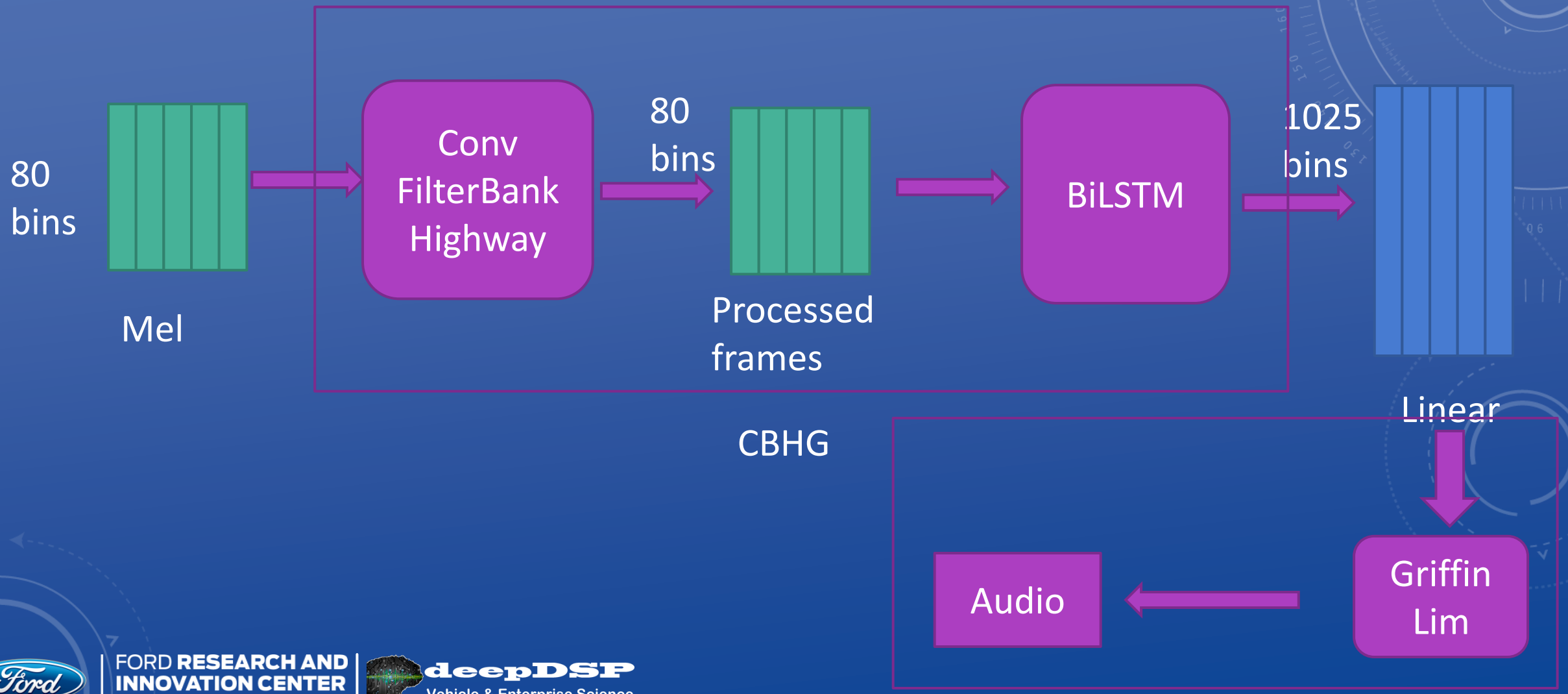
Reconstruction



Simple network (LSTM)

SUPPLEMENTARY SLIDES

FIGURE 1


80
bins
Mel
Conv
FilterBank
Highway
80
bins
Processed
frames
BiLSTM
1025
bins
Linear
CBHG
Griffin
Lim
Audio

GAN LOSS ARCHITECTURE

FIGURE 2: EXISTING FRAMEWORK FOR LOSSES

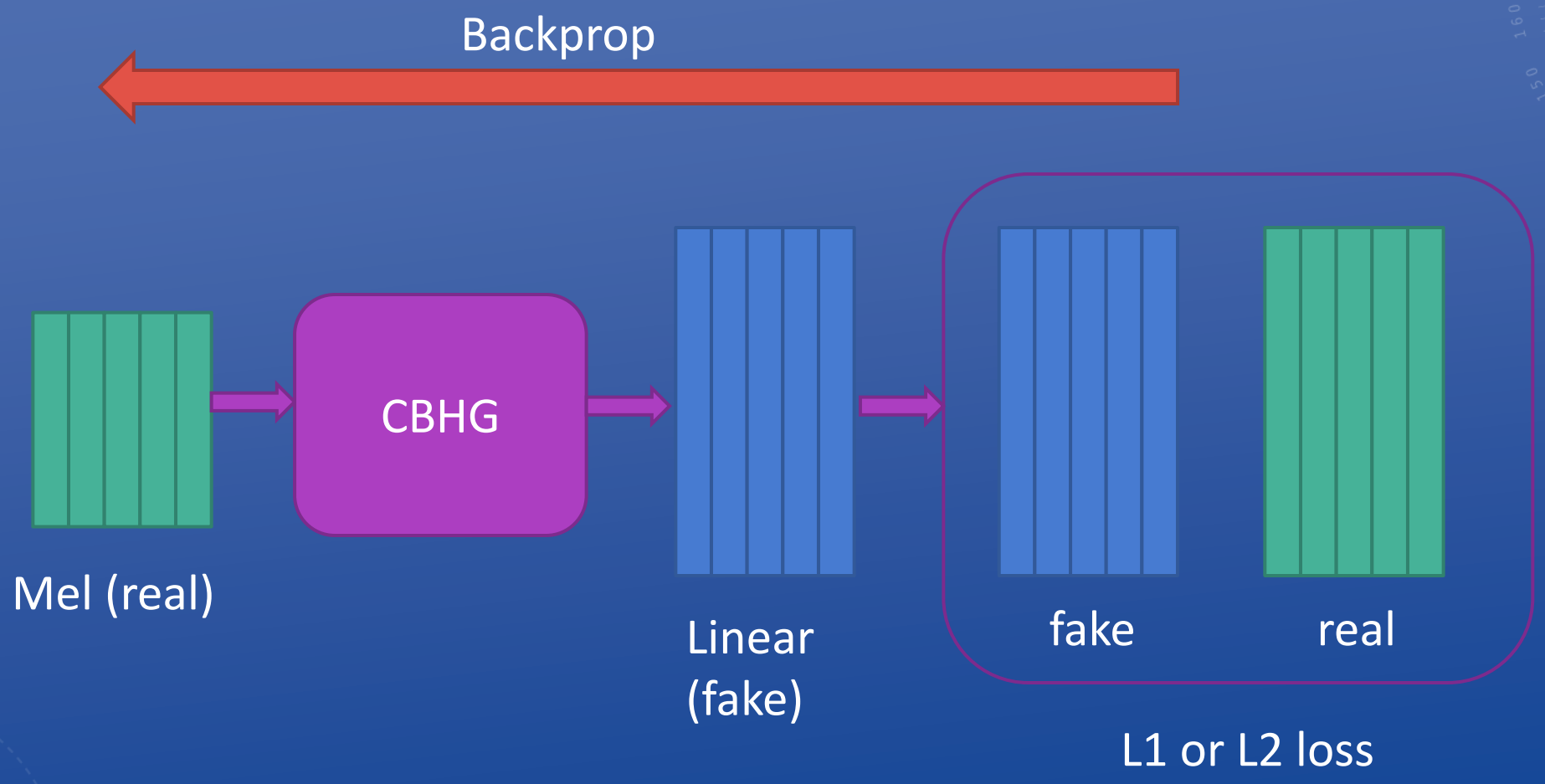


FIGURE 3: PROPOSED FRAMEWORK WITH GAN LOSSES

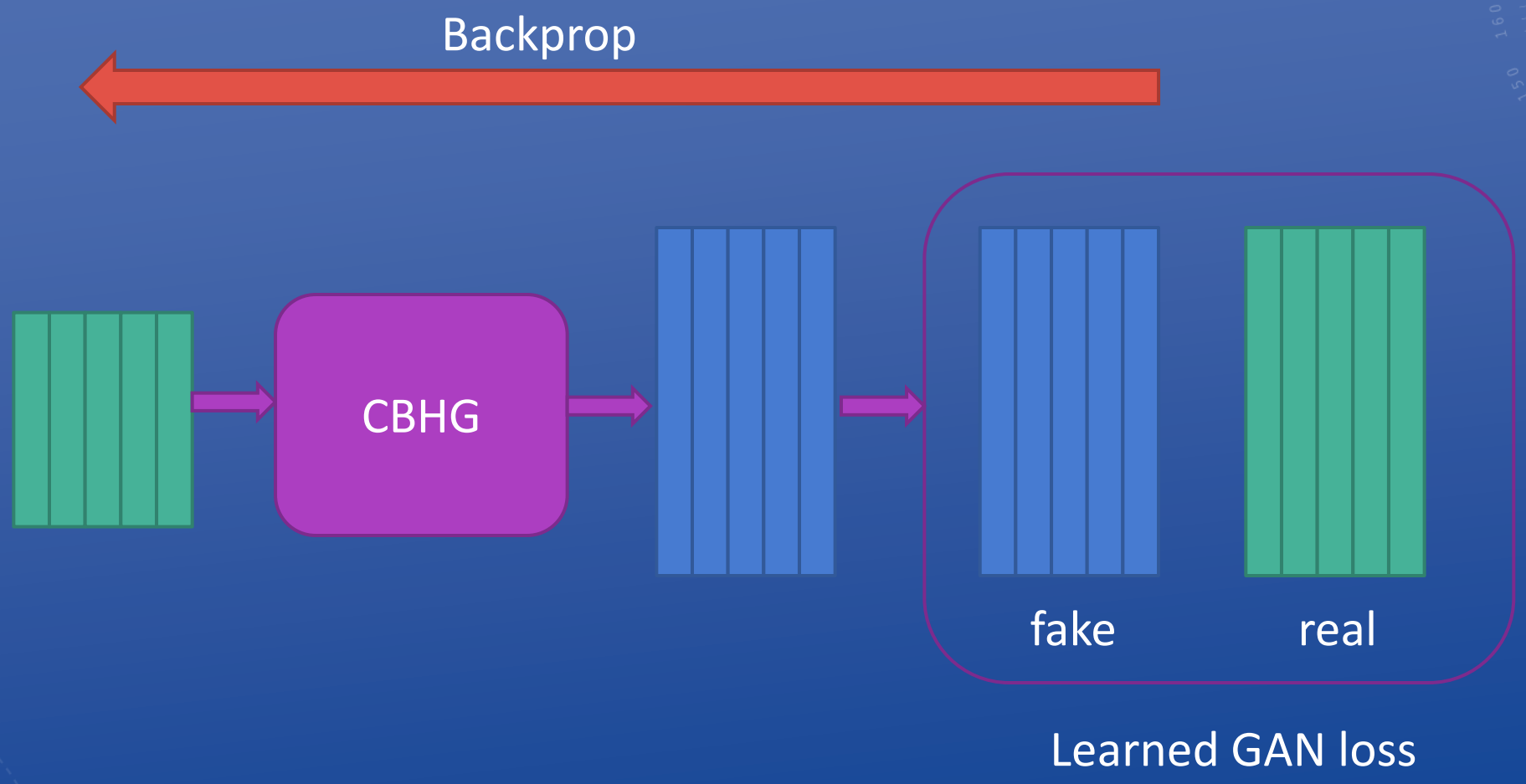


FIGURE 5: GAN DISCRIMINATOR DESIGN

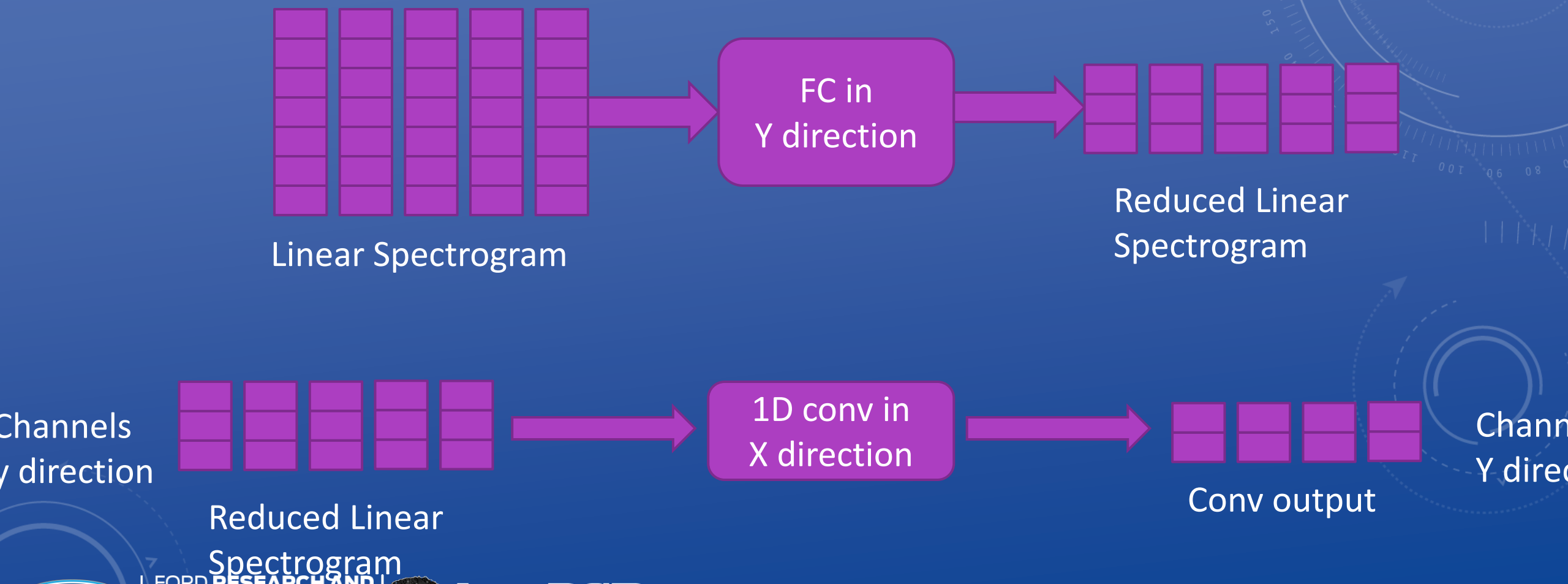


FIGURE 4: OVERVIEW OF SYSTEM ARCHITECTURE WITH GAN LOSSES