Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks

Interspeech 2019, Graz, Austria

Ryan Eloff, André Nortje, Benjamin van Niekerk, Avashna Govender, Leanne Nortje, Arnu Pretorius, Elan van Biljon, Ewald van der Westhuizen, Lisa van Staden, Herman Kamper

Stellenbosch University, South Africa & University of Edinburgh, UK

https://github.com/kamperh/suzerospeech2019

Advances in speech recognition

Advances in speech recognition

- **Addiction to text**: 2000 hours transcribed speech audio; $\sim$350M/560M words text [Xiong et al., TASLP'17]

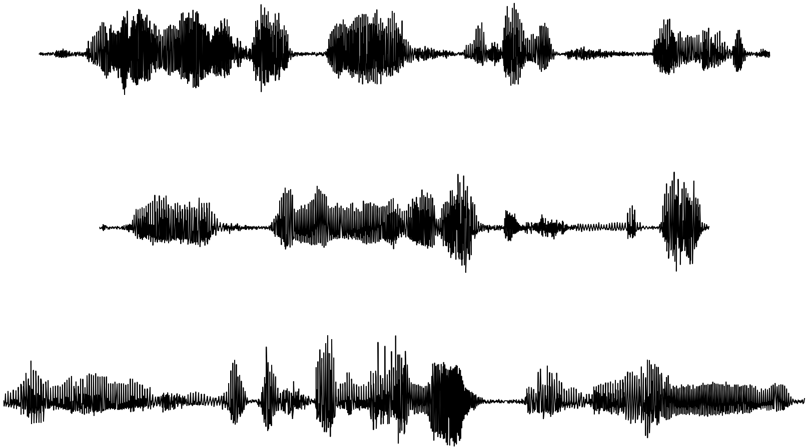
Advances in speech recognition

- **Addiction to text**: 2000 hours transcribed speech audio; $\sim$350M/560M words text [Xiong et al., TASLP'17]
- Sometimes not possible, e.g., for unwritten languages

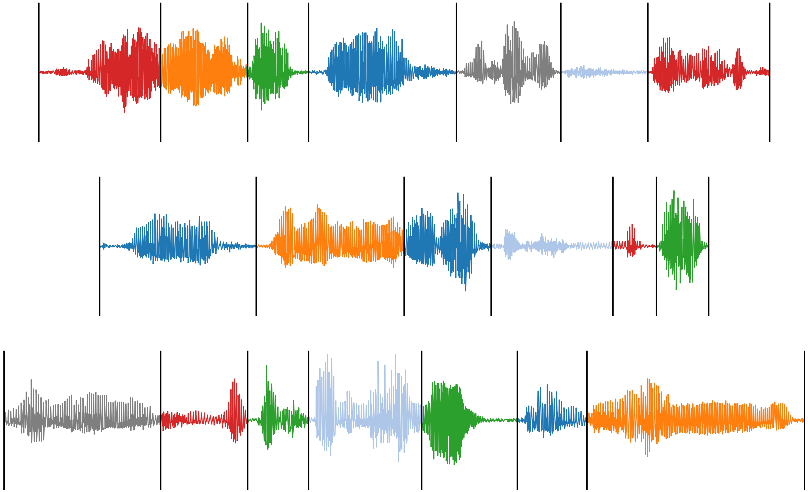
Advances in speech recognition

- **Addiction to text**: 2000 hours transcribed speech audio; $\sim$350M/560M words text [Xiong et al., TASLP'17]
- Sometimes not possible, e.g., for unwritten languages
- Very different from the way human infants learn language

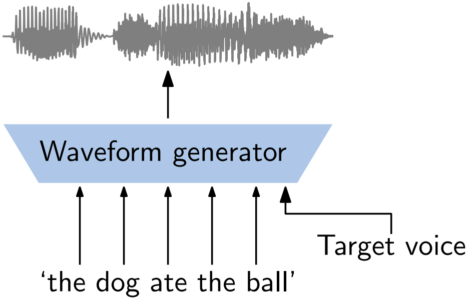
Zero-Resource Speech Challenges (ZRSC)



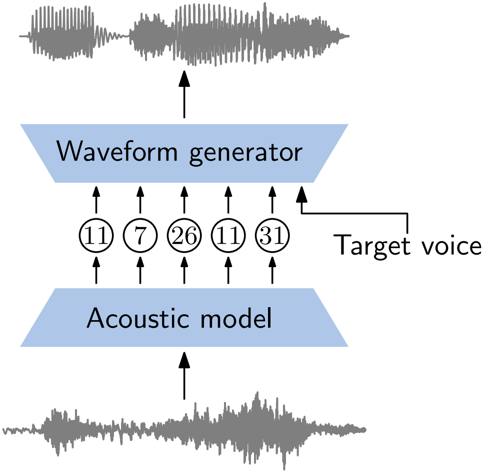
Zero-Resource Speech Challenges (ZRSC)



ZRSC 2019: Text-to-speech without text



ZRSC 2019: Text-to-speech without text



What do we get for training?

What do we get for training?

No labels

What do we get for training?

No labels :)

What do we get for training?

No labels :)

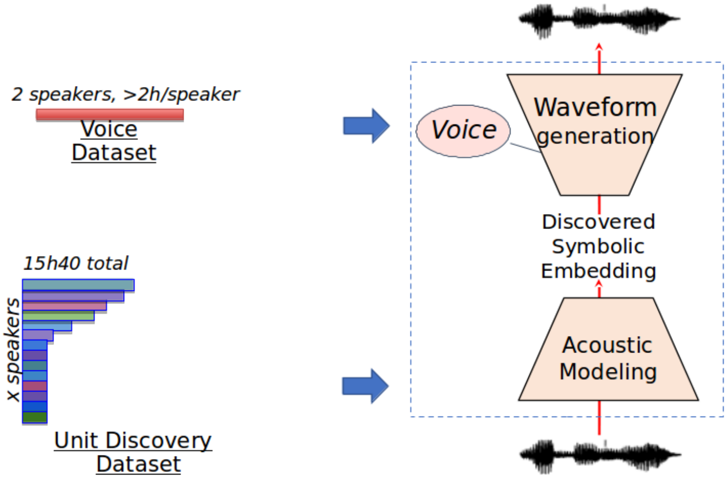
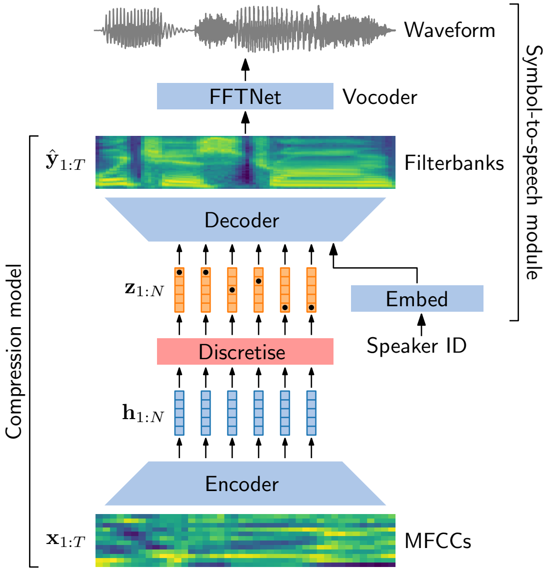
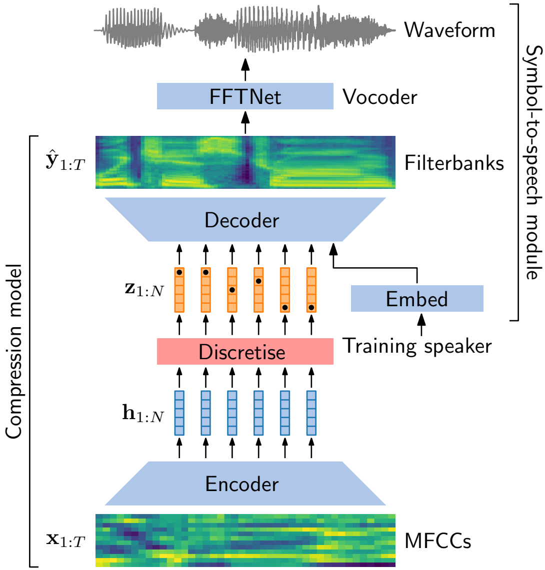


Figure adapted from: `http://zerospeech.com/2019`

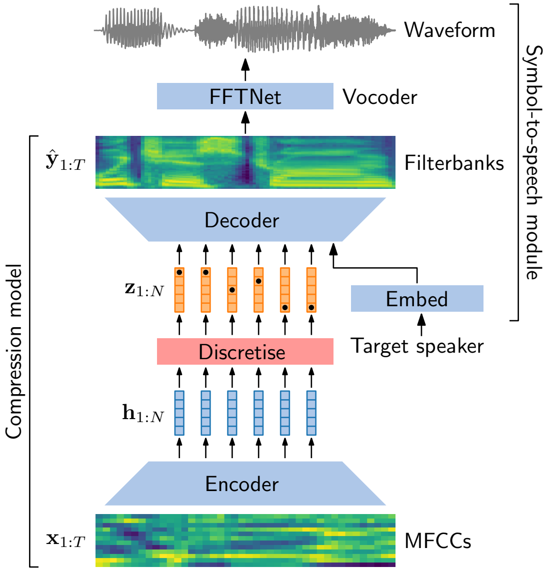
Approach: Compress, decode and synthesise



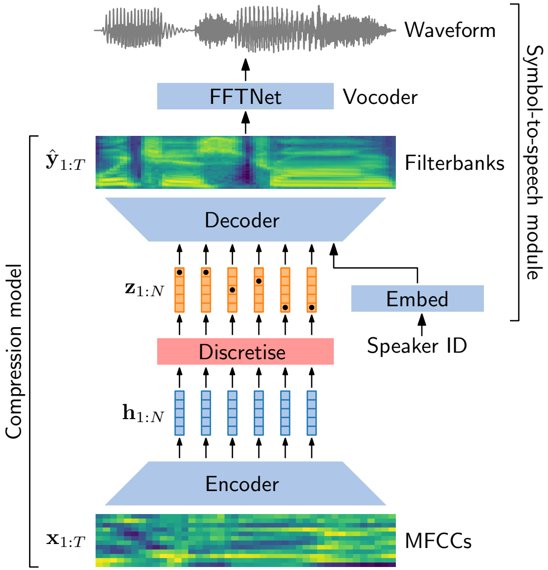
Approach: Compress, decode and synthesise



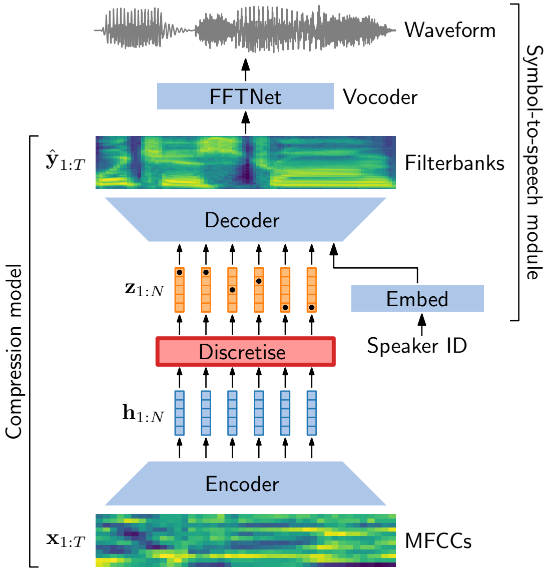
Approach: Compress, decode and synthesise



Approach: Compress, decode and synthesise



Approach: Compress, decode and synthesise



Discretisation methods

- Straight-through estimation (STE) binarisation:

$$\begin{array}{c}\mathbf{h}\\ \begin{bmatrix}0.9\\-0.1\\0.3\\0.7\\-0.8\end{bmatrix}\end{array} \xrightarrow{\text{threshold}} \begin{array}{c}\mathbf{z}\\ \begin{bmatrix}1\\-1\\1\\1\\-1\end{bmatrix}\end{array}$$

- Categorical variational autoencoder (CatVAE):

$$\begin{array}{c}\mathbf{h}\\ \begin{bmatrix}0.9\\-0.1\\0.3\\0.7\\-0.8\end{bmatrix}\end{array} \xrightarrow{\frac{e^{(h_k+g_k)/\tau}}{\sum_{k=1}^{K} e^{(h_k+g_k)/\tau}}} \begin{array}{c}\mathbf{z}\\ \begin{bmatrix}0.86\\0.01\\0.02\\0.11\\0.00\end{bmatrix}\end{array}$$

- Vector-quantised variational autoencoder (VQ-VAE):

$$\begin{array}{c}\mathbf{h}\\ \begin{bmatrix}0.9\\-0.1\\0.3\\0.7\\-0.8\end{bmatrix}\end{array} \xrightarrow[\text{embedding } \mathbf{e}]{\text{Choose closest}} \begin{array}{c}\mathbf{z}\\ \begin{bmatrix}0.8\\-0.2\\0.3\\0.5\\-0.6\end{bmatrix}\end{array}$$

Neural network architectures

- Encoder: Convolutional layers, each layer with a stride of 2
- Decoder: Transposed convolutions mirroring encoder
- Waveform generation: FFTNet autoregressive vocoder
- Also experimented with WaveNet: Sometimes gave noisy output
- Bitrate: Set by number of symbols K and number of striding layers

Evaluation

Human evaluation metrics:

- Mean opinion score (MOS)
- Character error rate (CER)
- Similarity to the target speaker's voice

Evaluation

Human evaluation metrics:

- Mean opinion score (MOS)
- Character error rate (CER)
- Similarity to the target speaker's voice

Objective evaluation metrics:

- ABX discrimination
- Bitrate

Evaluation

Human evaluation metrics:

- Mean opinion score (MOS)
- Character error rate (CER)
- Similarity to the target speaker's voice

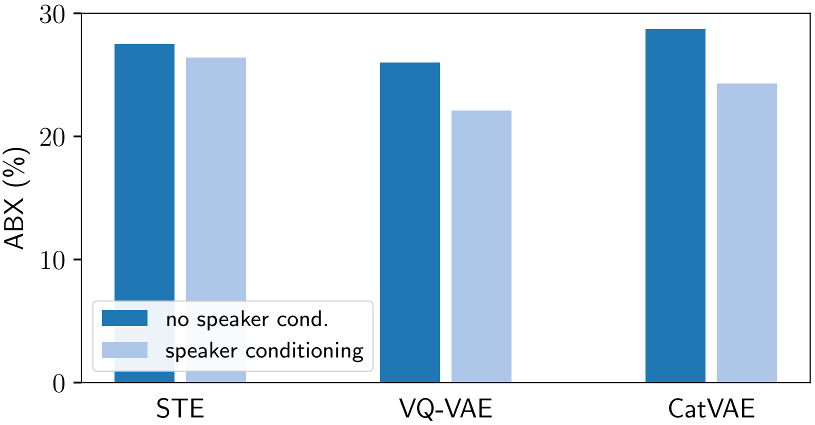
Objective evaluation metrics:

- ABX discrimination
- Bitrate

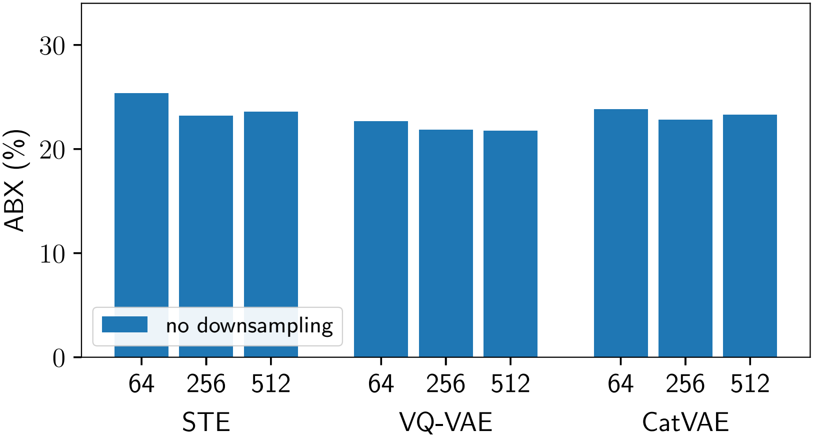
Two evaluation languages:

- English: Used for development
- Indonesian: Held out "surprise language"

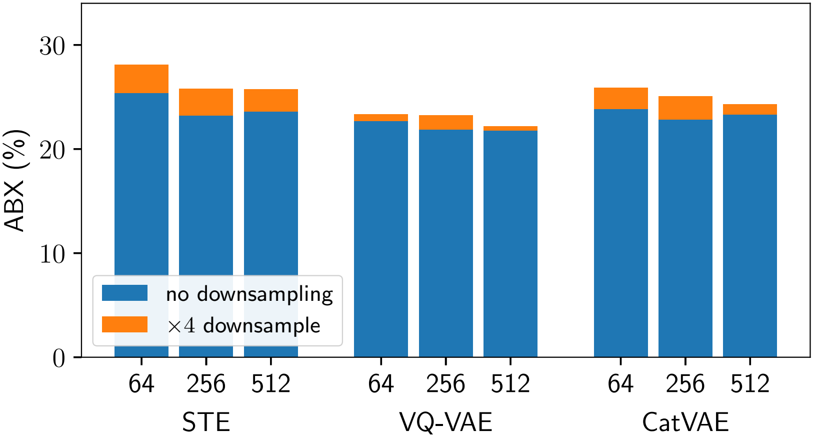
ABX on English with speaker conditioning



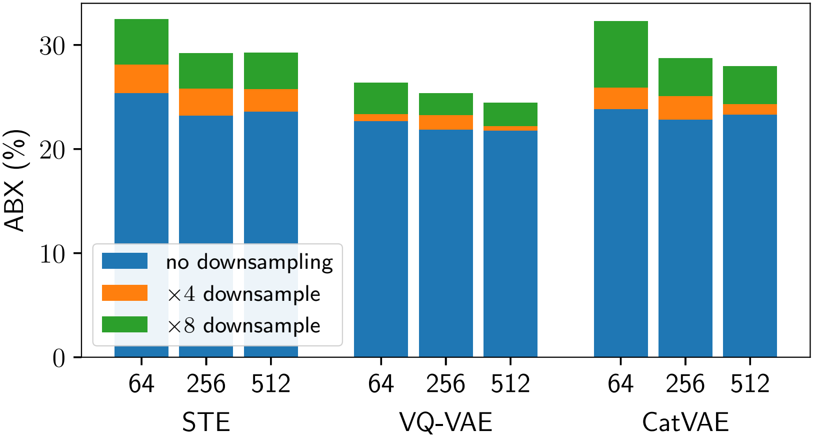
ABX on English for different compression rates



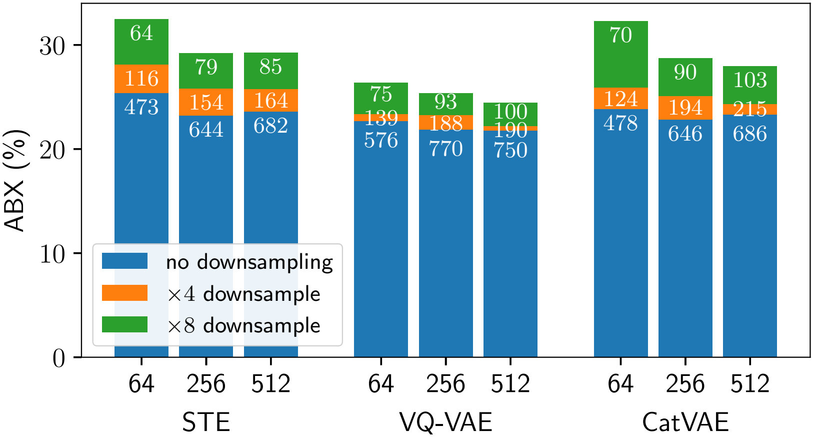
ABX on English for different compression rates



ABX on English for different compression rates



ABX on English for different compression rates



Official evaluation results

Model	CER (%)	MOS [1, 5]	Similarity [1, 5]	ABX (%)	Bitrate
English:					
DPGMM-Merlin	75	**2.50**	**2.97**	35.6	**72**
VQ-VAE-x8	75	2.31	2.49	25.1	88
VQ-VAE-x4	**67**	2.18	2.51	**23.0**	173
Supervised	44	2.77	2.99	29.9	38
Indonesian:					
DPGMM-Merlin	62	**2.07**	**3.41**	27.5	75
VQ-VAE-x8	**58**	1.94	1.95	17.6	**69**
VQ-VAE-x4	60	1.96	1.76	**14.5**	140
Supervised	28	3.92	3.95	16.1	35

Synthesised examples

Model	Input	Synthesised output	Target speaker
English:			
VQ-VAE-x4	Play	Play	Play
VQ-VAE-x4-new		Play	
VQ-VAE-x4	Play	Play	Play
VQ-VAE-x4-new		Play	
Indonesian:			
VQ-VAE-x4	Play	Play	Play
VQ-VAE-x4-new		Play	
VQ-VAE-x4	Play	Play	Play
VQ-VAE-x4-new		Play	

Conclusions

- Speaker conditioning consistently improves performance
- Different discretisation methods are similar (VQ-VAE slightly better)
- Different models difficult to compare because of bitrate
- Future: Does discritisation actually benefit feature learning?

Why do we have ten authors on this paper?

Ryan
Eloff

André
Nortje

Benjamin
van Niekerk

Avashna
Govender

Leanne
Nortje

Arnu
Pretorius

Elan van
Biljon

Ewald van der
Westhuizen

Lisa van
Staden

Herman
Kamper

https://github.com/kamperh/suzerospeech2019 `https://github.com/kamperh/suzerospeech2019`

(Update coming soon)

Straight-through estimation (STE) binarisation

- STE binarisation:

  $z_k = 1$ if $h_k \geq 0$ or $z_k = -1$ otherwise

$$\begin{array}{c} \\ \\ \\ h_4 \\ \\ \end{array} \overset{\mathbf{h}}{\begin{bmatrix} 0.9 \\ -0.1 \\ 0.3 \\ 0.7 \\ -0.8 \end{bmatrix}} \xrightarrow{\text{threshold}} \overset{\mathbf{z}}{\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}} \begin{array}{c} \\ \\ \\ z_4 \\ \\ \end{array}$$

- For backpropagation we need: $\dfrac{\partial J}{\partial \mathbf{h}}$

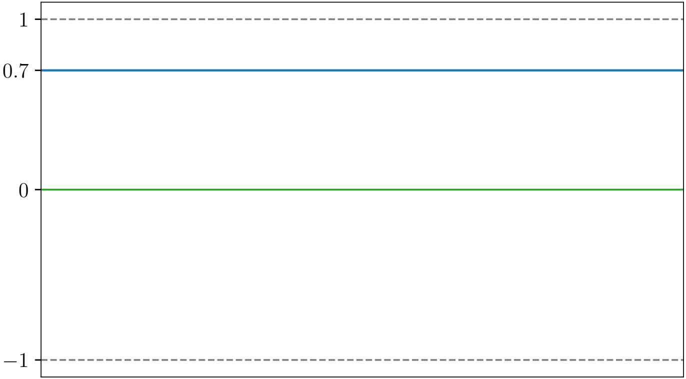
- For single element: $\dfrac{\partial J}{\partial h_k} = \dfrac{\partial z_k}{\partial h_k} \dfrac{\partial J}{\partial z_k}$

- What is $\dfrac{\partial z_k}{\partial h_k}$ with $z_k = \text{threshold}(h_k)$? Cannot solve directly

- Idea: If $z_k \approx h_k$ then we could use $\dfrac{\partial J}{\partial h_k} \approx \dfrac{\partial J}{\partial z_k}$

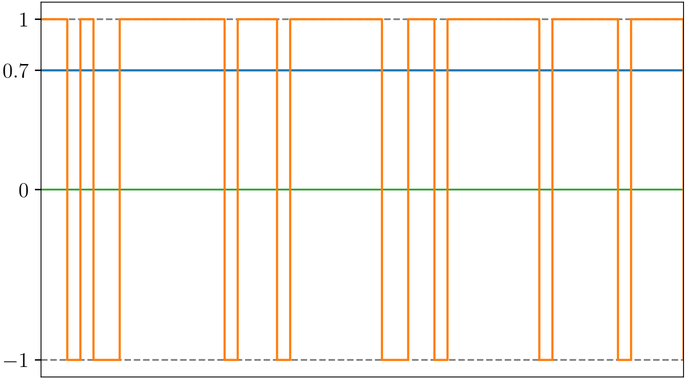
Straight-through estimation (STE) binarisation

As an example, let us say $h_k = 0.7$:



Straight-through estimation (STE) binarisation

Instead of direct thresholding, let us set $z_k = 1$ with probability 0.85 and $z_k = -1$ with probability 0.15:



Estimated mean of z_k over 500 samples: 0.668

Straight-through estimation (STE) binarisation

- So, instead of direct thresholding, we set $z_k = h_k + \epsilon$, where ϵ is sampled noise:

$$\epsilon = \begin{cases} 1 - h_k & \text{with probability } \frac{1+h_k}{2} \\ -h_k - 1 & \text{with probability } \frac{1-h_k}{2} \end{cases}$$

- Since ϵ is zero-mean, the derivative of the expected value of z_k is: $\dfrac{\partial \mathbb{E}[z_k]}{\partial h_k} = 1$

- Therefore, gradients are passed unchanged through the thresholding operation: $\dfrac{\partial J}{\partial \mathbf{h}} \approx \dfrac{\partial J}{\partial \mathbf{z}}$