An Investigation of Optimal Pretraining Methods for Speech Language Models

Eman Ansar advised by **Bhiksha Raj**
Carnegie Mellon University Qatar

To access the final report:

Abstract

- **Problem:** High variability in speech data poses challenges for *pretraining and tokenization.*
- **Approach:** Investigating different Autoencoders (AEs) and novel pretraining techniques.
- **Results:** Promising latent space representation with **DC-AEs on speech (PSNR-28dB)** and developed a **synthetic speech error dataset** yielding **78% classification accuracy**.

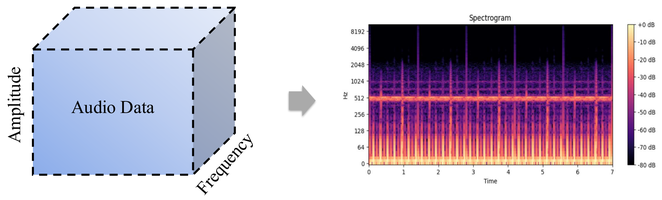
Introduction

How do speech models work?

Raw Audio (Waveform) → Feature Extraction (Spectrogram) → Modelling (Speech Model) → Output (Transcription/ Task)

- **Context and Motivation:** Speech recognition is transforming how humans interact with machines, enabling voice-driven applications in healthcare, education, and beyond. While models like Wav2Vec and DeepSpeech have advanced this field, they face **fundamental challenges** in **processing high-dimensional audio data efficiently and effectively**.

What type of data we are dealing with?



- Amplitude variations
- Frequency components
- Spectral Characteristics
- Temporal dynamics

- **Problem Statement:** Existing approaches in speech models heavily rely on tokenization and **pretraining strategies**, which are less explored compared to NLP and image-based methods. Techniques like Autoencoders (AEs) have shown promise but struggle with **latent space inefficiencies** and **scalability** when applied to speech.

Other Pre-training approaches:

a. Tokenization techniques (Scale Audio Tokenizer)
- Tokens reduced which leads to improved efficiency
- **Needs at least (~2TB) data**

b. Synthetic Dataset Approach (Error Correcting Data)
- Probability Density Analysis
- **Unclear I/O mappings for audio data**

Proposed Method and Description

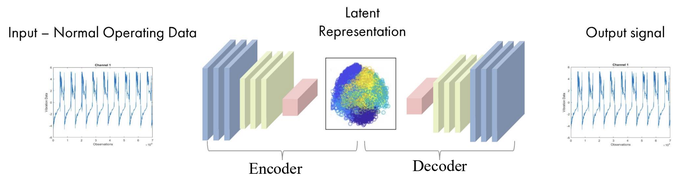
Why did we do preliminary studies with an image dataset? (MNIST)

- Standard dataset
- Baseline Metrics
- Rapid Testing



MNIST Experiments	Speech Application
Image reconstruction	Spectrogram reconstruction
Pixel noise	Environmental noise in audio
Pattern preservation	Phonetic feature preservation
Digit variations	Speaker variations

What are Auto Encoders and why do we need them?



What were some of the key experiments on the MNIST data and their findings?

1. Basic Autoencoder Performance:

- Established baseline reconstruction capability.
- Showed how autoencoders compress and reconstruct 2D patterns

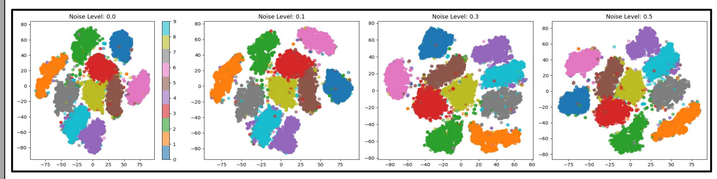
Original MNIST (t-SNE)
Distinct clusters for each digit in original data

Basic AE Latent Space (t-SNE)
Distinct clusters for each digit in original data

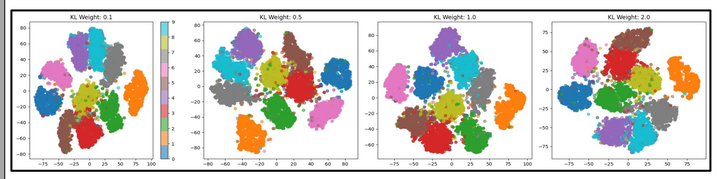
The basic autoencoder successfully preserves the clustering structure of the original data.

2. Other Autoencoder Experiments:

- Added Gaussian noise levels (0.0 → 0.6)
- Demonstrated how reconstruction quality degrades with noise



VAE Latent Space Robustness to Noise
t-SNE visualizations of VAE latent space with increasing Gaussian noise (0.0, 0.1, 0.3, 0.5 from left to right)



KL divergence Weight Effect on VAE Latent Space
t-SNE visualizations showing regularization effects with increasing KL weights (0.1, 0.5, 1.0, 2.0 from left to right)

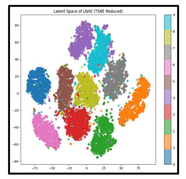
Key Insight: VAEs demonstrate remarkable resilience to input noise, maintaining distinguishable digit clusters even at high noise levels (0.5). This robustness suggests VAEs could effectively handle the natural variability and noise present in speech signals, making them promising candidates for speech representation learning.

Moving to Deep Compression Autoencoder:

How does DC-AE work and why DC-AE?

Input Image (high-res e.g. 1024 x 1024) → DC-AE encoder (High compression) → Very small latent space (e.g., 16x16) → DC-AE decoder (reconstructing back to high resolution)

- ✓ **19.1x faster**
- ✓ **Improved Quality (Better FID)**
- ✓ **Better handling of high-dim data**



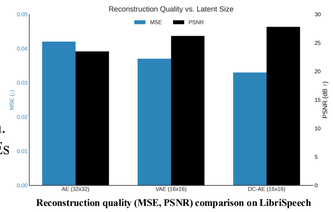
latent space learned by a DC-AE on MNIST

- Existing tokenization approaches are **computationally expensive** or require **extensive fine-tuning** on large datasets.
- DC-AEs use **residual autoencoding** to compress data efficiently, making them ideal for handling high-dimensional audio data.

Key Insight: From the graph, the **clear separation** between **digit clusters** highlights that the DC-AE successfully organizes similar inputs together in the latent space, enabling efficient and meaningful data representation.

How did DC-AE perform on the speech data? (LibriSpeech)

- Adapted DC-AE architecture for LibriSpeech spectrograms (128×128 → 16×16).
- Achieved **~28 dB PSNR** and preserved phoneme structures during reconstruction.
- Demonstrated compression and improved reconstruction quality over baseline AEs



Reconstruction quality (MSE, PSNR) comparison on LibriSpeech

Why and how did we need to create a new synthetic speech dataset?

- Created **10,000** synthetic samples by perturbing **TIMIT phoneme sequences**.
- Labeled samples into minor, moderate, severe error categories.
- Trained classifiers:
  - *CNN: 75% accuracy*
  - *VAE + MLP: 78% accuracy → Best performance*!

Model	Accuracy	F1 Score	Notes
Logistic Reg	62%	0.58	Baseline
CNN	75%	0.73	With spectrogram features
VAE+MLP	78%	0.76	Using latent embeddings from VAE

Performance Comparison of Different Models on Speech Error Classification.

Conclusion and Future work:

What did my research contribute?

- **DC-AEs** provide **efficient, compact representations** of speech data without sacrificing **reconstruction quality**.
- **VAEs** offer **better latent space generalization** compared to traditional AEs.
- **Synthetic error datasets** are viable for robustness training in speech LLMs.
- **Latent compressio**n and **synthetic robustness training** offer new paths toward scalable, efficient speech models.

What are the next steps to reach the end goal of democratizing speech technology?

a. Full-scale evaluation of VAE-enhanced DC-AEs on larger speech datasets (e.g., full LibriSpeech, TED-LIUM).

b. Expand synthetic error datasets with more diverse error types (e.g., deletion, insertion).

c. Integrate compressed latent representations into end-to-end ASR pipelines.

d. Compare performance and efficiency against the state-of-the-art methods

References and Acknowledgments:

1. Chen, J., Cai, H., Yang, S., Tang, H., Chen, J., Xie, E., Li, M., Lu, Y., & Han, S. (2024). Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint*, arXiv:2403.12345.
2. Kalyan, S., Rajasekharan, A., & Hariharan, K. (2021). A survey on tokenization techniques for deep learning models in NLP and speech processing. *ACM Computing Surveys, 54*(4), 1–35.
3. Shen, Y., Tenenbaum, J., et al. (2023). How to learn from mistakes on grade-school math problems. *arXiv preprint*, arXiv:2303.12345.
4. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning*, 1096–1103.

MEETING OF THE MINDS | Carnegie Mellon University Qatar Research