# MyFluentEcho

## Speech Assistant Designed to Support People Who Stutter
## “Transforming Disfluent Speech into Confident Voice”

**Carson Sager**

**ECEN5060 – Deep Learning**

**Electrical and Computer Engineering Department**

**Oklahoma State University**

**carson.sager@okstate.edu**

# Outline

- Background
- Design Introduction
- Technical Approach
- Implementation
- Results and Future Directions

# Background: Stuttering and Auditory Feedback

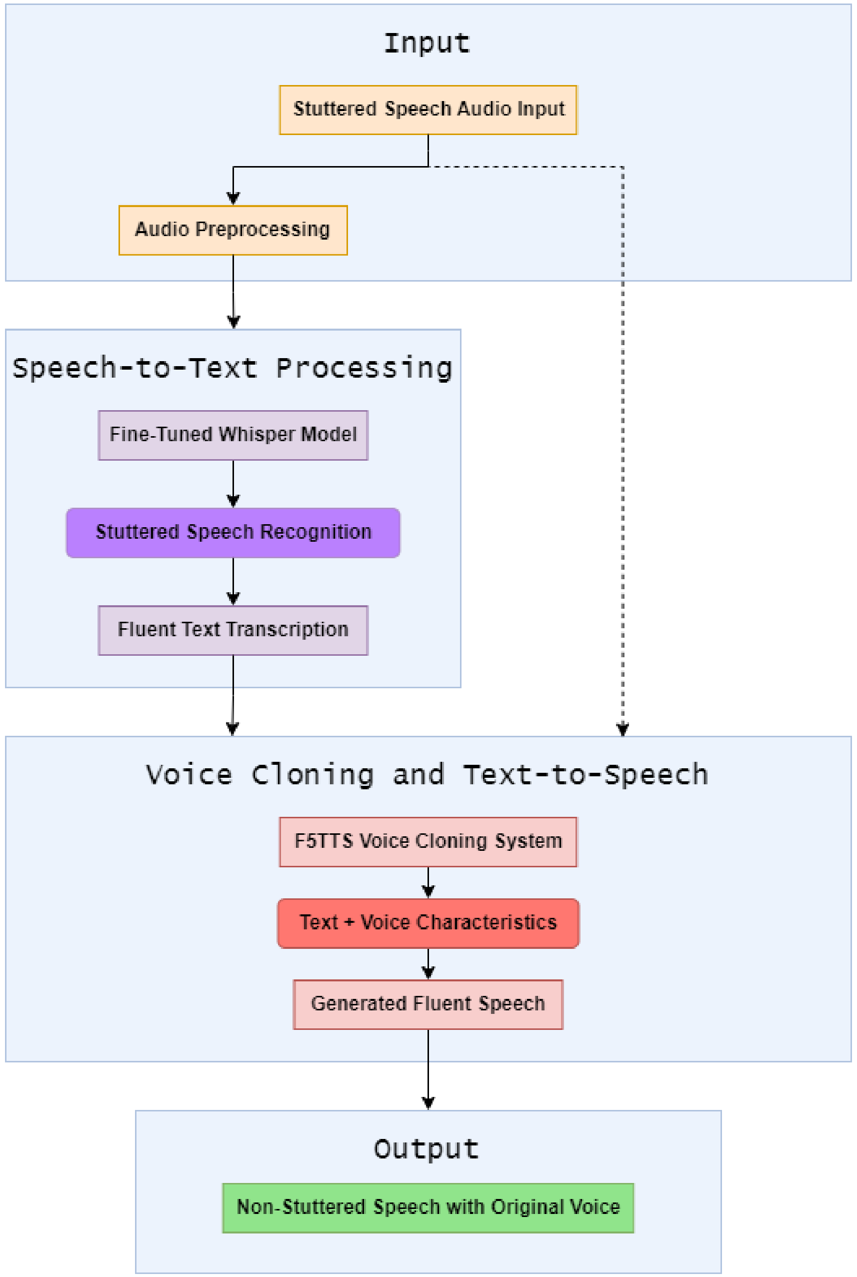
- Stuttering is a neurodevelopmental speech disorder affecting rhythm and flow of speech, often characterized by repetitions, prolongations, and blocks.
- For People Who Stutter (PWS), the act of speaking can be effortful, emotionally taxing, and socially limiting.
- Delayed Auditory Feedback (DAF): A technique that plays a speaker’s voice back to them with a slight delay.
  - Disrupts the typical auditory-motor loop, which may reduce the pressure to "monitor" one’s own speech in real-time—this often contributes to increased fluency.
  - May alter timing and rhythm cues in speech production, helping PWS shift to a more fluent speech pattern by slowing down articulation or improving pacing.
- Hypothesized drawback: DAF only uses the stuttered speech of the individual in its feedback

What if we developed a model that allows therapists to use DAF, with the feedback to the PWS being <u>fluent</u>?

# The Answer: MyFluentEcho

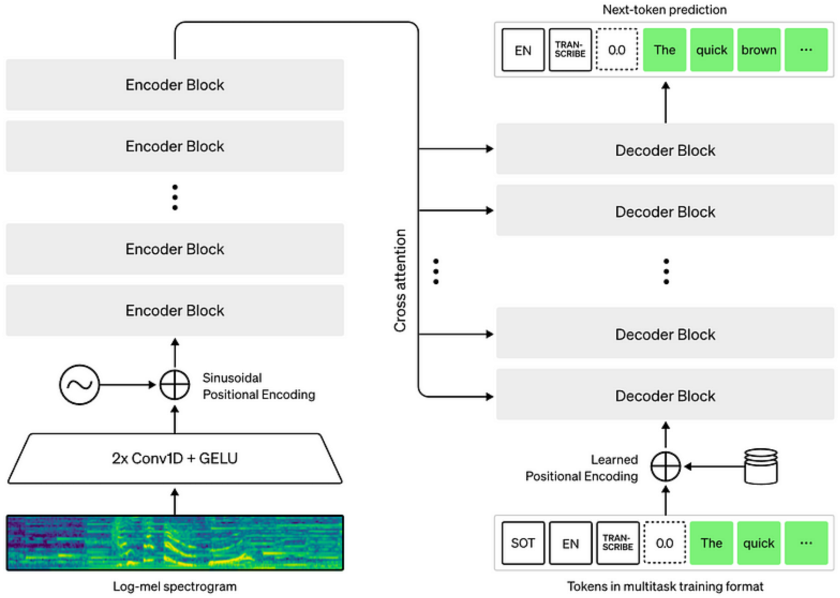
- Transforms stuttered speech into fluent output — using your own voice
- Combines speech recognition, disfluency correction, and voice cloning
- Provides natural, fluent auditory feedback as an alternative to the current DAF
- Supports fluent expression of ideas and self-confidence in a PWS’s ability to clearly speak
- *Back end:* **Whisper** for speech recognition and disfluency correction
    - ➢ Fine-tuned on the **FluencyBank Timestamped** dataset to improve stutter recognition
- *Front end:* **F5-TTS** for text-to-speech generation of the fluent text in the user’s voice

# Processing Diagram of MyFluentEcho


Input
Stuttered Speech Audio Input
Audio Preprocessing
Speech-to-Text Processing
Fine-Tuned Whisper Model
Stuttered Speech Recognition
Fluent Text Transcription
Voice Cloning and Text-to-Speech
F5TTS Voice Cloning System
Text + Voice Characteristics
Generated Fluent Speech
Output
Non-Stuttered Speech with Original Voice

# OpenAI-Whisper

- **Encoder-Decoder Transformer** - Built on the transformer architecture with an encoder that processes audio spectrograms and a decoder that generates text tokens
- **Audio Processing** - Converts audio into log-mel spectrograms with 80 frequency bands as input representation
- **Model Variants** - Available in different sizes from tiny (39M parameters) to large (1.55B parameters), with base model (74M) using 6 encoder and 6 decoder layers
- **Attention Mechanism** - Uses multi-headed attention to capture relationships between audio features and text, allowing the model to focus on relevant parts of the input
- **Multilingual Capabilities** - Trained on diverse datasets across 98 languages, enabling zero-shot transfer learning for specialized applications like stuttering speech recognition



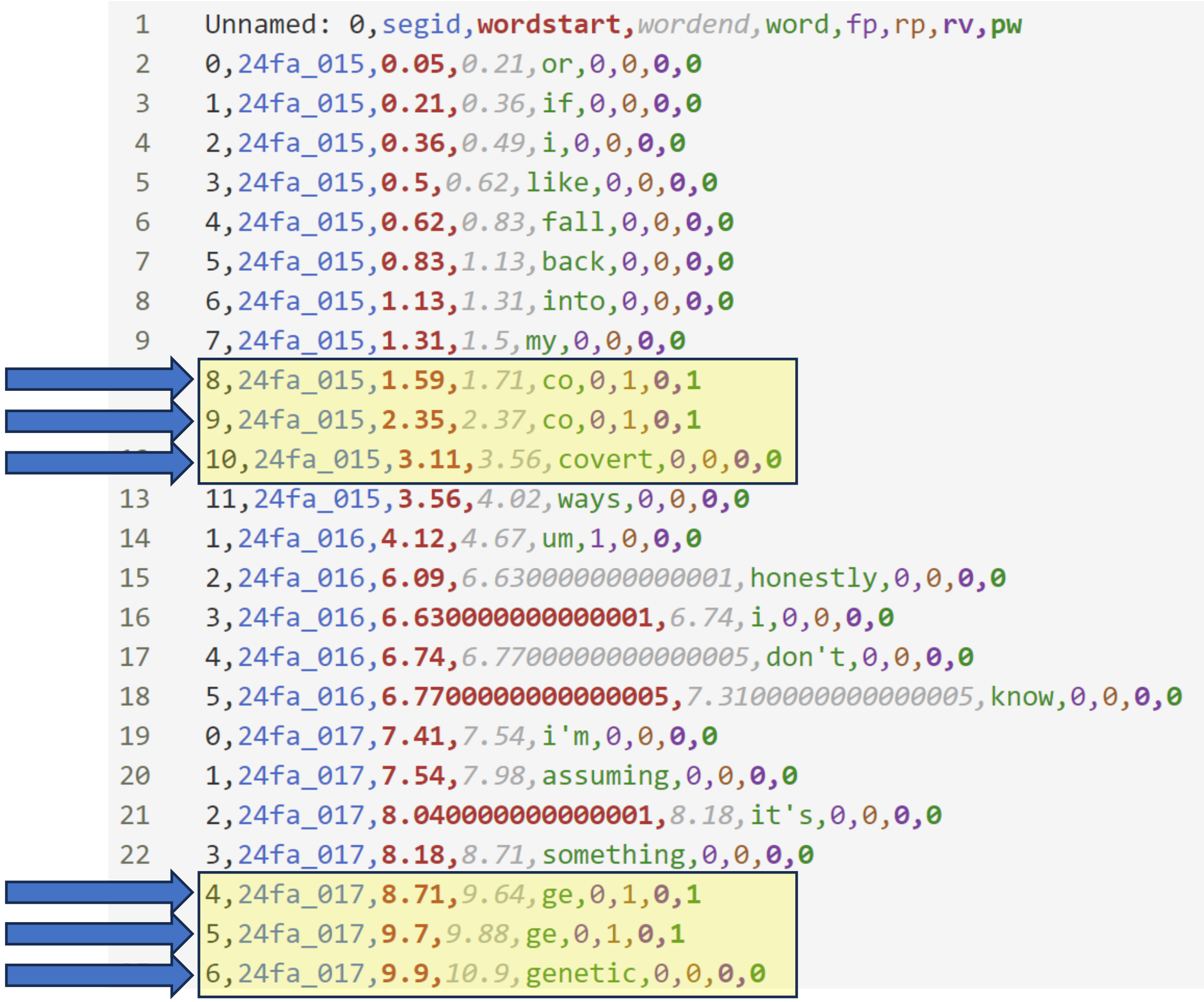
# FluencyBank Timestamped Dataset

- Overview:
  - Specialized corpus of stuttered speech recordings with detailed time-aligned annotations
  - Part of the larger FluencyBank initiative within the TalkBank project
  - Contains natural speech samples from individuals with various fluency disorders
- Key Features
  - Detailed annotations of different stuttering types:
    - fp: Filled pauses
    - rp: Sound/syllable repetitions*
    - rv: Word/phrase revisions
    - pw: Part-word repetitions*
- Fluency Bank originally only included Audio recordings, but Fluency Bank Timestamped developed the corresponding CSV annotation files
- Time-aligned transcriptions with start and end times for each word
- More than 3300 2- to 15-second audio segments from diverse speakers across different adult age groups

# Timestamped Data Manipulation

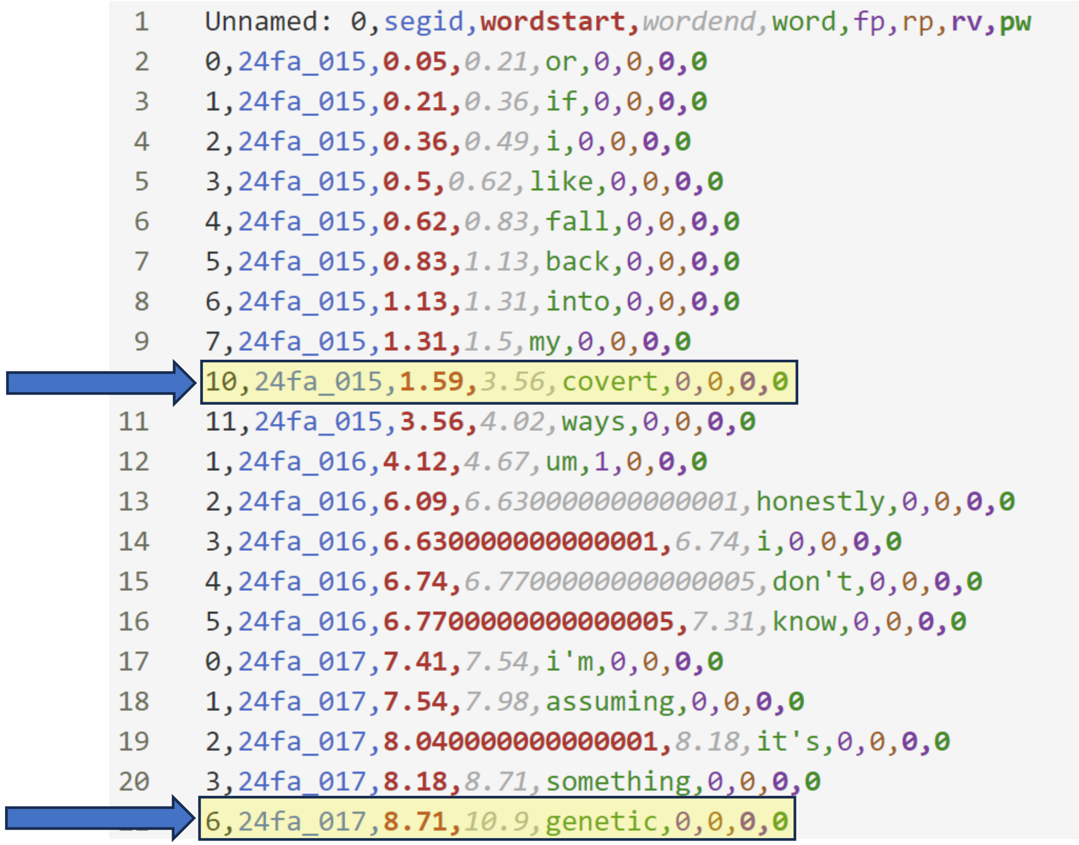
- The original purpose of the TimeStamped dataset was solely to examine the success of modern speech-to-text models (like Whisper) when presented with audio of PWS
  - ➢ No training involving this dataset was found
- Considering Whisper was trained on 30-second audio segments and the FluencyBank Timestamped set mostly contained 2- to 15-second clips, some data needed to be combined
- Since the shorter clips were uttered by the same participant if they contained the same prefix in the audio .wav and .csv label, 3 audio clips were combined at a time
- To compensate for this in the .csv files, the timestamped phrases also needed to be combined so that the timestamps were continuous
- Additionally, to properly train the model to detect and correct stuttering, the full disfluent audio needed to correspond to a single fluent phrase being said…

## Combined audio with stutter



```
Unnamed: 0,segid,wordstart,wordend,word,fp,rp,rv,pw
0,24fa_015,0.05,0.21,or,0,0,0,0
1,24fa_015,0.21,0.36,if,0,0,0,0
2,24fa_015,0.36,0.49,i,0,0,0,0
3,24fa_015,0.5,0.62,like,0,0,0,0
4,24fa_015,0.62,0.83,fall,0,0,0,0
5,24fa_015,0.83,1.13,back,0,0,0,0
6,24fa_015,1.13,1.31,into,0,0,0,0
7,24fa_015,1.31,1.5,my,0,0,0,0
8,24fa_015,1.59,1.71,co,0,1,0,1
9,24fa_015,2.35,2.37,co,0,1,0,1
10,24fa_015,3.11,3.56,covert,0,0,0,0
11,24fa_015,3.56,4.02,ways,0,0,0,0
1,24fa_016,4.12,4.67,um,1,0,0,0
2,24fa_016,6.09,6.630000000000001,honestly,0,0,0,0
3,24fa_016,6.630000000000001,6.74,i,0,0,0,0
4,24fa_016,6.74,6.7700000000000005,don't,0,0,0,0
5,24fa_016,6.7700000000000005,7.3100000000000005,know,0,0,0,0
0,24fa_017,7.41,7.54,i'm,0,0,0,0
1,24fa_017,7.54,7.98,assuming,0,0,0,0
2,24fa_017,8.040000000000001,8.18,it's,0,0,0,0
3,24fa_017,8.18,8.71,something,0,0,0,0
4,24fa_017,8.71,9.64,ge,0,1,0,1
5,24fa_017,9.7,9.88,ge,0,1,0,1
6,24fa_017,9.9,10.9,genetic,0,0,0,0
```

## Combined audio without stutter



```
Unnamed: 0,segid,wordstart,wordend,word,fp,rp,rv,pw
0,24fa_015,0.05,0.21,or,0,0,0,0
1,24fa_015,0.21,0.36,if,0,0,0,0
2,24fa_015,0.36,0.49,i,0,0,0,0
3,24fa_015,0.5,0.62,like,0,0,0,0
4,24fa_015,0.62,0.83,fall,0,0,0,0
5,24fa_015,0.83,1.13,back,0,0,0,0
6,24fa_015,1.13,1.31,into,0,0,0,0
7,24fa_015,1.31,1.5,my,0,0,0,0
10,24fa_015,1.59,3.56,covert,0,0,0,0
11,24fa_015,3.56,4.02,ways,0,0,0,0
1,24fa_016,4.12,4.67,um,1,0,0,0
2,24fa_016,6.09,6.630000000000001,honestly,0,0,0,0
3,24fa_016,6.630000000000001,6.74,i,0,0,0,0
4,24fa_016,6.74,6.7700000000000005,don't,0,0,0,0
5,24fa_016,6.7700000000000005,7.31,know,0,0,0,0
0,24fa_017,7.41,7.54,i'm,0,0,0,0
1,24fa_017,7.54,7.98,assuming,0,0,0,0
2,24fa_017,8.040000000000001,8.18,it's,0,0,0,0
3,24fa_017,8.18,8.71,something,0,0,0,0
6,24fa_017,8.71,10.9,genetic,0,0,0,0
```

- the single word covers the entire time (wordend-wordstart) of the stutter + completed word in original csv

# Fine Tuning Training Strategy

- **Hugging Face Seq2SeqTrainer:** Utilized specialized trainer class designed for sequence-to-sequence models to handle the encoder-decoder architecture of Whisper
  - ➢ Built on PyTorch backend for tensor operations and gradient computation
  - ➢ Leveraged PyTorch's CUDA optimization for GPU acceleration
- **Seq2SeqTrainingArguments:** Configured comprehensive training parameters including learning rate (5e-5), batch size, gradient accumulation, and training steps
- **Custom Data Collator:** Implemented DataCollatorSpeechSeq2SeqWithPadding to handle variable-length audio inputs and tokenized text outputs
- **Evaluation Metrics Integration:** Defined custom compute_metrics function to calculate Word Error Rate during model checkpointing and observed training loss across fine-tuning
- **Hyperparameter Optimization:** Conducted multiple training runs to identify optimal configurations for stuttering speech recognition

# Model Comparisons and Evaluation

- **Word Error Rate (WER)** is a standard metric used to evaluate the performance of automatic speech recognition (ASR) systems like OpenAI's Whisper model (and our evolved model). It quantifies how accurately the model transcribes spoken language into text:

$$WER = \frac{S + D + I}{N}$$

*S = # of substitutions* *D = # of deletions* *I = # of insertions*

*N = Total # of words in the reference transcription*

| Whisper Model | VAL WER | Test WER | Test Improvement From Default |
|---|---|---|---|
| Default “Base” | - | 0.40003299 | - |
| Trained “Base” | 0.39480159 | 0.23045199 | 42.39% |
| Default “Small” | - | 0.36737050 | - |
| Trained “Small’ | 0.24992310 | 0.189706367 | 48.36% |

# Fine-Tuned Whisper Demonstration

**Example Test Set Audio:**

**Base Whisper Transcription:**

Bye bye, do stutter a lot, and so be it as long as my message is clear.

**Trained Whisper Transcription:**

But if I do stutter a lot then so be it as long as my message is clear.

# F5TTS: Voice Cloning Technology

**What is F5TTS?**

- *Fast, Few-shot, Fine-tunable, and Factorized Text-to-Speech (F5TTS)*
- Advanced neural voice synthesis system that enables high-quality voice cloning
- Capable of capturing speaker identity and voice characteristics from short audio samples
- Developed to generate natural-sounding speech that matches reference speaker's voice

**How It Works in Our System**

- Takes fluent text transcription from fine-tuned Whisper as input
- Uses the original stuttered speech as a voice reference sample
- Preserves the speaker's unique voice characteristics (accent, timbre, intonation)
- Generates new speech that maintains speaker identity without stuttering

**Key Benefits**

- Requires only a small sample of reference audio (few-shot learning)
- Maintains high naturalness in synthesized speech
- Preserves speaker identity effectively
- Enables fluent speech production

# System Demonstration

| | Original Audio: | MyFluentEcho: |
|---|---|---|
| Clip 1 (Male): | 🔊 | 🔊 |
| Clip 2 (Female): | 🔊 | 🔊 |

# Summary of Results

- Key Achievements
  - **Significant WER Reduction:** Fine-tuning Whisper reduced Word Error Rate by almost 50% on stuttered speech
  - **Disfluency Recognition:** Model successfully learned to identify and correct different types of stuttering patterns
  - **Enhanced Transcription Quality:** Fine-tuned model properly interprets stuttered speech as fluent text
  - **Effective Voice Preservation:** Combined with F5TTS, system maintains speaker's voice characteristics while removing disfluencies
  - **Complete Fluency Pipeline:** Created end-to-end system from speech recognition to fluent voice generation
- Impact for People Who Stutter (PWS)
  - Provides natural, fluent auditory feedback as an alternative to traditional Delayed Auditory Feedback
  - Maintains personal voice identity while removing disfluencies
  - Creates potential for new therapeutic approaches using fluent speech models

# Future Work

- **Real-Time DAF Implementation:** Optimize model for low-latency processing to enable Delayed Auditory Feedback applications
- **Filled Pause Recognition:** Enhance model's ability to identify and correctly process hesitation phenomena (um, uh, er)
- **Revision Pattern Handling:** Develop techniques to detect and process word/phrase revisions typical in stuttered speech
- **Model Size Reduction:** Compress model for deployment on edge devices and mobile applications
- **Personalized Adaptation:** Create framework for rapid model adaptation to individual speech patterns

# Challenges and Considerations

## Disclaimers

- This project is proposed by a computer scientist, not a speech-language pathologist or therapist
- MyFluentEcho represents a technical proof-of-concept rather than a validated therapeutic tool

## Ethical Considerations

- Self-Image Impact: Potential psychological effects of hearing one's voice without stuttering
- Therapeutic Approach: Should complement rather than replace established speech therapy techniques
- Informed Consent: Users must understand the experimental nature of the technology

## Technical Limitations

- Real-Time Performance: Current processing latency exceeds requirements for true DAF applications
- Model Generalizability: Performance may vary across different stuttering patterns and individuals
- Voice Preservation Accuracy: Voice cloning might not capture all nuances of the original

# References

- Bernstein Ratner, N., & MacWhinney, B. (2018). "FluencyBank: A new resource for fluency research and practice." *Journal of Fluency Disorders, 56*, 69-80.
- Bayerl, S. P., Wagner, D., Bocklet, T., & Riedhammer, K. (2022). "FluencyBank Timestamped: Annotations and Processing for Stuttering Event Detection." Text, Speech, and Dialogue Conference Proceedings.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.*
- Wang, B., Chen, Z., Wu, J., Tao, J., & Wang, D. (2023). "F5TTS: Fast Text-to-Speech System with Fine-grained Feature Fusion." *Proceedings of ICASSP 2023.*
- Howell, P., & Sackin, S. (2000). "Speech rate modification and its effects on fluency reversal in fluent speakers and people who stutter." *Journal of Developmental and Physical Disabilities, 12, 291-315.*
- Lincoln, M., Packman, A., & Onslow, M. (2006). "Altered auditory feedback and the treatment of stuttering: A review." Journal of Fluency Disorders, 31(2), 71-89.
- Liu, Y., et al. (2023). "Whisper Fine-Tuning for Improved ASR Performance on Non-Standard Speech." *IEEE Transactions on Audio, Speech, and Language Processing*

# THANK YOU!

# Questions?