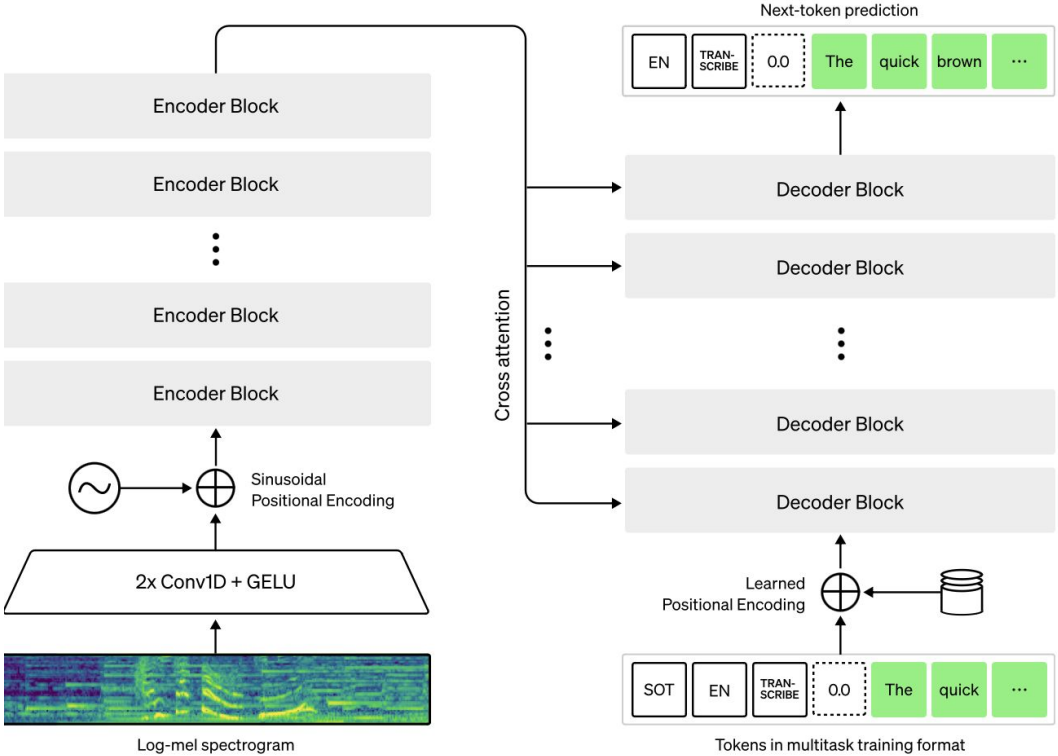
Insanely Fast Whisper

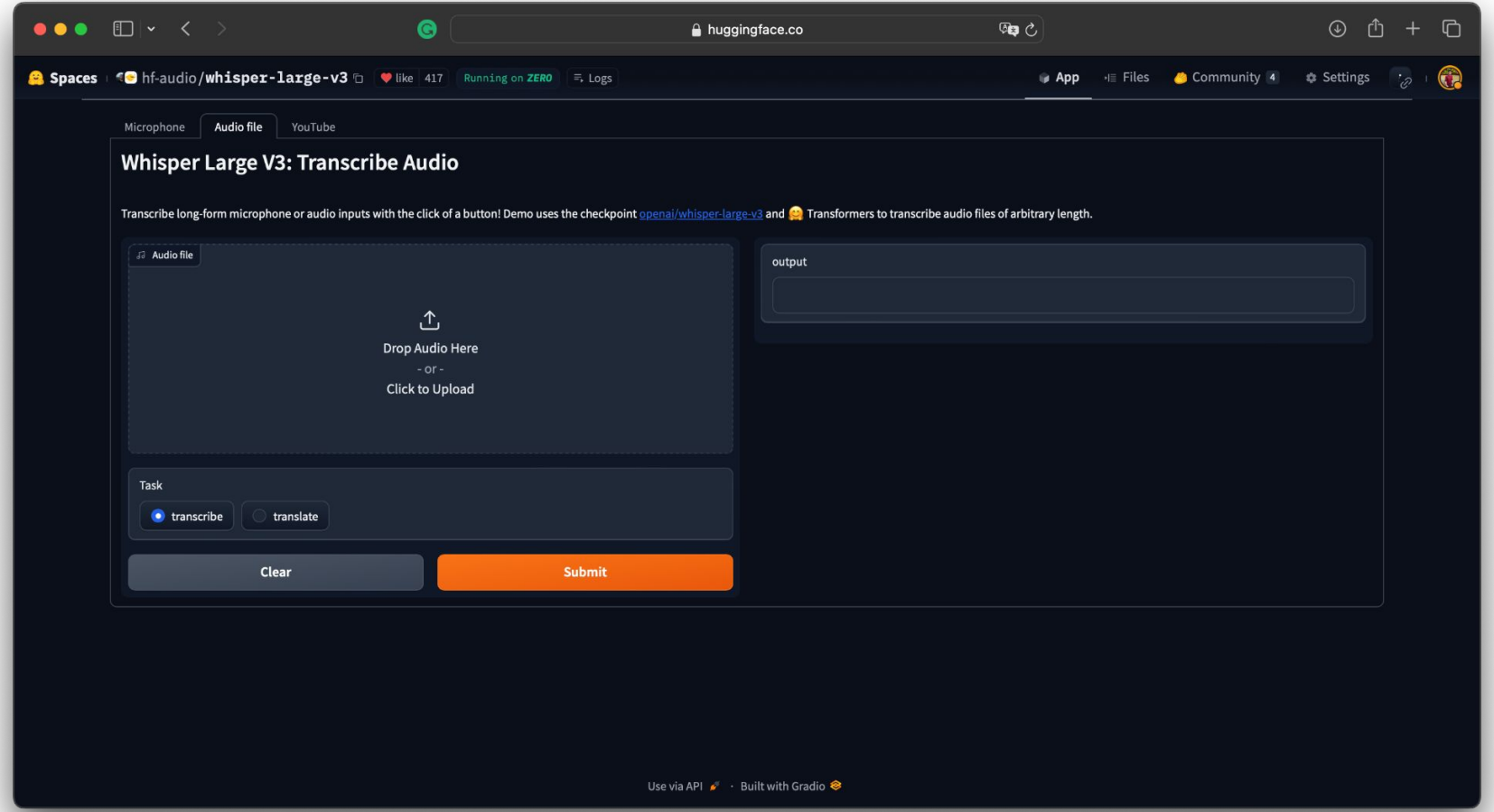
VB

What is Whisper?



1. Speech to Text
2. Trained on 5M hours
3. Multilingual
4. State of the Art (zero-shot)

[ref]


huggingface.co
Spaces
hf-audio/whisper-large-v3
like 417
Running on ZERO
Logs
App
Files
Community 4
Settings
Microphone
Audio file
YouTube
Whisper Large V3: Transcribe Audio
Transcribe long-form microphone or audio inputs with the click of a button! Demo uses the checkpoint openai/whisper-large-v3 and Transformers to transcribe audio files of arbitrary length.
Audio file
Drop Audio Here
- or -
Click to Upload
output
Task
transcribe
translate
Clear
Submit
Use via API
Built with Gradio

[ref]

Why optimise it?

1. Big fat model -> 1.5B parameter model
2. Real-time transcription
3. Reduce inference cost
4. Most importantly, it’s fun

Let’s optimise

1. fp16
2. fp16 + SDPA
3. fp16 + SDPA + Chunking
4. fp16 + SDPA + Chunking + Speculative Decoding
5. distil-whisper + fp16 + SDPA + Speculative Decoding + Chunking
6. and.. more

What do we measure?

Method	Time to Transcribe
fp16	
fp16 + SDPA	
fp16 + SDPA + Speculative Decoding	
fp16 + SDPA + Speculative Decoding + Chunking	
more…	

Note: Unless stated otherwise above methods do not incur a loss in performance.

fp16

1. Almost 2x as fast as fp32
2. Zero to negligible loss in performance [ref]
3. One line change `torch_dtype = torch.float16`

[ref]

Results

Method	Time to Transcribe
fp16	62s

*All results are from a Colab Free T4 GPU - transcribing a 10 min audio.

SDPA/ FA2

Scaled Dot Product Attention/ FA2 is a faster and more efficient implementation of the standard attention mechanism that can significantly speedup inference. [ref]

1. Parallelising the attention computation over sequence length
2. Partitioning the work between GPU threads to reduce communication and shared memory reads/writes between them

[ref]

Results

Method	Time to Transcribe
fp16	62s
fp16 + SDPA	60s

*All results are from a Colab Free T4 GPU - transcribing a 10 min audio.

Speculative Decoding

It is based on the idea that a smaller, faster model can often generate the same tokens as a larger, slower model.

The assistant model first generates a sequence of candidate tokens, and then the main model verifies these tokens through forward passes, ensuring the same outputs as if only the main model was used.

[ref]

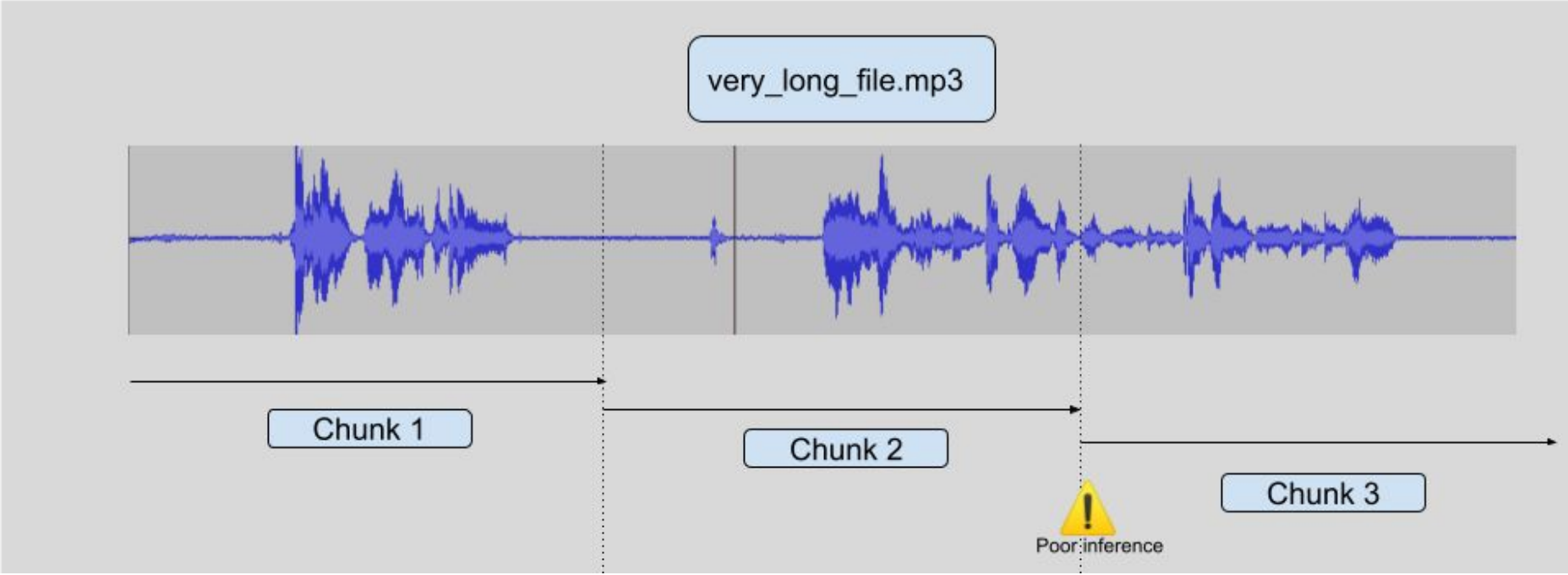
[ref]

Results

Method	Time to Transcribe
fp16	62s
fp16 + SDPA	60s
fp16 + SDPA + SD	37.9s

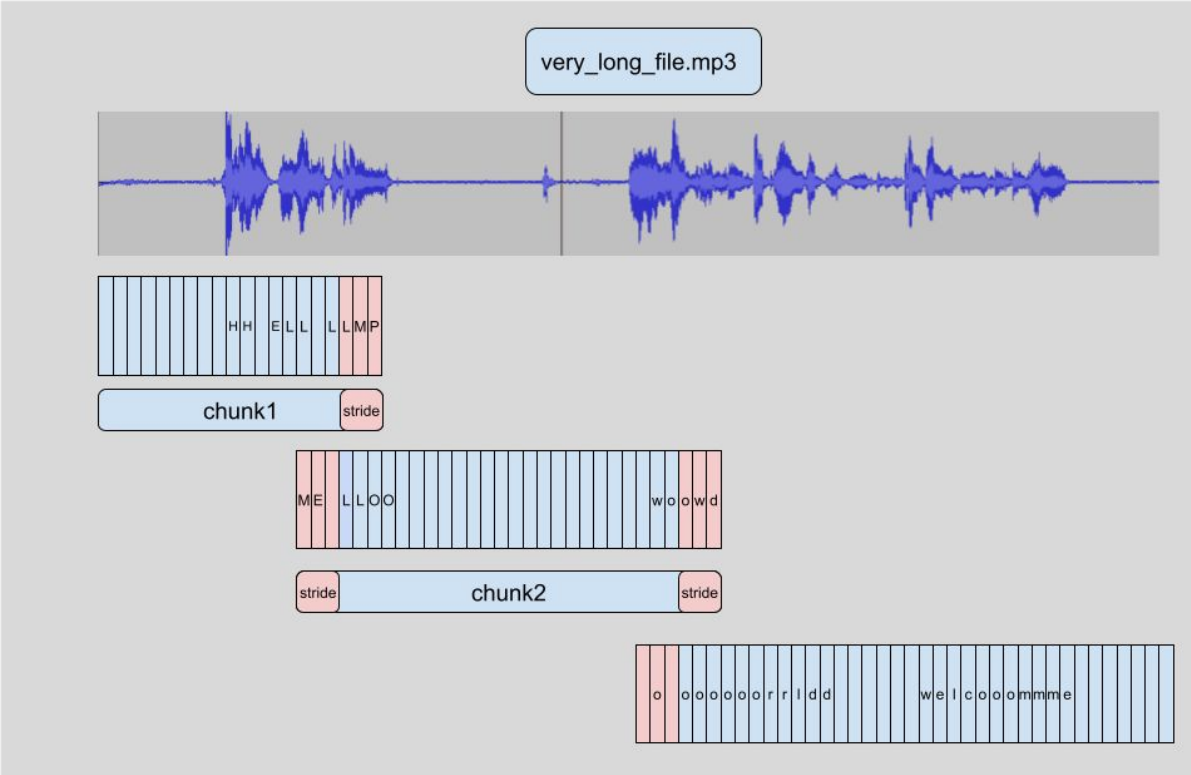
*All results are from a Colab Free T4 GPU - transcribing a 10 min audio.

Chunking



[ref]

Chunking



[ref]

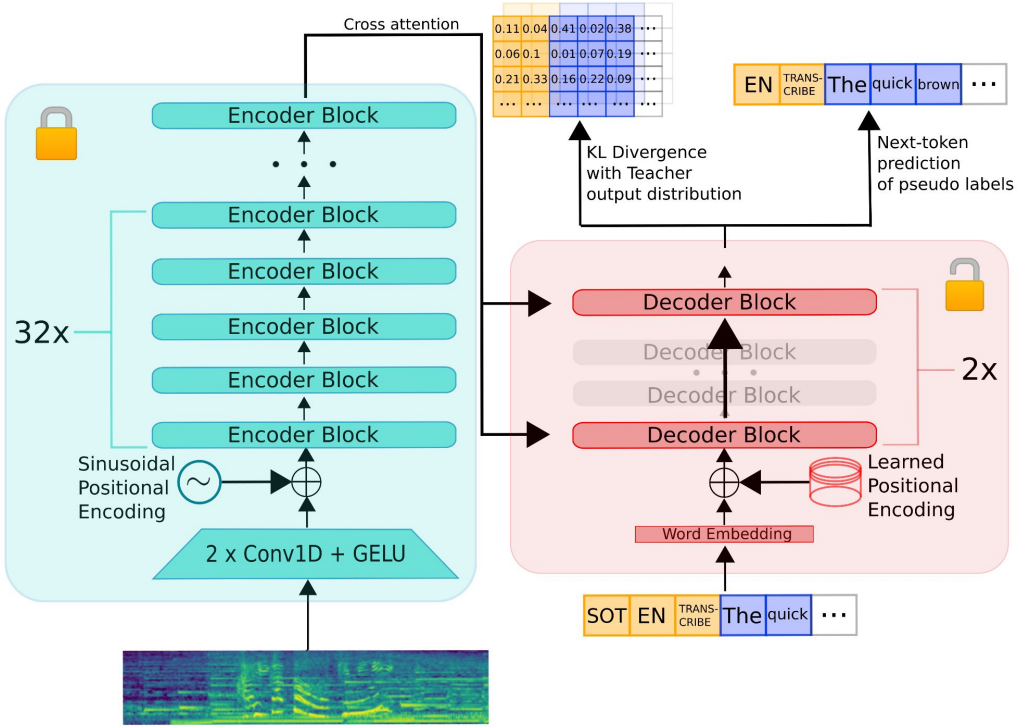
Results

Food for thought: why is it slower than just SD? :)

Method	Time to Transcribe
fp16	62s
fp16 + SDPA	60s
fp16 + SDPA + SD	37.9s
fp16 + SDPA + Chunking + SD	43.8s

*All results are from a Colab Free T4 GPU - transcribing a 10 min audio.

distil-whisper



1. Smaller Whisper
2. 6x faster, 2x lighter
3. English only
4. Within 0.5 WER

[ref]

[ref]

Results

Method	**Time to Transcribe**
fp16	62s
fp16 + SDPA	60s
fp16 + SDPA + SD	37.9s
fp16 + SDPA + Chunking + SD	43.8s
Distil-whisper + fp16 + SDPA + Chunking	**17.2s**

*All results are from a Colab Free T4 GPU - transcribing a 10 min audio.

Moar fastt ;)

1. Flash Attention 2 [ref]
2. Newer GPUs L40s/ A100s/ H100s
3. Quantisation - whisper.cpp/ faster-whisper
4. Short context Whisper [ref]

Takeaways

1. At the very least use SDPA & FA2 if your environment allows.
2. For bs=1 solutions use Speculative Decoding.
3. For rough translations use audio chunking.
4. For near-real time use cases - use short context.
5. Distil Whisper for use-case if possible.

Parting note

Don’t trust, verify.

Thank You!