

# 🤖 Agentic Demo — A1 (Cleaner) → A2 (Transcriber) → A3 (Cloner)

This demo runs your existing agents sequentially (no LangChain). It won't change your main app.

Upload audio (WAV) for cleaning



Drag and drop file here  
Limit 200MB per file • WAV

Browse files



sample-2.wav 432.6KB



YouTube URL or Video ID (both work)

<https://www.youtube.com/watch?v=N0-ChuOYWAE>



Run Agentic Pipeline (A1 → A2 → A3)

Saved uploaded file to temp\_agent\_audio.wav

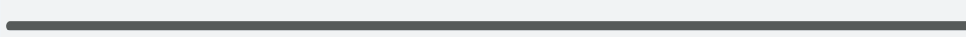
## Step 1 — Audio cleaning (Agent A1)

Audio cleaned!

Cleaned audio path: `cleaned_audio_agentic.wav`



0:00 / 0:10



## Step 2 — Transcription (Agent A2 simplified)

Transcript fetched!

Raw transcript (first 1500 chars)

you're present in gratitude you can't be  
anywhere else studies show that if  
you're having a thankful thought you  
can't have a worry filled thought at the  
same time if you're having a grateful  
thought you can't have an anxious  
thought at the same time gratitude and  
anxiety can't rent space in your mind at  
the same time but for gratitude to wo



## Step 3 — Voice cloning (Agent A3)

Voice cloning failed: CUDA error: device-side assert triggered CUDA kernel errors might be asynchronously reported at some other API call, so the stacktrace below might be incorrect. For debugging consider passing `CUDA_LAUNCH_BLOCKING=1` Compile with `TORCH_USE_CUDA_DSA` to enable device-side assertions.

Traceback (most recent call last):

```
File "C:\Users\ayaan\OneDrive\Desktop\tts-testing\chatterbox\testing\app_agentic.py", line 199, in
<module>
    cloned = agent_a3.clone_voice(str(CLEANED_PATH), transcript, str(CLONED_PATH))
File "C:\Users\ayaan\OneDrive\Desktop\tts-testing\chatterbox\agents\agent_a3_voice_cloner.py", line
11, in clone_voice
    wav = self.tts.generate(text=text, audio_prompt_path=cleaned_audio)
File "C:\Users\ayaan\OneDrive\Desktop\tts-testing\chatterbox\src\chatterbox\tts.py", line 246, in
generate
    speech_tokens = self.t3.inference(
File "C:\Users\ayaan\OneDrive\Desktop\tts-testing\env\lib\site-packages\torch\utils\_contextlib.py", line
116, in decorate_context
    return func(*args, **kwargs)
File "C:\Users\ayaan\OneDrive\Desktop\tts-testing\chatterbox\src\chatterbox\models\t3\t3.py", line 301,
in inference
    bos_token = torch.tensor([[self.hp.start_speech_token]], dtype=torch.long, device=device)
RuntimeError: CUDA error: device-side assert triggered
CUDA kernel errors might be asynchronously reported at some other API call, so the stacktrace below
might be incorrect.
```

For debugging consider passing `CUDA_LAUNCH_BLOCKING=1`

Compile with ``TORCH_USE_CUDA_DSA`` to enable device-side assertions.