GenAI-ML HW10

Speech Generation

TAs: 陳思齊 董家愷 林芷妤

ntu-gen-ai-ml-2025-fall-ta@googlegroups.com

Deadline: 2026/**01/09** 23:59:59 (UTC+8)

Outline

- Links
- Task description
- Grading
- Course grading release date
- Appendix (Task questions with its options)

Previous Course

- 【生成式AI時代下的機器學習(2025)】第十二講：語言模型如何學會說話 — 概述語音語言模型發展歷程
    - SUPERB: 語音上的自督導式學習模型居然十項全能？
    - Meta 語音對語音翻譯技術背後的黑科技

Links

Model-Related Materials：

- **Mimi** Model: https://huggingface.co/kyutai/mimi
- **CSM** Model: https://huggingface.co/sesame/csm-1b

- Reference paper (**Mimi** codec): https://kyutai.org/Moshi.pdf
- technical blog post (**CSM** model): https://www.sesame.com/research/crossing_the_uncanny_valley_of_voice

Links

Homework-Related Materials：

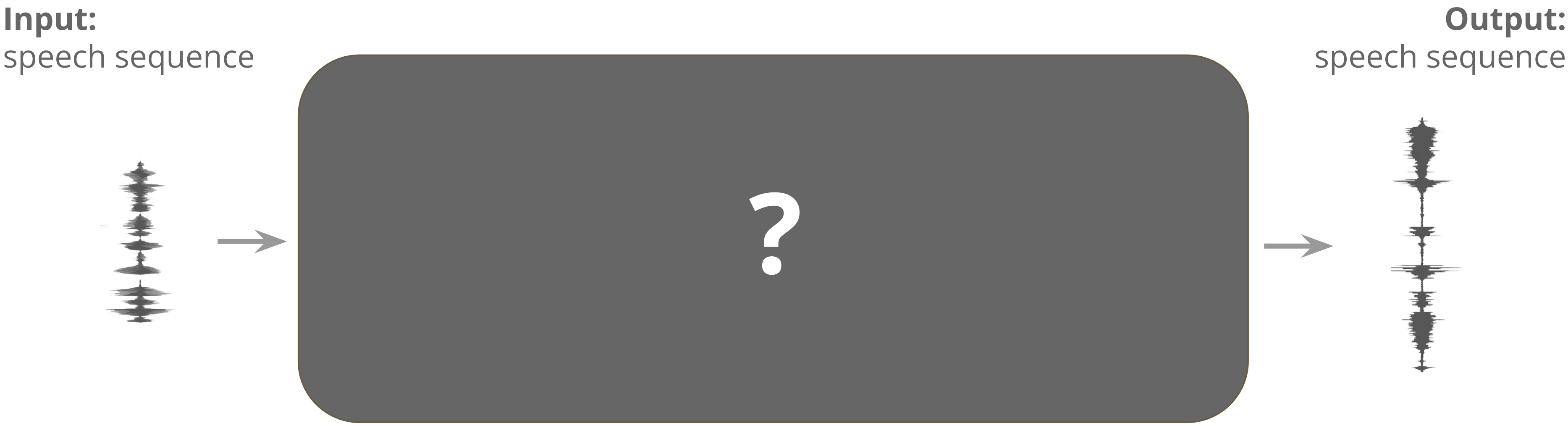
- NTU Cool: https://cool.ntu.edu.tw/courses/50706/quizzes/65482
- HW 10 Colab: https://colab.research.google.com/drive/1eSVN9UeZ-ITkA74EKzMtIu9lz3HhRgBf?usp=sharing

Goals of This Homework

- Learn how speech can be **represented as tokens** through the **Mimi model**.
- Learn how to **generate new speech content** using speech tokens through the **CSM model**.
- For this assignment, you only need to **answer 12 multiple-choice** questions on **NTU COOL**.

Task description

Goal: Speech Generation



Input:
speech sequence

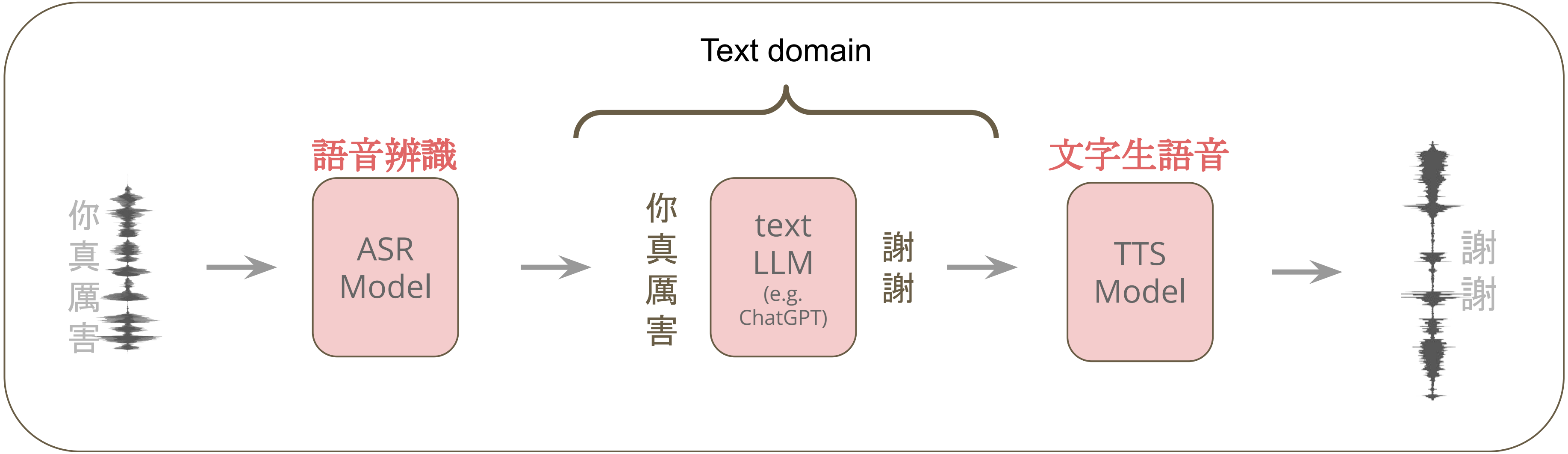
?

Output:
speech sequence

Task description

Speech Generation

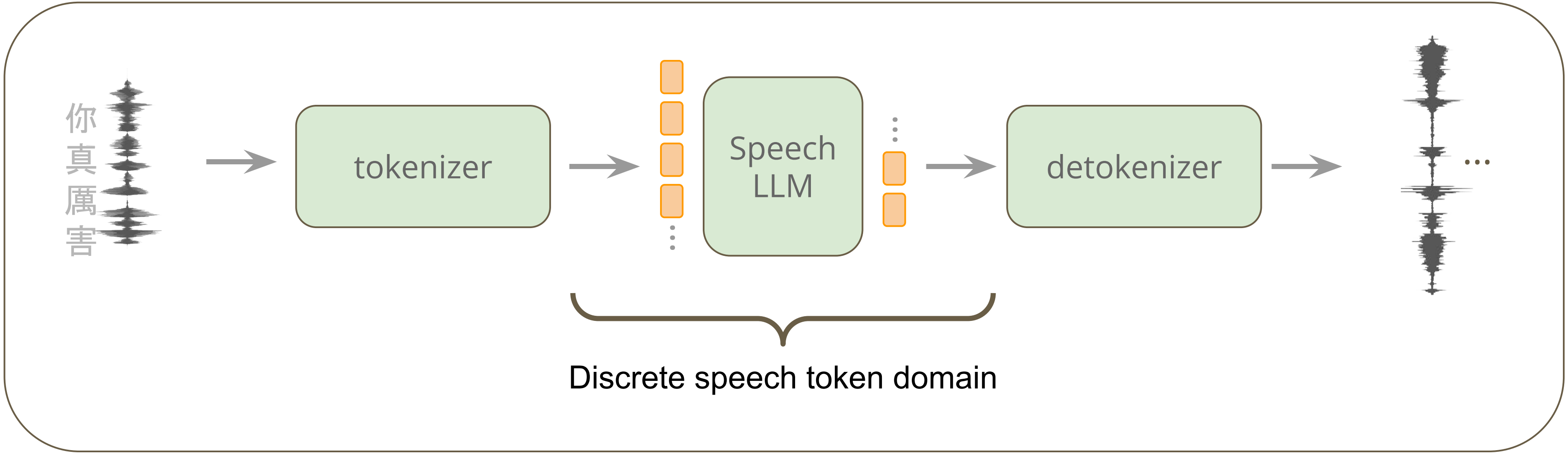
Naive Speech Model



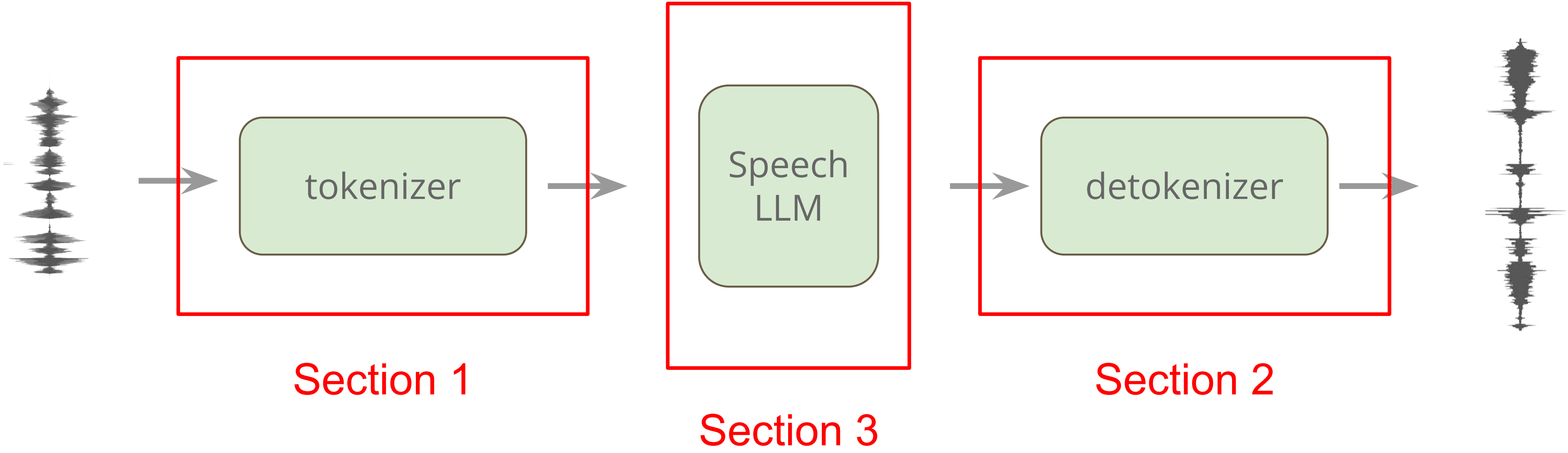
Task description

Speech Generation

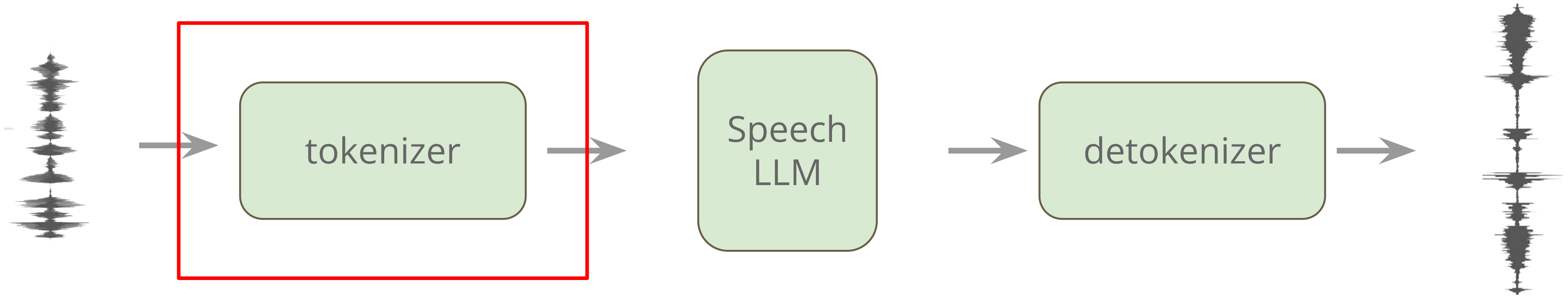
Speech token Model



Task description



Section 1: Tokenizer



Section 1

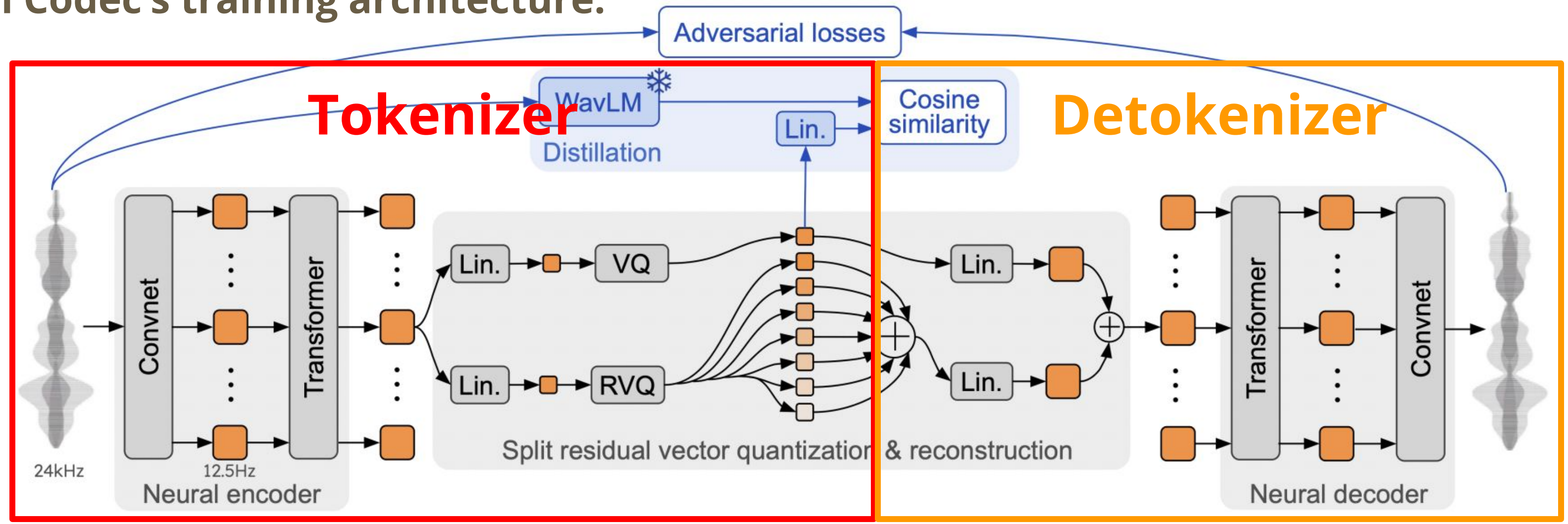
Section 1: Tokenizer (Mimi model)

Mimi Codec:

The tokenizer and detokenizer are learned jointly.

kyutai / mimi

Mimi Codec's training architecture:

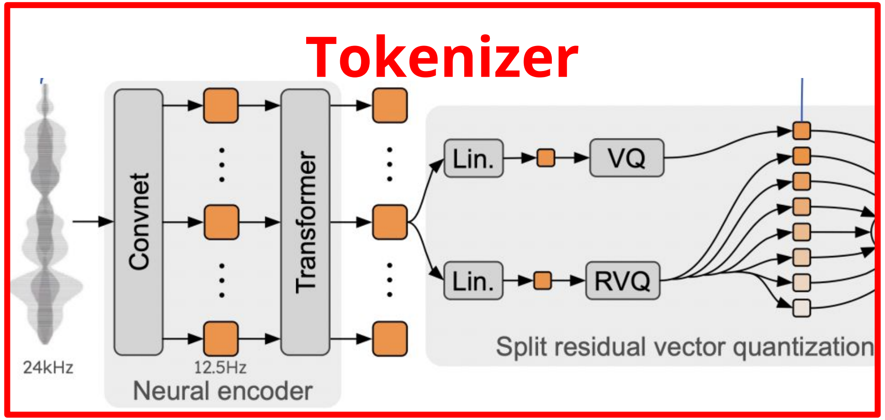


Section 1: Tokenizer

Mimi Codec:

In this assignment, **we're not training a model**. We'll simply use an already trained codec to observe how it behaves under different conditions.

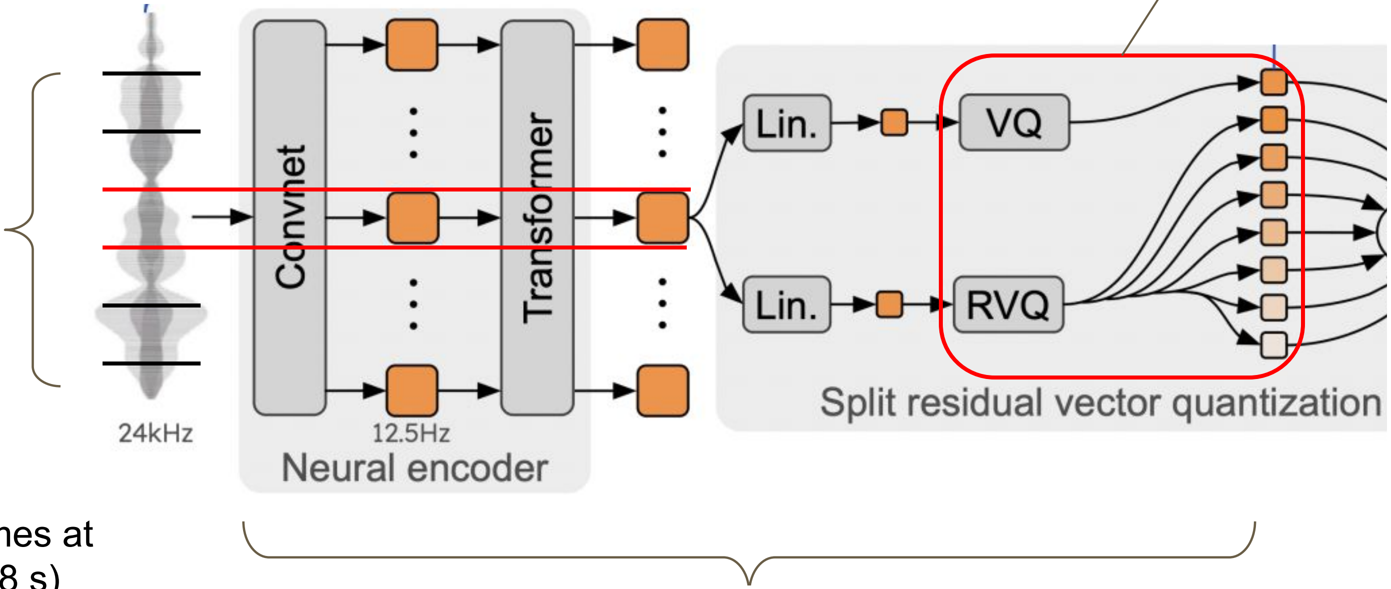
In this section (Tokenizer), we focus on
"**How** it behaves? ", and
"**What** a token looks like in this model? "



Section 1: Tokenizer

How does Mimi's model extract tokens from audio?

Then, each frame is quantized individually to obtain the final tokens.



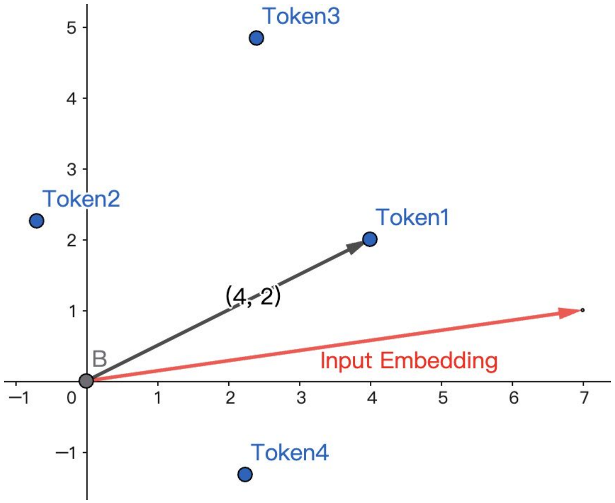
Split into frames at 12.5 Hz. (0.08 s)

Tokenizer

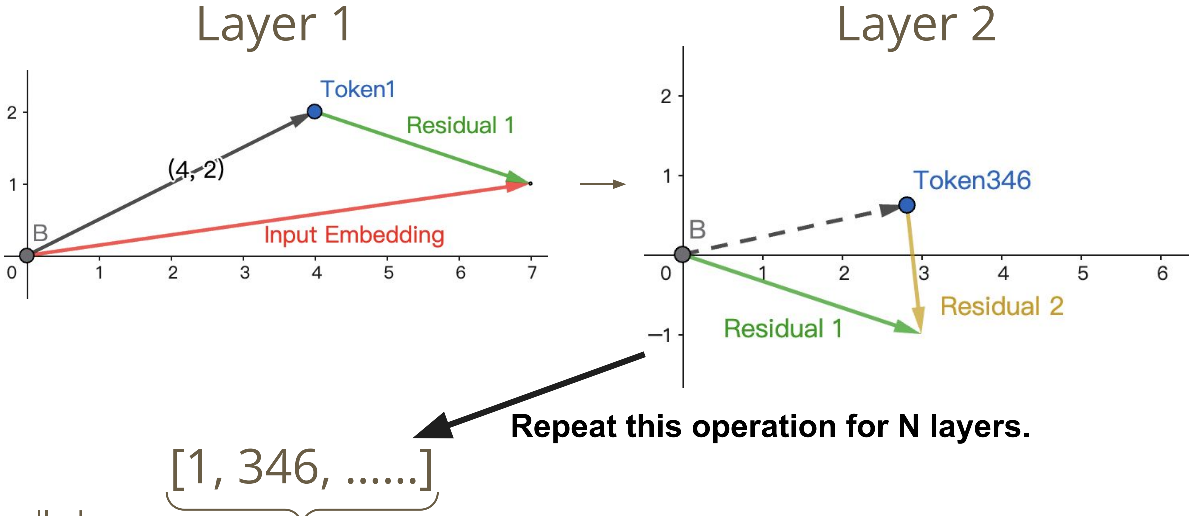
How does Mimi's model extract tokens from audio? (Conti.)

RVQ:
https://arxiv.org/abs/2210.13438
https://www.assemblyai.com/blog/what-is-residual-vector-quantization

Vector quantization(VQ):



Residual vector quantization(RVQ):



Repeat this operation for N layers.

[1, 346, ...]

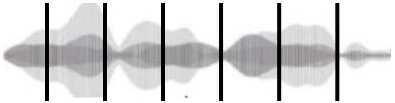
Note. Sometimes we encounter what are called **finer tokens**. Here, finer refers to tokens produced by the **deeper layers** of the RVQ.

In Mimi, the **number of layers** it passes through can be adjusted — by default, it goes up to 32 layers.

Section 1: Tokenizer

Comparing Mimi Tokens and Text Tokens

Mimi's Tokens



Segmented into **frames** by **equal time intervals**

$[\ \mathrm{emb}_1,\ \mathrm{emb}_2,\ \ldots\ldots\mathrm{emb}_{\mathrm{frames}}\ ]$

Residual vector quantization

$$\begin{pmatrix} 2002, & 1057, & \ldots\ldots & 392 \\ \vdots & & \ldots\ldots & \vdots \\ & & \ldots\ldots & \\ 914, & 2023, & \ldots\ldots & 1131 \end{pmatrix}$$

layers

Frames

Text Tokens

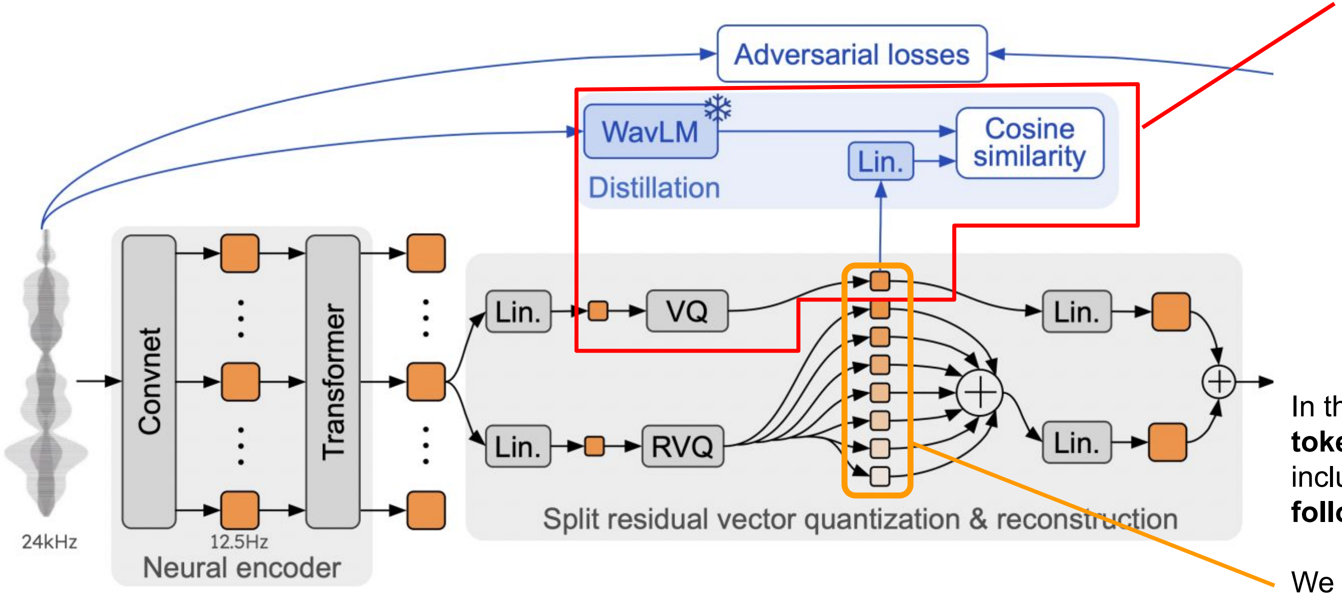
大|家|好

[27384, 46729, 53901]

Section 1: Tokenizer

In Mimi's model, one of the token layers is distilled **semantic token** from a self-supervised model (WavLM).

Previous course



In this homework, we put **all the tokens together** for discussion — including the one that was **trained to follow WavLM**.

We call that the **zero-th layer**, and the rest — from **layer 1 and onwards** are the **RVQ layers**.

Section 1: Tokenizer - Exercises

Task Descriptions:

Observations of Mimi’s token on different audios and understanding **how the model interprets** them.

Subtasks:

(1) Given the Mimi’s code of two audio files, what are their shapes? Compare their **last dimension**. **(Multiple Choice – Single Answer)**

(2) Given a **8-second audio** file, what would its **last dimension** be? **(Multiple Choice – Single Answer)**

(3) In Mimi’s codes, what does the **second dimension** (=32) represent? **(Multiple Choice – Single Answer)**

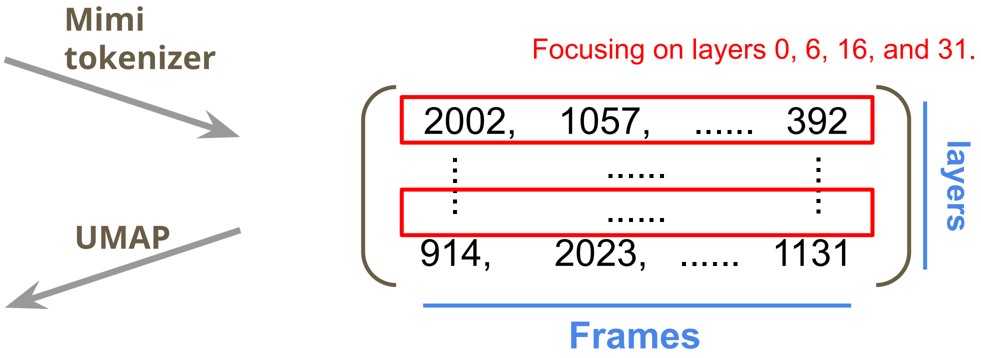
(4) Examine any single value in the code. What is its **data type**, and why? **(Multiple Choice – Single Answer)**

(5) In Mimi's code, for the **zero-th layer (codebook_layer = 0)**, what are the first **five** values of the embedding corresponding to the code ID is **2047** (0-indexed) ? **(Multiple Choice – Single Answer)**

Section 1: Tokenizer - Exercises

(6) Following the procedure on Colab, plot the **UMAP** classification maps of **32 layers** for different **emotion audio** at **layers 0, 6, 16, and 31.** Examine the results and answer the question.
(Multiple Choice – Multiple Answers)

1. Emotion dataset
Use the **EmoV-DB dataset** with predefined emotion categories.
(Amused, Angry, Disgusted, Sleepy, Neutral)
https://www.openslr.org/115/



Mimi tokenizer

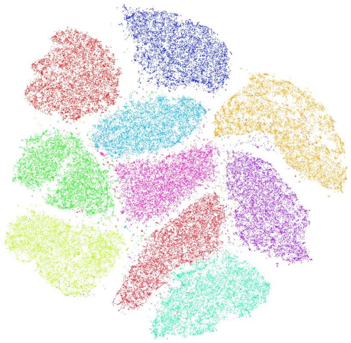
UMAP

Focusing on layers 0, 6, 16, and 31.

2002, 1057, ... 392
......
......
914, 2023, ... 1131

layers

Frames

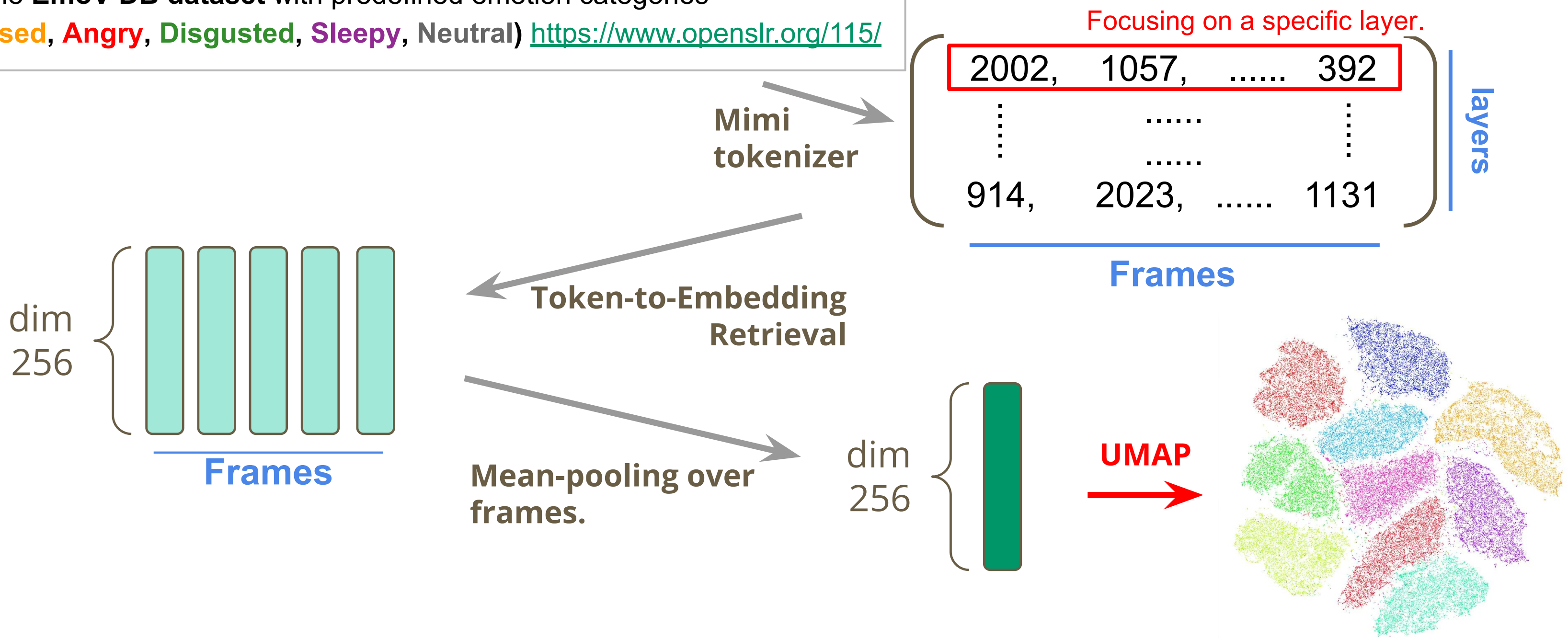


Q6 Data Preparation-

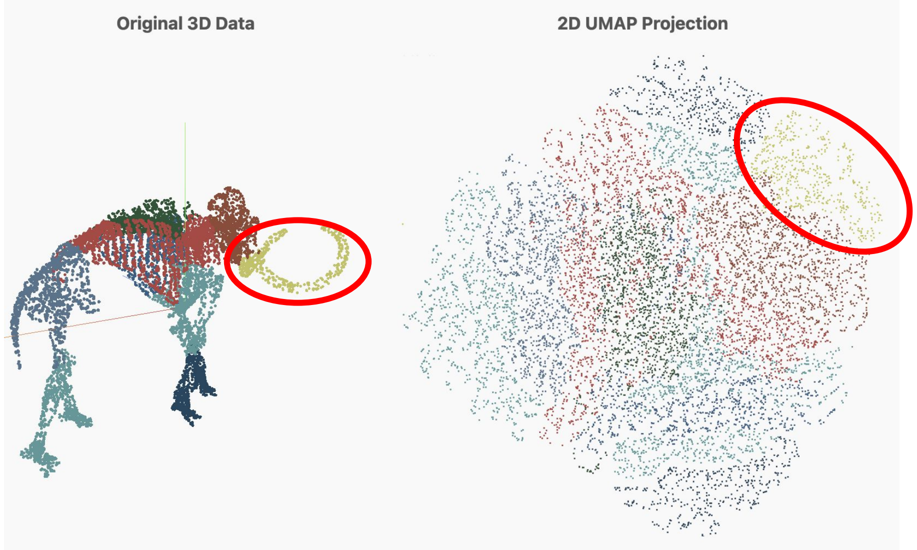
Conceptual Flow: One Audio Sample, Specific Layer

1. Emotion dataset

Use the **EmoV-DB dataset** with predefined emotion categories (**Amused**, **Angry**, **Disgusted**, **Sleepy**, **Neutral**) https://www.openslr.org/115/



What is UMAP?



UMAP is a **nonlinear dimensionality reduction** tool that is faster than t-SNE and better preserves global structure.
(Ref: Understanding UMAP)

The core goal of UMAP is to **project** complex, high-dimensional data into a **lower-dimensional space** while preserving the essential **neighborhood structure**.

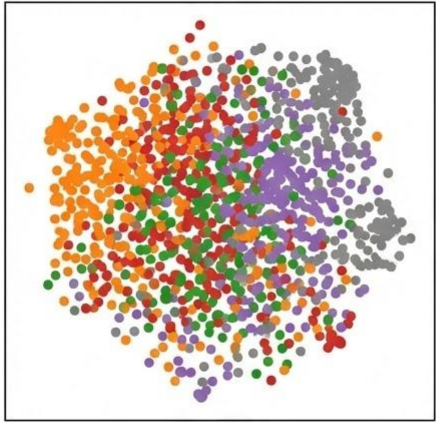
Q6 Interpretation

- Each 2D **UMAP figure** visualizes all data points from **a particular layer**.

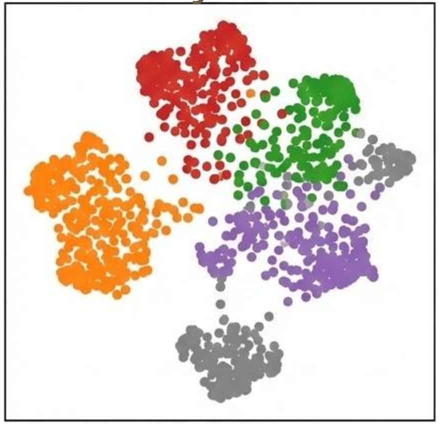
- Every **point** within the plot is **traceable** to a **specific audio file**.

(**Amused**, **Angry**, **Disgusted**, **Sleepy**, **Neutral**)

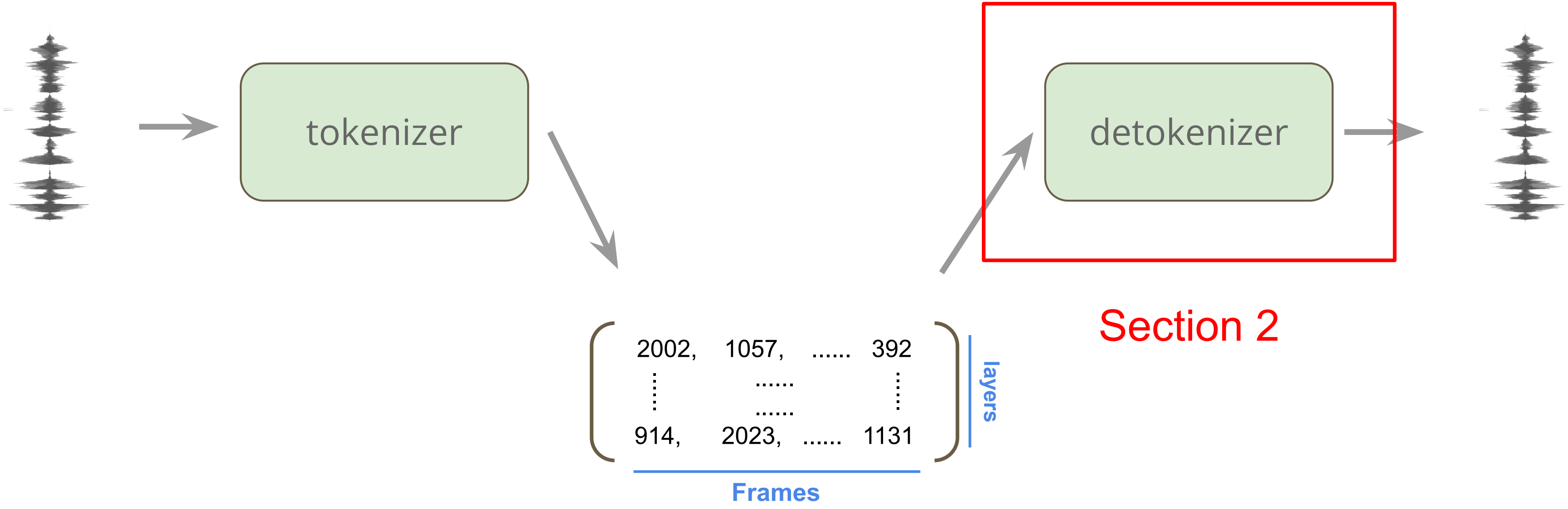
Layer *X*



Layer *Y*



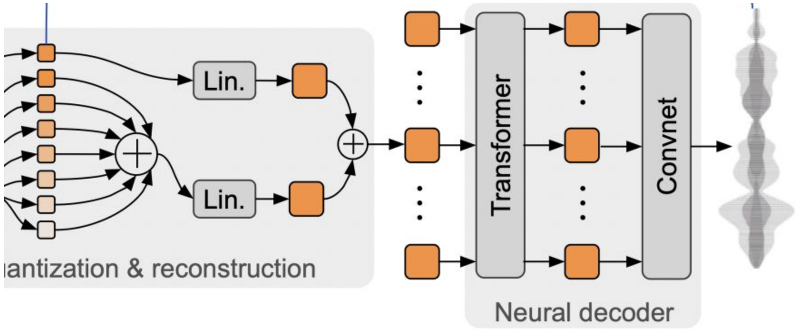
Section 2: Detokenizer



$$\begin{pmatrix} 2002, & 1057, & \ldots\ldots & 392 \\ \vdots & & \ldots\ldots & \vdots \\ & & \ldots\ldots & \\ 914, & 2023, & \ldots\ldots & 1131 \end{pmatrix}$$

Section 2: Detokenizer

$$\underbrace{\begin{pmatrix} 2002, & 1057, & \dots\dots & 392 \\ \vdots & \dots\dots & & \vdots \\ & \dots\dots & & \\ 914, & 2023, & \dots\dots & 1131 \end{pmatrix}}_{\text{Frames}} \Big| \, \text{layers} \longrightarrow$$



Reconstructing audio from **discrete tokens**

This step demonstrates the **benefit** of using the **token domain** rather **than the text domain**.

Since the **goal is to reconstruct the original audio**, good tokens should retain all essential information—emotion, prosody, speaker identity—so these features can be further processed.”

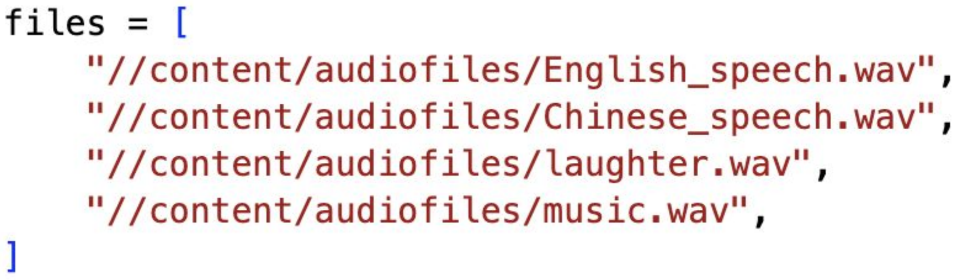
Section 2: Detokenizer - Exercises

Task Descriptions:

Decode(Reconstruct) Mimi’s code back to audios.

Subtasks:

(7-8) Listen to the original and decoded versions of the following audio types: **(a) English speech, (b) Chinese speech, (c) laughter, and (d) music.** Evaluate **how well** each type is reconstructed, then answer the provided question**s.(Multiple Choice – Single Answer)**

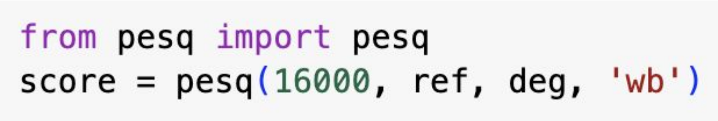


```
files = [
    "//content/audiofiles/English_speech.wav",
    "//content/audiofiles/Chinese_speech.wav",
    "//content/audiofiles/laughter.wav",
    "//content/audiofiles/music.wav",
]
```

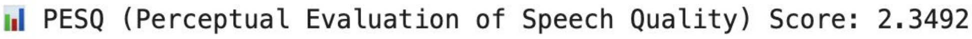
In this assignment, speech audio was generated using OpenAI.fm and ElevenLabs.

Section 2: Detokenizer - Exercises

(7-8) Listen to the original and decoded versions of the following audio types: **(a) English speech, (b) Chinese speech, (c) laughter, and (d) music.** Evaluate **how well** each type is reconstructed, then answer the provided question**s.(Multiple Choice – Single Answer)**



```python
from pesq import pesq
score = pesq(16000, ref, deg, 'wb')
```



```
📊 PESQ (Perceptual Evaluation of Speech Quality) Score: 2.3492
```

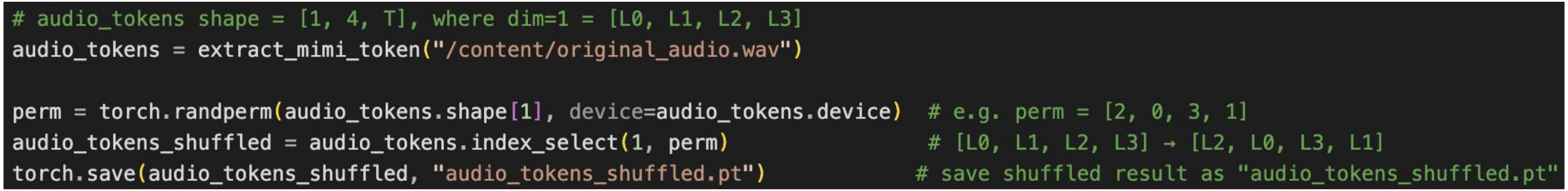
In this task, we additionally employ **PESQ** as an objective evaluation metric for each decoded audio against its original in (a)–(d).

Note. **PESQ** (Perceptual Evaluation of Speech Quality):
an objective metric comparing a degraded speech signal to its reference; outputs MOS-LQO ≈ −0.5–4.5 (higher = better); widely used in telecom/VoIP and speech codec evaluation.
(Ref: PESQ)

Section 2: Codec to Speech - Exercises Conti.

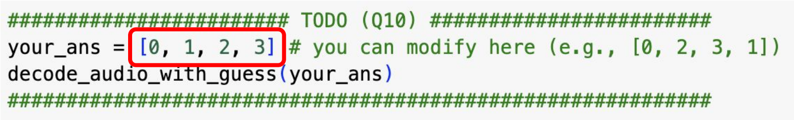
(9) In the code, there is an audio file encoded into **4 layers (**`audio_codes_shuffled.pt`) that has **been shuffled layer-wise**. Try to reorder it and recover the original spoken sentence. **(Multiple Choice – Single Answer)**

“Below is the process we used to construct a shuffled 4 layers file (audio_tokens_shuffled.pt).”



```
# audio_tokens shape = [1, 4, T], where dim=1 = [L0, L1, L2, L3]
audio_tokens = extract_mimi_token("/content/original_audio.wav")

perm = torch.randperm(audio_tokens.shape[1], device=audio_tokens.device)  # e.g. perm = [2, 0, 3, 1]
audio_tokens_shuffled = audio_tokens.index_select(1, perm)                 # [L0, L1, L2, L3] → [L2, L0, L3, L1]
torch.save(audio_tokens_shuffled, "audio_tokens_shuffled.pt")            # save shuffled result as "audio_tokens_shuffled.pt"
```



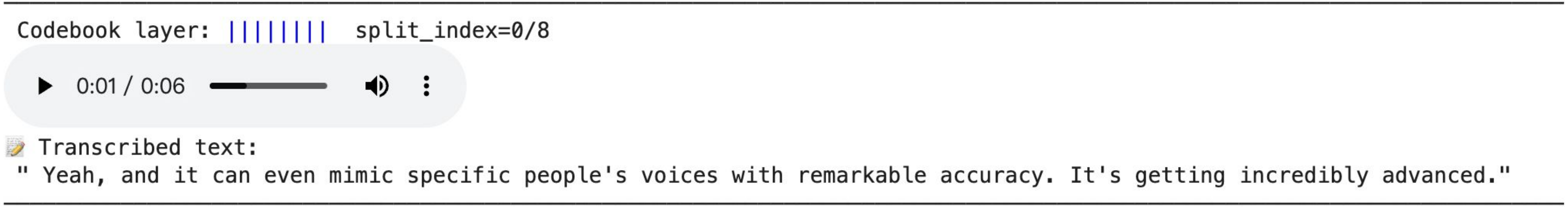
```
######################## TODO (Q10) ########################
your_ans = [0, 1, 2, 3] # you can modify here (e.g., [0, 2, 3, 1])
decode_audio_with_guess(your_ans)
############################################################
```

“Try to find the correct order (without repeating any sequence), among all 4! = 24 possible layer permutations.”

Section 2: Codec to Speech - Exercises Conti.

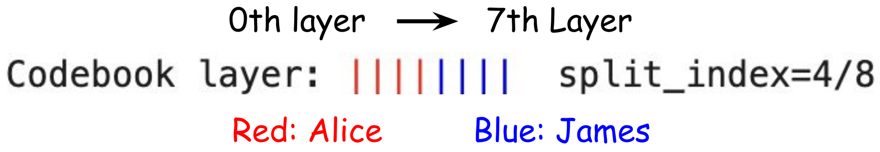
(10) Two equal-length(frames) audio files (Alice & James) are encoded into **8 layers**, then interleaved and decoded. Listen and select the correct statements.**(Multiple Choice – Multiple Answers)**

Example Result:



```
Codebook layer: ||||||||  split_index=0/8
```

```
📝 Transcribed text:
 " Yeah, and it can even mimic specific people's voices with remarkable accuracy. It's getting incredibly advanced."
```



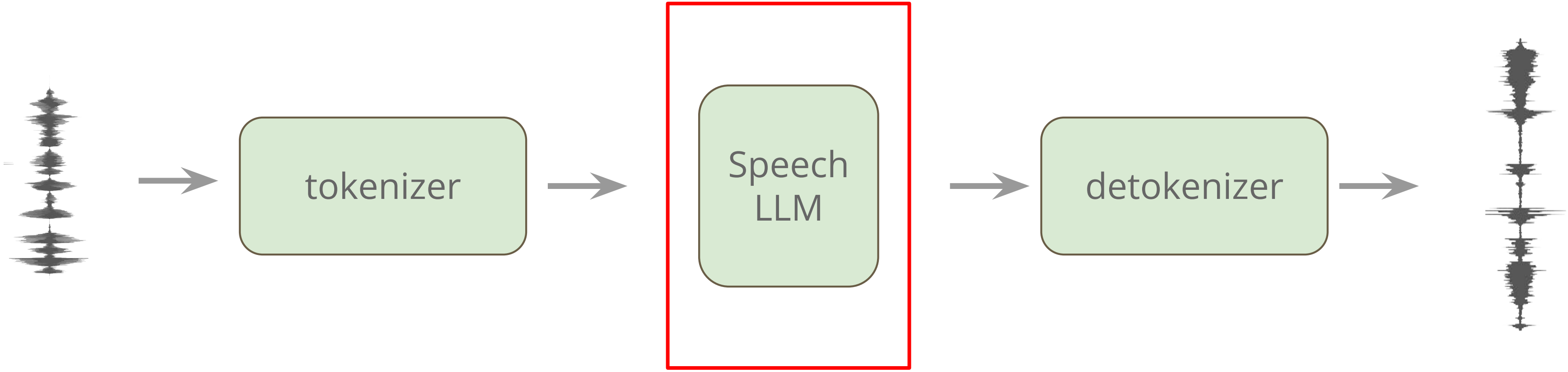
0th layer → 7th Layer

Codebook layer: |||||||| split_index=4/8

Red: Alice Blue: James

Note. In this task, each sample is **transcribed by Whisper (ASR model)** and shown alongside the audio as an additional **objective cue**.

Section 3: Backbone LLM

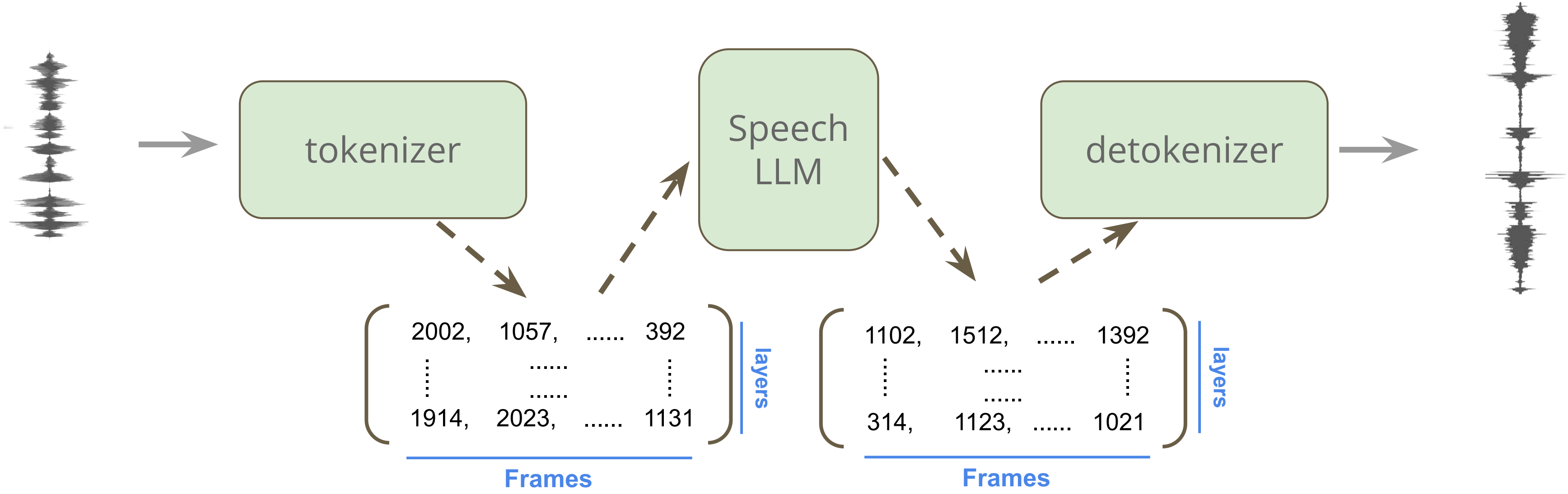


Section 3

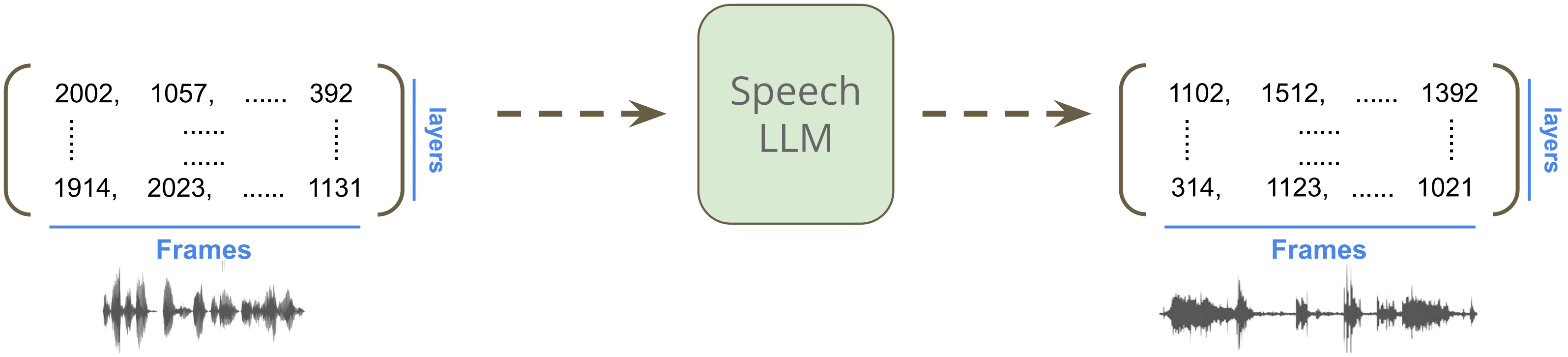
Section 3: Backbone LLM

After the previous two sections, we found that we were only trying to **reconstruct** the same audio file.

But... how can we make an Audio LLM generate an entirely **new** audio?



Section 3: Backbone LLM



2002, 1057, ... 392
⋮ .... ⋮
1914, 2023, ... 1131

1102, 1512, ... 1392
⋮ .... ⋮
314, 1123, ... 1021

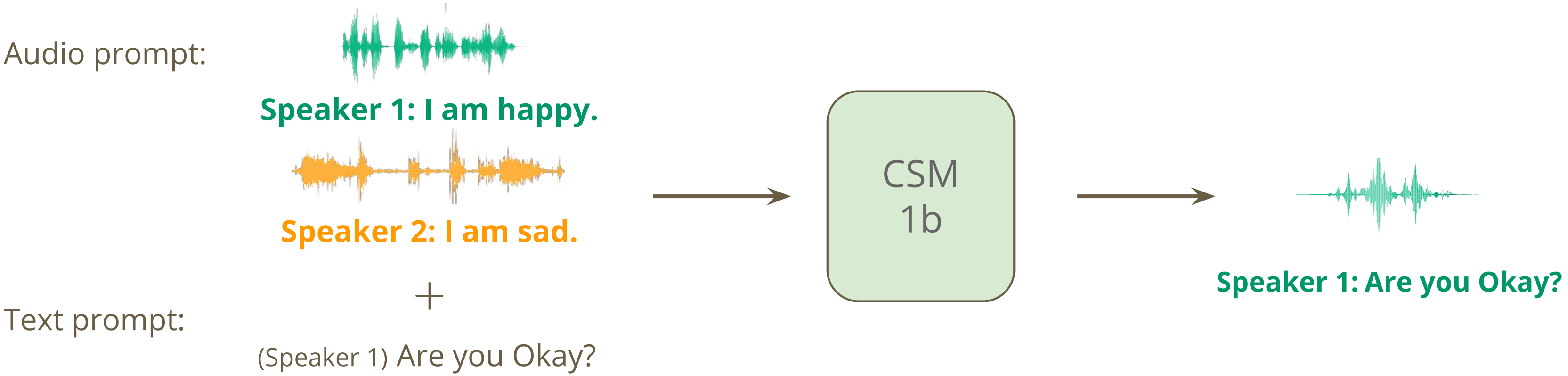
你好嗎？	Translation Task	How are you？
(Speaker 1) 你好嗎？	Voice conversion	(Speaker 2) 你好嗎？
你好嗎？	Audio dialogue	我很好。

This allows us to model speech-to-speech tasks in the **audio token domain**, just like how text-based LLMs predict tokens to generate language.

Section 3: Backbone LLM

In this assignment, we use CSM-1b as our backbone model.
https://www.sesame.com/research/crossing_the_uncanny_valley_of_voice

CSM-1b: is a speech generation model that combines **text** and **audio inputs** to achieve **speaker voice cloning**.
It **uses Mimi's tokens** as its audio token representation.



Audio prompt:

Speaker 1: I am happy.

Speaker 2: I am sad.

+

Text prompt:

(Speaker 1) Are you Okay?

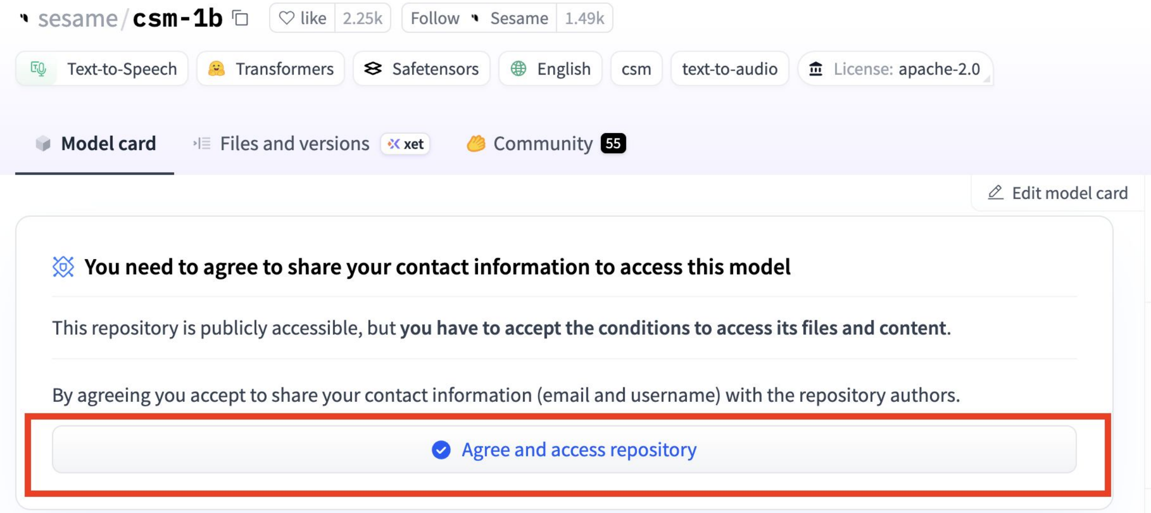
CSM 1b

Speaker 1: Are you Okay?

Section 3: Backbone LLM

- Obtaining Access to the CSM-1B Model on Hugging Face

- Agree to access the CSM-1b model on Hugging Face https://huggingface.co/sesame/csm-1b



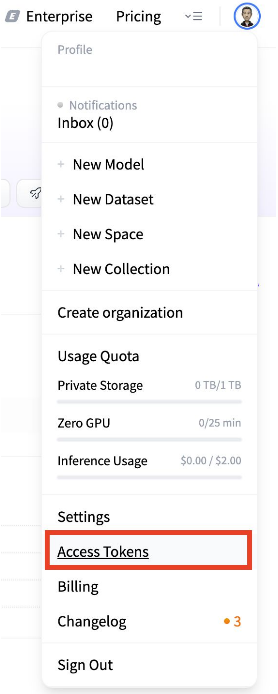
You need to agree to share your contact information to access this model

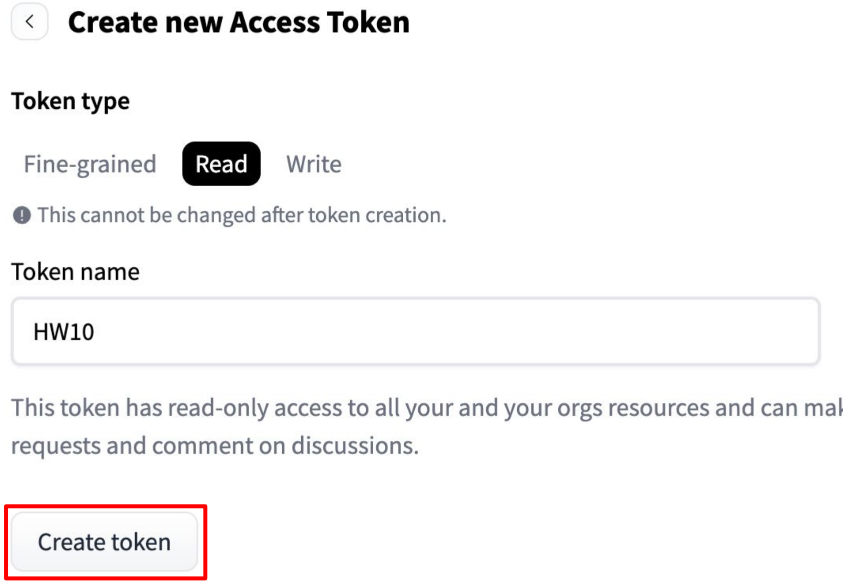
This repository is publicly accessible, but **you have to accept the conditions to access its files and content.**

By agreeing you accept to share your contact information (email and username) with the repository authors.

Obtaining Access to the CSM-1B Model on Hugging Face

- Obtain your personal access token from Hugging Face





Create new Access Token

Token type

Fine-grained [✓ Read] Write

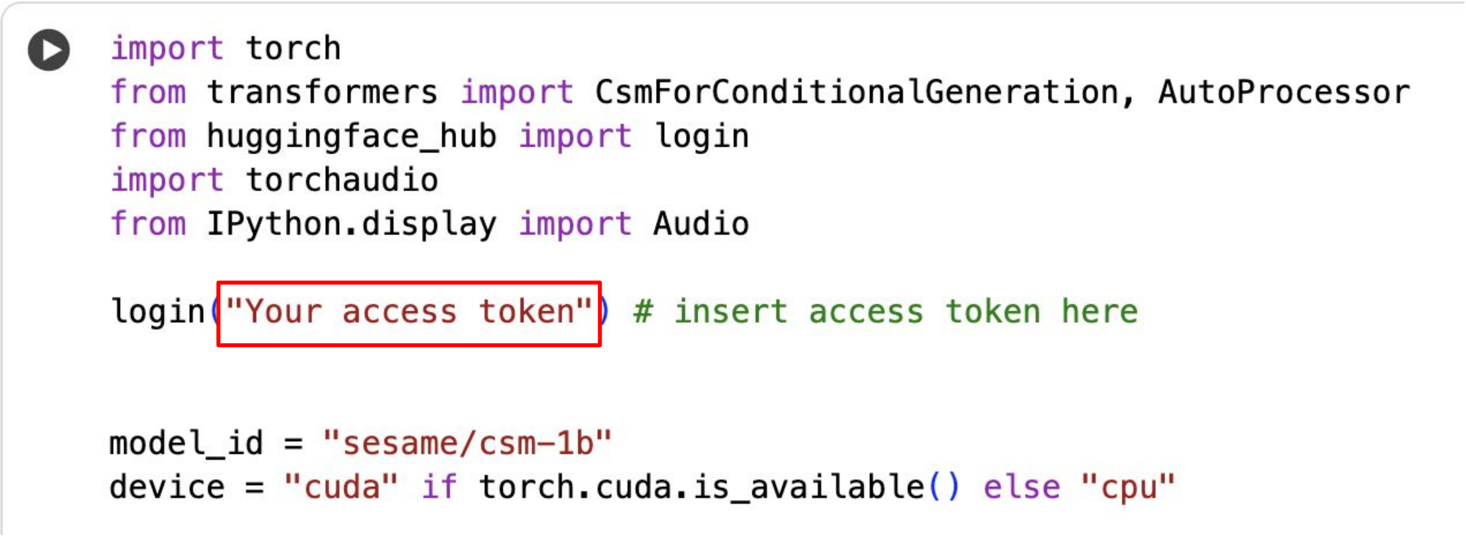
This cannot be changed after token creation.

Token name: HW10

This token has read-only access to all your and your orgs resources and can mal
requests and comment on discussions.

Obtaining Access to the CSM-1B Model on Hugging Face

- Input this token into the Colab environment



```
import torch
from transformers import CsmForConditionalGeneration, AutoProcessor
from huggingface_hub import login
import torchaudio
from IPython.display import Audio

login("Your access token") # insert access token here


model_id = "sesame/csm-1b"
device = "cuda" if torch.cuda.is_available() else "cpu"
```

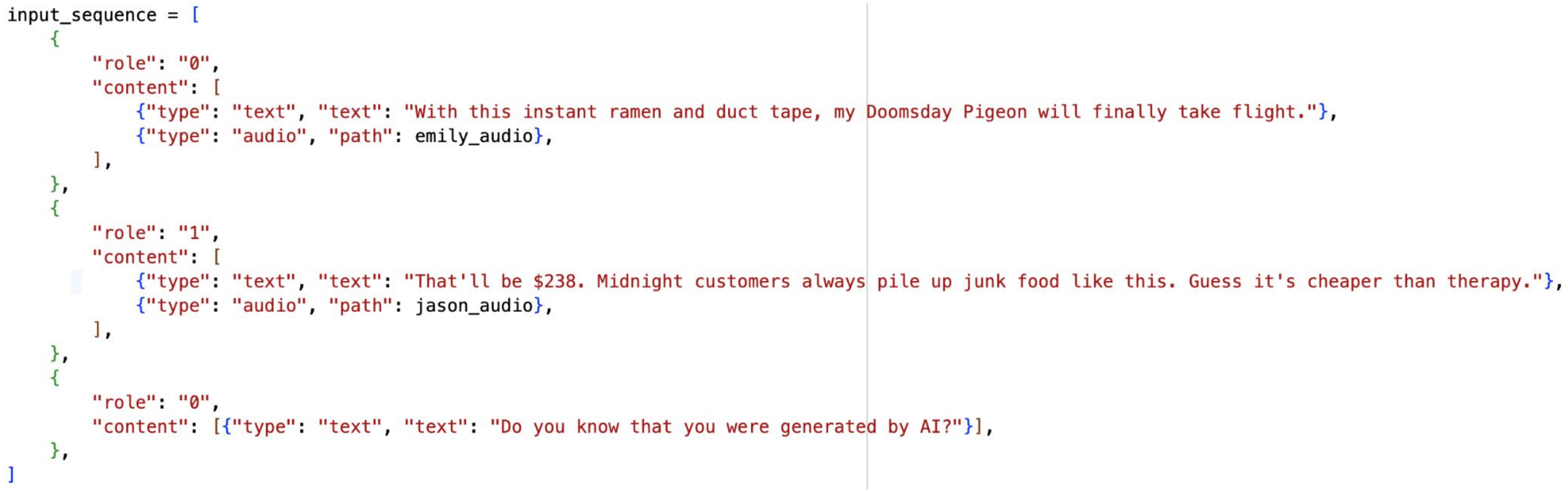
Section 3: Backbone LLM - Exercises

Task Descriptions:

Understanding how the model generate **entirely new audio** using Mimi’s code.

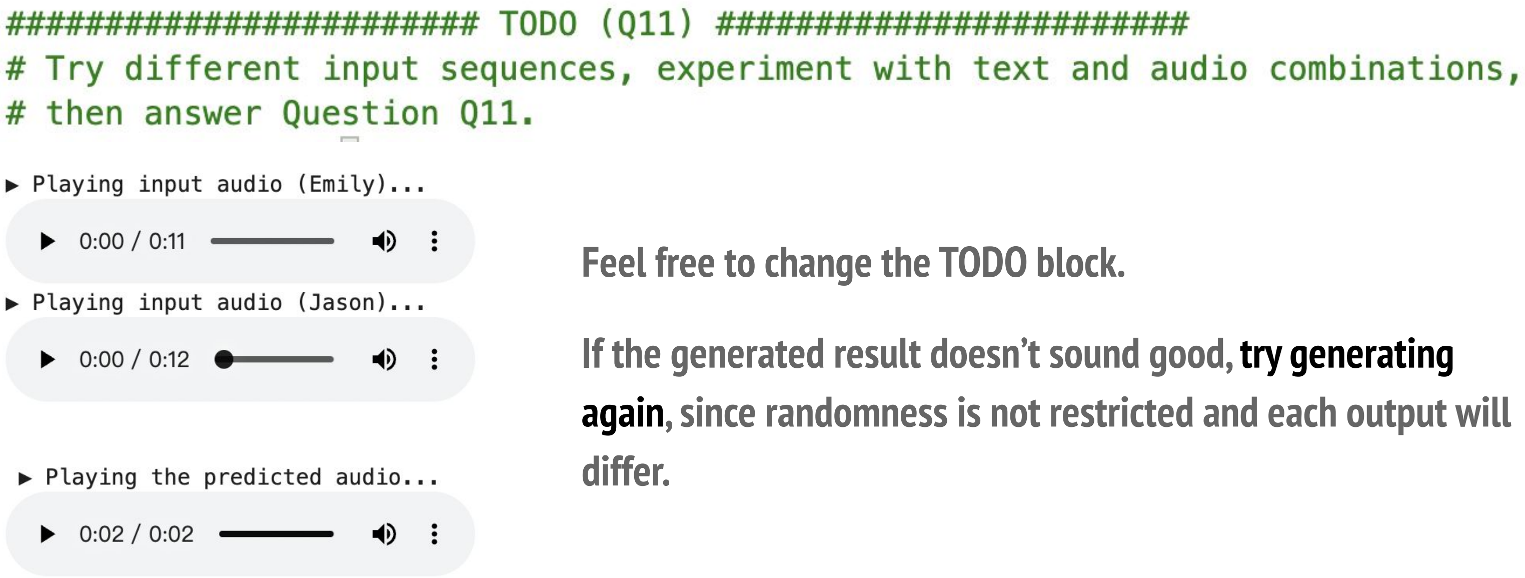
Subtasks:

(11) In Colab, try modifying the input_sequence. Given the input sequence shown in the image, what kind of audio output will be generated? **(Multiple Choice – Single Answer)**



```
input_sequence = [
    {
        "role": "0",
        "content": [
            {"type": "text", "text": "With this instant ramen and duct tape, my Doomsday Pigeon will finally take flight."},
            {"type": "audio", "path": emily_audio},
        ],
    },
    {
        "role": "1",
        "content": [
            {"type": "text", "text": "That'll be $238. Midnight customers always pile up junk food like this. Guess it's cheaper than therapy."},
            {"type": "audio", "path": jason_audio},
        ],
    },
    {
        "role": "0",
        "content": [{"type": "text", "text": "Do you know that you were generated by AI?"}],
    },
]
```

Section 3: Backbone LLM - Exercises



```
######################## TODO (Q11) ########################
# Try different input sequences, experiment with text and audio combinations,
# then answer Question Q11.
```

```
▶ Playing input audio (Emily)...
```

```
▶ Playing input audio (Jason)...
```

```
▶ Playing the predicted audio...
```

Feel free to change the TODO block.

If the generated result doesn't sound good, **try generating again**, since randomness is not restricted and each output will differ.

(12) Observe the Colab code to understand how the CSM model works, and choose the correct statement.

(Multiple Choice – Single Answer)

Submission & Deadline

- Submit your homework to **NTU Cool**
- 2026/**01/09** 23:59:59 (UTC+8)
- No late submission is allowed

Regulations

- You should NOT plagiarize
- Please protect your own work and ensure that your answers are not accessible to others. If your work is found to have been copied by others, you will be subject to the same penalties.
- Your final grade x 0.9 and get a score 0 for that homework if you violate any of the above rules first time (within a semester)
- Your will get F for the final grade if you violate any of the above rules multiple times (within a semester)
- Prof. Lee & TAs preserve the rights to change the rules & grades

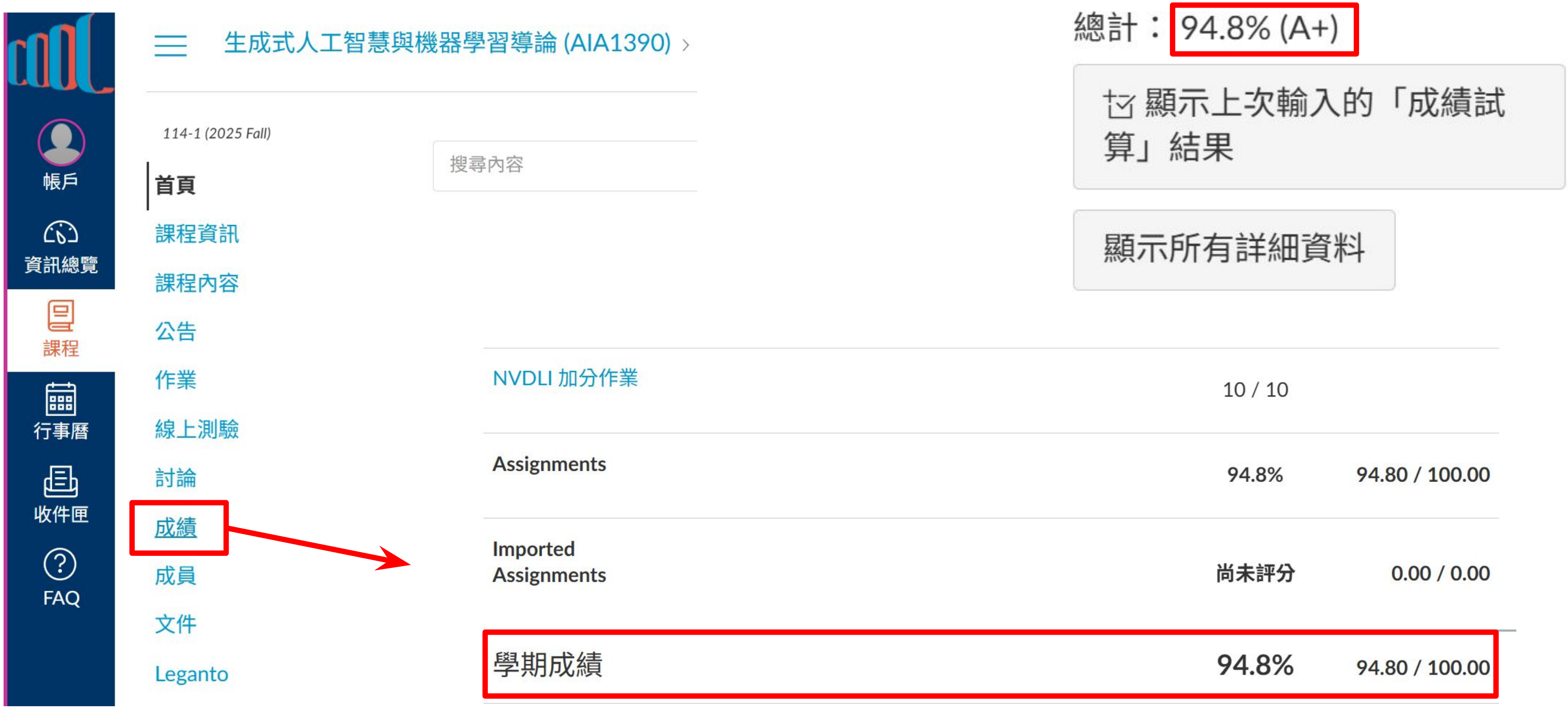
Grading Release Date

- This assignment consists of **12 questions**:
  10 single-choice and **2 multiple-choice**.
- Each **single-choice** question is worth **0.8 points** and each **multiple-choice** question is worth **1 point**, for a total of **10 points**.
- 多選題的計分方式為下:「每個選項占分為 **總分**(1)/**正確選項的個數**，答對分數減去答錯分數即為本題得分，若沒有選到正確或錯誤的，則不加分也不扣分，扣到0為止。」
- The grading of the homework will be released by 2026/**01/09** 23:59:59 (UTC+8)

[Important] - 學期總成績公布時間

- **本門課程學期總成績** 將於本次作業截止時刻當下 2026/**01/09** 23:59:59 公布。
- 公佈後有**兩天時間** 01/10 ~ 01/11讓同學確認分數，若對於總成績有疑義，請於這兩天寄信至助教信箱詢問。
- 截止時間為 2026/**01/11** 23:59:59 (UTC+8)。
- 學期總成績將於 2026/01/12 送出，成績送出後無法以任何理由要求更改。

NTU COOL 成績公布查詢



總計：94.8% (A+)

NVDLI 加分作業	10 / 10	
Assignments	94.8%	94.80 / 100.00
Imported Assignments	尚未評分	0.00 / 0.00
學期成績	94.8%	94.80 / 100.00

Early access to the course grade

- 課程預計1/12 送出成績。
- 如果有**「合理」**的理由需要提早知道成績，在這次作業十公告**且作業完成後**，我們可以優先改這些同學的作業。
- 請寄信至助教信箱，信件以 **[GenAI-ML 2025 Fall Early Grading]** 開頭並附上緣由與 NTU cool 上所使用之 email。在收到信件後，助教會立即批改您的作業，在您確認成績無誤後，即送出成績。
- 優先批改後無法再以任何形式修改答案。

If You Have Any Questions

- NTU Cool **HW10** 作業討論區
  - 如果同學的問題不涉及作業答案或隱私，請**一律使用**NTU Cool 討論區
  - 助教們會優先回答NTU Cool討論區上的問題
- Email: ntu-gen-ai-ml-2025-fall-ta@googlegroups.com
  - Title should start with [GenAI-ML 2025 Fall **HW10**]
  - Email with the wrong title will be moved to trash automatically
- TA Hours
  - 12/22, 12/29, 01/05 (Mon) 20:00 ~ 22:00
  - 12/26, 01/02, 01/09 (Fri) 17:30 ~ 19:30
  - Location: Google Meet (Link)

Appendix. (Task questions with its options)

(1)

選擇題：根據給定的兩秒、與四秒的音檔中 encode 出來的 Mimi’s tokens 之張量大小(.shape)為何？

- 2sAudio.wav = [1, 32, 120], 4sAudio.wav = [1, 32, 240]
- 2sAudio.wav = [1, 32, 25], 4sAudio.wav = [1, 32, 50]
- 2sAudio.wav = [1, 32, 2], 4sAudio.wav = [1, 32, 4]
- 2sAudio.wav = [1, 32, 2048], 4sAudio.wav = [1, 32, 2048]

Appendix. (Task questions with its options)

(2)

選擇題：比較前一題兩個音檔的最後一個維度，若一個 8 秒音檔被 encode 成 Mimi’s tokens，最後一個維度大小為多少？

- 8
- 150
- 100
- 75

Appendix. (Task questions with its options)

(3)

選擇題：Mimi’s tokens 的第二個維度為 32 代表什麼？

- frames，時間維度代表取出有 32 個frames
- num_codebooks（token layers），代表取出的 tokens 共有 32 層
- channels，代表取出有 32 個聲道
- batch_size，代表取出有 32 組資料

Appendix. (Task questions with its options)

(4)

選擇題：Mimi’s tokens 中的任一數值，其資料型態是什麼？為什麼？

- float32，因為 Mimi 的 tokens 實際上是直接儲存 embedding 向量的浮點數值。
- integer，因為 Mimi 採以 RVQ 將音訊離散化成為一個個對應到固定 embedding 上離散的 tokens。
- string，因為 Mimi 的 tokens 被設計成符號化的單位，因此是以字串形式表示。
- boolean，因為 Mimi 的 tokens 本質上是二元化的結果，只需用 0/1 來表達。

Appendix. (Task questions with its options)

(5)

選擇題：Mimi’s tokens 有多個層 (layers)，每個 token 對應特定 embedding。請問在第零層 codebook_layer(token layers) = 0，code ID 為 2047 (0-indexed)，其 embedding 前五個數值為？（四捨五入至小數點後四位）

- [0.3452, 0.0624, -0.1930, -0.5179, 0.1582]
- [-0.0055, 0.3850, 0.7027, -0.4732, 0.5290]
- [-0.2068, 0.3137, -0.9929, -0.0826, -0.0087]
- [-0.6255, 0.0448, -0.4475, 0.7799, -0.4781]

Appendix. (Task questions with its options)

(6)

多重選擇題：依照 colab 流程，畫出 [0, 6, 16, 31] 層 Mimi’s tokens embedding 在不同情緒標記下的 UMAP。依據結果選出正確敘述：

- UMAP 是一種非線性的降維視覺化工具。
- UMAP 中同樣顏色的點，代表被標記成同類型情緒的音檔。
- 比較 Layer 0 與 Layer 31，層數(Layer)越大，embedding 在情緒類別上的群集分離度更高。
- 比較 Layer 0 與 Layer 31，層數(Layer)越小，embedding 在情緒類別上的群集分離度更高。
- 仔細觀察 Layer 0 之散佈圖，該層能夠大致地將 Amused, Angry, Sleepy 分至不同群。
- 仔細觀察 Layer 0 之散佈圖，該層能夠大致地將 Disgusted, Sleepy, Neutral 分至不同群。
- 仔細觀察 Layer 6 之散佈圖，該層能夠大致地將 Amused, Angry 分群。
- 仔細觀察 Layer 6 之散佈圖，該層能夠大致地將 Sleepy, Angry 分群。

Appendix. (Task questions with its options)

(7)

選擇題：比較 TTS_English_speech.wav 與 TTS_Chinese_speech.wav，哪個敘述是正確的？

- 中文 TTS 效果顯著優於英文，顯示 Mimi 在中文語音上更有優勢。
- 英文與中文 TTS 都無法被有效重建，顯示 Mimi 並不適合跨語言語音任務。
- 兩者重建效果皆佳，顯示 Mimi 可同時處理中英文音訊資訊。
- 英文 TTS 效果顯著優於中文，顯示 Mimi 僅適合英文語音。

Appendix. (Task questions with its options)

(8)

選擇題：比較 laughter.wav 和 music.wav，哪一個音檔 decode 回來的效果最差，這可能代表什麼？

- laughter.wav 在重建的效果上最差，顯示 Mimi 無法辨識情緒相關的副語言訊號。
- laughter.wav 在重建的效果上最差，顯示 Mimi 僅適合處理語音而不適合非語言聲音。
- music.wav 在重建的效果上最差，顯示 Mimi’s tokens 若被後續模型使用，可能難以處理音樂。
- music.wav 在重建的效果上最差，顯示模型對長時序的音訊特別敏感，因此難以重建音樂。

Appendix. (Task questions with its options)

(9)

選擇題：在程式碼中，一個已經預先用 Mimi model encode 成 4 layers 的檔案 audio_codes_shuffled.pt，其層間順序被隨機打亂。請嘗試找出正確順序（不重複排列，共 4! = 24 個組合），並聆聽 decode 結果。音檔內容文字為？

- Generative AI can be challenging, but each experiment feels like exploring a completely different universe of thoughts.
- Generative AI is so much fun, and every experiment feels like discovering a brand-new world of ideas.
- Generative AI is quite powerful, and every experiment feels like stepping into a fresh landscape of imagination.
- Generative AI is truly exciting, and every new attempt feels like opening the door to another dimension of creativity.

Appendix. (Task questions with its options)

(10)

多重選擇題：首先，聆聽提供的兩個等長音檔 (Alice & James)，接著 encode 成 8 layers，並交錯合併再 decode，聆聽效果並根據實驗結果選出以下正確敘述。

- Mimi's tokens 的第零層在訓練上採對齊 WavLM 的 Semantic Token 的做法。
- 由於第零層的內容對齊的是 Semantic Token，故在八層中第零層被替代掉的情況下，就不可能有辦法 decode 回可以聽懂的語音內容。
- 即使第零層的資訊被取代了，僅透過後面幾層，decoder 也能夠良好地將音訊轉換回來。
- 音訊在 split_index = 2 時，音訊相對模糊不清。
- 從 split_index = 3 起，結果更接近第一個語音（Alice）
- 從 split_index=2～6 的音檔觀察發現「結果音檔由前幾層主導」，也就是即使 Alice 只佔少數層，解碼仍更像 Alice。
- 從 split_index=2～6 的音檔觀察發現「結果音檔由後幾層主導」，也就是即使 James 只佔少數層，解碼仍更像 James。

Appendix. (Task questions with its options)

(11)

選擇題:在 colab 內嘗試更改 input_sequence 的內容, 請問給定圖片內的 input_sequence 會得到怎麼樣的音檔？

- Jason 的聲音說 “Do you know that you were generated by AI?”
- Emily 的聲音說 “What? Seriously?”
- Emily 的聲音說 “Do you know that you were generated by AI?”
- Jason 的聲音說 “What? Seriously?”

Appendix. (Task questions with its options)

(12)

選擇題：觀察 Colab 程式碼，並理解 CSM 模型後，選出正確的敘述。

- 由於 CSM 模型以 Mimi's tokens 作為 audio token，故模型輸出的 audio token 可直接送回 Mimi model 解碼成音訊。
- 由於 CSM 模型還有再讓 Mimi model 經過後訓練，故模型輸出的 audio token 無法直接送回 Mimi model 解碼成音訊。
- 由於 CSM 模型並未與 Mimi model 同時訓練，故 CSM 模型輸出的 audio token 無法直接送回 Mimi model 解碼成音訊。
- 由於 CSM 模型與 Mimi 模型是兩個獨立的模型，CSM 生成出來的 token，傳入 Mimi 的 decoder 只會得出雜訊。