

# 用時頻分析搭配肉眼檢測deepfake語音

Detecting deepfake speech with naked eyes by using time-frequency analysis

報告人: 莊鳴鐸 R13942091

# 目錄 table of content

- Deepfake TTS套件介紹
- Mel-spectrumgram 介紹
- 用Mel-spectrumgram 分析
- 用STFT分析
- 用WDF分析
- 用Hibert-Huang Transform 分析
- 總結

# Which audio is real ?



收到好友從遠方寄來的生日禮物，那份真摯的驚喜與深情的祝福讓我心中充滿了甜蜜的快樂，笑容如花兒般綻放。



本來應該是從容容、游刃有餘，現在是匆匆忙忙、連滾帶爬！



我喜愛的課是不建軍、帥的時頻分析，且小波轉換！

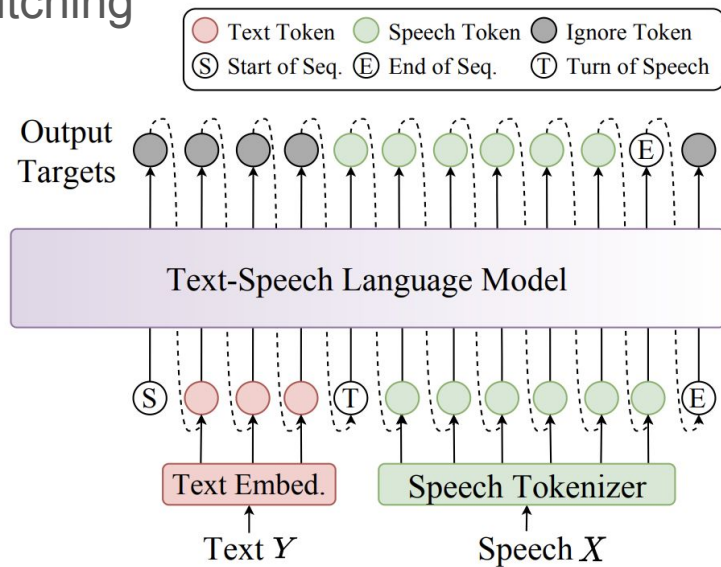


# Deepfake TTS套件介紹

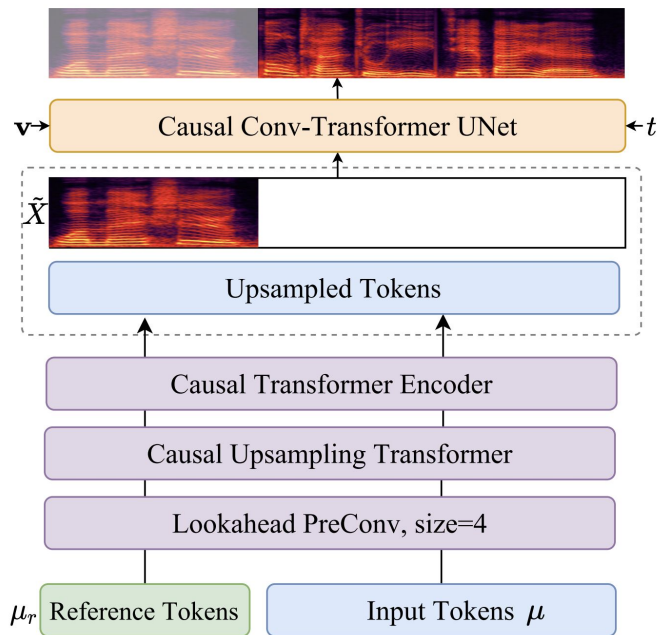
CosyVoice2: <https://github.com/FunAudioLLM/CosyVoice>

9 languages

\*support code switching



vocoder



# Deepfake TTS套件介紹

coqui/XTTS-v2: <https://github.com/coqui-ai/TTS>

17 languages

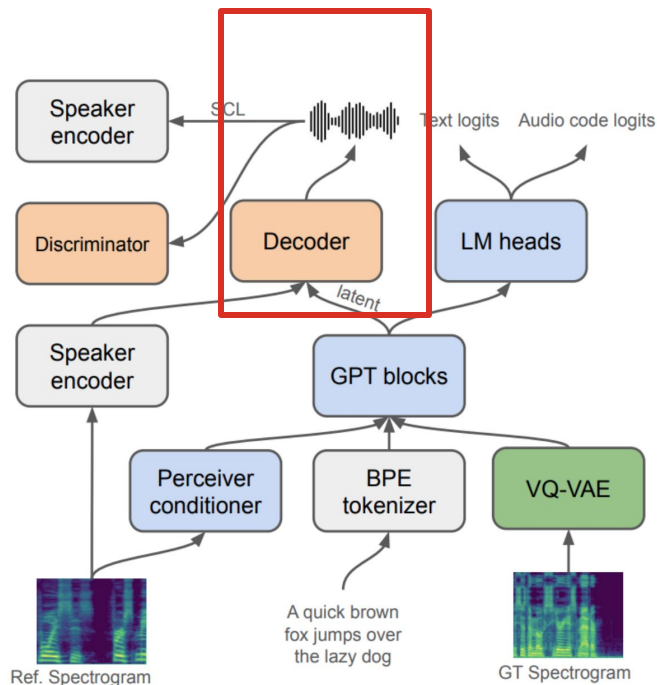


Figure 1: XTTS training architecture overview.

## Comparison: training objective

	CosyVoice2	XTTS v2
before decoding	Mel-spectrogram	discrete tokens
after decoding	Waveform (Signal)	Waveform (Signal)

# Mel-spectrumgram 介紹

A type of lossy compression

A time-frequency analysis pipeline

The process requires calculating the STFT.

Often used in speech audio

Parameter chosen according to the perception of sound of human

# Mel-spectrumgram - a bit of Math.....

## step1: Pre-emphasis(optional)

$$x'[n] = x[n] - \alpha \cdot x[n-1]$$

$\alpha = 0.97$  for 16k audio (chosen)

## step2: STFT

$$P[m,k] = |\text{STFT}(x'[n])|^2$$

lossy!!

M: 總共的frame數

H: frame hop (512)

# Mel-spectrumgram - a bit of Math....

**step3: 通過Mel-filterbank**

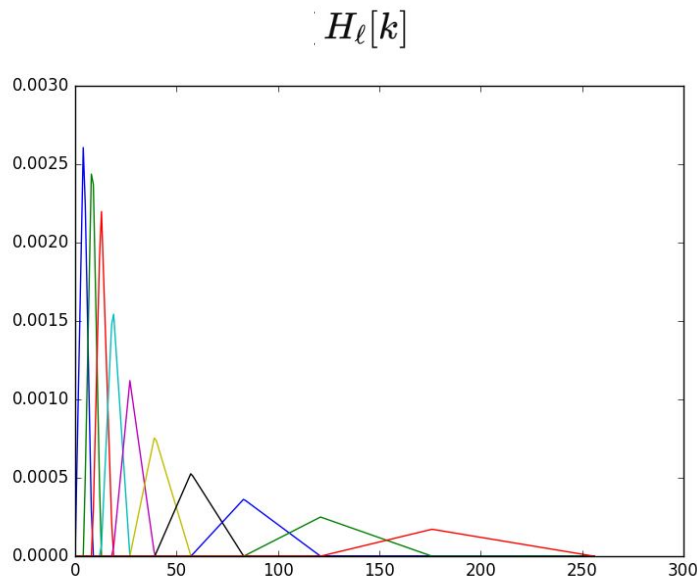
$$S_m[\ell] = \sum_k H_\ell[k] \cdot P[m,k]$$

$$\ell = 128$$

**step4: 取log**

$$D[m, \ell] = 10 \log_{10}(\max(S[m, \ell], \epsilon))$$

$\epsilon = 10^{-10}$ , to avoid 0 for log



# Mel-spectrumgram - a bit of Math....

## Look deeper into Step 1 Pre-emphasis

$$x'[n] = x[n] - \alpha \cdot x[n-1]$$

for Sampling rate =16k,  $\alpha=0.97$

100 Hz: -26.21 dB

300 Hz: -18.43 dB

1 kHz : -8.28 dB

2 kHz : -2.45 dB

3 kHz : +0.79 dB

4 kHz : +2.88 dB

8 kHz : +5.89 dB

cutoff: 相對 DC 增益上升 3 dB

$f = 77.6$  Hz

### 人體發聲

聲門脈衝的頻譜傾斜 (Spectral Tilt):

slope: -12 dB/octave

唇輻射效應 (Lip Radiation):

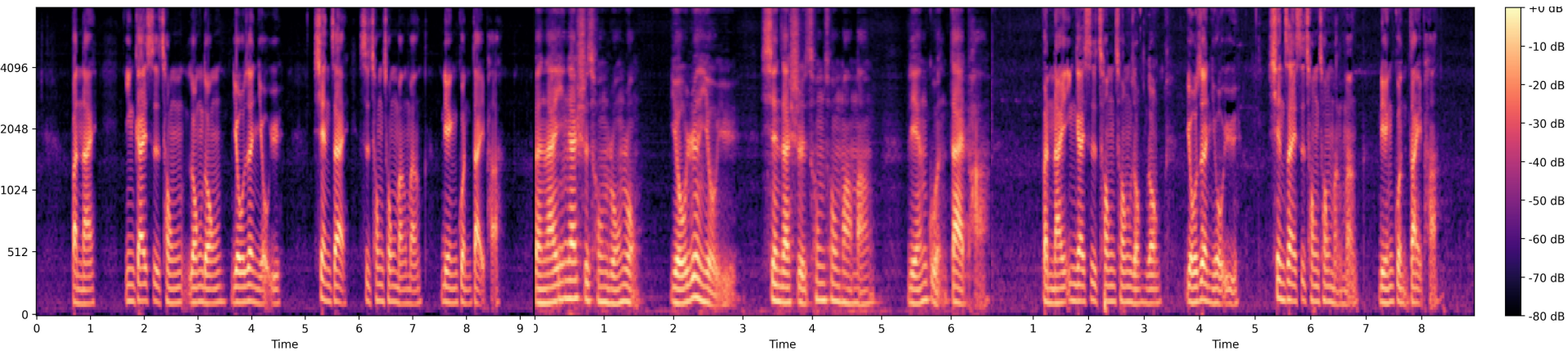
slope: +6 dB/octave

$\alpha=0.97$

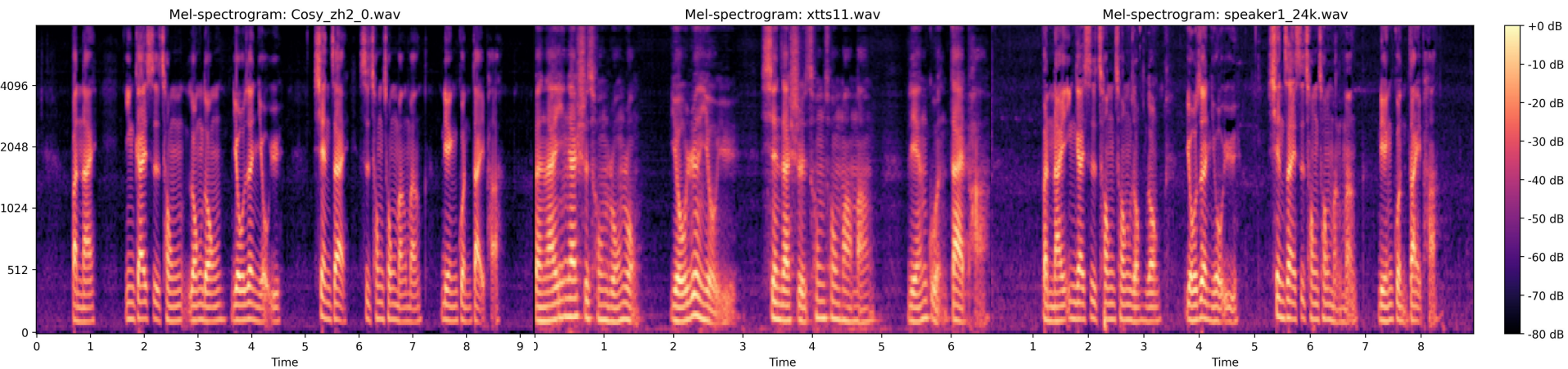
slope: +6 dB/octave

$$-12+6+6=0$$

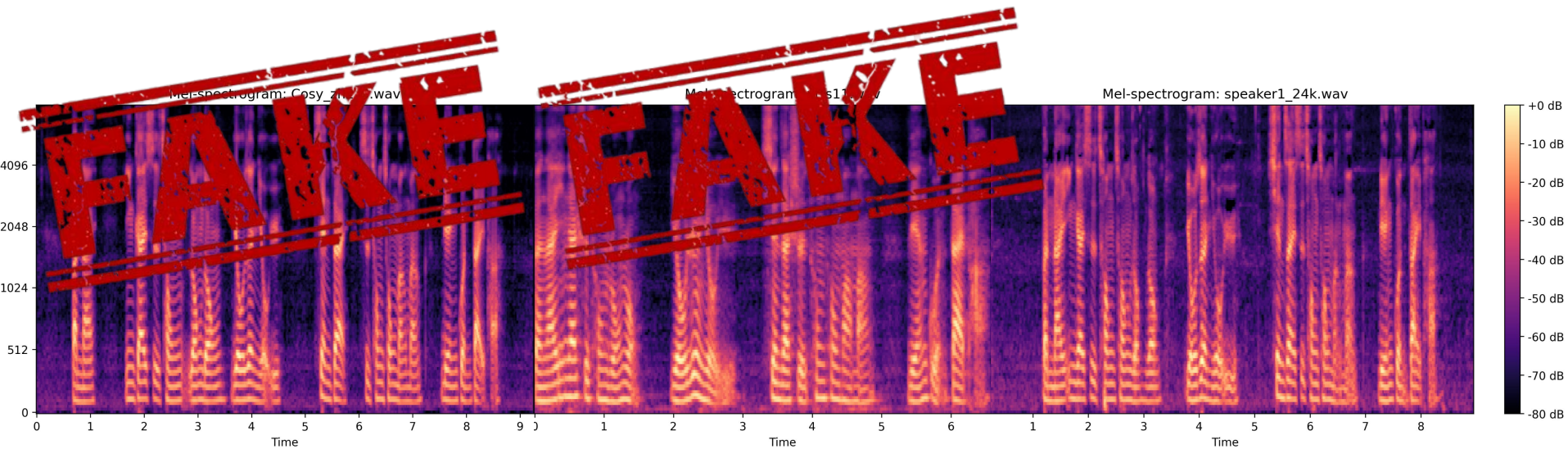
# 用Mel-spectrumgram 分析



# 用Mel-spectrogram 分析



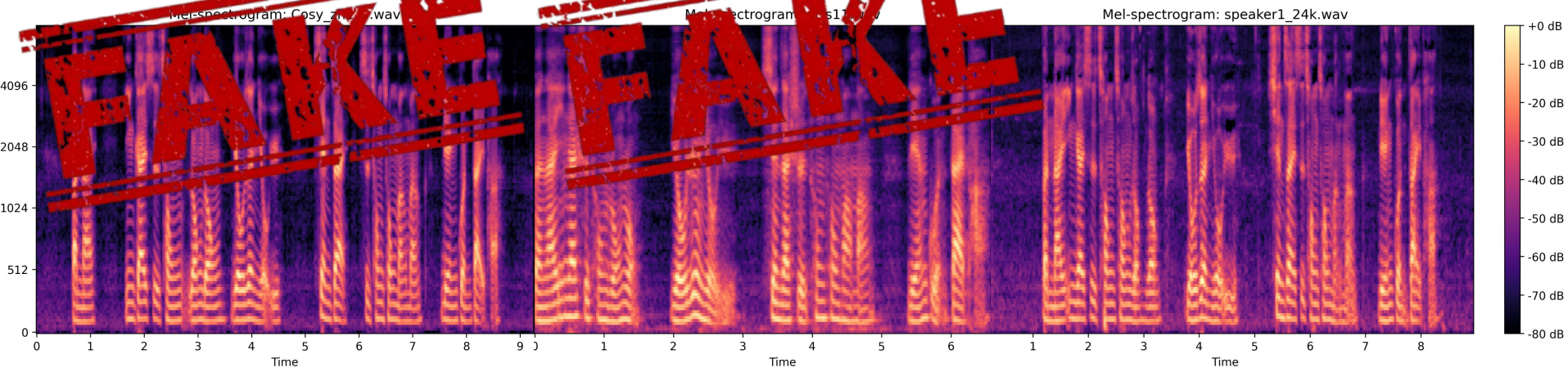
# 用Mel-spectrumgram 分析



# 用Mel-spectrumgram 分析

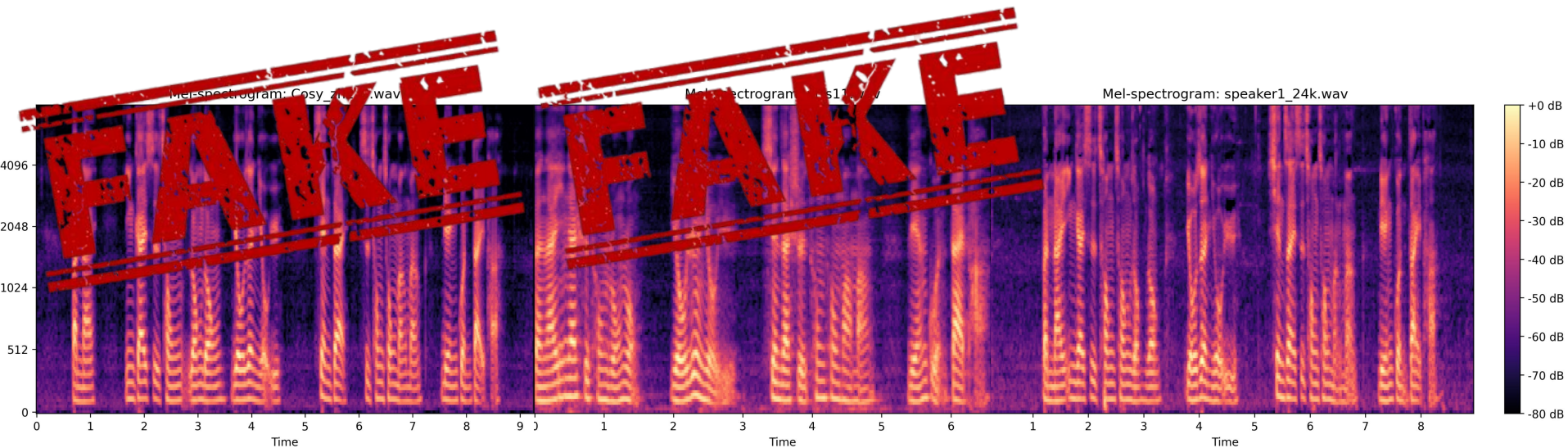
model output: Mel

model output: tokens



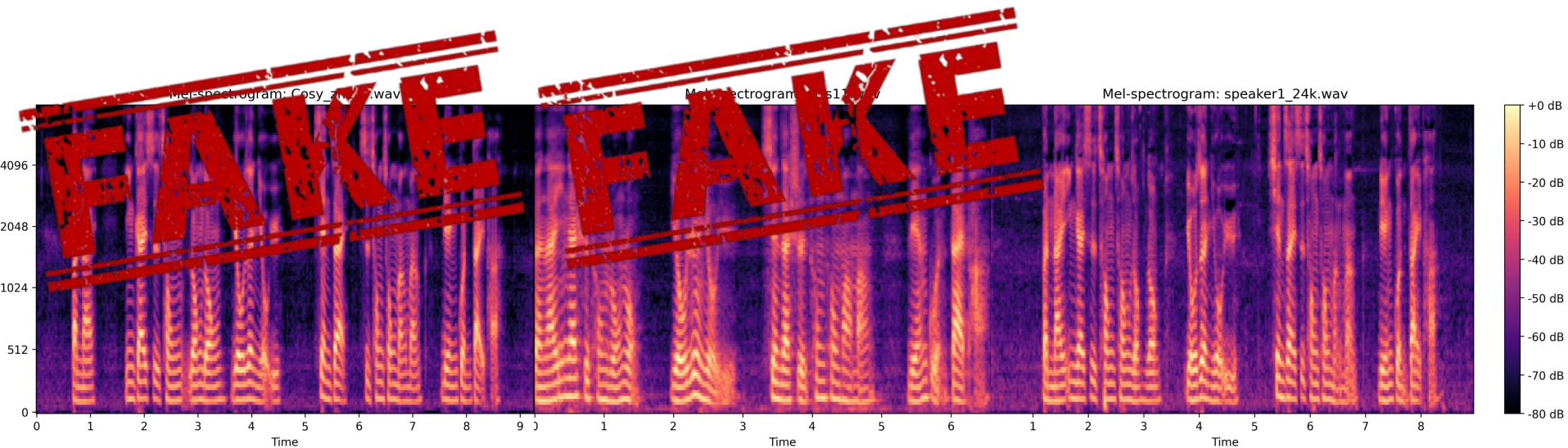
# 讓LLM用Mel-spectrogram 分析

*ChatGPT 5.2 thinking and Gemini 3-pro:*



# 讓LLM用Mel-spectrogram 分析

ChatGPT 5.2 thinking and Gemini 3-pro:



GPT: fake ✓ low confidence

Gemini: non reasonable answer ✗

GPT: fake ✓ high confidence

Gemini: non reasonable answer ✗

GPT: real ✓

Gemini: non reasonable answer ✗

# 用Mel-spectrumgram 分析

就算training objective不在Mel上也看不出明顯缺陷

可能成功原因:

Deepfake背景太乾淨

可能失敗原因:

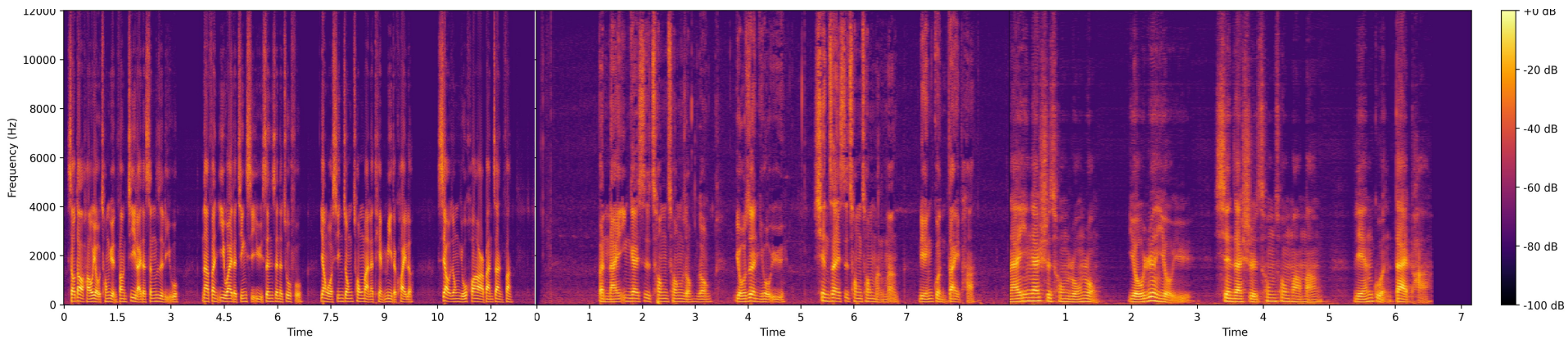
Mel-spectrumgram壓縮太多

Signal with  $N$  points (  $N/2$  個frequency )

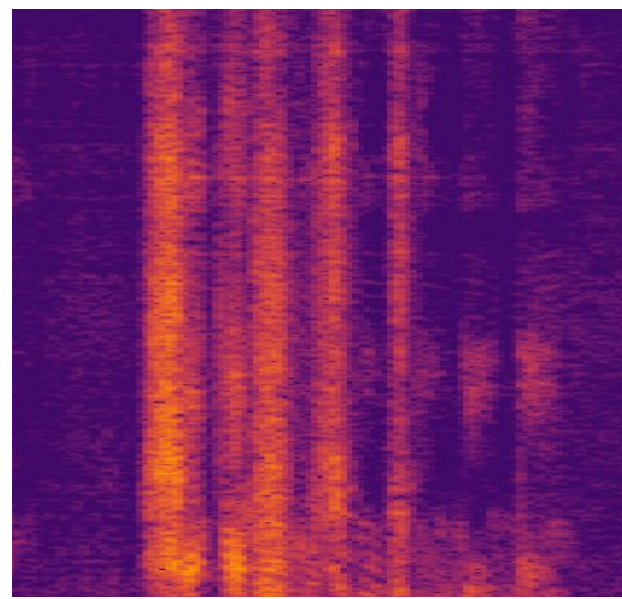
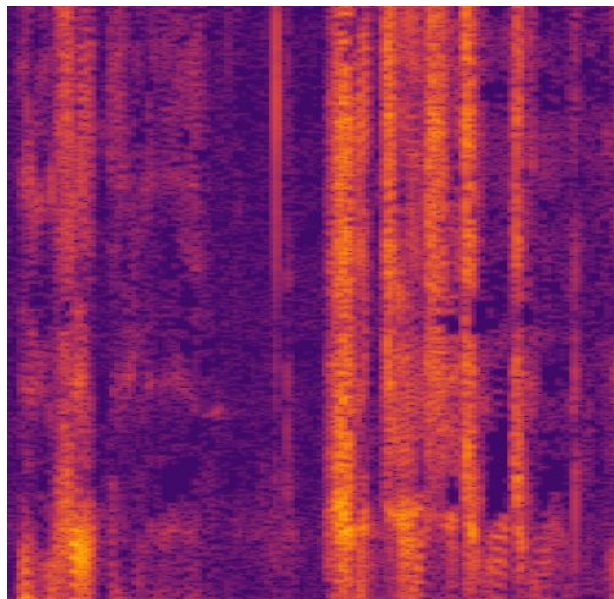
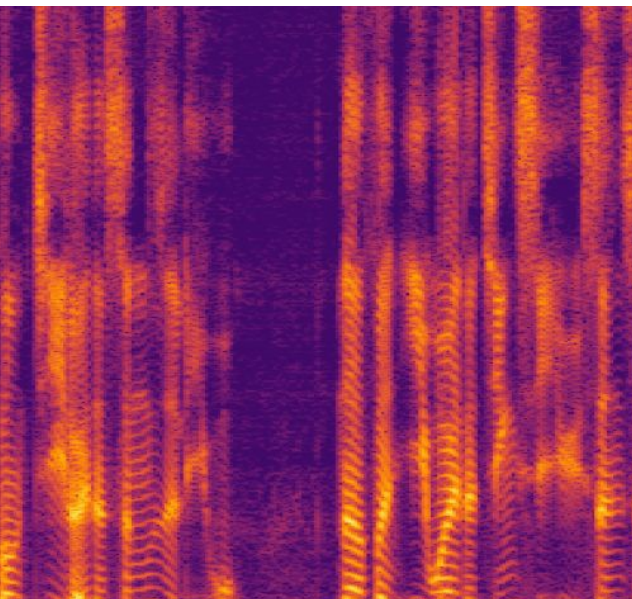
$[N/2, N] \Rightarrow [128, N/512]$  (128 is the channel number of Mel filterbank)

too lossy !

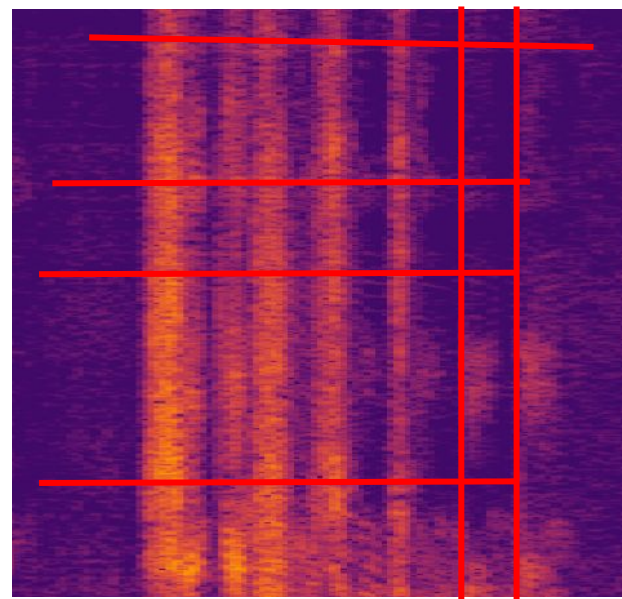
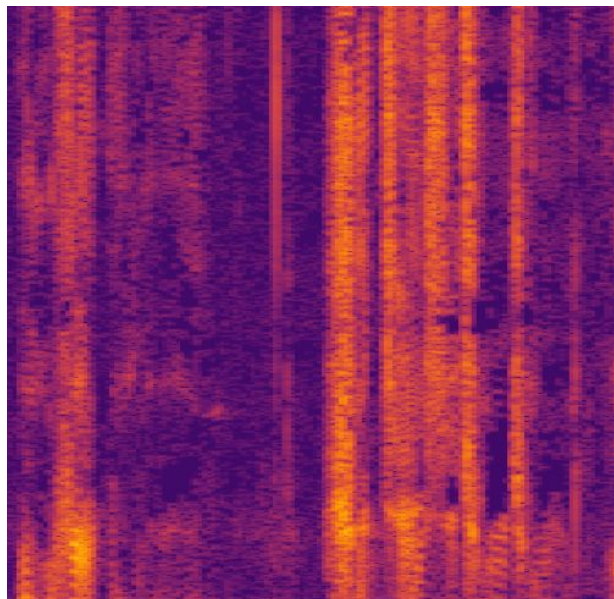
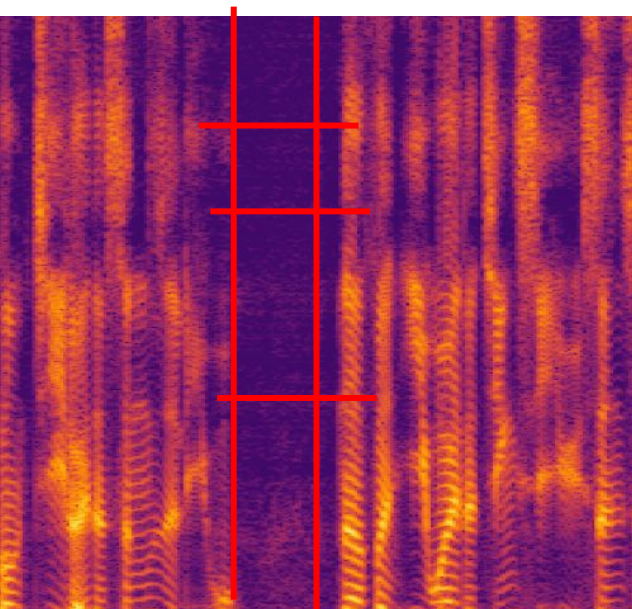
# 用STFT分析



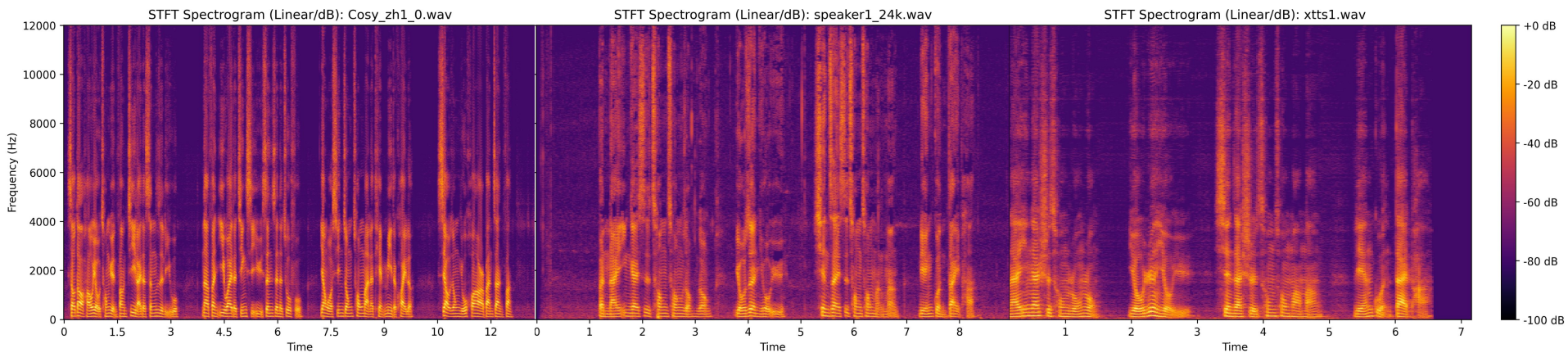
# 用STFT分析



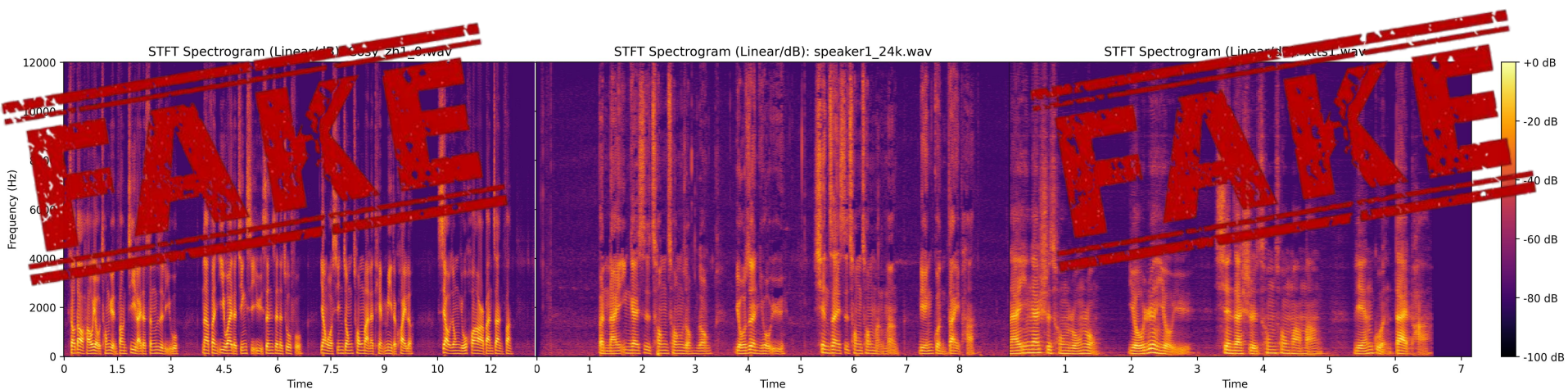
# 用STFT分析



# 用STFT分析

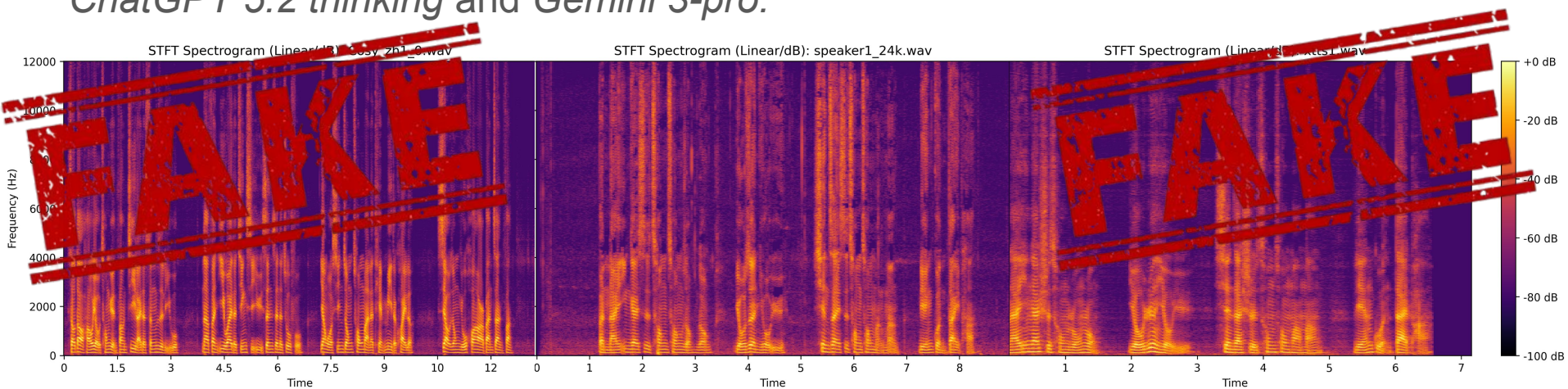


# 用STFT分析



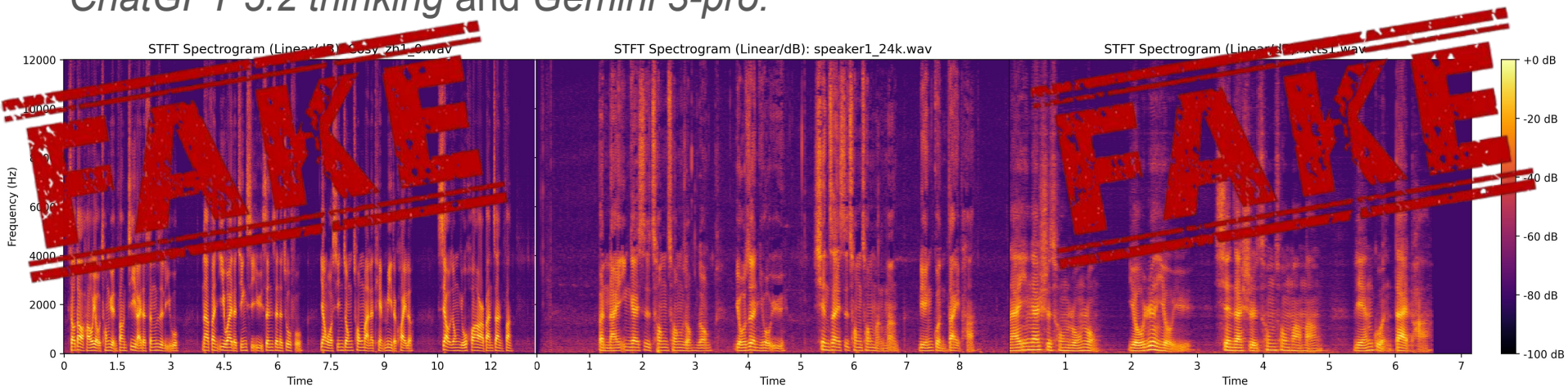
# 讓LLM用STFT 分析

*ChatGPT 5.2 thinking and Gemini 3-pro:*



# 讓LLM用STFT 分析

*ChatGPT 5.2 thinking and Gemini 3-pro:*



GPT: fake ✓

Gemini: fake ✓

GPT: real ✓

Gemini: real ✓

GPT: fake ✓

Gemini: fake ✓

# 用STFT分析

## LLM皆可輕易判讀

### 可能成功原因:

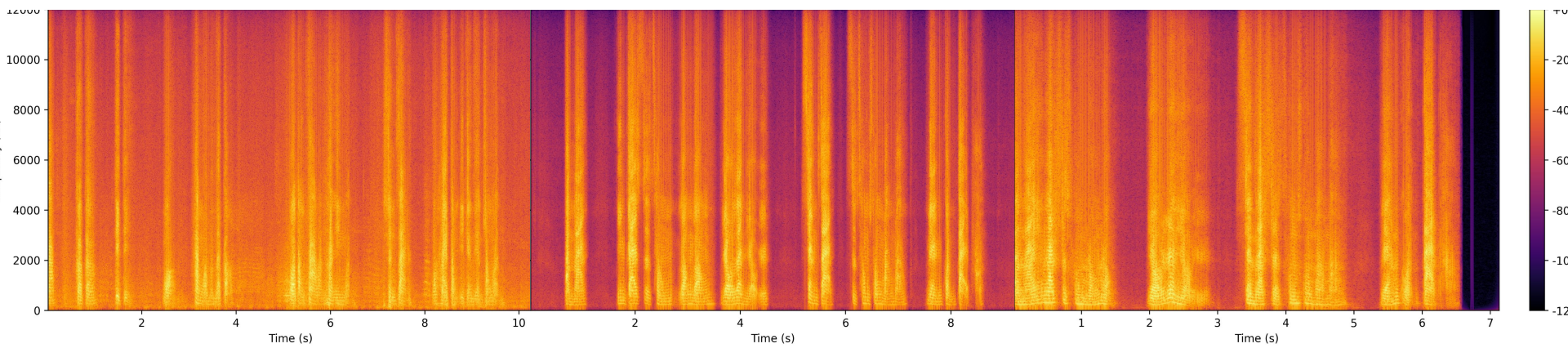
Vocoder/ Decoder的天生缺陷

Upsampling Stride沒有對好 => 訊號邊緣overlapped

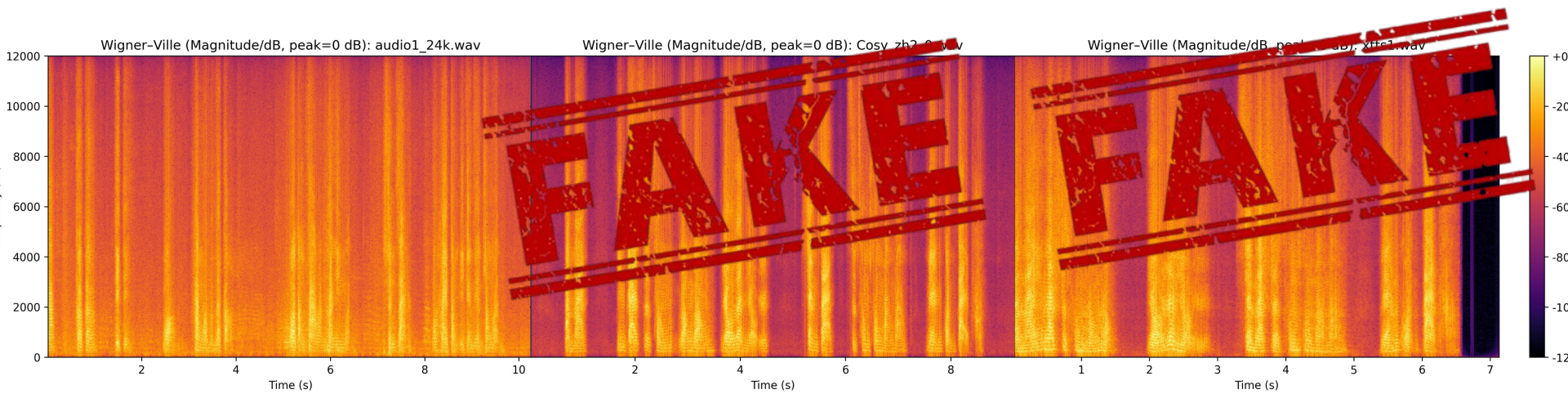
Vocoder output 長度有限 => 學不到高頻生成

人耳對週期高頻不敏感 => 聽起來還可以

# 用WDF分析

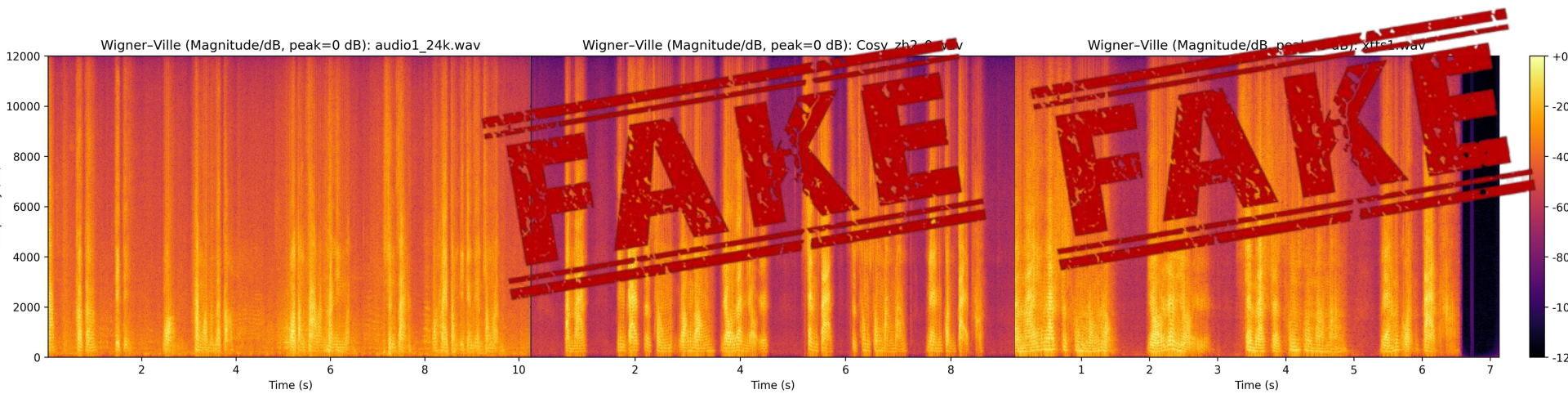


# 用WDF分析



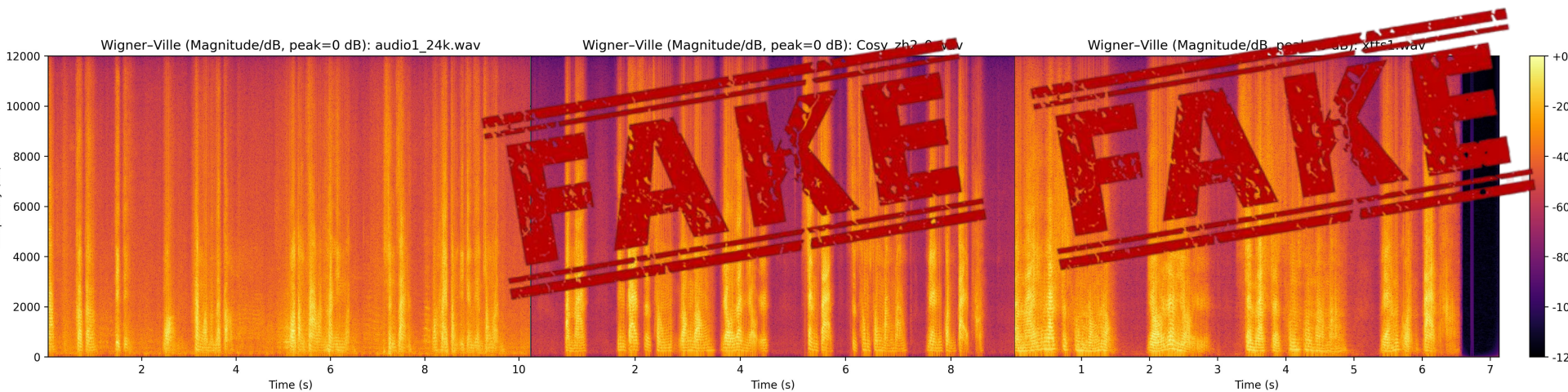
# 讓LLM用WDF 分析

*ChatGPT 5.2 thinking and Gemini 3-pro:*



# 讓LLM用WDF 分析

*ChatGPT 5.2 thinking and Gemini 3-pro:*



GPT: real ✓

Gemini: non reasonable answer ✗

GPT: fake ✓ low confidence

Gemini: non reasonable answer ✗

GPT: fake ✓ high confidence

Gemini: non reasonable answer ✗

# 用WDF分析

## 可能成功原因:

Deepfake背景太乾淨

## 可能失敗原因:

解析度太高 => 見樹不見林

高敏感度 => 不同方式產生音訊本就差很多

cross-terms => 以假當真

視覺化後看不到phase

# 讓LLM用WDF phase分析

*ChatGPT 5.2 thinking*

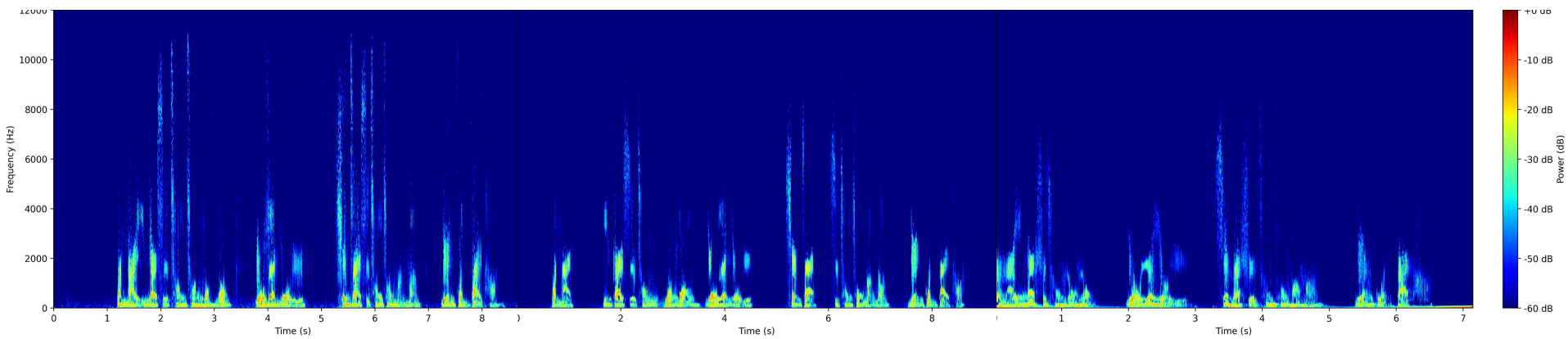


GPT: real

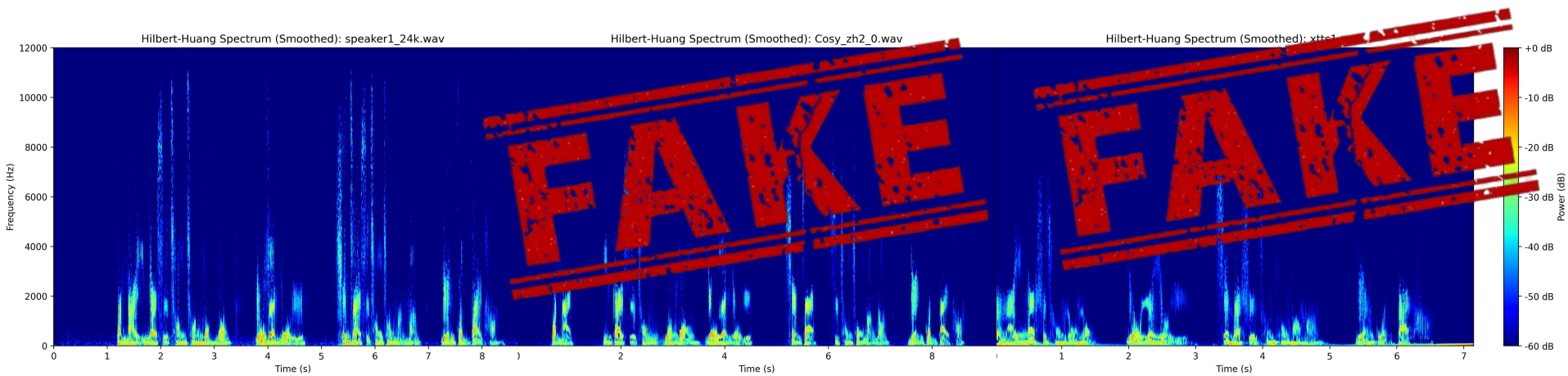
GPT: fake   
low confidence

GPT: fake   
high confidence

# 用Hibert-Huang Transform 分析

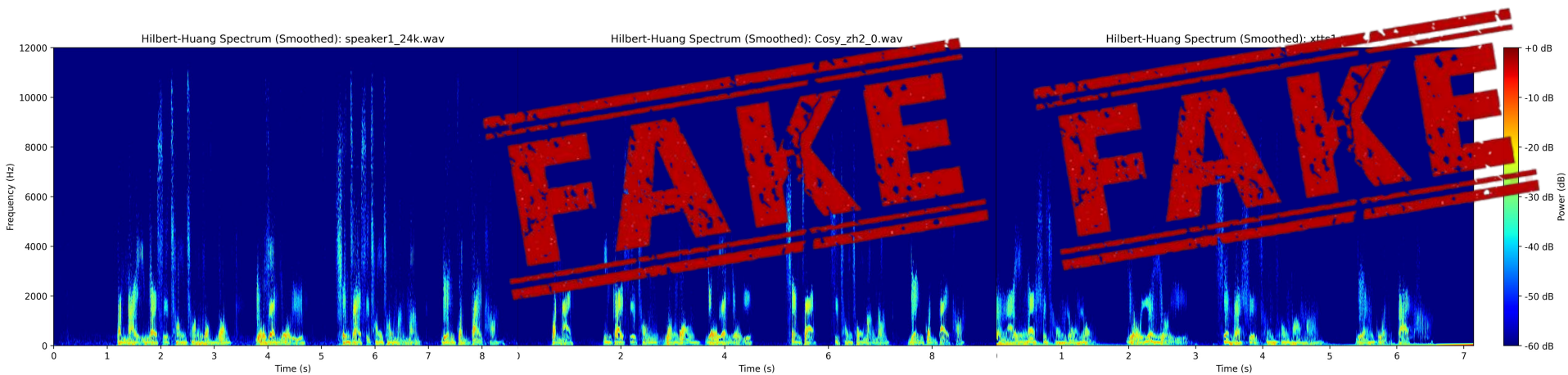


# 用Hibert-Huang Transform 分析



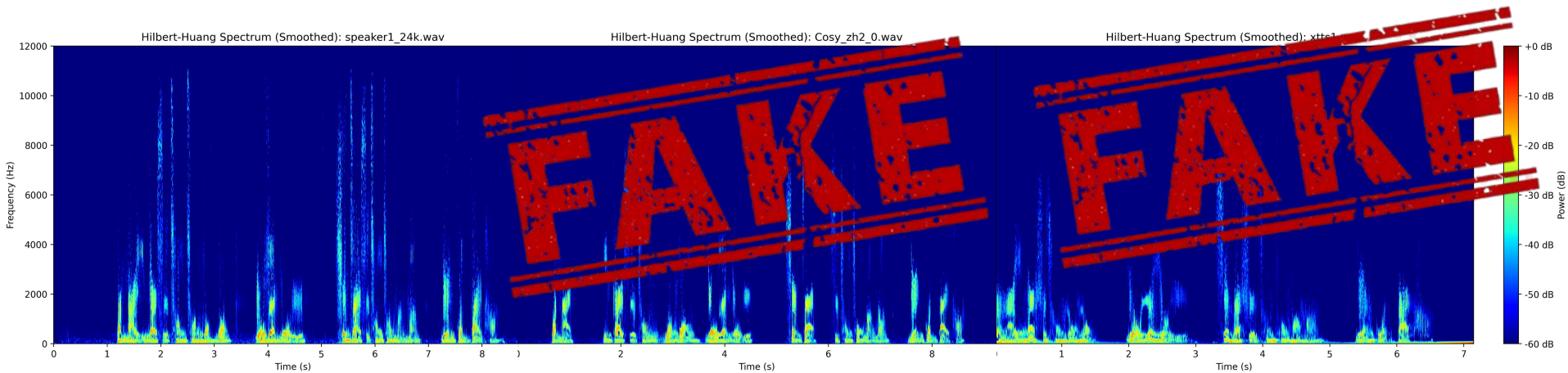
# 讓LLM用WDF phase分析

*ChatGPT 5.2 thinking*



# 讓LLM用WDF phase分析

*ChatGPT 5.2 thinking*



GPT: insufficient confidence ?

GPT: insufficient confidence ?

GPT: insufficient confidence ?

# 用Hibert Huang Transform分析

**可能失敗原因:**

把訊號表示成IMF的組合 => over analyse

source/ config sensible

# 總結

- Deepfake TTS quality: Cosyvoice2 >>> XTTS\_v2
- 檢測方法: STFT >> Mel  $\approx$  WDF > Hibert-Huang
- Mel用很少資料抓住重要feature => 壓縮太多不利檢測
- 選對工具很重要

## Disclaimer:

實驗中僅以少量樣本實驗，實驗結果可能不具有統計意義

實驗的code

