Thanks for laying that out clearly, Heiko — those 3 points give us a strong compass for shaping an alignment strategy that actually reflects what you’re trying to achieve.

Let’s break it down and address each piece intelligently:

**🧠 Your Goals Recap**

1. **Avoid duplicate lines with differing timestamps**
2. **Don't expect perfect word-for-word matches between Original and ASR**
3. **ASR should overwrite Original — treat it as the “source of truth”**

**🔧 What’s Going Wrong in the Current merge\_subtitles()**

* The original file's timestamps and segment structure are being preserved verbatim — even when they don’t align well with ASR.
* We're matching ASR → Original, but **you're asking to prioritize ASR**, not preserve Original's formatting.
* Duplicates appear because multiple Originals sometimes map vaguely to the same ASR segment due to fuzzy fallback logic.

**✅ Strategy Shift: Use ASR Blocks as Anchors**

We flip the logic:

**For each ASR block → find a matching Original block (if any) → use ASR’s timestamps and text in merged output.**

This ensures:

* No duplicated lines.
* You always keep ASR’s structure.
* Optionally blend in Original text when needed (e.g. for annotations or comparison later).

**💡 Next Move**

Let’s rewrite merge\_subtitles() to:

* Iterate over each ASR line
* Try to find the **closest original line** (by timestamp or fuzzy match if needed)
* Use the **ASR block’s timing and text** in the final output
* Only optionally include Original text (e.g. for diffing or debugging)

Would you like the merged file to just be ASR output, or are you looking for something like this:

1

00:00:01,000 --> 00:00:03,000

ASR: Hello, welcome to the show.

Orig: Hello and welcome.

2

00:00:03,500 --> 00:00:06,000

ASR: Let's begin our topic...

Orig: Let's start the topic...

That second format is helpful for human editors, but if you want a clean .srt ready for playback, we’d just keep the ASR block as the final say.

Let me know how “transparent” vs. “clean” you want the output to be — and I’ll craft the next version accordingly. 🎯📄🧠

Excellent point, Heiko — those [sound cues] and descriptive annotations are crucial for accessibility and definitely worth preserving. Let’s evolve your sync logic to honor that:

**🎯 Updated Strategy for merge\_subtitles()**

We'll now align subtitles with these principles:

1. **Use ASR structure** as the sync backbone (timestamps + text)
2. **Match and optionally merge any comments** from the original (in [brackets])
3. **Preserve original comments** unless they overlap/conflict with spoken lines

**✅ Implementation Plan**

Here's how we’ll do it:

**🧩 Step 1: Extract Comments from Original Blocks**

We detect lines that are either:

* Entirely comment lines: [Door slams]
* Inline comments, like: "Let’s go [whispers]"

We keep a separate list of these, along with their timing windows.

**🧩 Step 2: While Building from ASR Blocks**

* Use the ASR block’s text + timestamps
* Look for any comment-only original blocks whose time overlaps and add them as their own line — **unless they collide closely with the spoken text**
* Optionally allow "stacking":  
  If [Applause] appears right before a spoken ASR line, you can include both.

**🧠 Output Example**

1

00:00:00,500 --> 00:00:01,700

[Door closes]

2

00:00:01,800 --> 00:00:04,100

Let’s begin our journey.

3

00:00:04,300 --> 00:00:06,000

[Music fades]

This keeps commentary in place without overwriting what ASR has determined to be accurate speech.

Would you like to:

* Allow [comments] to appear **as separate blocks** before/after ASR speech?
* Or try merging them **into ASR text** when timestamp overlap is close?

I can write the logic either way — even give you a toggle if you’re feeling fancy. Let’s make this merge engine sing like a stagehand with perfect cue timing. 🎭🔊🧠