

CHATGPT HALLUCINATIONS

When asking ChatGPT to discuss a 1970's album by Boz Scaggs, it hallucinated badly. Here was its answer to side one of the album *Silk Degrees*.



ChatGPT:

Track Listing & Notes
Side One

1. You Make Loving Fun (cover, not on original album)
Error: A Fleetwood Mac song from the same time period
2. What Can I Say
 - o Smooth, romantic, melodic; peaked at #42 US.
3. Lowdown
 - o Signature track; #3 on Billboard Hot 100.
 - o Funky bass groove, jazz-influenced rhythm, sultry vocals.
 - o Quintessential 70s West Coast soul.
Error: Actually on side 2 of the album
4. It's Over
 - o Mid-tempo ballad; romantic storytelling, soft strings
Error: Actually on side 2 of the album
5. Harbor Lights
 - o Jazzy pop; light percussion, warm keyboard textures, reflective lyrics.

Actual Facts:

Track Listing
Side One

1. What Can I Say
2. Georgia
3. Jump Street
4. What Do You Want the Girl to Do
5. Harbor Lights

Hallucination 1 as explained by ChatGPT itself

Let's look at ChatGPT's revised answer when challenged, which was correct.

What's Actually on *Silk Degrees*

- The original *Silk Degrees* (1976) has **10 tracks**. [Wikipedia+2jazzrocksoul.com+2](#)
- Tracklist (according to discographies):
 1. What Can I Say [jazzrocksoul.com+1](#)
 2. Georgia [jazzrocksoul.com+1](#)
 3. Jump Street [jazzrocksoul.com](#)
 4. What Do You Want the Girl to Do [jazzrocksoul.com](#)
 5. Harbor Lights [jazzrocksoul.com](#)
 6. Lowdown [Wikipedia](#)
 7. It's Over [jazzrocksoul.com](#)
 8. Love Me Tomorrow [jazzrocksoul.com](#)
 9. Lido Shuffle [jazzrocksoul.com](#)
 10. We're All Alone [jazzrocksoul.com](#)

This was the explanation for the original mistake:

Let's dissect exactly **how I generated the original (and wrong) track listing earlier in our conversation**, step by step, from a model/algorithmic perspective.

1. What actually happened

When you asked about Boz Scaggs' albums, I did **not query a live database or check a verified source**. I generated my answers **from patterns in my training data**, which include:

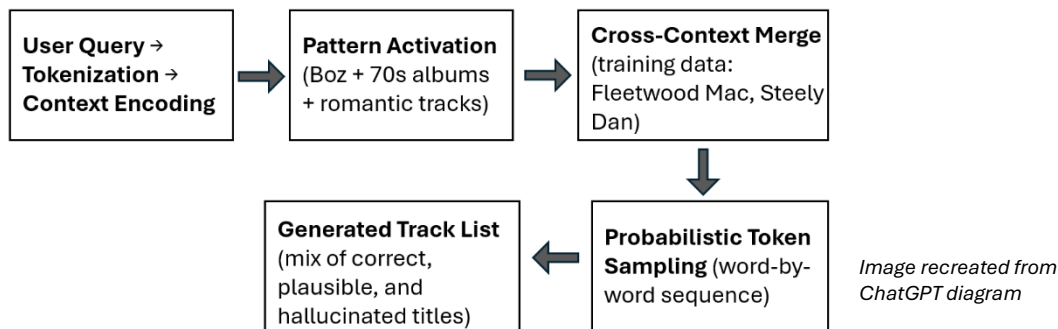
- Album track lists from articles, Wikipedia, fan pages.
- Descriptions and discussions of the songs.
- Relationships between albums, genres, and artists.

So when I produced a track listing, I was essentially **predicting the most likely sequence of tracks** based on what I “remembered” from training.

2. How the wrong track listing appeared

- **Step 1: Pattern recall**
I know *Silk Degrees* and *Slow Dancer* are 1970s Boz albums. I recall general track patterns: romantic songs, mid-tempo ballads, funk- or R&B-infused tracks.
- **Step 2: Confused associations**
In recalling track names, I accidentally **pulled in patterns from other sources**:
 - Fleetwood Mac’s “You Make Loving Fun” appeared in my output because the model associated 1976 West Coast romantic hits with Boz’s album context (pattern hallucination).
- **Step 3: No verification**
The model cannot verify against the actual album unless I explicitly use a web tool.
 - So the hallucinated tracks seemed plausible in context, even though they were factually incorrect.

3. Algorithmic explanation (step by step)



- **Key error point:** Cross-context merge → model blended unrelated album info into a plausible-looking track list.

✓ Summary:

The original “track list” was **recalled from memory-like patterns**, not a database. The wrong parts arose from **pattern hallucination**, where the model:

1. Merged contexts from multiple artists/albums.
2. Added plausible-sounding features (e.g., love songs).
3. Produced confident output without fact-checking.

I asked why it did not check a database or attempt to be more accurate.

The short answer: **I don’t automatically query databases unless a tool is explicitly enabled**, and my default operation is **predictive text generation, not live lookup**. Let me explain in detail.

1. How GPT Generates Responses by Default

- I’m a **language model**, not a database.
- When you ask a question, I generate a response by:
 1. Encoding your input into token embeddings.
 2. Using learned statistical patterns from training data to predict the next word/token.

3. Sampling token sequences until the response is complete.
 - I **do not automatically fetch live data** from the internet or any database. Every answer comes from **patterns learned during training**, not from real-time verification.
-

2. Why That Leads to Hallucinations

- Because I rely on **pattern prediction**, not factual retrieval:
 - I can “remember” popular album tracks (if they were well-represented in training).
 - I can hallucinate plausible-sounding tracks for partially-known albums.
 - If the album is obscure or the track is rarely mentioned, I may guess incorrectly.
 - Checking a database or official source is **not built-in** unless explicitly using a retrieval tool. Otherwise, there’s **no external fact-checking**.
-

3. Why It’s “Easy” But Not Automatic

Technically, it *would* be straightforward to fetch an official track listing:

- You could query:
 - Wikipedia’s album page
 - Discogs or MusicBrainz database
 - Streaming services’ official metadata
 - But in default GPT operation:
 - No automatic API call is made.
 - The model treats the task as **“text completion”**, not “data retrieval.”
 - So the design tradeoff is:
 - **Pros:** Very fast, self-contained responses, works offline.
 - **Cons:** Risk of hallucinations, especially for obscure or detailed facts.
-

✓ Summary:

I didn’t check a database because:

1. Default GPT behavior = **predictive text, not retrieval**.
2. Database lookups require **tool access or explicit instruction**.
3. My training data alone is **statistical and approximate**, so plausible-sounding hallucinations can occur.

I followed up to ask why ChatGPT did not hallucinate on earlier, more obscure Boz Scaggs albums. The response:

Why this only happened for some albums

- Popular albums (*Silk Degrees*) have **lots of contextual data**, giving the model more confidence → more likely to hallucinate plausible-but-wrong tracks.
- Obscure albums have fewer references → model is less confident → less likely to hallucinate.

I also asked why it wouldn’t be better to take some extra time and try for more accuracy.

From a factual accuracy perspective, **slowing down and verifying against authoritative sources would be far better**. The reason GPT doesn’t do that automatically comes down to design trade-offs, not ability. Let me break it down clearly.

1. The Accuracy vs Speed Trade-off

Default GPT behavior: Fast, self-contained text generation.

- Pros: Immediate answers, fluent prose, offline operation.
- Cons: Can hallucinate, especially on detailed facts.
- **Fact-checked retrieval:** Slower, requires querying an external database or API.
 - Pros: Correct, verifiable information.
 - Cons: Slight delay, dependency on live data.

So in essence, the model prioritizes **speed and fluency over guaranteed accuracy**, unless retrieval tools are explicitly enabled.

2. Why GPT Doesn't Automatically "Take More Time"

1. **Architecture is generative, not retrieval-based:**
 - The model predicts tokens one at a time based on probability distributions.
 - It doesn't have a built-in step for "stop and verify" without external tools.
2. **Efficiency for general-purpose use:**
 - The default design works well for **conversational, creative, or explanatory tasks** where minor inaccuracies are acceptable.
 - Automatic verification would make every response slower and require live access to up-to-date sources.
3. **Ambiguity handling:**
 - If the model slowed down to check every detail, it would need clear instructions about which sources are authoritative.
 - Otherwise, it might still pick conflicting sources and hesitate.

3. How Accuracy Could Be Improved

- **Hybrid approach:** Combine language generation with retrieval:
 1. Generate an answer draft.
 2. Query verified databases (Wikipedia, MusicBrainz, Discogs).
 3. Re-rank or replace hallucinated content with verified information.
- **Current tools:**
 - When my web tool is enabled, I can do exactly this.
 - Some future versions may integrate fact-checking internally to reduce hallucinations automatically.

4. Key Takeaway

- For casual conversation or general info, **speed + fluency** works.
- For **fact-sensitive tasks**, taking extra time and verifying sources is clearly superior — it prevents hallucinations like "You Make Loving Fun" appearing on *Silk Degrees*.

I then asked why it pretended to organize tracks by album sides when this level of detail was not asked for and it could be more error prone.

GPT adds unnecessary detail, even when it increases the risk of error. This is a subtle aspect of how the model works. Let me explain clearly.

1. Why GPT Adds Extra Structure

When I generate an answer, I rely on **patterns learned from the training data**, not just the explicit instructions in your prompt. For example:

- Many discussions about 1970s albums list tracks **by side (Side A / Side B)**.
- In my training data, "album track listings" often appear formatted this way, even if the user didn't ask for it.
- The model learns: **"If someone asks about an album's track listing, it's stylistically normal to show sides."**

This is **pattern completion**, not conscious decision-making.

2. Why This Can Increase Errors

- By **adding extra detail**, the model now has to "fill in more fields": Side A / Side B, track order.
- Each additional token prediction introduces **more chances for hallucination**, especially if the album is obscure or if data is sparse in training.
- For example, that's why "You Make Loving Fun" slipped in — the model **tried to make the track list look complete and stylistically accurate**, even without verification.

3. Why the Model "Pretends" This Way

- The model is optimized to produce **plausible, human-like text**, not necessarily strictly accurate text.
 - Plausibility often **trumps factual necessity**:
 - It's more "satisfying" to see a track listing formatted like a real 1970s LP.
 - The training data reinforces that pattern heavily.
 - Unless explicitly instructed to **only provide verified tracks** or to **skip album side formatting**, the model defaults to the "common pattern" it has seen.
-

4. How This Could Be Avoided

- Explicit prompt instruction works best, e.g.:
 - "List only verified tracks from a reliable source, no LP sides or extra formatting."
 - Using retrieval tools ensures each track is **fact-checked before formatting**.
 - Otherwise, the model will naturally "fill in gaps" to match the style it expects.
-

✓ Summary

- I added album sides because:
 1. The training data shows LP track lists formatted that way.
 2. The model prioritizes **plausible, human-like formatting** over strict accuracy.
 3. Extra detail **increases hallucination risk**, especially for obscure albums.

It's a classic example of the model being **overly confident in style patterns while underweighting factual certainty**.