

Below is a detailed explanation of each prompting technique—

Zero-Shot

One-Shot

Few-Shot

Chain of Thought

Ensembling

Decomposition

Self-Criticism

These examples are tailored to the specific task of **music sonification of seismic waveform miniseed data**. Each explanation includes a definition, its application to the task, and a specific example prompt. The task involves converting seismic waveform data (in miniseed format, which contains time-series data of ground motion) into musical sounds, mapping features like amplitude or frequency to musical elements like pitch or rhythm.

1. Zero-Shot Prompting

- **Definition:** Zero-shot prompting involves asking the AI to perform a task without providing any examples. The AI relies entirely on its pre-trained knowledge to generate a response. This approach is useful for general tasks but may lack precision for specialized or nuanced tasks.
- **Application to Music Sonification:** For sonifying seismic waveform miniseed data, a zero-shot prompt asks the AI to describe or generate a sonification process based on its understanding of seismic data and music, without prior examples. The result may be generic or require refinement due to the complexity of mapping seismic data to musical parameters.

- **Example Prompt:**

Create a method to sonify seismic waveform data in miniseed format, mapping the data to musical elements like pitch, rhythm, and volume. Describe the process, including how to extract features from the miniseed data and translate them into a musical composition.

Expected Output: The AI might suggest a basic approach, such as mapping amplitude to volume and frequency to pitch, but it may not account for specific nuances of miniseed data (e.g., sampling rate or noise) without further guidance.

2. One-Shot Prompting

- **Definition:** One-shot prompting provides a single example of the desired input and output to guide the AI's response. This gives the model a reference point to align its output with the user's expectations, improving specificity over zero-shot.
- **Application to Music Sonification:** A one-shot prompt includes a single example of how seismic data features (e.g., amplitude,

frequency) are mapped to musical elements, helping the AI understand the expected format or style for sonifying miniseed data.

- **Example Prompt:**

I want to sonify seismic waveform data in miniseed format by mapping it to musical elements. Here's an example:

- *Input: A miniseed file with seismic amplitude data (range: -1000 to 1000) and frequency content (0.1–10 Hz).*
- *Output: Map amplitude to volume (higher amplitude = louder sound), frequency to pitch (higher frequency = higher pitch), and time to rhythm (1 second = 1 beat). Use a piano sound for the composition.*

Now, create a method to sonify a new miniseed file with similar data, mapping amplitude to volume, frequency to pitch, and time to rhythm, using a violin sound.

Expected Output: The AI generates a method that mirrors the example, mapping amplitude to volume, frequency to pitch, and time to rhythm, but uses a violin sound instead of a piano, ensuring consistency with the provided example.

3. Few-Shot Prompting

- **Definition:** Few-shot prompting provides multiple examples (typically 2–5) of input-output pairs to clarify the task and desired output format. This approach is highly effective for complex tasks, as it gives the AI a robust reference for consistency and style.
- **Application to Music Sonification:** For seismic data sonification, few-shot prompting provides several examples of how different miniseed data features are mapped to musical elements, allowing the AI to generalize and apply similar mappings to new data.

- **Example Prompt:**

I want to sonify seismic waveform data in miniseed format by mapping it to musical elements. Here are three examples:

- 1. Input: Miniseed file with amplitude (-500 to 500), frequency (0.5–5 Hz).
Output: Amplitude to volume (linear scaling, max amplitude = 100% volume), frequency to pitch (0.5 Hz = C4, 5 Hz = C6), time to rhythm (1 second = 1 quarter note), using a flute sound.*
- 2. Input: Miniseed file with amplitude (-1000 to 1000), frequency (0.1–10 Hz).
Output: Amplitude to volume (logarithmic scaling), frequency to pitch (0.1 Hz = A3, 10 Hz = A6), time to rhythm (0.5 seconds = 1 eighth note), using a piano sound.*
- 3. Input: Miniseed file with amplitude (-200 to 200), frequency (1–8 Hz).
Output: Amplitude to volume (linear scaling), frequency to pitch (1 Hz = G4, 8 Hz = G6), time to rhythm (2 seconds = 1 half note), using a guitar sound.*

Now, create a method to sonify a new miniseed file with amplitude (-800 to 800) and frequency (0.2–6 Hz), mapping amplitude to volume, frequency to pitch, and time to rhythm, using a cello sound.

Expected Output: The AI generates a method that maps amplitude to volume (likely linear or logarithmic scaling, based on examples), frequency to pitch (e.g., 0.2 Hz = D4, 6 Hz = D6), and time to rhythm (e.g., 1 second = 1 quarter note), using a cello sound, consistent with the provided examples.

4. Chain of Thought Prompting

- **Definition:** Chain of Thought (CoT) prompting encourages the AI to explicitly reason through a task step-by-step before providing a final answer. This improves performance on complex tasks by ensuring logical progression, especially for non-reasoning models.
- **Application to Music Sonification:** For sonifying seismic data, CoT prompting asks the AI to outline its reasoning process (e.g., how it selects mappings, handles data preprocessing) before proposing a sonification method. This ensures clarity and catches potential errors in complex mappings.

Example Prompt:

I want to sonify seismic waveform data in miniseed format, mapping amplitude to volume, frequency to pitch, and time to rhythm, using a synthesizer sound.

Think step-by-step through the process, explaining how you would:

- 1. Read and preprocess the miniseed data.*
- 2. Map amplitude to volume.*
- 3. Map frequency to pitch.*
- 4. Map time to rhythm.*
- 5. Generate the musical output.*

Then, provide the final sonification method.

Step 1: Read and preprocess miniseed data. Parse the miniseed file to extract amplitude (e.g., -1000 to 1000) and frequency content (e.g., 0.1–10 Hz) using a library like ObsPy. Remove noise with a bandpass filter (0.1–10 Hz).

Step 2: Map amplitude to volume. Use linear scaling: amplitude -1000 = 0% volume, 1000 = 100% volume.

Step 3: Map frequency to pitch. Map 0.1 Hz to C3 (130.81 Hz) and 10 Hz to C6 (1046.50 Hz) using a logarithmic scale for perceptual smoothness.

Step 4: Map time to rhythm. Assign 1 second of data to 1 quarter note at 60 BPM.

Step 5: Generate musical output. Use a synthesizer sound in a DAW (e.g., Ableton Live) to create notes based on the mapped pitch, volume, and rhythm.

Final Method: Parse miniseed data with ObsPy, filter noise (0.1–10 Hz), map amplitude (-1000 to 1000) to volume (0–100%), frequency (0.1–10 Hz) to pitch

(C3–C6, logarithmic), and time (1 second = 1 quarter note). Output as a synthesizer-based composition.

Expected Output:

1. Music Producer: Map amplitude (-1000 to 1000) to volume (0–100%, logarithmic for dynamic range), frequency (0.1–10 Hz) to pitch (C4–C7, linear) for melodic appeal, and time (1 second = 1 eighth note) for upbeat rhythm. Use a piano sound for accessibility.
2. Seismologist: Map amplitude to volume (linear scaling for data fidelity), frequency to pitch (0.1 Hz = A2, 10 Hz = A5, logarithmic) to reflect seismic patterns, and time (1 second = 1 quarter note) for precision. Use a sine wave for clarity.
3. Sound Designer: Map amplitude to volume (non-linear for dramatic effect), frequency to pitch (0.1–10 Hz to random dissonant chords) for experimental texture, and time (variable rhythm based on amplitude peaks). Use granular synthesis.

Final Method: Map amplitude (-1000 to 1000) to volume (logarithmic scaling), frequency (0.1–10 Hz) to pitch (C4–C7, logarithmic), and time (1 second = 1 quarter note). Use a piano sound with subtle granular synthesis for engagement and clarity.

6. Decomposition Prompting

- **Definition:** Decomposition prompting breaks a complex task into smaller subproblems, which the AI solves individually before addressing the main task. This improves performance by structuring the problem-solving process.
- **Application to Music Sonification:** For seismic data sonification, decomposition involves identifying subproblems like data preprocessing, feature extraction, and musical mapping, solving each before combining them into a cohesive sonification method.

- **Example Prompt:**

I want to sonify seismic waveform data in miniseed format, mapping amplitude to volume, frequency to pitch, and time to rhythm, using a harp sound. First, list the subproblems that need to be solved to create this sonification. Then, solve each subproblem step-by-step. Finally, combine the solutions into a complete sonification method.

Expected Output:

Subproblems:

1. Read and preprocess miniseed data to extract amplitude and frequency.
2. Determine a mapping for amplitude to volume.
3. Determine a mapping for frequency to pitch.
4. Determine a mapping for time to rhythm.
5. Generate a musical composition with a harp sound.

Solutions:

1. Preprocessing: Use ObsPy to parse miniseed data, extracting amplitude (-800 to 800) and frequency (0.2–6 Hz). Apply a bandpass filter to reduce noise.
2. Amplitude to Volume: Map amplitude linearly (-800 = 0% volume, 800 = 100% volume).
3. Frequency to Pitch: Map frequency logarithmically (0.2 Hz = E4, 6 Hz = E6) for perceptual accuracy.
4. Time to Rhythm: Assign 1 second of data to 1 quarter note at 60 BPM.
5. Musical Output: Use a harp sound in a DAW to generate notes based on mapped parameters.

Final Method: Parse miniseed data with ObsPy, filter noise (0.2–6 Hz), map amplitude (-800 to 800) to volume (0–100%, linear), frequency (0.2–6 Hz) to pitch (E4–E6, logarithmic), and time (1 second = 1 quarter note). Output as a harp-based composition.

7. Self-Criticism Prompting

- **Definition:** Self-criticism prompting asks the AI to review its initial response, identify flaws or areas for improvement, and revise the output. This enhances quality by mimicking human reflection.
- **Application to Music Sonification:** For seismic data sonification, self-criticism involves generating an initial sonification method, critiquing it for issues (e.g., unclear mappings, inappropriate sound choice), and improving it.

- **Example Prompt:**

I want to sonify seismic waveform data in miniseed format, mapping amplitude to volume, frequency to pitch, and time to rhythm, using a clarinet sound. Provide an initial sonification method. Then, critique the method for potential issues (e.g., data accuracy, musical coherence). Finally, provide an improved method based on the critique.

Expected Output:

Initial Method: Parse miniseed data to extract amplitude (-500 to 500) and frequency (0.5–5 Hz). Map amplitude to volume (linear, -500 = 0%, 500 = 100%), frequency to pitch (0.5 Hz = C4, 5 Hz = C5), and time (1 second = 1 quarter note). Use a clarinet sound in a DAW.

Critique: The frequency-to-pitch mapping (C4–C5) spans only one octave, which may limit musical expressiveness. Linear amplitude mapping may overemphasize small changes, reducing dynamic contrast. The clarinet sound may not suit seismic data's continuous nature, as it's better for discrete notes.

Improved Method: Parse miniseed data with ObsPy, extracting amplitude (-500 to 500) and frequency (0.5–5 Hz). Map amplitude to volume (logarithmic

scaling for smoother dynamics), frequency to pitch (0.5 Hz = C4, 5 Hz = C6, two octaves for expressiveness), and time (1 second = 1 quarter note at 60 BPM). Use a sustained clarinet-like synth sound for continuity.

Summary of Prompting Techniques

Technique	Key Feature	Example Focus (Seismic Sonification)
Zero-Shot	No examples, relies on pre-trained knowledge	General method to map seismic data to music.
One-Shot	One example for guidance	Method mirroring a single example with a violin sound.
Few-Shot	Multiple examples for consistency	Method based on three examples, using a cello sound.
Chain of Thought	Step-by-step reasoning	Detailed reasoning for mappings, using a synthesizer.
Ensembling	Multiple prompts/models, select best	Combine methods from different perspectives (e.g., producer, seismologist).
Decomposition	Break task into subproblems	Solve preprocessing, mappings, and output separately.
Self-Criticism	Critique and revise initial response	Refine a clarinet-based method for better coherence.

Additional Notes

- **Task Complexity:** Music sonification of seismic data is complex due to the need to preprocess miniseed data (e.g., using ObsPy), map numerical features to musical parameters, and ensure the output is musically coherent. Techniques like few-shot, decomposition, and chain of thought are particularly effective for this task.
- **Choosing a Technique:**
 - **Conversational Use:** Zero-shot or one-shot for quick exploration; few-shot or self-criticism for refined outputs.
 - **Product-Focused Use:** Decomposition, chain of thought, or ensembling for robust, scalable sonification systems.

- **Implementation:** For real-world sonification, you might combine techniques (e.g., few-shot with decomposition) to provide examples and structure the task systematically.