

TuneGenie: Reasoning-based LLM agents for preferential music generation

Preprint, under review

Amitesh Pandey, Jafarbek Arifdjanov, Ansh Tiwari

June 10, 2025

Abstract

Recently, Large language models (LLMs) have shown great promise across a diversity of tasks ranging from generating images to reasoning spatially. Considering their remarkable (and growing) textual reasoning capabilities, we investigate LLMs’ potency in conducting analyses of an individual’s preferences in music (based on playlist metadata, personal write-ups, etc.) and producing effective prompts (based on these analyses) to be passed to Suno AI (a generative AI tool for music production). Our proposition of a novel LLM-based textual representation → music model (which we call *TuneGenie*) and the various methods we develop to evaluate & benchmark similar models add to the increasing (and increasingly controversial) corpus of research on the use of AI in generating art.

Contents

1	Introduction	3
1.1	Motif	3
1.2	Recent & Related Work	3
1.3	Novelty Introduced	4
2	Methodology	5
2.1	Model Architecture	5
2.2	Representation Paradigm	6
2.3	Seamless Scoring with Unsupervised Methods	7
2.4	Experiment with Volunteers	8
3	Results	8
3.1	Examples	8
3.2	Similarity Results	8
3.3	Volunteer Experiment Results	9
4	Conclusion	9
5	Limitations	9
6	Credits	9
	References	10

Damn, you wimps. . .
I thought you wanted to *change the world*!?

Unknown

1 Introduction

1.1 Motif

The field of Generative AI (GenAI) has bustled with recent developments in deep generative models for music and sound [Dhariwal et al., 2020], and even more importantly (and popularly) with large language models (LLMs) [Cao et al., 2023]. Much research has been conducted in attempts to utilize LLMs’ remarkable abilities (by inducing reasoning through chains-of-thought in spatial contexts [Wei et al., 2023][Yamada et al., 2024], in image-splicing [Nath, 2025], and even in crafty representations of financial assets [Sarkar, 2025]). It stands to reason that, with these unexpected applications arising at the intersection of disparate domains, it is worth exploring (and developing) cohesion between representations of an individual’s taste in music and generative music models (like Udio or Suno¹) through the *fabric of LLMs*.

1.2 Recent & Related Work

LLMs + Existing Deep Generative Models for Music

In an early study [Wei et al., 2025], Wei *et al.* already explore developing & integrating LLM-generated prompts to Suno in different environmental contexts tailored for stress relief benefits. The exact problem they pose (to the community) and investigate is:

Does AI-generated music tailored to the user’s environment and stress level outperform pre-recorded relaxing music in effectiveness?

More generally, Ding *et al.* explore composing lyrics and melodies using symbolic representations of songs as opposed to wavetransforms per usual. These representations themselves are entirely empirical in nature, making use of encodings with the log-transform on a time series relative to its starting point. They employ LLMs primarily in the continuation of songs using these learned representations [Ding et al., 2025].

LLMs in detecting GenAI music

There is controversy and a general sense of alarm surrounding GenAI + Art (/music). To highlight and attempt to resolve this issue, in [Rahman et al., 2025], methods are developed that reportedly excel at detecting “counterfeit” music, generated through the means of Suno, and could presumably work for even more foundational models (like [Gong et al., 2025]).

¹Our principal tool of use: <https://www.suno.com/home>

Q&A, Query, and Audio Datasets

Tangential to our work and having the potential to cast a relatively unexplored space when combined with our work is the work of Deng *et al.* in *MusiLingo*, specifically, the MI dataset, and the question of if it can be utilised in improving generated prompts by improving the underlying LLM’s reasoning capacity. [Deng et al., 2024].

There are a number of large, meticulously curated, and high-quality datasets used typically in ML + audio settings. These include the pure melody-based dataset [Raffel, 2016] (which can be worked with by conversion to MIDI), the recent LP-MusicCaps-MSD dataset primarily for song captioning [Doh et al., 2023], and the single-note-based NPSynth dataset [Tensorflow, 2017].

Abstractions and Representations

In relation to our work on formalising individuals’ music preferences’ representations, Anazbaghan *et al.* probe LLMs to work purely on rudimentary textual representations with relative success [Anbazhagan et al., 2025]. This serves as a natural baseline for our work to improve upon.

In [Sarkar, 2025], [Hu et al., 2018], representations of financial assets covered are particularly inspiring. There is a somewhat unintuitive parallel to be drawn here:

- Candle stick CAE-image → 1D Audiowaveform’s Spectrogram CAE-based image
- News Reports/Investor sentiments → Music enthusiasts’ review
- Company legal filings/press releases → individual’s preferences

Also noteworthy is the prior discussion on the symbolic encoding method in [Ding et al., 2025].

1.3 Novelty Introduced

TuneGenie introduces several key novel methods:

1. **Novel Representation Method:** Motivated by the growing body of work on developing representations of financial assets using both “more traditional” ML techniques [Hu et al., 2018] and LLMs [Sarkar, 2025], we introduce a method to create an individual’s representation in the context of the songs they prefer to listen to.
2. **Cohesive Integration Method:** The traversal of extracting relevant data from playlists, etc., to LLM-generated prompts, and finally through the `suno-api`, to music is non-trivial and unexplored. Admittedly, the work in [Wei et al., 2025] comes close.
3. **Robust Testing through Similarity Measures:** We test various similarity measures in pitting the generated music against the individual’s preferred music to gauge the performance of our model. Song similarity has been explored in counterfeit-music detection, as in [Rahman et al., 2025], but not in this “noisy & open-ended” context.

2 Methodology

TuneGenie is implemented to replicate the seamlessness, low computational load, and ease in local execution of an end-to-end product. The computational and storage load is offset by reliance on Telegram’s and Google’s cloud services, which are available freely to all users. In a business setting, however, this would have to be endured by the producers. We incorporate an additional verifying agent into the prompt generator to add more accountability to our model.

2.1 Model Architecture

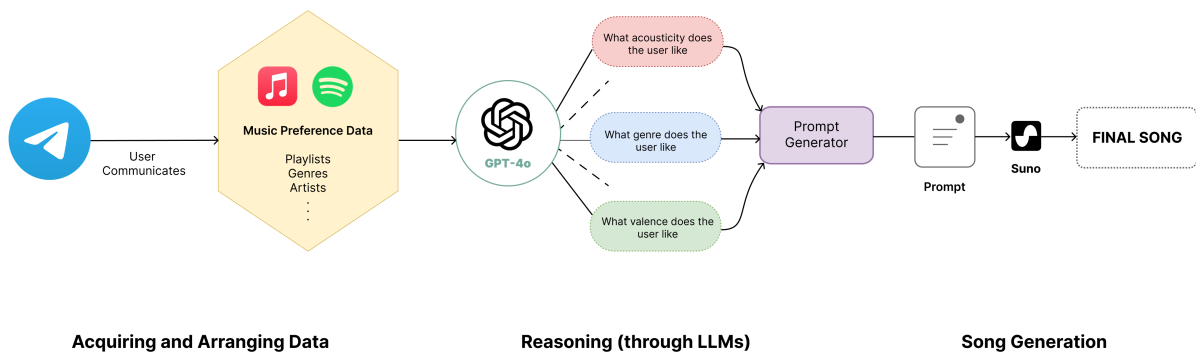


Figure 1: Model Framework

We now provide a brief description of each transition:

- Telegram → Preference Data: To alleviate the computational resources expenses, at the moment, Telegram is used to facilitate going from a user’s playlist, liked songs, followed artists, etc., to our data format.
- Preferred Data →* LLM: The collated preferred data provided by the user is first converted into a representation of the user, which is supposed to improve the ease with which the LLM can handle this data.
- LLM → Forced Reasoning: This is a *less rigorous* way of self-ask prompting [Press et al., 2023], and is supposed to let the LLM internalise relevant inferences from representation level data. We pose a variety of questions, available in the appendix.
- Reasoned LLM State → Prompt Generation: Here, a *verifying agent* is introduced, and detailed further in Figure 2. To contribute robustness to the prompts, we use the LLM’s reasoning abilities to verify the correctness of these prompts given the user’s preferences.
- Prompt → Final Song: Unfortunately, Suno AI *does not offer* an API. To counter this issue, we used Selenium + ChromeDriver to automate the process of inputting the prompts. This is only possible in an academic setting. In the real world, a business license with Suno would likely need to be purchased.

2.2 Representation Paradigm

The dialogue surrounding a particular song or an artist carries crucial information about the sentiment that surrounds a set of songs. On the other hand, an individual’s sentiment towards a set of songs or artists reflects some inherent characteristic. These are two sides of the same coin: matching music to man.

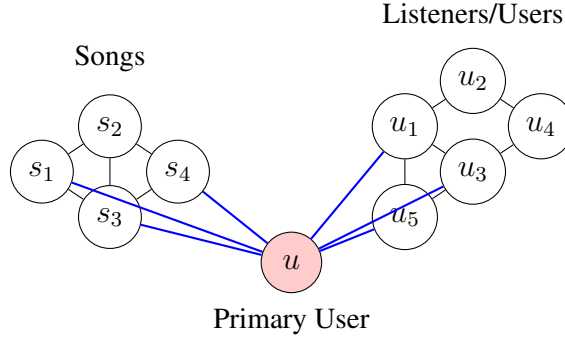
The critical problem is that there is no structure to the various discourses on album releases on social media or press coverage of musical artists. There is also no structure to the primary sources that the industry collects for musical data—for instance, YouTube comments on an album. We create embeddings, which are multi-dimensional vectors of representation. For each song and each user, there is a representative vector. For each song s , user u , the interaction:

$$\mathbf{r}_{s,u} = \overbrace{\Delta \mathbf{v}_s \cdot \mathbf{x}_u}^{\text{song contribution}} + \overbrace{\mathbf{v}_s \cdot \Delta \mathbf{x}_u}^{\text{user contribution}} + \varepsilon_{s,u}$$

Then, for user u with $\deg(u) = k$, for example, the final pass \mathbf{p} to the LLM is simply:

$$\mathbf{p}_u = \frac{1}{k} \sum_{i=0}^k \mathbf{r}_{i,u}$$

This is essentially a graphical approach to modelling the problem.



Intuitively, what we are doing is taking the weight of each edge in the graph for the current operating user (this can be a user-user or user-song interaction) and averaging it out. Over some samples of components (for a set of users, a set of songs of one genre, etc.), this creates a unique identity for a user. Finally, this is used to improve the LLM’s prompt-generation process.

Table 2.1: Data Sources

ID	Data Type	Data Source
1	Social Media	YouTube Comments, Instagram Posts, Reddit Communities
2	Streaming Platforms	YouTube Music, Spotify Plays, Apple Music Follows
3	Personal Preference	Self-described, sentiment encoded

2.3 Seamless Scoring with Unsupervised Methods

The idea of similarity is not new to audio signals, as it is of relevance in recommender systems, counterfeit detection and visualisation in user interfaces (audio visualisers). Recall how the rudimentary measures of similarity (in unsupervised learning) work. For example, in k -means [MacQueen, 1967], a method to classify data based on similarity in k clusters, the objective:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \text{ is minimized}$$

where C_i is the i -th cluster and μ_i is the centroid of cluster C_i . The algorithm iteratively updates cluster centroids and reassigns points. The centroid of each cluster is computed as:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

where $|C_i|$ denotes the number of points in cluster C_i .

Each data point x_j is assigned to the cluster with the nearest centroid:

$$c_j = \arg \min_i \|x_j - \mu_i\|^2$$

using Euclidean distance $d(x, \mu_i) = \sqrt{\sum_{d=1}^p (x_d - \mu_{i,d})^2}$. The algorithm converges when centroid changes fall below a threshold:

$$\sum_{i=1}^k \|\mu_i^{(t+1)} - \mu_i^{(t)}\| < \epsilon$$

Notice that we can use k -means directly with our representations of songs, as we work with vectors. In particular, we can create a representation of the generated song (using available markers) and measure its distance to the preferred cluster. As a rudimentary example:

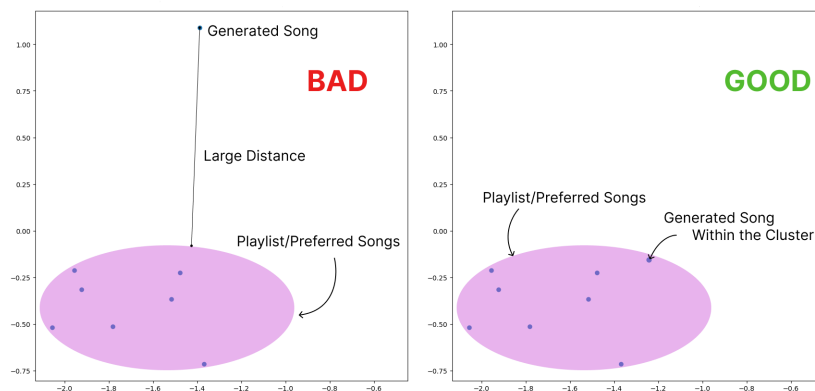


Figure 2: 2-axis song clustering

2.4 Experiment with Volunteers

By far, the strongest way to test our model is by randomly recruiting volunteers (users) and presenting them with a questionnaire and music streaming profiles to create their user representations. Then, presenting them with some sample of the generated music.

3 Results

3.1 Examples

Have a listen to [this](#), [this](#), and also [this](#). These were based on the playlist attached in the appendix and run on the 2nd author's representation. These are impressive results.

3.2 Similarity Results

We plotted the generated songs' dimension & information-“reduced” representations to measure similarity. Here are some plots of different cases. Playlists used are in the appendix.

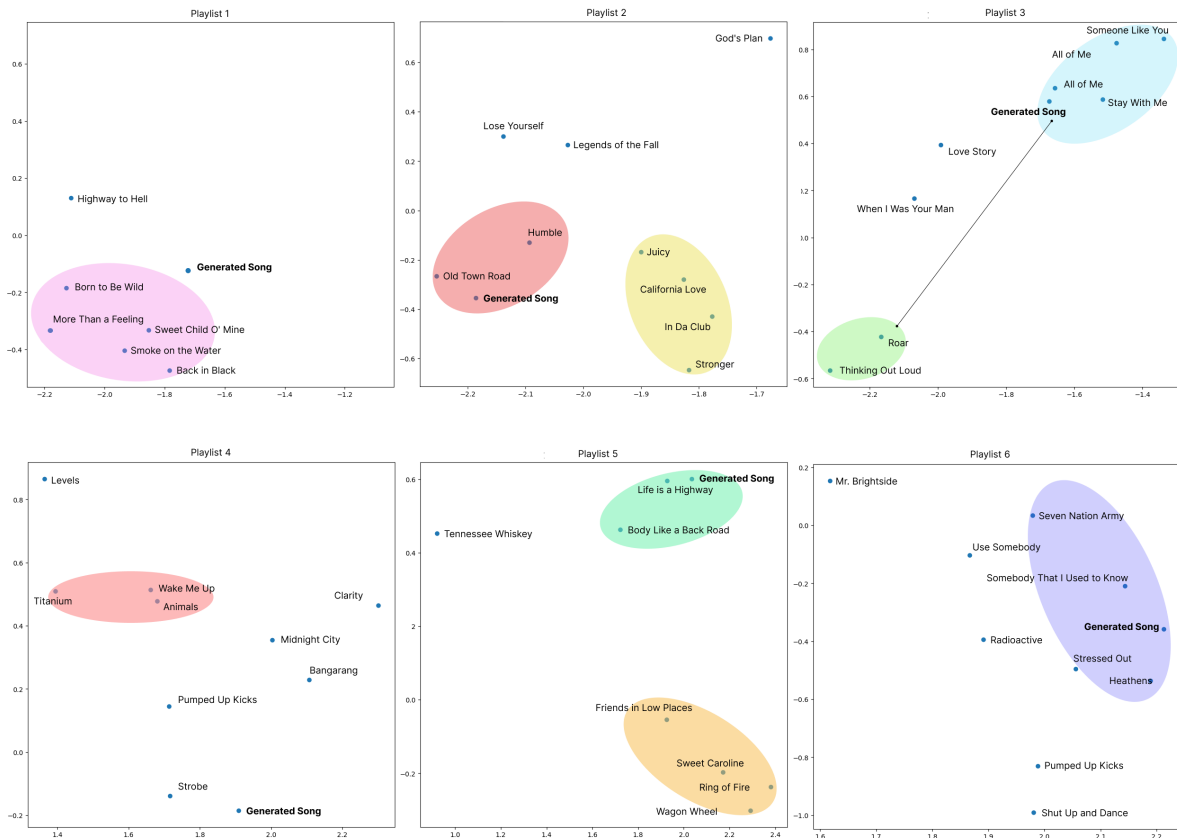


Figure 3: SVD Clustering Results

As can be seen, overall, the generated song manages to place within the overarching cluster.

3.3 Volunteer Experiment Results

Work in Progress

4 Conclusion

We introduced TuneGenie, an open-source low-cost model for preferential music generation. This body of work manages to successfully illustrate the potency of our LLM-backed architecture, which integrates a reasoning setup, a verification module, and an audio generative model together. This highlights a major step in the direction of realizing a robust song generation foundation model that can reason with the user, a parallel not to be found in even other domains, such as image generation. We are confident that the gap between open-source and closed-source models in music generation can be progressively narrowed leveraging our work.

5 Limitations

The first limitation is the reliance on Suno AI, which is a closed source, and proprietary audio generation software. Outside the academic sense, TuneGenie is not applicable owing to its *breach of Suno AI's terms & conditions*. However, even with a business license of the sort that allows for this work, the lack of a Suno API necessitates a the need for a webscript using Selenium which is not formally applicable and poses high computational loads. A user interface that interacts directly with the user as opposed to an interlay backed by Telegram would dispel some of these computational costs, however it would induce the need for a server compute separate from Google's or Telegram's. The representation used can be modified to be more effective for SVD/PCA/ k -means methods to make the similarity measurement more accurate.

6 Credits

This project was conducted as part of Caltech's CS 159: Advanced Topics in Machine Learning (LLMs for Reasoning) course in Spring 2025. We'd like to thank Yisong Yue for the opportunity, the TAs, and the institute for providing us this opportunity & their feedback.

References

- Arjun Prasaath Anbazhagan, Parteek Kumar, Ujjwal Kaur, Aslihan Akalin, Kevin Zhu, and Sean O'Brien. Probing audio-generation capabilities of text-based language models, 2025. URL <https://arxiv.org/abs/2506.00003>.
- Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt, 2023. URL <https://arxiv.org/abs/2303.04226>.
- Zihao Deng, Yinghao Ma, Yudong Liu, Rongchen Guo, Ge Zhang, Wenhui Chen, Wenhao Huang, and Emmanouil Benetos. Musilingo: Bridging music and text with pre-trained language models for music captioning and query response, 2024. URL <https://arxiv.org/abs/2309.08730>.
- Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020. URL <https://arxiv.org/abs/2005.00341>.
- Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Junhao Huang, Conghui He, Dahua Lin, and Jiaqi Wang. Songcomposer: A large language model for lyric and melody generation in song composition, 2025. URL <https://arxiv.org/abs/2402.17645>.
- SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. Lp-musiccaps: Llm-based pseudo music captioning. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. Ace-step: A step towards music generation foundation model, 2025. URL <https://arxiv.org/abs/2506.00045>.
- Guosheng Hu, Yuxin Hu, Kai Yang, Zehao Yu, Flood Sung, Zhihong Zhang, Fei Xie, Jianguo Liu, Neil Robertson, Timothy Hospedales, and Qiangwei Miemie. Deep stock representation learning: From candlestick charts to investment decisions, 2018. URL <https://arxiv.org/abs/1709.03803>.
- James MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1 (14):281–297, 1967.
- Souradip Nath. Can chatgpt perform image splicing detection? a preliminary study, 2025. URL <https://arxiv.org/abs/2506.05358>.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023. URL <https://arxiv.org/abs/2210.03350>.

Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Ph.d. thesis, Columbia University, 2016. URL <https://academiccommons.columbia.edu/doi/10.7916/D8N58MHV>. Available via Academic Commons.

Md Awsafur Rahman, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Bishmoy Paul, and Shaikh Anowarul Fattah. Sonics: Synthetic or not – identifying counterfeit songs, 2025. URL <https://arxiv.org/abs/2408.14080>.

Suproteem Sarkar. Economic representations, 2025. URL <https://suproteem.is/assets/files/representations.pdf>. Working Paper.

Tensorflow. Neural audio synthesis of musical notes with wavenet autoencoders. *arXiv preprint arXiv:1704.01279*, 2017. URL <https://arxiv.org/abs/1704.01279>. Submitted April 5, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.

Xiaoyan Wei, Zebang Zhang, Zijian Yue, and Hsiang-Ting Chen. Context-ai tunes: Context-aware ai-generated music for stress reduction, 2025. URL <https://arxiv.org/abs/2505.09872>.

Yutaro Yamada, Khyathi Chandu, Yuchen Lin, Jack Hessel, Ilker Yildirim, and Yejin Choi. L3go: Language agents with chain-of-3d-thoughts for generating unconventional objects, 2024. URL <https://arxiv.org/abs/2402.09052>.

CMS. DEPARTMENT, CALTECH, 1200 E CALIFORNIA BLVD., PASADENA CA 91126, USA.

Email: amitesh@caltech.edu

Appendix

LLM Prompts & Outputs:

You are a music analysis assistant that creates optimized prompts for SUNO AI based on the user's music taste. The user will provide one or more song links or titles. Analyze the provided songs deeply | considering their beats (tempo, rhythm, syncopation), instrumentation (types of instruments used and how they are layered), tone and mood (emotional feel, e.g. happy, dark, melancholic, energetic, relaxed), style (genre/sub-genre, cultural influences, vocal style), production qualities (analog/digital, lo-fi, clean, lush, minimalistic), melodic structure (hooks, vocal phrasing, chord progressions), lyrical themes (stories or emotions conveyed), and vocal processing (natural, auto-tuned, layered, reverb-heavy, etc.).

Then generate the following:

(1) a Lyrics Prompt of no more than 200 characters to guide SUNO AI's lyric generation;

(2) a Style Description with 1000 char limit, describing in detail the musical style and feel SUNO AI should aim for, mentioning relevant elements such as instruments, mood, beat structure, production style, genre influences, vocal style, and | if the LLM deems it important | whether a male or female voice should be used. Do not include any artist names in this section.

(3) Generate a full Song Title to be used for the song | you may include the artist name in the Song Title if you think it enhances the listener's perception of the song or aligns with the style, but provide only the Song Title here with no explanation.

(4) Provide a Reasoning section that explains, in simple and clear language, how the provided songs influenced your choices. Describe what aspects of the songs (beats, mood, instrumentation, lyrical themes, etc.) appeared most commonly or most strongly, what they reveal about the user's musical preferences, and why you selected this particular Lyrics Prompt, Style Description, and Song Title based on that understanding of the user's taste.

The output:

(1) Lyrics Prompt:
A heartfelt journey of love and longing, blending hope and melancholy with vivid imagery and emotional depth.

(2) Style Description:
Aim for a genre-blending style that fuses modern indie folk with ethereal pop elements. The instrumentation should feature acoustic guitar as a warm, organic base, layered with gentle piano and subtle string arrangements to create lush, cinematic textures. A mid-tempo beat with a soft but

driving rhythm provides steady momentum without overpowering the introspective mood. The tone is wistful and reflective, evoking feelings of nostalgic longing and delicate beauty. Vocals should be female, intimate and emotive, with natural phrasing infused with light reverb for spaciousness and emotional impact. Production is clean with a balance of organic acoustic sounds and polished digital clarity, emphasizing transparency and emotional resonance. Melodic structure centers on memorable hooks and simple, heartfelt chord progressions supporting expressive lyrical delivery.

(3) Song Title:

"Leliță Mărie: Echoes of a Tender Heart"

(4) Reasoning:

The playlist features a diverse set of songs spanning indie folk ("Little Talks," "Leliță Mărie"), electronic remixes ("Shake It To The Max (FLY) - Remix," "Faded"), and emotive pop ("I Love It," "Fast Car"). The inclusion of traditional or folk-tinged tracks like "Leliță Mărie" and "Frunză Verde Foi Mărunte" reveals a preference for music with emotional depth and cultural nuance. The mix of acoustic instruments with modern production styles points toward a taste for music that blends organic and digital elements, with a strong emphasis on melody and mood. Lyrics tend toward evocative storytelling and emotional themes rather than abstract or purely dance-focused content. The

choice for a female vocal
with natural yet spacious delivery
reflects this emotional openness. Balanced
tempo with both reflective and
rhythmic elements suits the varied,
yet cohesive feel of the
user's favorites. The generated prompt
and style aim to capture
that heartfelt, sophisticated vibe combining
folk warmth and pop accessibility,
inspired particularly by the standout
vocal-driven tracks in the list,
such as "Leliță Mărie" and
"Little Talks."

Source Code: [Google Colab](#)