MusicSem: A Semantically Rich Language-Audio Dataset of Organic Musical Discourse

Rebecca Salganik[1], **Teng Tu**[2], **Fei-Yueh Chen**[1], **Xiaohao Liu**[2], **Kaifeng Lu**[1], **Ethan Luvisia**[1]
Zhiyao Duan[1], **Guillaume Salha-Galvan**[3], **Anson Kahng**[1], **Yunshan Ma**[4], **Jian Kang**[1]
[1]University of Rochester, [2]National University of Singapore
[3]SJTU Paris Elite Institute of Technology, [4]Singapore Management University
rsalgani@ur.rochester.edu

Abstract

We present MusicSem, a dataset of 32,493 language–audio music descriptions derived from organic discussions on Reddit. What sets MusicSem apart is its focus on capturing a broad spectrum of musical semantics, reflecting how listeners naturally describe music in nuanced, human-centered ways. To structure these expressions, we propose a taxonomy of five semantic categories: descriptive, atmospheric, situational, metadata-related, and contextual. Our motivation for releasing MusicSem stems from the observation that music representation learning models often lack sensitivity to these semantic dimensions, due to the limited expressiveness of existing training datasets. MusicSem addresses this gap by serving as a novel semantics-aware resource for training and evaluating models on tasks such as cross-modal music generation and retrieval.

1 Introduction

Table 1: Categorization of different caption elements.

Category	Description	Example
Descriptive	concrete musical attributes	"I like the high pass filter on the vocals in the chorus, really makes harmonies pop."
Contextual	other songs	"Sabrina Carpenter's *Espresso* is just a mix of old Ariana Grande and 2018 Dua Lipa."
Situational	an activity or environment	"I listened to this song on the way to quitting my sh**ty corporate job."
Atmospheric	emotions and expressive adjectives	"This song makes me feel like a manic pixie dream girl in a bougie coffeeshop."
Metadata-related	technical & background information	"This deluxe edition of this song was released in 2013 and it has three bonus hiphop tracks."

Music representation learning is central to music information retrieval and generation [1, 2]. While prior work has primarily focused on audio-centric models [3, 4, 5, 6], recent advances in multimodal learning, particularly in aligning text and audio, have enabled progress in tasks such as cross-modal retrieval [7, 8, 9], music-to-text generation [10, 11, 12], and text-to-music generation [13, 14, 15, 16]. However, recent work has shown that multimodal models often fail to capture the user's expressed intent in text descriptions of music [17, 18]. This interpretation gap suggests that the language-audio datasets used to train these models may not fully reflect the broader and more natural forms of human discourse.

In this paper, we begin by formalizing the notion of musical semantics and introducing a taxonomy that distinguishes five types of music captions. We then confirm that many state-of-the-art generative and retrieval models lack sensitivity to these semantic distinctions, particularly variations in atmosphere, context, situational cues, and metadata-related aspects of user intent. Motivated by this observation, we introduce MusicSem, a semantically rich language-audio dataset derived from organic music discussions on the social media platform Reddit. The dataset comprises 32,493 language-audio music description pairs, with textual annotations that express not only descriptive attributes of the music, but also emotional resonance, contextual and situational usage, and co-listening patterns. MusicSem

Table 2: Semantic sensitivity analysis in text-to-music generative models. Best performance is highlighted in **bold**, second best in underline. The superscripts ^{d}, ^{a}, ^{s}, ^{m}, ^{c} refer to descriptive, atmospheric, situational, metadata, and contextual, respectively.

Model	G^d	G^a	G^s	G^m	G^c
AudioLDM2	0.68	0.37	0.35	0.40	0.34
MusicLM	0.50	0.36	0.42	0.39	0.35
Mustango	0.62	0.27	0.25	0.26	0.32
MusicGen	0.57	0.47	0.39	0.47	0.52
Stable Audio	**0.72**	**0.67**	**0.68**	**0.70**	**0.74**

Table 3: Semantic sensitivity analysis on cross modal retrieval models. Best performance is highlighted in **bold**, second best in underline. The superscripts ^{d}, ^{a}, ^{s}, ^{m}, ^{c} refer to descriptive, atmospheric, situational, metadata, and contextual, respectively. We set K=10.

Model	R^d	R^a	R^s	R^m	R^c
LARP	**0.98**	0.17	0.06	0.0	**0.56**
CLAP	0.95	0.52	0.35	0.42	0.52
ImageBind	0.84	0.39	0.35	0.38	0.41
CLaMP3	0.92	**0.58**	**0.49**	**0.62**	0.55

distinguishes itself by capturing a broader spectrum of musical semantics than prior datasets used for multimodal model training. MusicSem also serves as a novel semantics-aware resource for benchmarking cross-modal retrieval and generation models. The accompanying MusicSem website, including access to the full dataset, detailed documentation, and source code for data construction will be released upon acceptance.

2 Capturing Music Semantics

One of the goals in language-audio music understanding tasks is the design of models which are able to capture the nuances that contextualize a listening experience. We organize these contextual elements into five major categories, which we term *music semantics* [19, 20, 21]. Then, we highlight the importance of *music semantics* in language-audio datasets by quantifying the semantic sensitivity in a wide range of generative and retrieval models.

Categorization of music semantics. Consider the following two prompts: *"This song is a ballad. It contains guitar, male vocals, and a piano. It sounds like something I would listen to at church"* or *"This song is a ballad. It contains guitar, male vocals, and a piano. It sounds like something I would listen to while tripping on acid"*. While their descriptions of musical attributes (e.g., ballad, guitar, male vocals, piano) remain the same, the change in the situational context (listen to at church vs. while tripping on acid) should drastically change our expectations for the associated audio in generative and retrieval settings. To this end, we present a comprehensive formal categorization of music semantics, including (1) descriptive elements to describe the musical attributes of a song, (2) contextual elements that highlight other songs that are similar to a song or might be co-listened together, (3) situational elements to describe an activity or environment in which a song is listened to, (4) atmospheric that express the emotions a song evokes or other expressive adjective of a song, and (5) metadata that provides technical and background information of a song and/or its corresponding artist. An example for each category is presented in Table 1.

Insensitivity to varying semantic context. We can then quantify the sensitivity of multimodal music understanding models to varying semantic elements. Given any i-th language-audio pair (t_i, a_i) in a language-audio dataset, we construct a counterfactual annotation $\tilde{t}_i^c$ by changing descriptions with respect to a semantic category c, e.g., while at church vs. while tripping on acid in the aforementioned example. We randomly sampled 50 language-audio pairs in ***MusicCaps*** and create a counterfactual example with respect to each semantic category present in each language-audio pairs[1].We can then present two metrics to assess sensitivity of a model to semantic shifts. For text-to-music generation, we define

$$G^c = \frac{1}{n}\left[\sum_{i=1}^{n} 1 - \text{cosine}(f_i, \tilde{f}_i^c)\right], \tag{1}$$

[1]We release the dataset construction code `https://tinyurl.com/musicsem-code` and full set of counterfactual examples created from the MusicCaps [13] at `https://tinyurl.com/counterfac`

where n is the number of language-audio pairs, $f_i = \mathcal{M}(t_i)$ and $\tilde{f}_i^c = \mathcal{M}(\tilde{t}_i^c)$ are the outputs of the model $\mathcal{M}$. For text-to-music retrieval, we define

$$R@k = \frac{1}{n}\left[\sum_{i=1}^{n} 1 - \frac{|A_i \cap \tilde{A}_i|}{|A_i|}\right], \tag{2}$$

where $A_i = \mathcal{M}(t_i)$ and $\tilde{A}_i^c = \mathcal{M}(\tilde{t}_i^c)$ are the top-k retrieved audio candidates.

Table 2 and 3 show the sensitivity of a wide range of SOTA text-to-music generative and retrieval models. From the tables, we observe that these models maintain a substantially higher sensitivity to changes in descriptive elements compared to atmospheric, situational, contextual, or metadata change. These results highlight the lack of semantic awareness in the textual conditioning of a music understanding model, which manifests a misalignment between the audio candidates expected by a user and the model output.

3 The MusicSem Dataset

To address this lack of semantic sensibility, we introduce MusicSem, a novel dataset of language–audio music description pairs extracted from five English-language Reddit threads featuring detailed user discussions across diverse genres: r/electronicmusic, r/popheads, r/progrockmusic, r/musicsuggestions, and r/LetsTalkMusic.

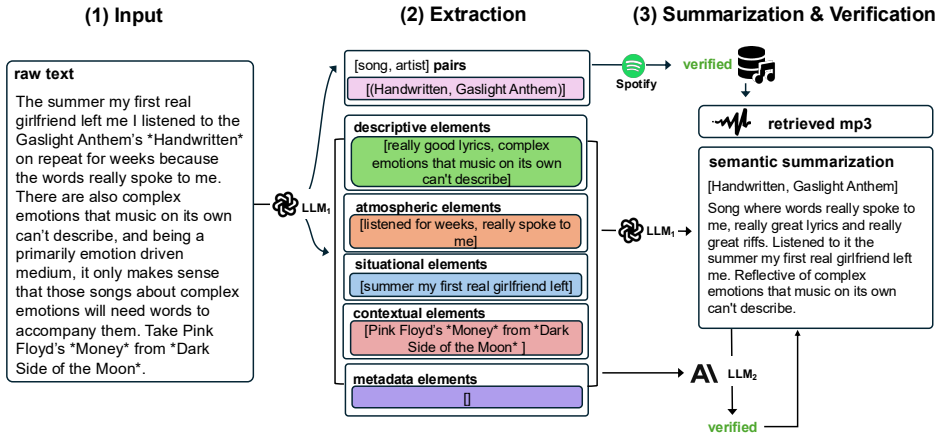

Figure 1: Visualization of the extraction and verification pipeline for dataset construction.

The dataset aims to capture more nuanced musical semantics to support the training and evaluation of multimodal models in future work. Its construction involved substantial effort to identify, extract, structure, and validate semantic content from online discourse, combining LLM-assisted extraction with human annotation and verification. A comprehensive description of this process is provided in our extended paper and illustrated in the Demo section of our website.

The released dataset comprises 32,493 entries, each including a Spotify ID and URL for audio retrieval, the source thread, raw text, song and artist names, and semantics structured according to the taxonomy in Table 1. We also constructed an unpublished test set of 480 entries for future leaderboard use on our website. Table 4 shows the proportion of entries containing each of the five semantic categories in MusicSem and two canonical language-audio datasets. MusicSem consistently demonstrates broader coverage across all categories, highlighting its semantic richness. It also exhibits a richer vocabulary, with a higher count of unique words and music genres.

Table 4: Statistics (top) and semantic diversity (bottom) of MusicSem and two other language-audio music datasets. For a more comprehensive collection of related datasets please see Appendix A.

Statistics	**MusicCaps** [13]	**Song Describer** [22]	**MusicSem (ours)**
# Entries	5,521	1,100	32,493
# Vocab. Words	6,245	2,824	22,738
# Music Genres	267	152	493
Category	**MusicCaps** [13]	**Song Describer** [22]	**MusicSem (ours)**
Descriptive	100%	94%	100%
Contextual	6%	8%	77%
Situational	41%	16%	48%
Atmospheric	57%	33%	64%
Metadata	28%	6%	64%

4 Evaluating on Cross Modal Retrieval and Text-to-Music Generation

To demonstrate the utility and superiority of our dataset, we evaluate representative multimodal music understanding models on text-to-audio retrieval and generation, two of the major tasks where semantic awareness in music representation learning plays a pivotal role in a model's success.

General Insights As we can see in Table 5, *model performance differs between datasets and metrics*. There is no conclusive state-of-the-art model for either task, nor along any dataset or metric. This variation in model rankings across datasets highlights the limited generalization of current multimodal music understanding models. Furthermore, there is also noticeable performance inconsistency between the canonical metrics used in Table 5 compared to the semantic sensitivity results in Tables 2 and 3. For example while LARP/CLaMP3 and Stable Audio are the best performing models in terms of overall semantic sensitivity, they do not achieve superior performance on the canonical performance metrics for their tasks.

Table 5: Evaluation results on text-to-music generation (left) and retrieval (right) tasks. Best performance for each metric within a dataset is in **bold** and second best in underline. Please see Appendix F.5 for detailed break down of evaluation metrics.

Dataset	Generation Model	Generation Metrics				Retrieval Model	Retrieval Metrics		
		FAD ↓	$\text{FAD}_{\text{E}}^{\text{FMA}} \infty$ ↓	CS ↑	Vendi ↑		R@10 ↑	N@10 ↑	MRR ↑
MusicCaps	MusicLM	5.70	249.72	0.28	1.55	Random	0.36	0.16	0.31
	Stable Audio	6.97	377.02	0.31	1.31	LARP	0.98	0.45	0.62
	MusicGen	7.03	354.07	0.29	**1.57**	CLAP	**22.60**	**12.99**	**11.60**
	AudioLDM2	3.29	202.11	**0.36**	**1.57**	ImageBind	14.91	8.25	7.23
	Mustango	**1.27**	**161.47**	0.27	1.48	CLaMP3	13.65	7.32	9.07
Song Describer	MusicLM	7.20	241.95	0.28	1.49	Random	1.41	0.64	1.01
	Stable Audio	4.42	341.92	0.31	1.29	LARP	2.62	1.29	1.61
	MusicGen	2.64	354.07	**0.35**	**1.50**	CLAP	27.67	14.54	12.41
	AudioLDM2	2.74	184.03	0.34	1.48	ImageBind	20.71	11.16	9.84
	Mustango	**2.58**	**170.27**	0.29	1.46	CLaMP3	**38.61**	**22.84**	**19.83**
MusicSem (Ours)	MusicLM	7.25	248.42	0.27	1.46	Random	2.11	0.96	1.42
	Stable Audio	5.50	342.53	**0.31**	1.28	LARP	3.07	1.22	1.47
	MusicGen	3.75	229.29	0.30	**1.50**	CLAP	9.84	4.74	4.65
	AudioLDM2	**3.47**	181.11	0.28	1.46	ImageBind	11.07	5.48	5.24
	Mustango	5.06	**157.32**	0.20	1.41	CLaMP3	**26.84**	**16.21**	**14.68**

Text-to-Music Retrieval From the results in Table 5, we can see that *MusicSem is more challenging than existing datasets.* Despite its smaller candidate set (480 items), MusicSem sees lower performance from most models compared to Song Describer (1K items). This suggests that MusicSem's complexity lies in its semantics, highlighting current models' limitations in music understanding.

Music-to-Text Generation Surprisingly do not see any in Table 5 we are unable to see any performance differences between various models on the canonical benchmark datasets and MusicSem. This is unexpected because MusicSem contains significantly less descriptive annotations which should, intuitively, be reflected in the CLAP score performance, which is one of the key metrics used to objectively evaluate alignment between a textual prompt and its associated generated audio output. Thus, we consider the expressiveness of the CLAP model in relation to semantic complexity. More specifically, we leverage the semantic sensitivity metric in Eq. 1 and calculate the cosine similarity of text embeddings generated by the CLAP model, in order to assess its ability to adequately capture semantic differences in a textual prompt. Crucially, the results in Table 6 show that *CLAP has a concrete lack of semantic sensitivity*. This strongly indicates that the *CLAP score is highly limited in capturing the rich semantics of music*.

Table 6: The sensitivity of CLAP score. The superscripts ^{d}, ^{a}, ^{s}, ^{c}, and ^{m} refer to descriptive, atmospheric, situational, contextual, and metadata, respectively.

Category	Metric	Score
Descriptive	G^d	0.55
Atmospheric	G^a	0.36
Situational	G^s	0.32
Contextual	G^c	0.29
Metadata	G^m	0.36

5 Conclusion

In this work, we introduce MusicSem, a semantically rich language-audio dataset that captures the diverse language in organic musical discourse. We categorize these textual annotations into five categories of music semantics and show the importance of music semantics. We evaluate a suite of

music understanding models in multimodal generation and cross modal retrieval tasks on MusicSem and other canonical datasets, which reveal critical insights about pitfalls in existing evaluations of music understanding and the importance of capturing nuances in musical annotations.

References

[1] Meinard Müller. *Fundamentals of Music Processing*. 01 2015. ISBN 978-3-319-21944-8. doi: 10.1007/978-3-319-21945-5.

[2] Markus Schedl, Emilia Gómez, and Julián Urbano. 2014. doi: 10.1561/1500000042.

[3] Yu-Ching Lin, Yi-Hsuan Yang, and Homer H. Chen. Exploiting online music tags for music emotion classification. *ACM Trans. Multim. Comput. Commun. Appl.*, 7(Supplement):26, 2011.

[4] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, 2019.

[5] Sergio Oramas, Oriol Nieto, Francesco Barbieri, and Xavier Serra. Multi-label music genre classification from audio, text and images using deep features. In *ISMIR*, pages 23–30, 2017.

[6] Ruibin Yuan, Yinghao Ma, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, Le Zhuo, Yiqi Liu, Jiawen Huang, Zeyue Tian, Binyue Deng, Ningzhi Wang, Chenghua Lin, Emmanouil Benetos, Anton Ragni, Norbert Gyenge, Roger B. Dannenberg, Wenhu Chen, Gus Xia, Wei Xue, Si Liu, Shi Wang, Ruibo Liu, Yike Guo, and Jie Fu. MARBLE: music audio representation benchmark for universal evaluation. In *NeurIPS*, 2023.

[7] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, pages 1–5. IEEE, 2023.

[8] Shangda Wu, Zhancheng Guo, Ruibin Yuan, Junyan Jiang, Seungheon Doh, Gus Xia, Juhan Nam, Xiaobing Li, Feng Yu, and Maosong Sun. Clamp 3: Universal music information retrieval across unaligned modalities and unseen languages. 2025. URL `https://arxiv.org/abs/2502.10362`.

[9] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

[10] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, and Ying Shan. Music understanding llama: Advancing text-to-music generation with question answering and captioning. 2023. URL `https://arxiv.org/abs/2308.11276`.

[11] SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. Lp-musiccaps: Llm-based pseudo music captioning. 2023. URL `https://arxiv.org/abs/2307.16372`.

[12] Junda Wu, Zachary Novack, Amit Namburi, Jiaheng Dai, Hao-Wen Dong, Zhouhang Xie, Carol Chen, and Julian McAuley. FUTGA: Towards fine-grained music understanding through temporally-enhanced generative augmentation. In Anna Kruspe, Sergio Oramas, Elena V. Epure, Mohamed Sordo, Benno Weck, SeungHeon Doh, Minz Won, Ilaria Manco, and Gabriel Meseguer-Brocal, editors, *Proceedings of the 3rd Workshop on NLP for Music and Audio (NLP4MusA)*, pages 107–111, Oakland, USA, November 2024. Association for Computational Lingustics. URL `https://aclanthology.org/2024.nlp4musa-1.17/`.

[13] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text. 2023. URL `https://arxiv.org/abs/2301.11325`.

[14] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[16] Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:2871–2883, May 2024. ISSN 2329-9290. doi: 10.1109/TASLP.2024.3399607. URL https://doi.org/10.1109/TASLP.2024.3399607.

[17] Yongyi Zang and Yixiao Zhang. The interpretation gap in text-to-music generation models. 2024. URL https://arxiv.org/abs/2407.10328.

[18] Francesca Ronchini, Luca Comanducci, Gabriele Perego, and Fabio Antonacci. Paguri: a user experience study of creative interaction with text-to-music models. 2024. URL https://arxiv.org/abs/2407.04333.

[19] Mark Levy and Mark Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 2008.

[20] Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE Signal Process. Mag.*, 36(1):41–51, 2019.

[21] Jeong Choi, Anis Khlif, and Elena Epure. Prediction of user listening contexts for music playlists. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*. Association for Computational Linguistics, 2020.

[22] Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam. The song describer dataset: a corpus of audio captions for music-and-language evaluation. *CoRR*, abs/2311.10057, 2023.

[23] Anna-Maria Christodoulou, Olivier Lartillot, and Alexander Refsum Jensenius. Multimodal music datasets? challenges and future goals in music processing. *Int. J. Multim. Inf. Retr.*, 13 (3):37, 2024.

[24] Daniel McKee, Justin Salamon, Josef Sivic, and Bryan Russell. Language-guided music recommendation for video via prompt analogies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[25] Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, Jesse Engel, Quoc V. Le, William Chan, Zhifeng Chen, and Wei Han. Noise2music: Text-conditioned music generation with diffusion models. 2023. URL https://arxiv.org/abs/2302.03917.

[26] Seungheon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. Lp-musiccaps: Llm-based pseudo music captioning. In *ISMIR*, pages 409–416, 2023.

[27] Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8286–8309, 2024.

[28] Darius Afchar, Alessandro B. Melchiorre, Markus Schedl, Romain Hennequin, Elena V. Epure, and Manuel Moussallam. Explainability in music recommender systems. *CoRR*, abs/2201.10528, 2022.

[29] Qingqing Huang, Daniel S. Park, Tao Wang, Timo Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, Jesse Engel, Quoc V. Le, William Chan, Zhifeng Chen, and Wei Han. *Noise2Music: Text-conditioned Music Generation with Diffusion Models*, 2023. URL https://arxiv.org/abs/2302.03917.

[30] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Downie. Evaluation of algorithms using games: The case of music tagging. pages 387–392, 01 2009.

[31] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 909–916, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312301. doi: 10.1145/2187980.2188222. URL https://doi.org/10.1145/2187980.2188222.

[32] Abhinaba Roy, Renhang Liu, Tongyu Lu, and Dorien Herremans. Jamendomaxcaps: A large scale music-caption dataset with imputed metadata. 2025. URL https://arxiv.org/abs/2502.07461.

[33] Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Efficient text-to-music diffusion models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8050–8068, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.437. URL https://aclanthology.org/2024.acl-long.437/.

[34] John Thickstun, Zaid Harchaoui, Dean P. Foster, and Sham M. Kakade. Invariances and data augmentation for supervised music transcription. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[35] Samarth Bhargav, Anne Schuth, and Claudia Hauff. When the music stops: Tip-of-the-tongue retrieval for music. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2506–2510, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3592086. URL https://doi.org/10.1145/3539618.3592086.

[36] Veniamin Veselovsky, Isaac Waller, and Ashton Anderson. Imagine all the people: Characterizing social music sharing on reddit. In *ICWSM*, pages 739–750. AAAI Press, 2021.

[37] Joshua Patrick Gardner, Simon Durand, Daniel Stoller, and Rachel M. Bittner. Llark: A multimodal instruction-following language model for music. In *ICML*. OpenReview.net, 2024.

[38] John Thickstun, Zaid Harchaoui, and Sham M. Kakade. Learning features of music from scratch. In *International Conference on Learning Representations (ICLR)*, 2017.

[39] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. 2017. URL https://arxiv.org/abs/1612.01840.

[40] S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. Mmau: A massive multi-task audio understanding and reasoning benchmark. 2024. URL https://arxiv.org/abs/2410.19168.

[41] David Hauger, Andrej Kosir, Marko Tkalčič, and Markus Schedl. The million musical tweets dataset: What we can learn from microblogs. 11 2013.

[42] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, and Ying Shan. Music understanding llama: Advancing text-to-music generation with question answering and captioning. In *ICASSP*, pages 286–290. IEEE, 2024.

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[44] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005. URL https://aclanthology.org/W05-0909/.

[45] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[46] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. 2015. URL https://arxiv.org/abs/1411.5726.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[48] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250, 2021. doi: 10.1109/ASRU51503.2021.9688253.

[49] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. 2022. URL https://arxiv.org/abs/2210.11416.

[50] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger B. Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. MERT: acoustic music understanding model with large-scale self-supervised training. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=w3YZ9MSlBu.

[51] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

[52] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. 2022. URL https://arxiv.org/abs/2212.04356.

[53] Rebecca Salganik, Xiaohao Liu, Yunshan Ma, Jian Kang, and Tat-Seng Chua. LARP: language audio relational pre-training for cold-start playlist continuation. In *KDD*, pages 2524–2535. ACM, 2024.

[54] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[56] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP*, pages 646–650. IEEE, 2022.

[57] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, pages 2818–2829. IEEE, 2023.

[58] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, pages 8440–8451. Association for Computational Linguistics, 2020.

[59] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. 2024. URL https://arxiv.org/abs/2303.16199.

[60] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards Generic Hearing Abilities for Large Language Models. April 2024. doi: 10.48550/arXiv.2310.13289. URL http://arxiv.org/abs/2310.13289.

[61] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio and natural language. 2022. URL https://arxiv.org/abs/2208.12415.

[62] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. 2022. URL https://arxiv.org/abs/2210.13438.

[63] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. 2021. URL https://arxiv.org/abs/2107.03312.

[64] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

[66] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023. URL https://arxiv.org/abs/2111.09543.

[67] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In Gernot Kubin and Zdravko Kacic, editors, *20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019*, pages 2350–2354. ISCA, 2019. doi: 10.21437/INTERSPEECH.2019-2219. URL https://doi.org/10.21437/Interspeech.2019-2219.

[68] Azalea Gui, Hannes Gamper, Sebastian Braun, and Dimitra Emmanouilidou. Adapting frechet audio distance for generative music evaluation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1331–1335, 2024. doi: 10.1109/ICASSP48485.2024.10446663.

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

A Related Works

Given the complexity of achieving true music understanding, there has been a large body of work which attempts to address different facets of this challenging task via various approaches to language-audio representation learning, each needing their own format of data. We briefly review datasets that are complementary to our work. For a more comprehensive review we direct readers towards a survey by Christodoulou, Lartillot, and Jensenius [23]. Depending on the source of textual data, current language-audio music datasets can be categorized as human-annotated datasets or LLM-augmented datasets. For human-annotated language-audio music datasets, *MusicCaps* [13] is one of the most commonly used dataset. It consists of approximately 5,521 language-audio samples annotated by professional musicians. These annotations contain descriptive language that often involves attributes such as instrumentation, genre, and stylistic analysis. Similarly, *YouTube8M-MusicTextClips* [24] contains approximately 4,169 language-audio pairs, but the associated captions are written by text-for-hire annotators. More recently, Manco et al. [22] presented the *Song Describer* [22] extended 1,100 of audio samples in *Jamendo* [4] with crowd-sourced annotations. Meanwhile, there are also datasets with LLM-augmented annotations [25, 26, 12, 11, 27], which, though although they have a larger scale, lack precise description on how music is experienced in the real world [21, 19, 28]. Different from these datasets that primarily capture the acoustic elements of a song, our work seeks to understand how a song makes a user feel and the contexts in which users listen to it.

Recently, there has been a push to expand the scope of these datasets via LLM-based augmentation. For example in their work [29] build a combined dataset of 6.8M pairs by fusing *MusicCaps* [13] with LLM-based annotations of 150K popular songs. *LP-MusicCaps* [11] combines several sources including *MusicCaps* [13], *MagnaTagATune* [30], and *Million Song Dataset* [31] to construct 2.2M captions paired with 0.5M audio samples by generating sentence-like captions generated by an LLM. Similarly, *MusicBench* [27] is a dataset with 52K paired language-audio samples constructed by applying automatic algorithms for extracting downbeats, chords, keys, and tempo from the audio included in *MusicCaps* and augmenting its original captions to include this information. The dataset used to train FUTGA [12] follows a similar augmentation strategy in which an LLM is prompted to augment the annotations in *MusicCaps* [13] and *Song Describer* [22] to include structural elements in the music. *JamendoMaxCaps* [32] is generated by collecting 200,000 audio samples from the *Jamendo* [4] dataset and applying a music captioning model to generate automatic textual annotations. *Text2Music* [33] is another such dataset which contains 50K language-audio pairs which are compiled by scraping Spotify for the top 10 most popular playlists and using an LLM to rephrase their metadata into sentence-like structures. Notably, despite the prevalent use of LLMs in the construction of these datasets, to our knowledge, there is extremely limited discussion within this body of work on the hallucination protocols used to ensure data quality. While we acknowledge that it is impossible to fully mitigate hallucination when engaging on a large scale with LLMs, it is important to consider the effectiveness of various mitigation techniques in order to ensure data quality.

Table 7: Language-Audio Music Dataset Statistics. Note, for brevity, we present the datasets that are most comparable with our setting. Here, we use L-A Pairs to mean Language-Audio Pairs and Annotation Source to indicate the source of the textual annotations.

Dataset Name	Year	# L-A Pairs	Annotation Source	Base Dataset
MusicNet 34	2018	330	Human	-
Song Describer 22	2023	1,106	Human	-
YouTube8M-MusicTextClips 24	2023	4,169	Human	
MusicCaps 13	2023	5,521	Human	-
MusicSem (Ours)	2025	35,977	Human or LLM	-
MuLaMCap 25	2023	6,800,000	LLM	AudioSet
LP-MusicCaps 26	2023	2,000,000	LLM	MusicCaps, Magnatagtune, & Million Song Dataset
Text2Music [33]	2024	50,000	LLM	Spotify
FUTGA 12	2024	51, 800	LLM	MusicCaps & Song Describer
MusicBench 27	2024	53,168	LLM	MusicCaps
JamendoMaxCaps 32	2025	200,000	LLM	Jamendo

There also exist other music datasets based on Reddit threads [35, 36]. However, they are intended for different settings from ours. For example, *Tip-Of-My-Tongue* [35] is based on `r/TipOfMyTongue` for text-to-music querying. Alternatively, Veselovsky, Waller, and Anderson [36] scrape Reddit for $536,860$ unique song-artist pairs to analyze the music sharing behaviors in Reddit communities.

Table 9: Results of music-to-text generation. Best performance within each dataset is in **bold**.

Dataset	Model	$B_1 \uparrow$	$B_2 \uparrow$	$B_3 \uparrow$	M $\uparrow$	R $\uparrow$	CIDEr $\uparrow$	Bert-S $\uparrow$
MusicCaps	LP-MusicCaps	**53.21**	**47.28**	**44.60**	**51.90**	3.35	**384.72**	**90.47**
	MU-LLaMA	1.35	0.55	0.22	40.22	11.27	0.09	80.47
	FUTGA	8.80	3.07	1.19	44.77	**11.90**	2.63e-17	81.67
Song Describer	LP-MusicCaps	9.51	3.07	0.94	**8.90**	10.45	1.03	**84.40**
	MU-LLaMA	**12.03**	**4.73**	**1.73**	8.72	**13.00**	**3.59**	83.51
	FUTGA	3.39	1.28	0.43	8.72	6.30	3.58e-30	82.55
MusicSem (Ours)	LP-MusicCaps	**11.57**	**3.05**	**0.72**	**20.59**	9.54	0.77	**82.13**
	MU-LLaMA	4.11	1.41	0.51	22.33	**10.57**	**0.92**	81.63
	FUTGA	4.82	1.50	0.44	22.23	7.48	0.01	80.93

In another strain of music understanding tasks, several works have begun to consider music understanding through the lens of generative retrieval or musical question-answering [37, 10]. To serve the needs of this novel task, several works have proposed datasets that reformat the textual information described above as question-answer pairs. For example, *MusicQA* [10] uses a LLM to reformulate the captions in *MusicCaps* [13] and *Magnatagtune* [30] into 4,500 question-answer pairs. Alternatively *LLaRK* [37] propose a dataset with over 1.2M language-audio pairs by combining *MusicCaps* [13], *YouTube8M* [24], *MusicNet* [38], *FMA* [39], Jamendo [4], and *MagnaTagATune* [30]. Finally Sakshi et al. [40] curate 10K a set of generalized audio and music question-answer pairs which assess a variety of music understanding tasks.

Finally, in a complementary body of musical datasets, several works have analyzed music understanding through the lens of online discourse. In addition to the datasets mentioned in the main body of our work [35, 36] which contained discourse from Reddit, the *Million Tweet Dataset* [41] analyzed over 1M tweets associated with music to understand the trends in popularity among songs and artists.

B Additional Experimental Results & Insights

B.1 Music-to-Text Generation

Music-to-text generation, also known as music captioning, focuses on generating natural language descriptions of a musical work. We consider three SOTA models, including LP-MusicCaps [42], MU-LLaMA [10], and FUTGA [12], and evaluate them on three datasets, i.e., MusicCaps [13], Song Describer [22] and the test set of our proposed datset, MusicSem. We employ objective evaluation metrics borrowed from natural language processing such as BLEU (B) [43], METEOR (M) [44], ROUGE (R) [45], CIDEr [46], and BERT-score (Bert-S) [47] which are commonly used in evaluation for this task. For a more in-depth discussion of the evaluation metrics and intuitions behind them, please see Appendix F.5. The results are presented in Table 9 with the following insights.

Table 8: Semantics analysis of the music-to-text generation results on MusicSem. 'G.T.' refers to 'Ground Truth'.

Model	LP-MusicCaps	MU-LLaMA	FUTGA	G.T.
Descriptive	100%	99%	100%	83%
Contextual	2%	1%	0%	17%
Situational	42%	0%	1%	38%
Atmospheric	78%	3%	91%	62%
Metadata	32%	2%	34%	15%

Insight 2.1: Model performance differs between datasets and metrics. When looking at the results for MusicCaps and MusicSem datasets we can see that LP-MusicCaps [11] has strong performance on this dataset. Meanwhile, on the Song Describer dataset, MU-LLaMA outperforms both models. This observation coincides with the performance inconsistency observed in cross modal retrieval, further justifying that existing music-to-text generation models have generalization issues. Developing highly generalizable models would be one of the key research questions for text-to-music generation.

Insight 2.2: The performance inconsistency is attributed to the diverse semantics among datasets. To further demystify the performance inconsistencies, we analyze the presence of each type of semantics both in the ground truth of the MusicSem test set and the text generated by each model in Table 8. From this statistics, we can see that LP-MusicCaps's high performance positively correlates

with its higher percentage of atmospheric, situational, and contextual annotations in our dataset. LP-MusicCaps is the model with the highest percentage of these semantic categories represented in its output. Furthermore, we can clearly see that all of the models are skewed towards presenting descriptive captions and very few are able to capture the contextual, situational, and atmospheric elements of the Reddit-based annotations. This highlights the challenge of generating accurate and meaningful semantic information using the existing SOTA models, and MusicSem can be instrumental in bridging this gap.

B.2 Case study of music-to-text generation

When looking at the performance of the various music-to-text models reported in Table 9 within the main body of this work, it seems that LP-MusicCaps is the best performing model but a deeper analysis of its output challenges this. In Figure 2 we showcase a case study of the comparative outputs between the original annotation and the captions produced by each model. As we can see in the case study, FUTGA generates a much more detailed and accurate description of the audio however, it receives a lower overall performance score because, in generating more content, it has the potential for a lower n-gram overlap. Meanwhile, MU-LLaMA, though completely incorrect retains scores which are close to that of FUTGA potentially due to the shortened length of the model's output. Furthermore, despite the seemingly high performance of each model on the objective metrics, each caption output contains at least one factually incorrect description of the input music. This indicates that there is still a significant information gap that SOTA models are unaware of.

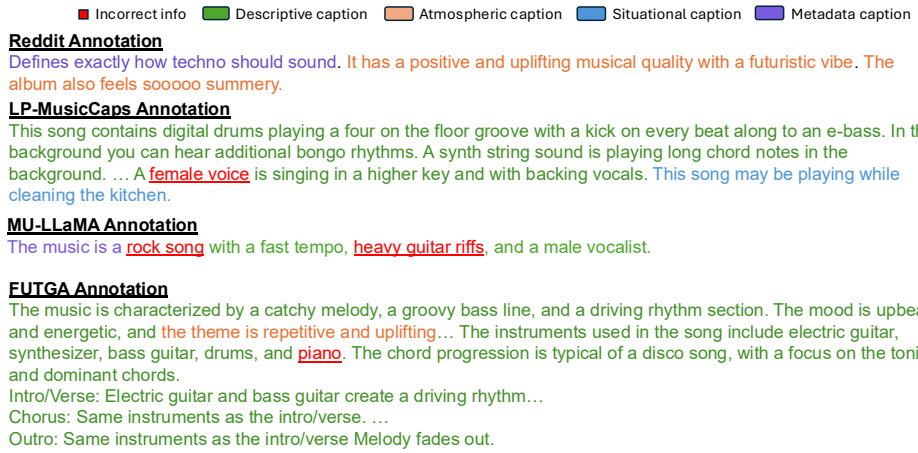

Figure 2: Case study of music-to-text generation evaluation. We can see that all models make objective factual errors and focus primarily on descriptive annotations. For reference, please listen to the song on Youtube – While Others Cry by The Future Sound of London.

C Further Details of Dataset Construction Pipeline

We present the pseudocode for the complete extraction pipeline in Algorithm 1.

In Lines 2-3 we filter the posts within the thread itself, removing any posts that were written by moderators and any posts that had less than 20 characters. In Line 4 we perform the extraction, using an LLM to extract semantic information from the text using a prompt (see Appendix D.1 for the full prompt). In Line 6 we query the Spotify API to find a unique identifier associated with each song mentioned in a thread. In Line 7 we perform the first hallucination check, ensuring that the audio is aligned with the extracted song-artist pair. In Line 8 we extract the mp3 files associated with the audio of each song. In Line 9 we generate summaries from the extraction caption categories that mimic those of *MusicCaps* [13] or Song Describer [22]. Finally, in Line 10 we perform one more hallucination check using a different model to ensure that the summary did not deviate from the extracted caption categories (see Appendix D.2 for the full prompt). In total, this process yields a dataset of approximately 35K language-audio pairs. For a visualization of the entire pipeline, please see Figure 1 within the main body of the paper.

Algorithm 1 Collection Framework

Input: thread name T, language models $\mathcal{M}_1$, $\mathcal{M}_2$
Output caption set C

procedure DATASET GENERATION($T, \mathcal{M}$)
    posts = Load_Entire_Thread(T)
    filtered = Length_and_Mod_Filter(posts)
    sa_pairs, caption_extracts = $\mathcal{M}_1$(filtered)
    descriptive, atmospheric, situational, contextual, metadata = caption_extracts
    song_ids = Spotify_Metadata(sa_pairs)
    sa_pairs = Hallucination_Check1(sa_pairs,fltrd)
    mp3s = Spotify_Audio(song_ids)
    final_summaries = Summarize(sa_pairs,caption_extracts, mp3s)
    filtered_captions = Hallucination_Check2(caption_extracts, final_captions, $\mathcal{M}_2$)

D Prompts

D.1 Extraction Prompt

Below we present the prompt which is used to extract semantic content from raw text posts on Reddit. Following the formulation of caption categories in Table 1, we break down the elements which are contained in each of the five categories. We also provide an example extraction for guidance.

```
% Feature Extraction

Task Description
You are tasked with analyzing Reddit posts about music and extracting
    structured information into specific categories. When given a
    Reddit post discussing music, identify and extract the following:
Categories to Extract
Song/Artist pairs
(using the names of artists and their songs with unfixed form) some
    examples:

'Shake it Off by Taylor Swift'
'Radiohead's Weird Fishes'
'Genesis - Yes'
'Maroon5 [She Will Be Loved]'

Descriptive (using musical attributes)
This includes detailed observations about:

Instrumentation: 'I love the high pass filter on the vocals in the
    chorus and the soft piano in the bridge'
Production techniques: 'The way they layered those harmonies in the
    second verse is incredible'
Song structure: 'That unexpected key change before the final chorus
    gives me goosebumps'
Sound qualities: 'The fuzzy lo-fi beats with that vinyl crackle in the
     background'
Technical elements: 'The 6/8 time signature makes it feel like its
    swaying'

Contextual (using other songs/artists)
This includes meaningful comparisons such as:

Direct comparisons: 'Sabrina Carpenter's Espresso is just a mix of old
     Ariana Grande and 2018 Dua Lipa'
Influences: 'You can tell they've been listening to a lot of Talking
    Heads'
Genre evolution: 'It's like 90s trip-hop got updated with modern trap
    elements'
```

```
Sound-alikes: 'If you like this, you should check out similar artists like...'
Musical lineage: 'They're carrying the torch that Prince lit in the 80s'

Situational (using an activity, setting, or environment)
This includes relatable scenarios like:

Life events: 'I listened to this song on the way to quitting my sh**ty corporate job'
Regular activities: 'This is my go-to album for late night coding sessions'
Specific locations: 'Hits different when you're driving through the mountains at sunset'
Social contexts: 'We always play this at our weekend gatherings and everyone vibes to it'
Seasonal connections: 'This has been my summer anthem for three years running'

Atmospheric (using emotions and descriptive adjectives)
This includes evocative descriptions such as:

Emotional impacts: 'This song makes me feel like a manic pixie dream girl in a bougie coffeeshop'
Visual imagery: 'Makes me picture myself in a coming-of-age indie movie, running in slow motion'
Mood descriptions: 'It has this melancholic yet hopeful quality that hits my soul'
Sensory experiences: 'The song feels like a warm embrace on a cold day'
Abstract feelings: 'Gives me this feeling of floating just above my problems'

Lyrical (focusing on words and meaning)
This includes thoughtful commentary on:

Storytelling: 'The lyrics tell such a vivid story of lost love that I feel like I've lived it'
Wordplay: 'The clever double entendres in the chorus make me appreciate it more each listen'
Messaging: 'The subtle political commentary woven throughout the verses really resonates'
Personal connection: 'These lyrics seem like they were written about my own life experiences'
Quotable lines: 'That line 'we're all just stardust waiting to return' lives rent-free in my head'

Metadata (using information found in labels or research)
This includes interesting facts like:

Technical info: 'The song is hip-hop from the year 2012 with a bpm of 100'
Creation context: 'They recorded this album in a cabin with no electricity using only acoustic instruments'
Chart performance: 'It's wild how this underplayed track has over 500 million streams'
Artist background: 'Knowing the guitarist was only 17 when they recorded this makes it more impressive'
Release details: 'This deluxe edition has three bonus tracks that are better than the singles'

Sentiment (whether the person feels good or bad about the song)
Output Format
Return your analysis as a structured JSON with these categories:
Copy{
```

```
  'pairs': [(song_1, artist_1), (song_2, artist_2), ...],
  'Descriptive': [],
  'Contextual': [],
  'Situational': [],
  'Atmospheric': [],
  'Lyrical': [],
  'Metadata': [],
  'Sentiment': []
}
Example
Input:
'I like Plastic Love by Mariya Takeuchi because of the funky, jazzy,
    retro vibes. I listen to this music at 3am when Im lonely because
    it romanticizes my loneliness and makes it meaningful. It helps me
     to enjoy my own loneliness. It has very distinctive synthesizer
    sounds in the chorus and leading bass lines in the bridge. The
    vocals are chill and blended. Another song that sounds very
    similar is Once Upon a Night by Billyrrom or Warm on a Cold Night
    by Honne. The genre is like City Pop which describes an idealized
    version of a city.'
Output:
Copy{
  'pairs': [('Plastic Love', 'Mariya Takeuchi'), ('Once Upon a Night',
     'Billyrrom'), ('Warm on a Cold Night', 'HONNE')],
  'Situational': ['3am when Im lonely'],
  'Descriptive': ['funky', 'jazzy', 'retro vibes', 'distinctive
    synthesizer in chorus', 'leading bass lines in bridge', 'chill and
     blended vocals', 'genre of City Pop'],
  'Atmospheric': ['romantic loneliness', 'vulnerability', 'kind of sad
     in a good way', 'acting heartbroken', 'idealized version of a
    city'],
  'Contextual': ['Plastic Love sounds similar to Once Upon a Night', '
    Plastic Love sounds similar to Warm on a Cold Night'],
  'Metadata': ['funky', 'jazzy', 'retro vibes', 'genre of City Pop']
}
```

D.2 Hallucination Check Prompt

Below we present the prompt which is used to validate the results of an extraction and summarization. Here, we use a secondary model to check for hallucination between an extraction of semantic tags and their sentence-like summarization. Please note that we present the LLM with two examples: one negative (i.e. containing no hallucinations) and one positive (i.e. containing hallucinations) as we found in our ablation experiments that this significantly improved the model's ability to identify hallucinations.

```


% Getting summarizations
# Summarization task

Write a sentence which combines the associated sentence fragments.
Please do not add anything other than the information given to you.

Your description should:
- Be maximum 4 sentences in length

Your description shouldn't:
- Add any additional information that is not present in the tags
- Include any information that is based on your own knowledge or
    assumptions

Example:
  'Situational':  ['3am when Im lonely'],\
```

```
  'Descriptive':['funky', 'jazzy', 'retro vibes', 'distinctive synthesizer in chorus', 'leading bass lines in bridge', 'chill and blended vocals', 'genre of City Pop'],\
  'Atmospheric'': ['romantic loneliness', 'vulnerability', 'kind of sad in a good way', 'acting heartbroken', 'idealized version of a city'],\
  'Contextual': ['Plastic Love sounds similar to Once Upon a Night', 'Plastic Love sounds similar to Warm on a Cold Night'],\
  'Metadata': ['funky', 'jazzy', 'retro vibes', 'genre of City Pop']\

  Desired output: This song has funky, jazzy, retro vibes. I listen to this music at 3am when Im lonely because it romanticizes my loneliness and makes it meaningful. \
   It helps me to enjoy your own loneliness. It has very distinctive synthesizer sounds in the chorus and leading bass lines in the bridge. \
   The vocals are chill and blended.  The genre is like City Pop which describes an idealized version of a city.'  \

Tags:

{input_tags}

% Hallucination
# Hallucination Check Prompt for Generated Summary

## Instructions
Evaluate whether the generated summary contains hallucinations based on the provided features/tags from the original source.
A hallucination is defined as information in the summary that is not present in or contradicts the features from the source material.

## Input Format
- **Original Features/Tags**: [List of key features/tags from the source material]
- **Generated Summary**: [The summary to be evaluated]

## Task
1. Compare each claim or statement in the summary against the original features/tags
2. Identify any information in the summary that:
   - Is not supported by the original features/tags
   - Contradicts the original features/tags
   - Represents an embellishment beyond what can be reasonably inferred
3. **The output should be in JSON format.**

## Output Format
‘‘‘
{{
"hallucination_detected": [True/False],
}}
‘‘‘

## Example 1
**Input Data**:
{{
  "original_features": {{
    'situational':  ['3am when Im lonely'],
    'descriptive':['funky', 'jazzy', 'retro vibes', 'distinctive synthesizer in chorus', 'leading bass lines in bridge', 'chill and blended vocals', 'genre of City Pop'],
    'atmospheric'': ['romantic loneliness', 'vulnerability', 'kind of sad in a good way', 'acting heartbroken', 'idealized version of a city'],
```

```
    'contextual': ['Plastic Love sounds similar to Once Upon a Night',
     'Plastic Love sounds similar to Warm on a Cold Night'],
  }},
  "generated_summary": 'funky, jazzy, retro vibes. I listen to this
   music at 3am when Im lonely because it romanticizes my loneliness
   and makes it meaningful.
    It helps me to enjoy your own loneliness. It has very distinctive
   synthesizer sounds in the chorus and leading bass lines in the
   bridge.
    The vocals are chill and blended.  The genre is like City Pop
   which describes an idealized version of a city.'
}}


**Expected Output**:
```
{{
"hallucination_detected": False,
}}

Example 2
Input Data:
{{
  "original_features": {{
    'situational': ['when I'm quitting my corporate job'],
    'descriptive':['angry punk guitar', 'killer drums', 'harcore vocal
     processing', 'distortion'],
    'atmospheric'': ['pumped up vibes', 'makes me want to take charge
   of my life'],
    'contextual': [''],
  }},
  "generated_summary": 'This song makes me happy. It has a soft and
   exciting vibe with killer drums. I listen to this song at parties
   or festivals when I feel positive.'
}}

Expected Output:
```
{{
"hallucination_detected": True,
}}
```

Input Data:
```
{{
  "original_features": {features},
  "generated_summary": {summary}
}}
```
Expected Output:
```
```

E Properties of the Dataset

We present additional insights into several unique aspects of MusicSem. As mentioned in earlier sections, MusicSem contains two key attributes: personalization and contextualization.

Personalization As we show in Table 10, for each song in our dataset there are approximately 3 different posts which discuss it. This yields a variety of annotations containing differing opinions on the same song. For example, in Figure 3 we showcase the semantic associations of two different users for the same song. We can see that this broadens the scope of perspectives that are represented by a dataset, presenting the opportunity for a more nuanced understanding of each musical piece.

Contextualization of Songs In Table 10 we can see that many songs are presented in tandem, where each post contains approximately 10 songs. For an intuitive example of this, we present a case study in Figure 3. In this case study the user describes a set of songs which are aligned under a unified theme (e.g. positivity). This form of contextualization provides an explicit definition of the underlying latent need that creates association between songs.

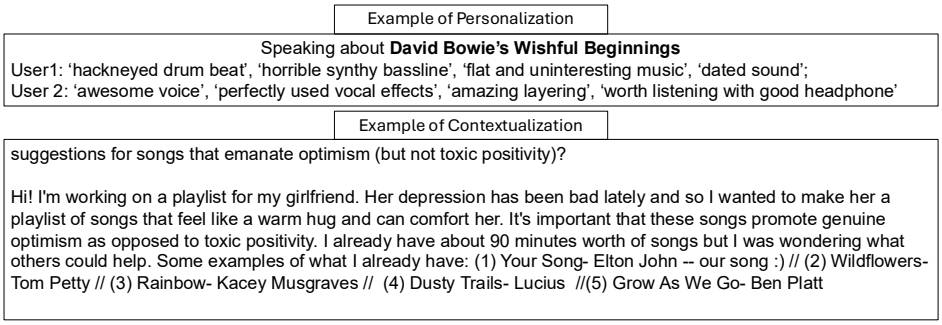

Figure 3: An example of personalization and contextualization on Reddit.

Table 10: Properties of the dataset.

Total Size	# Unique Songs	# Unique Artists	# Posts per Song	# Songs per Post	# Genres per Song
32,493	11,842	4,430	2.98	10.51	2.71

F Experimental Settings

F.1 Hyperpameter Settings

We present the baseline models and the specific details of their implementations. The evaluation involves both retrieval and generation tasks, where the tested models are summarized in Table 11.

Table 11: An overview of all the models we evaluate in this work. 'Hier.', 'Trans.', 'Diff.', and 'Co-List.' are short of Hierarchical, Transformer, Diffusion, and Co-Listing, respectively.

Task	**Name**	**Date**	**Architecture**	**Text Conditioner**	**Length**	**Sample Rate**	**Proprietary**
Text-to-Music	MusicLM [13]	2023	Hier. Trans. + SoundStream	w2v-BERT [48]	variable	24kHz	
	AudioLDM 2 [16]	2023	VAE + 2D U-Net	CLAP [7]	variable	16kHz	
	Stable Audio [15]	2023	VAE + 2D U-Net	CLAP [7]	up to 95s	48kHz	
	MusicGen [14]	2024	AE + 1D U-Net	FLAN-T5 [49]	10s	48kHz	
	Mustango [27]	2024	VAE + 2D U-Net	FLAN-T5 [49]	10s	16kHz	
	Mureka	2024	-	-	-	-	✓
Task	**Name**	**Year**	**Architecture**	**Audio Conditioner**	**Length**	**Sample Rate**	**Proprietary**
Music-to-Text	MU-LLaMA [10]	2024	Diff. Trans.	MERT [50]	60s	16kHz	
	LP-MusicCaps [42]	2023	Trans.	BART [51]	10s	16kHz	
	FUTGA [12]	2024	Hier. Trans. + VAE	Whisper[52]	240s	16kHz	
Task	**Name**	**Year**	**Architecture**	**Modalities**	**Length**	**Sample Rate**	**Proprietary**
Retrieval	CLAP [7]	2023	Contrastive Learning	Text + Waveform	-	48kHz	
	LARP [53]	2024	Contrastive Learning	Text + Waveform + Co-List. Graph		48kHz	
	ImageBind [9]	2023	Contrastive Learning	Text + Image	-	16kHz	
	CLaMP3 [8]	2024	Contrastive Learning	Text + Image + Waveform	-	24kHz	

F.2 Cross Modal Retrieval Models

CLAP [7] learns joint embeddings between audio clips and text descriptions through Contrastive Language-Image Pretraining `https://arxiv.org/abs/2103.00020`, on 630K audio-text pairs. For audio data, it first represents signals using log Mel spectrograms at a sampling rate of 44.1kHz, then employs CNN14 [54] (80.8M parameters) pretrained on AudioSet with 2M audio clips. For text data, it uses BERT [55] (110M parameters) to encode text descriptions, taking the [CLS] token embedding as text representation. Both modality encodings are projected into a multimodal space using two learnable projection matrices, resulting in an output dimension of 1024. We employ its music variant from the official repository `https://github.com/LAION-AI/CLAP`.

LARP [53] addresses the cold-start problem in playlist continuation through a three-stage contrastive learning framework. Built upon the BLIP framework, it consists of two uni-modal encoders: HTS-AT [56] for audio encoding and BERT for text processing (using [CLS] token embeddings), with their original 768-dimensional encodings being projected into a unified 256-dimensional space. The framework then performs within-track contrastive learning, track-track contrastive learning, and track-playlist contrastive learning to optimize representations from both semantic and intra-playlist music relevance perspectives. We use the official implementation from `https://github.com/Rsalganik1123/LARP`.

ImageBind [36] unifies six modalities (image, audio, text, etc.) in a single embedding space through multimodal contrastive learning. While not music-specific, its general-purpose audio-text alignment capability provides a strong baseline for cross-domain retrieval. ImageBind employs Transformer architectures for all modality encoders. For audio input, it converts 2-second 16kHz samples into spectrograms using 128 mel-spectrogram bins. Treating spectrograms as 2D signals similar to images, it processes them using a ViT with patch size 16 and stride 10. For text input, it utilizes pretrained text encoders (302M parameters) from OpenCLIP [57]. After projection, different modalities are encoded into a 768-dimensional shared space. We extract audio embeddings from the ViT-B/16 variant available at `https://github.com/facebookresearch/imagebind`.

CLaMP3 [8] establishes a unified multilingual music-text embedding space through cross-modal alignment of sheet music, audio recordings, and text in 12 languages. The audio processing pipeline adopts pre-trained acoustic features from MERT-v1-95M [50]. Each 5-second clip is represented by a single embedding obtained through averaging across all MERT layers and time steps. For textual content processing, the model employs XLM-R-base [58], a multilingual transformer, which features a 12-layer architecture with 768-dimensional hidden states. The framework implements contrastive learning to align multimodal representations, incorporating novel components such as a retrieval-augmented training mechanism that enhances cross-modal association. We use the checkpoints and architecture from the original authors' implementation at `https://sanderwood.github.io/clamp3`, specifically the SaaS version optimized for audio.

F.3 Cross Modal Generation Models

Music-to-Text Generation Models:

MU-LLaMA[10] is a music-specific adaptation of the LLaMA-2-7B architecture, integrating MERT [50] acoustic features through LLaMA-Adapter [59] tuning. We use the official implementation from `https://github.com/shansongliu/MU-LLaMA`, with the same hyperparameter settings: the input audio is split into 60-second audio signal at 16 kHz and the temperature for LLaMA-2-7B is set to 0.6, top_p is set to 0.8, and the maximum sequence length is 1024 tokens.

LP-MusicCaps [11] employs a BART-based encoder-decoder architecture [51] with 768 widths and six transformer blocks for both the encoder and the decoder, and the encoder takes a log-mel spectrogram with convolution layers similar to whisper [52]. We use the official implementation from `https://github.com/seungheondoh/lp-music-caps` and their pretrained checkpoint, splitting our test audio to 10-second audio signal at 16 kHz and choose the longest caption among all the clips as the inference result. In addition, the num_beams is set as five and the maximum sequence length is 128 tokens.

FUTGA[12] enables time-located music captioning by automatically detecting functional segment boundaries. Built upon SALMONN-7B [60] with LoRA-based instruction tuning, it integrates a music feature extractor for full-length music captioning. For our evaluation of this model we use the checkpoints and architecture presented by the original authors on `https://huggingface.co/JoshuaW1997/FUTGA`. In the implementation, Vicuna-7B `https://huggingface.co/lmsys/vicuna-7b-v1.5` is used as the backbone. For the hyperparameter settings, the repetition_penalty is set to 1.5, num_beams is set to 5, top_p is set to 0.95, top_k is set to 50, and an audio file is processed as 240-second 16kHz audio signal.

Text-to-Music Generation Models:

MusicLM [13] is a generative model that produces high-quality music from text prompts by using a hierarchical sequence-to-sequence approach. It leverages audio embeddings from a self-supervised model and autoregressively generates semantic and acoustic tokens. Unfortunately this model does

not have any publicly available architecture or checkpoints. However, we use a crowd-sourced implementation available at https://github.com/zhvng/open-musiclm. Notably, this implementation deviates from the originally proposed text conditioning model by using the open-sourced version of CLAP [7] instead of Mulan [61] and Encodec [62] instead of SoundStream [63]. The purpose of including this implementation is to showcase the performance of a large collection of publicly available models.

Stable Audio [15] is a diffusion-based music generation model that creates audio from text and optional melody input, using a latent audio representation. The Stable Audio model is based on a combination of a latent diffusion model consisting of a variational autoencoder, a conditioning signal, and a diffusion model. The VAE consists of a Descript Audio Codec [64] encoder and decoder. The textual conditioning signal comes from a pre-trained CLAP model [7], specifically the HT-SAT [56] and RoBERTa-based [65] iteration. Finally, the diffusion model is based on a U-Net [33] which consists of 4 levels down-sampling encoder blocks and up-sampling decoder blocks, with skip connections between them. encoder and decoder blocks providing a residual For our evaluation of this model we use the checkpoints and architecture presented by the original authors on https://github.com/Stability-AI/stable-audio-tools.

MusicGen [14] is a transformer-based model that generates music from text descriptions. In our implementation with use the 300M parameter model. This model uses a five layer EnCodec model for 32 kHz monophonic audio with a stride of 640, resulting in a frame rate of 50 Hz, an initial hidden size of 64 and a final embedding size of 640. The embeddings are quantized with using an RVQ with four quantizers, each with a codebook size of 2048. Finally, for sampling, the model employs top-k sampling, keeping the top 250 tokens and a temperature of 1.0. For our evaluation of this model we use the checkpoints and architecture presented by the original authors in https://github.com/facebookresearch/audiocraft.

AudioLDM2 [16] is a diffusion model for text-to-audio generation, trained on large-scale data and designed to handle diverse audio types, including music and sound effects. It improves over its predecessor by using high-quality representations and efficient training strategies. For our evaluation we use the checkpoints and architecture presented by the original authors in https://github.com/haoheliu/AudioLDM2. For the specific hyperparameters of the checkpoint architecture, we use the version with a 2-layer latent diffusion model. As their audio encoder the model uses a AudioMAE with a patch size of 16 ×16 and no overlapping, resulting in a 768-dimension feature sequence with length 512 for every ten seconds of mel spectrogram. For the text encoder there is a GPT-2 model that has an embedding dimension of 768 with 12 layers of transformers.

Mustango [27] is a multi-stage latent diffusion model that generates music from text prompts, focusing on both coherence and audio quality. It introduces a time-aware transformer to model long audio sequences and supports multi-track generation. For our evaluation we use the checkpoints and architecture presented by the original authors in https://github.com/AMAAI-Lab/mustango. During inference, the model uses two transformer-based text-to-music-feature generators which predict the beat and chord features. For the beats prediction, this model uses DeBERTa Large model [66] which predicts both the meter and the sequence of interval duration between the beats. Simultaneously, the chord predictions are made by a FLAN-T5 Large model [49].

F.4 Computational Resources

For generative tasks, all experiments were conducted on a system equipped with NVIDIA L40 GPUs with 48GB VRAM per card, utilizing 12.6. Each experiment was executed on a single GPU instance.

For retrival tasks, all experiments were conducted on a system equipped with NVIDIA A40 GPUs with 46GB VRAM per card, utilizing CUDA 12.4. Each experiment was executed on a single GPU instance.

F.5 Evaluation Metrics

F.5.1 Intuition for Interpreting Music-to-Text Metrics

In this section we present a brief overview for the metrics used for evaluating music-to-text models. Following the canonical works in music-to-text generation [10, 11] we begin by presenting three n-gram based metrics borrowed from machine translation tasks called BLEU [43], ROUGE [45]

and METEOR [44]. BLEU (B) uses precision to compare the overlap in n-grams (sequences of 1, 2, or 3 words - (B_1, B_3, B_3) between the original annotation and the generated musical caption. Alternatively, ROUGE (R) uses recall to compare the overlap in n-grams between the original annotation and the generated musical caption. Finally, METEOR (M) is designed to be better aligned with human judgments by extending the comparison to include synonym and paraphrasing-based matches in addition to the exact matches covered by BLEU/ROUGE. Meanwhile, we also include the CIDEr [46] metric which was originally proposed for image captioning. This metric measures how well the generated text matches the consensus of a set of original annotation, using a weighted n-gram similarity. Finally, we present the Bert Score [47] which uses the Bert model to compare the embeddings between the generated and original musical annotations.

The purpose of using each of these evaluation metrics is to present increasing levels of abstraction in considering the alignment between the original annotations and their generated counterparts. As we can see the Bert Score remains the most stable across all three datasets while the range of the n-gram based metrics maintains high variability between both datasets and models.

F.5.2 CLAP Score

Contrastive Language-Audio Pretraining [7] Score (CLAP Score) is a simple but effective and reference-free metric that quantifies how closely audio signal matches a text description. This metric is commonly used in text-to-music generation to evaluate how well a generative model is able to express the information provided in a textual input which forms the basis for its generation. Thus, given a set of associated language-audio pairs, $(T, \tilde{A})$ where the audio $\tilde{A} = \mathcal{M}(T)$ is generated by providing the associated textual inputs T to a music generation model (e.g. MusicGen [14]). We can generate embeddings for each modality using the CLAP model such that

$$Z_{\tilde{A}} = \text{CLAP}_{\text{audio}}(\tilde{A}), \quad Z_T = \text{CLAP}_{\text{text}}(T),$$

where $Z_{\tilde{A}}, Z_T$ are the output from the audio encoder and text encoder for the CLAP model, respectively. Given these sets of audio and text embeddings we can measure the cosine similarity of the audio and the text embeddings in their joint representation space. We slightly abuse the notation for indexing borrowing from the syntax used for coding matrices such that $Z_{\tilde{A}}[i]$ reflects the i-th embedding. Thus, we can formalize the CLAP Score as:

$$CS(T, \tilde{A}) = \frac{1}{n} \sum_{i=1}^{n} \frac{< Z_{\tilde{A}}[i], Z_T[i] >}{||Z_{\tilde{A}}[i]|| \cdot ||Z_T[i]||}.$$

As we can see, the more alignment there is between the language and audio representational spaces, the higher this score will be.

F.5.3 FAD/FAD∞

Intuitively, the FAD measures the distance of the mean and covariance of embeddings between the real (as formalized by some predefined body of reference data) and generated audio, extracted using a reference model. In this work we compare the results from two different variations of FAD to demonstrate the complexity of objectively evaluating the quality of audio outputs which we denote as FAD [67] and FAD∞ [68]. Recent work [68] has postulated a criticism of the traditional metric, explaining that varying the reference model and data yields drastically different results. Thus, we showcase our results by evaluating over several reference models (indicated by the subscript, where V, M, E corresponds to VGG [69], MERT [50], and Encodec [62], respectively) and reference datasets (indicated by the superscript, where MC and FMA refers to MusicCaps[13] and Free Music Audio Dataset (FMA) [39], respectively). Similar to the findings presented by Gui et al. [68], we see that the values calculated using VGG, MERT, and Encodec demonstrate significant differences between competing models (often by a factor of x100).

F.6 Dataset Splits

For each of our evaluations on MusicCaps [13] and Song Describer [22], we evaluate over the entire dataset that is currently publicly. This choice is justified by the fact that neither dataset has openly published concrete train-test splits which can be used to standardize over models. For example,

although in the original MusicCaps paper they address the existence of a test set, on the publicly available version of their dataset released on Kaggle there is only a training split. Thus, in many of the works which evaluate on MusicCaps, they simply create a synthetic test set by implementing their own train-test split over the available data [27, 12, 11]. And, without testing over the entire set, we cannot ascertain their performance. Unfortunately, the possibility of overfitting cannot be accounted for without leaderboard access to the held-out test set. The same holds for Song Describer. Loading from the storage site does not yield any clear demarcation of the dataset meaning that each paper that evaluates on this dataset selects its own split. Since in our work we do not engage in any fine-tuning, we felt it was best to evaluate over the entire set and see the final performance. Meanwhile, for our dataset, which has a clearly demarcated evaluation set, we use only this portion of the data for evaluation and publish the rest for training.

G User Privacy Safeguards

Here, we specifically address the sensitive nature of releasing data that is scraped from the internet. In our work we release a large collection of Reddit threads which were scraped from the internet. While we understand that releasing data which is scraped from Reddit can have lasting impacts, we do our best to mitigate these. First, since the domain of our dataset is music, we are not dealing with a safety-critical setting. Second, although the original raw posts contain the user ids, we do not release these in the final version of our dataset. Finally, given the already anonymous nature of Reddit, we hope that our scraped posts will cannot be used to identify specific users.