UNSUPERVISED LEAD SHEET GENERATION VIA SEMANTIC COMPRESSION

Zachary Novack *Nikita Srivatsan* *Taylor Berg-Kirkpatrick* *Julian McAuley*

University of California San Diego

ABSTRACT

Lead sheets have become commonplace in generative music research, being used as an initial compressed representation for downstream tasks like multitrack music generation and automatic arrangement. Despite this, researchers have often fallen back on deterministic reduction methods (such as the skyline algorithm) to generate lead sheets when seeking paired lead sheets and full scores, with little attention being paid toward the quality of the lead sheets themselves and how they accurately reflect their orchestrated counterparts. To address these issues, we propose the problem of *conditional lead sheet generation* (i.e. generating a lead sheet *given* its full score version), and show that this task can be formulated as an unsupervised music compression task, where the lead sheet represents a compressed latent version of the score. We introduce a novel model, called Lead-AE, that models the lead sheets as a discrete subselection of the original sequence, using a differentiable top-k operator to allow for controllable local sparsity constraints. Across both automatic proxy tasks and direct human evaluations, we find that our method improves upon the established deterministic baseline and produces coherent reductions of large multitrack scores.

Index Terms— symbolic music generation, lead sheet generation, transformers, latent variable models, music reduction

1. INTRODUCTION

With the goal of communicating high-level musical information quickly and efficiently, **lead sheets**, or compressed scores that contain the musically important notes of a larger arrangement and high-level chord symbols (see Fig. 1), are one of the most common musical representations used by modern musicians. Recently, many in the larger AI Music community have looked to using lead sheets for their efficient representation, whether for generating lead sheets from scratch [1, 2, 3] or using lead sheets as input to larger music generation and arrangement systems [4, 5, 6, 7]. Despite this growing interest, there is a data sparsity issue, with few publicly available datasets containing high quality lead sheets, and fewer still having paired lead sheet–full arrangement data [8]. This has led many researchers to fall back to deterministic reduction methods like the skyline algorithm when needing paired score–lead sheet data [3, 6, 9], and to use these heuristic algorithms for constructing datasets themselves [8]. Furthermore, this trend has exacerbated an over-reliance in generative music on *Pop* music generation and arrangement, which contains simpler musical structures that algorithms like skyline have shown strong results on [3, 10, 11], as opposed to more complex music like classical or jazz. Considering this lack of attention to the quality of the ground truth lead sheets themselves, this raises the question: how can we model compressed music (i.e. lead sheets) that captures the important information from full scores across diverse genres?

In this work we thus propose the problem of *conditional lead sheet generation*, where the goal is to generate the lead sheet given the corresponding full score. Given the lack of paired lead sheets and full scores, we design our approach around the intuition that lead sheets carry a sparse selection of notes with the highest information content; thus, we can aim to select a subset of notes and chords from the original score that maximize reconstruction of the original score, making the process entirely unsupervised. Inspired by work in the text domain on discrete latent variable modeling [12, 13, 14], we propose **Lead-AE**, a novel Auto-Encoder–like setup wherein the encoder leverages a transformer encoder with a differentiable top-k operator to target specific notes of the input piece, which are then passed to an encoder-decoder transformer to regenerate the initial sequence. On automatic proxy evaluations through reconstruction and melody recovery, as well as both audio and sheet music–based human evaluations, Lead-AE consistent outperforms the strong deterministic baseline. Our proposed model opens up new avenues for work in conditional lead sheet generation for multitrack music, and can potentially improve the capabilities of existing generation and accompaniment models. Our source code is available at github.com/zacharynovack/lead-ae.

2. RELATED WORK

A number of previous works have focused on *unconditional* lead sheet generation (i.e. from scratch), using both traditional RNN architectures [1, 2] and modern transformers [2, 3, 15, 7]. Here, the authors generally rely on either the Wikifonia dataset or the WJazzD [16] dataset, which both only contain lead sheets with no paired scores. Recently, there has been a number of works using lead sheets as the first step in a larger

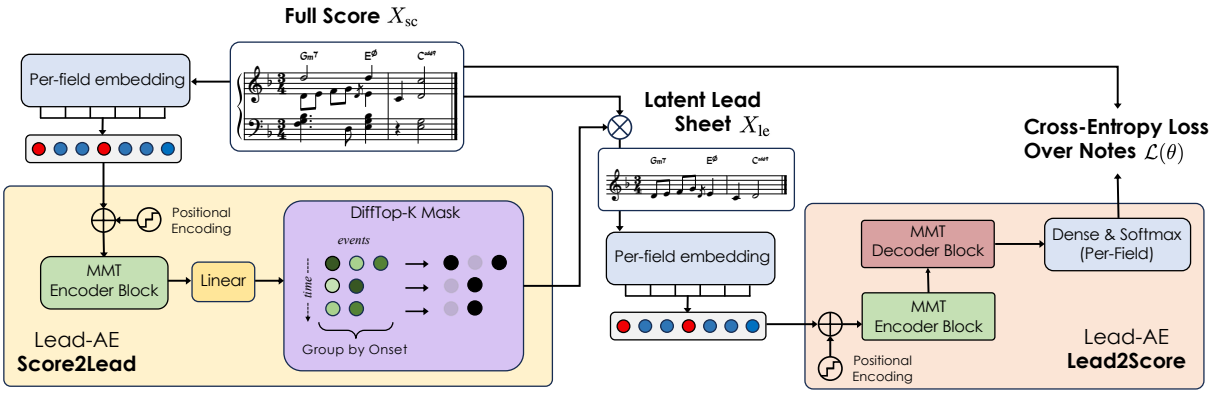

Fig. 1. Overview of Lead-AE. Score2Lead uses a transformer encoder and a differentiable top-k mask to select specific notes and chords from the sequence. Lead2Score uses an encoder-decoder transformer to recreate the score given the lead sheet.

music generation system, in both long-form structured music generation [6] and accompaniment [4, 5]. In all such works, the lead sheets only serve as an initial "planning" step for unconditional generation, and are not concerned with how well the lead sheet directly summarizes the full score output.

Using discrete latent spaces where the input and latent representations share the same structure (e.g. text) has been well-studied in the NLP community, being used to improve translation tasks [13], story generation [12], and text debiasing [14]. To our knowledge, ours is the first work to extend this framework to the symbolic music domain, directly modeling our latent space as a symbolic music sequence (rather than an abstract continuous or vector-quantized space).

3. PROPOSED METHOD

3.1. Data Representation

We follow the data representation put forth by the state-of-the-art Multitrack Music Transformer (MMT) [17] for efficient modeling of dense multitrack music. Each note is represented as a multidimensional input with type, beat, position, pitch, duration and instrument fields, which reduces the number of tokens needed to represent a given polyphonic sequence (as opposed to tokenizations like REMI+ [18]), and maintains a 1-to-1 mapping from each musical note to a given token (see [17] for more detailed information).

In order to model the symbolic chord labels present in lead sheets, we use an off-the-shelf chord extractor based on state-of-the-art work in chord recognition [19, 20]. We assume an overly dense set of chord labels, with chord tokens extracted for every beat, and thus part of the learning problem also includes selecting which chords to keep. Every chord $\mathbf{c}_i$ is represented as $\mathbf{c}_i = (c_i^{\text{type}}, c_i^{\text{beat}}, c_i^{\text{position}}, c_i^{\text{root}}, c_i^{\text{quality}})$, where c_i^{root} denotes the pitch class and c_i^{quality} denotes the chord type (see Fig. 1 for an example of chord symbols).

3.2. Learning Objective

Conditional Lead Sheet Generation is at its core a sequence translation problem, with the goal of converting a given multitrack music sequence X_{sc} to a reduced lead sheet sequence X_{le}, where X_{sc} is the set of original notes $\mathbf{x}_i$ and densely notated chords $\mathbf{c}_i$ and $X_{\text{le}} \subseteq X_{\text{sc}}$. As lead sheets reduce the total number of events from the full score yet maintain a consistent *local* coherence in terms of information density, we require that X_{le} resides in the constraint set $\mathcal{C}_k(X_{\text{sc}})$:

$$\mathcal{C}_k(X_{\text{sc}}) = \{X_{\text{le}} \subseteq X_{\text{sc}} : |X_{\text{le}}[i \in o]| = k, \forall o \in O\}, \quad (1)$$

i.e. that for every unique onset $o \in O$ in the score, X_{le} can only include k events that share each onset.

To our knowledge there are no large scale datasets of paired lead sheets and full scores available aside from the POP909 [8] dataset, which only contains around 1K songs, is limited to pop music, and only has automatically-labeled chords. In order to circumvent this, we note that lead sheets can be viewed as a form of "musical compression": while the form of the data is the same, lead sheets are *semantically* compressed, containing only the most important parts of the original score for easy communication. Thus, we can formulate conditional lead sheet generation as an *unsupervised* compression task, with the aim of learning a Score2Lead encoder to squeeze X_{sc} into a latent reduction X_{le} such that a learned Lead2Score decoder can reconstruct X_{sc} as accurately as possible, with the following objective:

$$\mathcal{L}(\theta) = -\mathbb{E}_{P_{\text{enc}}(X_{\text{le}}|X_{\text{sc}})}[\log P_{\text{dec}}(X_{\text{sc}} \mid X_{\text{le}})], \quad (2)$$

which is equivalent to the reconstruction term from the standard VAE objective. Note that this expectation is not tractable, as it requires summing over the combinatorial space of lead sheets X_{le} from P_{enc} that are in $\mathcal{C}_k$; however, we show in the next section that we can approximate this expectation using a differentiable top-k operator.

3.3. Lead-AE Formulation

Score2Lead Architecture. Lead sheet generation requires a model to learn which of the N notes and C extracted chords should be included in the latent lead sheet to maximize score reconstruction. We use an encoder Multitrack Music Transformer (MMT) [17] module, where the model attends to the *entire* score input, by taking in the multidimensional sequence as the sum over each field's learned embeddings, with a learnable absolute positional embedding [21]. The latent outputs $\mathbf{h} \in \mathbb{R}^{(N+C)\times d}$ (where d is the latent dimension) are then passed through a linear layer, that squeezes each latent representation into a single score per token $s_i \in \mathbb{R}$.

Objective Approximation. Like in VAEs, the expectation in Eq. 2 is intractable. In order to handle this as well as the local sparsity constraints from Eq. 1, we approximate the top-k distribution in each onset using a differentiable top-k operator. Specifically, after grouping the sequence of s_i by shared note onsets, we can apply the top-k operator from [22], amounting to iterative applications of the Gumbel-Softmax trick [23] along each onset to produce approximate top-k probabilities. We then use the straight-through estimator to convert the soft probabilities to a hard mask, and multiply this with our initial input sequence to generate our latent lead sheet X_{le}, which is gauranteed to obey Eq. 1 (see the left of Fig. 1 for an overview of the system). We enforce the entire lead sheet to have the same unified instrument to further compress the musical representation. At inference time, we deterministically select the top-k notes per onset.

Lead2Score Architecture. The L2S module follows a standard encoder-decoder transformer [21] using the MMT framework [17], where X_{sc} is autoregressively generated given the latent lead sheet X_{le}. Thus, the entire Lead-AE system is trained to minimize $\mathcal{L}(\theta)$ across each field.

Module Pretraining. Training the entire Lead-AE system end to end from scratch is difficult, as the approximate gradients of the discrete latent lead sheet can throw off early training. Thus, we warm start Lead-AE with the skyline algorithm (which selects the highest pitch note at each onset), using the skyline reduction as the input for L2S and as a supervised target mask for S2L.

4. RESULTS

4.1. Experimental Setup

Though an optimal assessment of our model would involve evaluating Lead-AE by how well the latent lead sheet matches human–produced lead sheets, no datasets exist to our knowledge that have paired scores and human–annotated lead sheets *including* chords. We thus run our automatic evaluation over two proxy tasks: the overall reconstruction accuracy of the full score using the Symbolic Orchestral Database (SOD) [24], and how well Lead-AE can recover human–annotated *melodies* from POP909 [8]. Beyond this, we turn to two parallel human evaluations, which are able to directly assess the quality of Lead-AE's outputs. SOD contains 357 hours of multitrack orchestral music, while POP909 contains 60 hours of Chinese pop songs. We adopt an 0.8/0.1/0.1 train-val-test split for both datasets. Each MMT block in Lead-AE has 4 attention layers with a hidden dimension of 512 and 8 attention heads. We use a maximum sequence length of 1024 and a maximum beat of 256. For all experiments, we randomly augment the data by shifting the starting beat and pitch shifting the sequence within a ± 6 semitone range. All models are trained for a maximum of 1000 epochs, or if the validation performance stagnates for 20 epochs. For inference, we use top-k sampling and enforce a monotonicity constraint for the type and beat fields to ensure coherent outputs.

4.2. Automatic Evaluation

We first evaluate Lead-AE's performance on reconstructing the original score from the learned lead sheets (as a proxy for evaluating the lead sheets directly). We use the deterministic skyline algorithm with all chords included as a baseline (i.e. just training an MMT encoder-decoder model with the fixed skyline lead sheets as input), as it has shown competitive results in melody extraction [3, 10]. For Lead-AE (LAE), we experiment with two varieties: one that maintains the same local sparsity as the skyline algorithm at $k = 1$ event per onset (or 2 if there is a chord present), and a more flexible model that uses a *fractional* k at each onset, taking $\lceil 10\% \rceil$ of the notes at a given onset to allow the model to adapt to the local density of the score. Given previous work on music translation [25], we report the reconstruction MuTE metric, which measures the F1 score per time-step, as well as the global Jaccard (Jac.) similarity, between the original and reconstructed (R) sequences. We additionally report both metrics over the output *pitch-classes* (PC), which compresses the sequence to a single octave. In Table 1, Lead-AE outperforms the skyline model with the same sparsity constraint in all metrics, and notably the variable sparsity model (which includes slightly higher note density and lower chord density) dominates both skyline baseline and the fixed sparsity Lead-AE. Thus, we use the variable sparsity model for all further evaluations.

We also evaluate how well Lead-AE accurately recovers the ground truth melodies in POP909 without seeing them during training. Since POP909 has long phrases of each song with no melody, we focus specifically on the melody–present sections, as both Lead-AE and skyline are designed to pick *which* notes should be included, not whether notes should be included at all. We finetune both Lead-AE ($k = 0.1$) and the skyline baseline on POP909, and report the MuTE scores between the generated *lead sheet* (excluding chords) and the ground truth melodies. In Table 2, Lead-AE performs as good as skyline (which has strong performance, echoing [3, 10]) in recovering the ground truth melodies.

Model	Note Density	Chord Density	(R) MuTE	(R) PC-MuTE	(R) Jac.	(R) PC-Jac.
Skyline	37%	100%	59.95	75.76	42.75	63.92
LAE ($k = 1$)	37%	100%	<u>62.14</u>	<u>78.21</u>	<u>45.21</u>	<u>67.00</u>
LAE ($k = 0.1$)	39%	95%	**64.04**	**79.48**	**46.59**	**67.92**

Table 1. Average quantitative results on SOD test set across 3 random seeds, showing the global note and chord densities of the latent lead sheets, and the MuTE and Jaccard results between the original and reconstructed scores.

Model	(LS) MuTE	(LS) PC-MuTE
Skyline	77.93	81.86
LAE ($k = 0.1$)	**77.94**	**82.04**

Table 2. MuTE scores on POP909 between latent lead sheets and the ground-truth melodies.

4.3. Human Listening Study

While the automatic metrics presented are informative, it is worth noting that they are at best a proxy for human judgement. Therefore, in order to assess the quality of the generated lead sheets *directly*, we conducted two rounds of human evaluation where annotators were asked to rank the outputs of our system against the skyline baseline based on an audio rendering and the sheet music respectively. For the listening study, we recruited 12 participants at varying levels of musical maturity (from novice to professional musicians). We randomly selected 11 short excerpts, all from the validation set of SOD. For each excerpt, users were asked to listen to an audio rendering of the full score, and then were presented both the audio corresponding to the skyline lead sheet and our Lead-AE–generated lead sheet. We rendered all chord symbols in root position starting from C3, and all scores were recorded using the same piano patch no matter the instrument, in order to remove timbre differences from the evaluation. The systems were not labeled and their order was randomized.

Participants were asked to rank the lead sheets along two parallel criteria: **(1)** Which lead sheet more *accurately* captured important musical information from the original score? **(2)** Which lead sheet sounded more musically *fluent* and coherent, irrespective of the original? They were also allowed to answer that the two were too similar to distinguish, although this was discouraged. As shown in the left half of Table 3, listeners strongly preferred Lead-AE's reductions over the skyline, and only rarely felt that they were too similar to distinguish. The proportions were similar for both accuracy and fluency. Per-song an average of 62.1% of annotators agreed with the majority answer for accuracy and 60.6% for fluency, indicating a reasonable degree of inter-annotator agreement.

4.4. Human Reading Study

While assessing lead sheet quality based on an audio rendering is useful as music is experienced aurally, lead sheets are generally intended to be *read* by musicians rather than learned by ear. Furthermore, sheet music allows an experienced musician to quickly analyze a piece based on learned visual patterns and compositional conventions without having to frequently relisten. Therefore to supplement the listening study, we also conducted a round of human evaluation in which annotators were asked to rank the lead sheets based on the sheet music alone. In some ways this can be considered a closer simulation of the eventual downstream use case for this task. For this *reading* study we consulted 6 professional musicians, and presented them with 5 4–bar excerpts, also taken from the SOD validation set. They were similarly asked to rank Lead-AE against skyline based on both accuracy and fluency.

Model	Listening		Reading	
	Acc.	Flu.	Acc.	Flu.
Skyline	26.5	28.0	6.7	20.0
LAE ($k = 0.1$)	**57.6**	**56.8**	**80.0**	**56.7**
Too similar	15.9	15.2	13.3	23.3

Table 3. Results from the human studies, showing proportional annotator preference for our system against the skyline.

On the right half of Table 3 we show the results from this evaluation. We see that human experts once again preferred our system, and by an even larger margin on the criteria of accuracy. An average of 80.0% of annotators agreed with the majority on accuracy, and 56.7% for fluency. When asked to justify their answers, annotators often stated that the skyline omitted important melodic or harmonic information which Lead-AE was better able to capture. The results of this reading study along with the listening study above corroborate our findings via the automatic metrics and indicate that Lead-AE is capable of generating significantly more accurate and functional lead sheets compared to the skyline baseline.

5. CONCLUSION

In this work we have presented Lead-AE, a novel architecture for modeling conditional lead sheet generation without relying on paired lead sheet–full score data. On both automatic and human evaluations, Lead-AE consistently improves upon the determinstic baseline, and thus may inspire new directions for generating higher quality lead sheets and improve the downstream quality of larger arrangement systems.

6. REFERENCES

[1] Cedric De Boom, Stephanie Van Laere, Tim Verbelen, and Bart Dhoedt, "Rhythm, chord and melody generation for lead sheets using recurrent neural networks," in *Machine Learning and Knowledge Discovery in Databases: ECML PKDD*. Springer, 2020.

[2] Dimos Makris, Kat R Agres, and Dorien Herremans, "Generating lead sheets with affect: A novel conditional seq2seq framework," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021.

[3] Shih-Lun Wu and Yi-Hsuan Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," *arXiv preprint arXiv:2008.01307*, 2020.

[4] Jingwei Zhao and Gus Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," *arXiv preprint arXiv:2108.11213*, 2021.

[5] Li Yi, Haochen Hu, Jingwei Zhao, and Gus Xia, "Accomontage2: A complete harmonization and accompaniment arrangement system," *arXiv preprint arXiv:2209.00353*, 2022.

[6] Shih-Lun Wu and Yi-Hsuan Yang, "Compose & embellish: Well-structured piano performance generation via a two-stage approach," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[7] Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," *arXiv preprint arXiv:2109.00663*, 2021.

[8] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia, "Pop909: A pop-song dataset for music arrangement generation," *arXiv preprint arXiv:2008.07142*, 2020.

[9] John Thickstun, David Hall, Chris Donahue, and Percy Liang, "Anticipatory music transformer," *arXiv preprint arXiv:2306.08620*, 2023.

[10] Yi-Hui Chou, I Chen, Chin-Jui Chang, Joann Ching, Yi-Hsuan Yang, et al., "manrt-piano: large-scale pretraining for symbolic music understanding," *arXiv preprint arXiv:2107.05223*, 2021.

[11] Yo-Wei Hsiao and Li Su, "Learning note-to-note affinity for voice segregation and melody line identification of symbolic music data.," in *ISMIR*, 2021, pp. 285–292.

[12] Harsh Jhamtani and Taylor Berg-Kirkpatrick, "Narrative text generation with a latent discrete plan," *arXiv preprint arXiv:2010.03272*, 2020.

[13] Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick, "A probabilistic formulation of unsupervised text style transfer," *arXiv preprint arXiv:2002.03912*, 2020.

[14] Fatemehsadat Mireshghallah and Taylor Berg-Kirkpatrick, "Style pooling: Automatic text style obfuscation for improved classification fairness," *arXiv preprint arXiv:2109.04624*, 2021.

[15] Yi Zou, Pei Zou, Yi Zhao, Kaixiang Zhang, Ran Zhang, and Xiaorui Wang, "Melons: generating melody with long-term structure using transformers and structure graph," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[16] "The weimar jazz database," https://jazzomat.hfm-weimar.de/.

[17] Hao-Wen Dong, Ke Chen, Shlomo Dubnov, Julian McAuley, and Taylor Berg-Kirkpatrick, "Multitrack music transformer," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[18] Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann, "Figaro: Generating symbolic music with fine-grained artistic control," *arXiv preprint arXiv:2201.10936*, 2022.

[19] Junyan Jiang, Ke Chen, Wei Li, and Gus Xia, "Large-vocabulary chord transcription via chord structure decomposition.," in *ISMIR*, 2019, pp. 644–651.

[20] Brian McFee and Juan Pablo Bello, "Structured training for large-vocabulary chord recognition.," in *ISMIR*, 2017, pp. 188–194.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[22] Sang Michael Xie and Stefano Ermon, "Reparameterizable subset sampling via continuous relaxations," *arXiv preprint arXiv:1901.10517*, 2019.

[23] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[24] Léopold Crestel, Philippe Esling, Lena Heng, and Stephen McAdams, "A database linking piano and orchestral midi scores with application to automatic projective orchestration," in *ISMIR*, 2017.

[25] Matan Gover and Oded Zewi, "Music translation: Generating piano arrangements in different playing levels," in *ISMIR*, 2022.