SCALING ANTICIPATORY MUSIC TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We seek to build capable generative models for symbolic music at the largest scale supported by the limited data available. To this end, we apply recently proposed *data-constrained scaling laws* to symbolic music and conduct an empirical scaling study on an improved variant of the anticipatory music transformer. The study informs the training configuration for a 1.14 billion parameter model, which achieves state-of-the-art likelihood metrics on the Lakh MIDI dataset and exhibits strong performance in human evaluation, even against closed source commercial systems like Suno v6. The model demonstrates strong extrapolative performance on a new test set of music composed in 2026, and exhibits intriguing long-context generative capabilities. However, the model's final test loss diverges from the scaling prediction, raising methodological questions about data-constrained scaling laws, at least in the symbolic music domain. We release the model checkpoint, source code, scaling law runs, and the aforementioned 2026 test set, and emphasize that all training data are fully open.

1 INTRODUCTION

Generative models for symbolic music are bottlenecked by limited data. One of the largest public research datasets, GigaMIDI (Lee et al., 2025), contains fewer than 1.5 million documents. By contrast, the FineWeb (Penedo et al., 2024) dataset of natural language has over 50 billion documents and its super set, Common Crawl, has over 300 billion (Common Crawl Foundation, 2026). In computer vision, ImageNet contains over 14 million images (Deng et al., 2009) and Re-LAION-5B has 5.5 billion image-text pairs (LAION, 2024). Although there are domains with even less data, a 1.5 million document corpus means symbolic music is constrained not by compute, but by data. Our practical question is then, *how do we get the most from the data we have?*

To best leverage the data, we improve the generative modeling recipe for symbolic music from datasets and data mixtures to tokenization, model architecture, and optimizers. We first modify the data representation proposed in the anticipatory music transformer (AMT-1) (Thickstun et al., 2024) and show our changes improve likelihood-based metrics. Specifically, we resolve a problematic interdependency between sequence content and vocabulary size and impart new control capabilities by a modified form of anticipation. Next, we gather proven, domain-agnostic improvements from frontier-scale generative models and the GPT-2 speedrunning community (Karpathy, 2025) to construct an architecture that further increases performance in fewer steps.

We show by IsoFLOP analysis (Hoffmann et al., 2022) that AMT-1 was severely data-constrained. In addition to consolidating some of the largest public datasets available for this work, our result shows that infinite-data style scaling laws are inappropriate for symbolic music. Thus, our domain mandates using 'data-constrained scaling laws', leading us to fit all loss prediction equations proposed in Muennighoff et al. (2023) and Lovelace et al. (2026). We use their proposed scaling law forms to predict the loss minimizing configuration in the multi-epoch regime. While our results support the claim in Lovelace et al. (2026) that an additive penalty is more accurate than exponential decay for predicting loss in this setting, the scaling law forms conflict with one another. Our fitted data-constrained scaling law equations disagree in their extrapolations and in their qualitative descriptions of scaling behavior.

Nevertheless, we use this scaling analysis to obtain the **Anticipatory Music Transformer 2 (AMT-2)**, a 1.14 billion parameter symbolic music model. We find that the scaling law equation that best predicts the loss of this model does not have the best goodness of fit metrics, and is not the suggested

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

form proposed in Lovelace et al. (2026). This evaluation-result mismatch raises methodological questions about how best to evaluate data-constrained scaling laws, at least in the symbolic music domain. Regardless of these difficulties forecasting model quality, this model achieves state of the art likelihood metrics on the test split of the Lakh MIDI dataset (Raffel, 2016) and exhibits a clear win against its predecessor AMT-1 while tying larger models like MuPT (Qu et al., 2024) and the commercial closed source music generation platform Suno.

We mitigate train/test contamination in our evaluations by curating a new test dataset of symbolic music written after June 4th, 2026, we which use as conditioning prompts for our human evaluations and in our automatic evaluations. Moreover, we find that the distributional divergence metric for symbolic music, the Fréchet Music Distance (FMD) (Retkowski et al., 2025) does not correlate with our human evaluations, which raises concerns about its use as a standard metric in our domain. This leads us to a less obvious, yet effective protocol of using the Mauve Audio Divergence (MAD) Huang et al. (2025) as a symbolic music metric. We find that the MAD better correlates with human evaluation results.

Taken together, our work presents an advance of the practice of training, scaling, and evaluating symbolic music generative models. We release our code, our new evaluation dataset, our scaling law fitting runs, and our 1.14 billion parameter model checkpoint. We emphasize that all training data and evaluations are open, which is not the case for some models we compare against.

2 Related work

We focus on symbolic music with note-wise conditioning because it affords composers precision, flexible editing after generation, and creative freedom in sound design, aligning it more to preserving human agency (Chen, 2025). Unlike an audio-based generative model (Dhariwal et al., 2020; Copet et al., 2023; Agostinelli et al., 2023), a symbolic music model returns a sequence of notes instead the sound they make. Consequently, note editing is trivial in symbolic representations like MIDI whereas in audio, note editing is not lossless (Celemony Software GmbH, 2026; Cano et al., 2019) and likened to 'unbaking a cake' by audio engineers (Bernsee, 2015). Complementary to control paradigms like *text-to-music* (Melechovsky et al., 2024; Wu et al., 2026; Suno, Inc.) and *image-to-music* (Kim et al., 2025b; Yi et al., 2026), *notes-to-notes* generation remains musically explicit (Thickstun et al., 2024) when descriptive adjectives are elusive (O'Toole, 2010; Talbert, 1979).

3 Data representation, data mixtures, and modeling

3.1 Symbolic music tokenizers

Because we are data-constrained, a symbolic music tokenizer should be permissive in what it can express. This motivates two related design choices: first, we use MIDI as our primary input and second, we avoid detecting musical semantics. Notes expressed in common symbolic music formats like ABC notation or Scores can be converted into MIDI but the reverse direction can be non-trivial (Cuthbert & Ariza, 2010). This makes MIDI a reasonable choice for maximizing corpus size. The reverse conversion from MIDI to Scores can be challenging because it requires inferring potentially ambiguous musical semantics (Murgul & Heizmann, 2025; Chen & Su, 2021). Numerous symbolic music tokenizers start with musically rich file formats and encode their contained semantics (e.g. meter or chords) directly in the vocabulary (Ren et al., 2020; Hsiao et al., 2021) primarily for conditional generation (Guo & Dixon, 2025; Qian et al., 2026; Pasquier et al., 2025). While sensible, these 'musically-aware' tokenizers must reliably detect these attributes if their input files do not state them, which is brittle for expressive MIDI and ultimately limits their coverage. Indeed, some tokenizers discard samples if meter detection heuristics fail (Qian et al., 2026).

We further seek a tokenization that supports diverse control patterns such as *in-filling* in a single design. Prior work bifurcates the vocabulary by maintaining distinct tokenizations for continuation and in-filling (Pasquier et al., 2025), use distinct tokenizations for single-track and multi-track music (Qian et al., 2026), require meter detection (Guo & Herremans, 2025), or they allow chord-level, but not not-level conditioning (Guo & Dixon, 2025). In contrast to prior work, we wish to simultaneously: control single-track and multi-track music generation at the note level without requiring music



Figure 1: **Tokenized music with block anticipation.** The top plot depicts notes and instruments in temporal order. The bottom plot shows when the temporal and sequential order diverge, i.e., when we apply anticipation. The sequence is 1024 tokens long and the anticipation interval δ is set to 10. On the bottom plot, we emphasize that the distance in sequence indices between the events (squares) and controls (diamonds) corresponds to a time difference of roughly δ, per the block-wise anticipatory order.

information retrieval in a single, unified design. We therefore build upon the tokenization proposed in Thickstun et al. (2024), which meets these criteria.

3.2 Time Relativization

AMT-1's tokenization couples the vocabulary size and time representable in a sequence, which is problematic for expressivity and long-context fine-tuning. The procedure assigns a token to each discretized onset time, meaning that a sequence containing times beyond what the vocabulary can express must be discarded or the vocabulary must increase. The latter work-around is impractical because one must retrain the model, but more subtly, a larger vocabulary relative to the hidden dimension d_{ff} in the final layer of a language model bottlenecks expressivity and gradients (Yang et al., 2018; Godey & Artzi, 2026). To overcome this problem, we use time relativization.

Time relativization enables a symbolic music tokenizer to express arbitrary onset times in a compact vocabulary. The idea is simple: introduce a `TICK` token that serves as a temporal reference post and define all subsequent onset times relative to it. For example, a sequence of absolute onset times may be $[0, 120, 270]$, but with time relativization with a time resolution of 100, the sequence becomes: $[\texttt{TICK}, 0, \texttt{TICK}, 20, \texttt{TICK}, 70]$. Consequently, time relativization bounds the onset vocabulary to the time between two ticks. This relativizing idea is inherent in music notation and is one of the main design choices in Qian et al. (2026); Pasquier et al. (2025). In contrast to these works, we place tick tokens at constant intervals of seconds, *not* musical beats or bars – this obviates meter detection and its challenges. Qualitatively, our large model, AMT-2 1.14B, produces expressive gestures of metric slow-down (*ritardando*) and speed-up (*accelerando*) without explicitly encouraging this behavior.

3.3 Anticipation

Anticipation is a token ordering that enables autoregressive models to condition on the *temporal* future by moving tokens *sequentially* backwards (Thickstun et al., 2024). An *anticipated* token is pushed backwards in sequence until it sits adjacent to tokens that occur approximately δ seconds

Table 1: **Our proposed sequence representation beats AMT-1 on the same dataset and architecture.** All models use the AMT-1 small configuration in Thickstun et al. (2024). Lower time-normalized Bits per Second (BPS) is better (Thickstun et al., 2019). The first two rows are from the original AMT-1 work and the bottom two use our proposed tokenization. We explain time resolution in Appendix A.1.

All Params	Non-Embedding	Vocab	Step	Tokenization	Time Resolution	BPS ↓
128m	85m	55k	100k	AMT-1	100	80.4
128m	85m	55k	100k	AMT-1	100	80.7
112m	58m	35.2k	100k	ours	100	78.1
112m	58m	35.1k	100k	ours	96	**77.5**

before it. With this design, causal-masked generative models retain the temporal past and future in their context as they tractably sample the present, allowing them to 'fill-in-the middle' (*in-filling*). To distinguish conditioning information from events, anticipated tokens use a distinct range in the vocabulary. Accordingly, we call the anticipated tokens *controls*, and past or present time tokens *events*.

Time relativization changes how anticipation works and enables new control patterns. Whereas AMT-1 anticipation interleaves events and controls according to a stopping rule, block anticipation places controls immediately after each tick and before any events following that tick. We call the group of [TICK, controls..., events...] a *block*, and it is analogous to a localized realization of seq2seq learning (Sutskever et al., 2014). Block anticipation places controls such that the events they condition are δ seconds away from the tick preceding the first control.

Cold start, simultaneous accompaniment and in-filling

AMT-1 does not support anticipation when there are no events in the context. This means anticipation cannot be used until one writes the song up to the point of generation – a restriction that inhibits idea generation. This is a consequence of the method of sequence packing the training data, which we fix by introducing such 'cold start' sequences deliberately. AMT-1 supports in-filling and accompaniment but not both simultaneously. The limitation arises similarly because sequences that exhibit both are not expressed in the training data. We remedy this by first deciding a subset of instruments to anticipate, and then uniformly sampling a time $> \delta$ in the current chunk and anticipating the subsequent tokens. These samples enable one to condition on both on an instrument and generate connecting music between sections. Figure 1 depicts a sample that designates drums as a control and also anticipates a span.

3.4 Data representation result

Table 1 shows our changes to time representation and anticipation improves the time-normalized Bits per Second (BPS) (Thickstun et al., 2019) by roughly 3.2 when all other settings are held constant. BPS is an automatic likelihood-based metric that is encoding agnostic, defined as the per-token cross entropy loss in nats per token normalized by the seconds in the evaluation set.

3.5 Model architecture and optimizer

We incorporate proven designs from language models into our symbolic music model. We start with a LLaMA 2 (Touvron et al., 2023)[1] architecture and detail our changes in Appendix A.5. Inspired by its impressive convergence speed for smaller models (Wen et al., 2025; Semenov et al., 2025), we use the Muon (Jordan et al., 2024) optimizer for matrix parameters in tandem with AdamW (Loshchilov & Hutter, 2019) for scalar parameters, as is the standard setting. We use the Moonlight RMS scale matching implementation of because it allows re-use of AdamW weight decay and learning rates, is stable up to frontier scale, and exhibits predictable scaling behavior (Liu et al., 2025). Additional details are also documented in Appendix A.5.

[1] In their own words, the LLaMA 3 architecture "does not deviate significantly" from LLaMA 2 (Grattafiori et al., 2024). Thus, LLaMA 2 is still a modern architecture even though it is not the latest iteration.

Table 2: **Updated architecture and optimizer improves likelihood-based metrics.** We surpass the best AMT-1 model in BPS. To control for the impact of extending the context from 1024 to 4096, hold the number of tokens processed per-optimizer step constant by reducing the batch size. The final two rows are obtained from the same model. During evaluation, we truncate the context 1024, which preserves the impact of the 'early token curse' (Press et al., 2021). We include the BPS achieved by the model when given full 4096 context in the final row but delineate it from smaller context results with a dotted line.

All Params	Non-Embedding	Vocab	Step	Tokenization	Context	BPS ↓
360m	302m	55k	100k	AMT-1	1024	74.4
360m	302m	55k	200k	AMT-1	1024	71.5
360m	302m	55k	800k	AMT-1	1024	69.7
780m	708m	55k	100k	AMT-1	1024	73.2
139m	84m	35.1k	400k	ours	1024	69.9
380m	308m	35.1k	100k	ours	1024*	**66.0**
380m	308m	35.1k	100k	ours	4096	51.1

The updated model and optimizer improves performance considerably. Table 2 shows that our diagnostic model in row five nearly matches the best reported AMT-1 model in half the steps and less than half the parameters. A parameter-matched model exceeds the performance of AMT-1 medium in one-eighth of the steps, achieving a BPS of 66.

4 Scaling laws

Neural scaling laws use relatively small, inexpensive model training results to estimate the final loss of a larger, expensive model run as a function of compute (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022). The *Chinchilla* scaling laws proposed in Kaplan et al. (2020) estimate the lowest-loss allocation of compute subject to $C = 6ND$, i.e. the total floating point operations (FLOPs) C is approximately equal to the product of the number of parameters N and total training tokens D (Kaplan et al., 2020). There is thus an optimal allocation of data D and parameters N for a compute budget C. Training configurations along this 'efficient frontier' are said to be compute optimal. As a corollary, one may use Chinchilla scaling laws characterize the optimality of past training recipes, which we do for AMT-1.

4.1 AMT-1 was data-constrained

IsoFLOP curves (Hoffmann et al., 2022) reveal AMT-1 spent excessive FLOPs on limited tokens. Even performing this analysis requires using some data-repeated runs for fitting, as data repetition is inevitable on the 1.7 billion training tokens of the Lakh MIDI dataset. Following the finding in Muennighoff et al. (2023) that data are effectively fresh up to four epochs, we exclude scaling runs from fitting only if they exceed this. We show the fitted IsoFLOP curves in the left-most graph of Figure 2 with the right-most plot showing how far AMT-1 veers off the efficient frontier. This motivates larger datasets and shows that data-infinite scaling laws are unsuitable for this domain. We need turn to data-constrained scaling laws to better understand how to scale AMT-2.

4.2 Data-constrained scaling laws for AMT-2

Data-constrained scaling laws relax the Chinchilla assumption of infinite data to characterize how pretraining loss behaves in finite data, multi-epoch settings (Muennighoff et al., 2023; Kim et al., 2025a; Lovelace et al., 2026; Qin et al., 2026). Slightly different from the finding the compute-optimal allocation, we instead seek the configuration of parameters N, unique pretraining tokens U_D, and dataset repetition R_D that achieves the lowest test loss.

We follow the method of Lovelace et al. (2026) to obtain parametric fits but we expand the grid of U_D for the single-epoch grid to fill nearly the entire dataset available. Our fill scaling law grid is detailed in Appendix Tables 6 and 7. Given that symbolic music sequences have lower entropy than

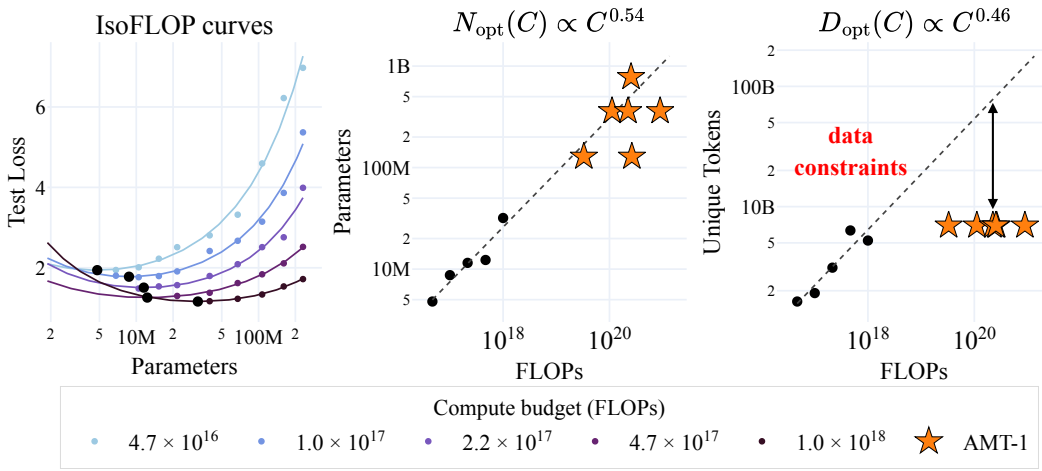

Figure 2: **AMT-1 was data-constrained, motivating data-constrained scaling laws.** We run an IsoFLOP scaling analysis for the AMT-1 training recipe as it was published without modification. The optima along each FLOP budget define the efficient frontier. The right-most plot shows that the AMT-1 models stray orders of magnitude off the compute-optimal allocation for data D_{opt}, suggesting they were over-trained and data-starved. $N_{\text{opt}} \propto C^a$ and $D_{\text{opt}} \propto C^b$ characterize how the optimal allocation of parameters N and data D scale with compute C. We obtain $a = 0.54, b = 0.46$ which is similar to the reported $a = 0.49, b = 0.51$ in Hoffmann et al. (2022)

those of natural language (Qu et al., 2024), exposing the single-epoch runs to nearly the entirety of available data was necessary to fully determine the value of E. Following Karpathy (2025), we fit scaling laws to three distinct parameter (N) accounting styles: all parameters, transformer matrices only, and non-embedding parameters. The four-parameter form proposed in Lovelace et al. (2026) (**Form E**, Appendix Equation 5) using all parameters as N has the highest R^2 for single epoch and multi-epoch samples. We this scaling law for extrapolation. We report all parametric form results with their coefficients and goodness of fit in Appendix Table 9.

Figure 3 depicts the the loss landscape and estimated by Form E along with all other parameterizations we fit. More intriguing is the diverging curvature of their frontiers. The 'bend' of the frontier expresses its scaling recommendation – Form A suggests scaling parameters over data, Form E suggests scaling data over parameters, and Form D suggests larger models overfit faster and that data repetition past a certain point is universally worse at all model sizes.

The optima of Form E is a prohibitively expensive 5.85 billion parameter model. We refine our search to the best we can feasibly train on our compute. We further limit our search to what we can predictably time, which is the largest model we trained for the scaling law: $N = 1.14$ billion. With these constraints, decide to train a **1.14 billion parameter** model for **seven epochs**. The Form E scaling law predicts this configuration should achieve a final test cross entropy loss of $\mathbf{0.7162}$. To contextualize this number, the Form E scaling law predicts that the best *possible* model for $N \leq 1.14$ billion is one that trains for 100 epochs, which it predicts would achieve a loss of 0.6812. We proceed to train our chosen feasible configuration and report its likelihood metrics in Appendix Table 8. The model obtains a BPS of 44.0 on Lakh test split given its full context of 4096, which to our knowledge, is state of the art. However, the run obtains a test loss of 0.840, far from the predicted 0.7162. Our loss is outside the 95% confidence interval of all data-constrained scaling laws we fit, which we visualize in Figure 4.

5 Evaluation

Thus far we have used BPS to compare AMT-2 against AMT-1, but seek to compare against other models using both human evaluation and non-likelihood, automatic evaluation metrics. We select four comparisons to assess model quality: MuPT, BEAT, AMT-1, and Suno v6, each to compare a specific design choice. We select MuPT because it has more parameters and a larger, though radically

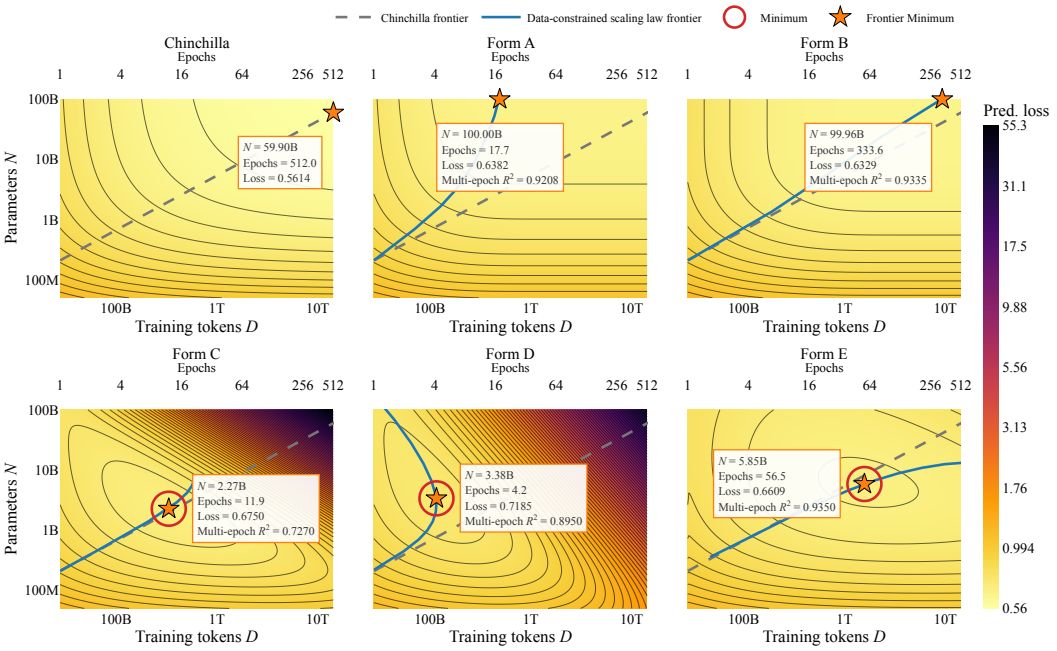

Figure 3: **AMT-2 data-constrained scaling laws tell conflicting stories about the same data, yet report comparable goodness of fit.** All graphs share the same axes for all dimensions with $U_D = 27.858$ billion tokens. The parameter count N and number of epochs $U_D \times e = D$ that it predicts has the lowest loss is marked with a star. Form B, Form D, and Form E report comparable Multi-epoch R^2 values of 0.9335, 0.8950, and 0.9350 respectively – yet their predicted optima are exa-FLOPs apart.

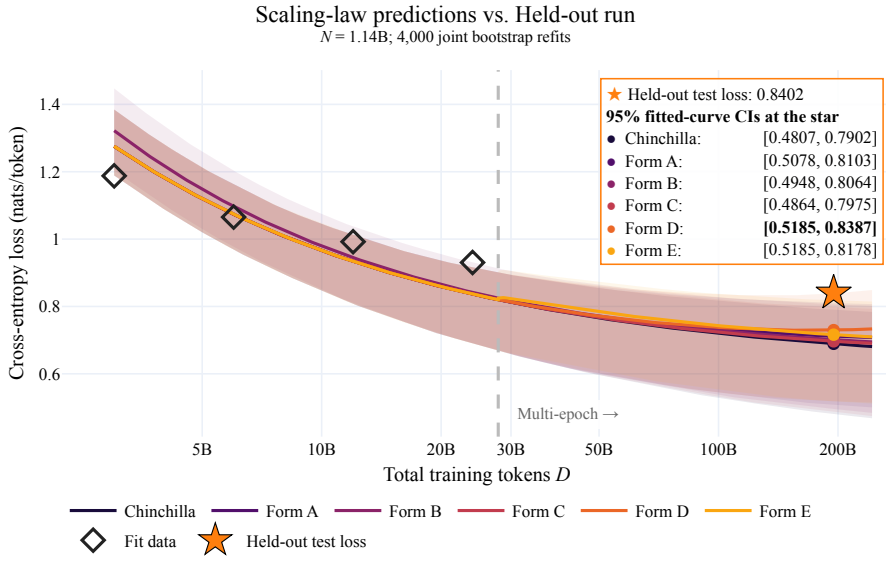

Figure 4: **No data-constrained scaling law extrapolates to AMT-2 1.14B.** For all points $N = 1.14$ billion. For our Held-out run $U_D = 27.86$ billion, but for the fit data, $U_D = D$. Loss equation Form D has the lowest error. We mark our held-out training run with a star and ran for seven epochs. We obtain confidence intervals for each parametric form by bootstrapping 4,000 fit runs, using the methodology in Besiroglu et al. (2024). The test loss falls outside the 95% confidence interval for all scaling law forms. Form D is closest to predicting our true loss value, but exhibited the second worst multi-epoch R^2 value.

Table 3: **Automatic evaluation metrics.** We use the default or published inference settings for each model when possible. We find that our model consistently generates higher quality samples at $p = 0.99$ by both MAD and FMD scores against our 2026 Held-out test set and the Lakh Test split. For the version of the 2026 Held-out test set that was quantized by BEAT's multi-track quantization procedure, $p = 1.0$ is better. FMD estimates stabilize past $1{,}000$ samples (Retkowski et al., 2025).

		MAD ↓		FMD ↓	
Model	Nucleus p	Score	Samples	Score	Samples
Reference: *2026 Held-out test set - Quantized*					
BEAT	0.95	0.1523	1,239	**108.413**	1,248
AMT-2	1.0	**0.1109**	1,234	194.165	1,248
AMT-2	0.99	0.1458	1,239	194.639	1,248
Reference: *Lakh Test (f)*					
AMT-1	0.95	0.0862	1,103	116.854	1,212
AMT-2	1.0	0.0438	1,103	115.373	1,208
AMT-2	0.99	**0.0279**	1,122	**114.129**	1,212

different training dataset of ABC notation; MuPT is a 1.97 billion parameters trained on 33.6 billion tokens. This work and others (Qu et al., 2024; Yuan et al., 2024) claim that ABC notation is better than MIDI for autoregressive neural sequence models. We select BEAT because their time-relativizing tokenization closely resembles ours (Qian et al., 2026); it is a more musically-aware realization of the time-relativization idea. Qualitatively, their demo samples are impressive. We select AMT-1 because we seek to measure whether our work yields audible improvement. This implicitly tests whether BPS correlates with better music. Finally, we compare against Suno v6 (Suno, Inc.) to measure against the state of the art in audio-based music generation.

5.1 2026 Held-out test set

To mitigate test/train contamination, we procured a dataset of symbolic music released after June 4th, 2026. Many papers in symbolic music, including BEAT to which we compare, train on all of Lakh MIDI or they produce their own iid splits. Even an iid test split can induce contamination, because there are many duplicates in Lakh (Choi et al., 2025). We describe our collection procedure in Appendix A.7.

5.2 Evaluation Protocol

We sample from each model and obtain a symbolic representation of its output. We detail the procedure for each model in Appendix A.10.

Automatic evaluation

We measure the quality of our model outputs by the MAUVE Audio Divergence (MAD) (Huang et al., 2025) and the Fréchet Music Distance (FMD) (Retkowski et al., 2025). Because the MAD is an audio metric, we convert our samples and reference distribution using FluidSynth with the same MuseScore 3 soundfont and we set all notes to the same velocity of 100. We note that MAD is expressly designed to more strongly correlate with human preferences of music, not just audio fidelity, and captures human musical preferences more strongly than the Fréchet Audio Distance (Kilgour et al., 2018).

We report our automatic results in Table 3 and note that the FMD scores do not agree with human evaluation results in Table 4 while MAD scores do agree in relative rankings.

Human evaluation

We compare our model with the others pairwise: ours vs. AMTv1, ours vs. BEAT, ours vs. MuPT, and ours vs. Suno, largely following the protocol in Thickstun et al. (2024) and Huang et al. (2018). We must convert all model outputs to audio so that survey participants can listen. We do this with FluidSynth. Suno already provides audio, which sounds like full, acoustically rich productions that would undoubtedly win in musicality. However, we are interested in the symbolic content of the

Table 4: **Human evaluators prefer AMT-2 over AMT-1 and BEAT.** AMT-2 closes the gap between AMT-1 and Suno. For each model, we collect 80 samples for 30 human participants to rate three times. The model name is hidden and order is randomized. We report p values via the Wilcoxon signed rank test following Thickstun et al. (2024) with Bonferroni correction. We note that AMT-2 conditions on five second prompts generated by MuPT, which forces AMT-2 to generate music in a distribution quite different from its own. This setting is the most favorable for MuPT, yet AMT-2 still ties in human preferences. Full sampling details for all models are in Appendix A.10.

Model A	Model B	Wins (A)	Ties	Wins (B)	p-value	p-bonferroni
AMT-2	AMT-1	**121**	49	70	0.0053	**0.0263**
AMT-2	BEAT	**127**	37	76	0.0027	**0.0133**
AMT-2	MuPT	**93**	59	88	0.6867	1.0000
AMT-2	Suno v6	100	40	100	0.9866	1.0000
AMT-1	Suno v6	94	30	**116**	0.2079	1.0000

music, not the sonic content – thus we transcribe the Suno output with MuScriptor (Rouard et al., 2026), and voice it with the same FluidSynth procedure as a sonic normalization.

All pairwise samples used for comparison share the same five second prefix, while the suffixes diverge. We present survey participants all four pairs in random order and further randomize the order of prompts. Participants are presented with an interface that asks them to chose which clip is more "conventionally musical", with ties allowed. The order of which model appears in which A/B slot is also randomized. We do not present the model names or how the music was obtained to survey participants; there is no information that could bias a decision towards one model or the other. Before taking the survey, participants must pass a four question screening that we have designed to measure their ability to discern dissonance and rhythmic irregularity.

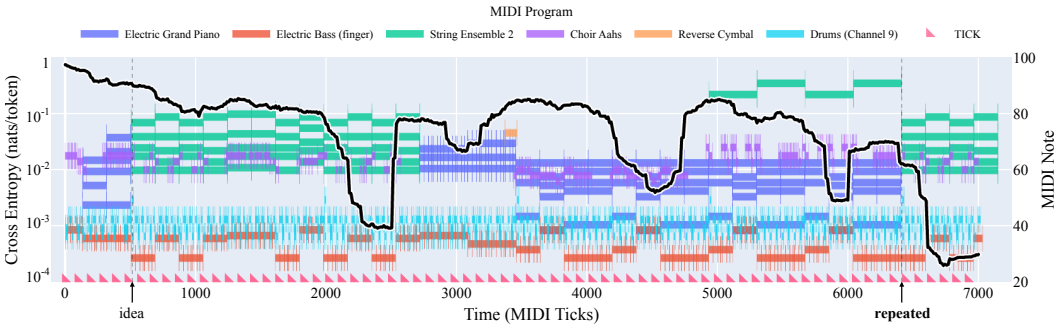

Figure 5: **AMT-2 1.14B shows evidence of musical form**. Conditioned on a prompt from our 2026 held-out test set, the model generates an idea voiced with a String Ensemble and then repeats its own output. This idea is not present in the conditioning prompt. When the idea repeats, the cross entropy rapidly falls near zero.

6 Conclusion

We present **Anticipatory Music Transformer 2 (AMT-2)**, a capable 1.14 billion parameter model informed by data-constrained scaling laws and contemporary practice. AMT-2's final test loss lies just barely beyond the 95% confidence interval of all scaling law forms we examine. Future work may explore the fitting methodology of parametric scaling laws in data-constrained settings, similar the scholarship for data-infinite settings (Besiroglu et al., 2024; Pearce & Song, 2024; Sardana et al., 2025; Weng, 2026). Given the limitations of automatic evaluation metrics, a more robust or comprehensive benchmark for symbolic music could be valuable. Finally, we show in Figure 5 that AMT-2 can generate coarse musical form, wherein it repeats its own material when continuing a prompt from our 2026 held-out test set.

AI USE STATEMENT

In this work, we have used generative AI tools to construct synthetic datasets, which we use for evaluation against a generative model in our chosen domain, described in full in Appendix A.8. None of the synthetic data are used for training. We use generative AI tools for figure and table creation, to create bibtex citation entries, and for drafting method descriptions in the appendix. We use generative AI to implement routine code for PyTorch, MIDI manipulation, and for understanding codebases of prior work; we write our tokenization and inference code by hand.

Generative AI tools identified only one research idea, which was to use the RMS-matched variant of Muon proposed in Liu et al. (2025) instead of the original Jordan et al. (2024) implementation, which we had been using. Generative AI tools were used to implement the website for our human feedback survey but the survey question design and its data analysis procedures are from prior work.

All other required disclosure tasks are not applicable.

We have reviewed all AI-assisted work. In addition to reviewing all generated source code, we manually inspect MIDI files and listen to their audio realization to ensure they are correct. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We commit to open-sourcing our source code used in tokenization, scaling law sweeping and fitting, and data processing. We will release our scaling law runs since this is required to fully reproduce our coefficients. We will also release our evaluation pipeline.

REFERENCES

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Apple Inc. Apple music style guide, 2025. URL https://help.apple.com/itc/musicstyleguide/en.lproj/static.html.

Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019. doi: 10.1109/MSP.2018.2869928.

Stephan Bernsee. Unbaking cakes since 1998, 2015. URL https://blogs.zynaptiq.com/bernsee/unbaking-cakes-since-1998/. Stephan Bernsee's Blog.

Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt, 2024. URL https://arxiv.org/abs/2404.10102.

Louis Bradshaw and Simon Colton. Aria-midi: A dataset of piano midi files for symbolic music modeling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=X5hrhgndxW.

Estefanía Cano, Derry FitzGerald, Antoine Liutkus, Mark D. Plumbley, and Fabian-Robert Stöter. Musical source separation: An introduction. *IEEE Signal Processing Magazine*, 36 (1):31–40, 2019. doi: 10.1109/MSP.2018.2874719. URL https://inria.hal.science/hal-01945345v1.

Celemony Software GmbH. *Melodyne 5 Studio – Stand-Alone: User Manual*. Celemony Software GmbH, 2026. URL https://helpcenter.celemony.com/M5/pdf/melodyneStudio5/en?env=standAlone.

Tsung-Ping Chen and Li Su. Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models. *Transactions of the International Society for Music Information Retrieval*, 4(1):1–13, 2021. doi: 10.5334/tismir.65. URL https://doi.org/10.5334/tismir.65.

Yinuo Chen. Postscript on the musics of control. In *1st International Workshop on Emerging AI Technologies for Music*, 2025. URL https://openreview.net/forum?id=Bwoq7ZH5cV.

Eunjin Choi, Hyerin Kim, Jiwoo Ryu, Juhan Nam, and Dasaem Jeong. On the de-duplication of the lakh MIDI dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

Y. Collet and M. Kucherawy. Zstandard Compression and the 'application/zstd' Media Type. RFC 8878, February 2021. URL https://www.rfc-editor.org/info/rfc8878/.

Common Crawl Foundation. Common Crawl Corpus. https://commoncrawl.org, 2026.

Michael Connor. The impossible music of black midi, September 2013. URL https://rhizome.org/editorial/2013/sep/23/impossible-music-black-midi/.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In J. Stephen Downie and Remco C. Veltkamp (eds.), *ISMIR*, pp. 637–642. International Society for Music Information Retrieval, 2010. ISBN 978-90-393-53813. URL http://dblp.uni-trier.de/db/conf/ismir/ismir2010.html#CuthbertA10.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 19822–19835. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a4d92e2cd541fca87e4620aba658316d-Paper.pdf.

Nathan Godey and Yoav Artzi. Lost in backpropagation: The lm head is a gradient bottleneck, 2026. URL https://arxiv.org/abs/2603.10145.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego

Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim

Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Rui Guo and Dorien Herremans. An exploration of controllability in symbolic music infilling. *IEEE Access*, 13:54873–54891, 2025. doi: 10.1109/ACCESS.2025.3554648.

Zixun Guo and Simon Dixon. Moonbeam: A midi foundation model using both absolute and relative music attributes. *arXiv preprint arXiv:2505.15559*, 2025. URL https://arxiv.org/abs/2505.15559.

Tatsunori Hashimoto. Stanford CS336: Language Modeling from Scratch, Spring 2025, Lecture 3: Architectures, Hyperparameters. YouTube, April 2025. URL https://www.youtube.com/watch?v=ptFiH_bHnJw. Stanford Online, accessed May 26, 2026.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1lYRjC9F7.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017. URL https://arxiv.org/abs/1712.00409.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 300–313, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs, 2021. URL https://arxiv.org/abs/2101.02402.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. Liger-kernel: Efficient triton kernels for LLM training. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*, 2025. URL https://openreview.net/forum?id=36SjAIT42G.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018. URL https://arxiv.org/abs/1809.04281.

Yichen Huang, Zachary Novack, Koichi Saito, Jiatong Shi, Shinji Watanabe, Yuki Mitsufuji, John Thickstun, and Chris Donahue. Aligning text-to-music evaluation with human preferences, 2025. URL https://arxiv.org/abs/2503.16669.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Andrej Karpathy. nanochat: The best chatgpt that $100 can buy, 2025. URL https://github.com/karpathy/nanochat.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute, 2025a. URL https://arxiv.org/abs/2509.14786.

Yewon Kim, Sung-Ju Lee, and Chris Donahue. Amuse: Human-AI collaborative songwriting with multimodal inspirations. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400713941. doi: 10.1145/3706598.3713818. URL https://doi.org/10.1145/3706598.3713818.

LAION. Releasing Re-LAION-5B: transparent iteration on LAION-5B with additional safety fixes. https://laion.ai/blog/relaion-5b/, 2024. Accessed: 2026-09-16.

Hyun Lee, Taehyun Kim, Hyolim Kang, Minjoo Ki, Hyeonchan Hwang, Sharang Han, Seon Joo Kim, et al. Commu: Dataset for combinatorial music generation. *Advances in Neural Information Processing Systems*, 35:39103–39114, 2022.

Keon Ju Maverick Lee, Jeff Ens, Sara Adkins, Pedro Sarmento, Mathieu Barthet, and Philippe Pasquier. The gigamidi dataset with features for expressive music performance detection. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 8(1):1–19, 2025.

Julian Lenz and Anirudh Mani. Pertok: Expressive encoding and modeling of symbolic musical ideas and variations, 2024. URL https://arxiv.org/abs/2410.02060.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982.

Phillip Long, Zachary Novack, Taylor Berg-Kirkpatrick, and Julian McAuley. Pdmx: A large-scale public domain musicxml dataset for symbolic music processing. In *ICASSP 2025-2025 IEEE international conference on acoustics, speech and signal processing (icassp)*, pp. 1–5. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Justin Lovelace, Christian Belardi, Srivatsa Kundurthy, Shriya Sudhakar, and Kilian Q. Weinberger. Prescriptive scaling laws for data constrained training, 2026. URL https://arxiv.org/abs/2605.01640.

Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8293–8316, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.459. URL https://aclanthology.org/2024.naacl-long.459/.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*, 2023.

Sebastian Murgul and Michael Heizmann. Beat and downbeat tracking in performance midi using an end-to-end transformer architecture, 2025. URL https://arxiv.org/abs/2507.00466.

NVIDIA. Transformer Engine Documentation. https://docs.nvidia.com/deeplearning/transformer-engine/, 2026.

OpenAI. Chatgpt. https://chatgpt.com/, 2026. URL https://chatgpt.com/. OpenAI, Inc. Version: ChatGPT-5.6-Sol.

Garson O'Toole. Quote origin: Writing about music is like dancing about architecture, November 2010. URL https://quoteinvestigator.com/2010/11/08/writing-about-music/. Quote Investigator.

Philippe Pasquier, Jeff Ens, Nathan Fradet, Paul Triana, Davide Rizzotti, Jean-Baptiste Rolland, and Maryam Safi. Midi-gpt: A controllable generative model for computer-assisted multitrack music composition, 2025. URL https://arxiv.org/abs/2501.17011.

Tim Pearce and Jinyeop Song. Reconciling kaplan and chinchilla scaling laws. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=NLoaLyuUUF.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Ofir Press, Noah A. Smith, and Mike Lewis. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5470–5483, Online, August 2021. Association for Computational Linguistics. URL https://aclanthology.org.

Lekai Qian, Haoyu Gu, Jingwei Zhao, and Ziyu Wang. Beat: Tokenizing and generating symbolic music by uniform temporal steps. In *Proceedings of the International Conference on Machine Learning*, 2026.

Tian Qin, Kimia Hamidieh, and David Alvarez-Melis. Bridging compute- and data-optimal pretraining, 2026. URL https://arxiv.org/abs/2607.25271.

Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xinrun Du, Shuyue Guo, Yiming Liang, Yizhi Li, Shangda Wu, Junting Zhou, Tianyu Zheng, Ziyang Ma, Fengze Han, Wei Xue, Gus Xia, Emmanouil Benetos, Xiang Yue, Chenghua Lin, Xu Tan, Stephen W. Huang, Jie Fu, and Ge Zhang. Mupt: A generative symbolic music pretrained transformer, 2024. URL https://arxiv.org/abs/2404.06393.

Colin Raffel. The lakh midi dataset. https://colinraffel.com/projects/lmd/, 2016. Accessed: 2026.

Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation, 2020. URL https://arxiv.org/abs/2008.07703.

Jan Retkowski, Jakub Stępniak, and Mateusz Modrzejewski. Frechet music distance: A metric for generative symbolic music evaluation, 2025. URL https://arxiv.org/abs/2412.07948.

Simon Rouard, Michael Krause, Axel Roebel, Carl-Johann Simon-Gabriel, and Alexandre Défossez. Muscriptor: An open model for multi-instrument music transcription, 2026. URL https://arxiv.org/abs/2607.08168.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws, 2025. URL https://arxiv.org/abs/2401.00448.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining, 2025. URL https://arxiv.org/abs/2509.01440.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Suno, Inc. Suno. URL https://suno.com/.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Bob Talbert. Bob talbert's quotebag. *Detroit Free Press*, pp. 19C, February 1979. February 18, 1979. Includes the quotation attributed to Martin Mull: "Writing about music is like dancing about architecture.".

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins,

Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

John Thickstun, Zaïd Harchaoui, Dean P. Foster, and Sham M. Kakade. Coupled recurrent models for polyphonic music composition. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 311–318, 2019. URL http://ismir.net.

John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *Transactions on Machine Learning Research*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017. URL http://arxiv.org/abs/1706.03762.

Avery Li-Chun Wang. An industrial-strength audio search algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pp. 7–13, Baltimore, Maryland, USA, October 2003.

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, 2020. URL https://arxiv.org/abs/2008.07142.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them, 2025. URL https://arxiv.org/abs/2509.02046.

Lilian Weng. Scaling laws, carefully. *lilianweng.github.io*, June 2026. URL https://lilianweng.github.io/posts/2026-06-24-scaling-laws/.

Wikipedia contributors. Black MIDI. https://en.wikipedia.org/wiki/Black_MIDI. Wikipedia, The Free Encyclopedia. Accessed: 2026-09-08.

Shih-Lun Wu, Dave Carlton, Ryan Miyakawa, Yoon Kim, Chris Donahue, and Cheng-Zhi Anna Huang. Midi-llm: Improving text-to-midi music generation via adapting large language models, 2026. URL https://arxiv.org/abs/2511.03942.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwZSG-CZ.

Xiaoyu Yi, Qi He, Gus Xia, and Ziyu Wang. Vitex: Visual texture control for multi-track symbolic music generation via discrete diffusion models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 14797–14801, 2026. doi: 10.1109/ICASSP55912.2026.11463443.

Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, Ziyang Ma, Liumeng Xue, Ziyu Wang, Qin Liu, Tianyu Zheng, Yizhi Li, Yinghao Ma, Yiming Liang, Xiaowei Chi, Ruibo Liu, Zili Wang, Pengfei Li, Jingcheng Wu, Chenghua Lin, Qifeng Liu, Tao Jiang, Wenhao Huang, Wenhu Chen, Emmanouil Benetos, Jie Fu, Gus Xia, Roger Dannenberg, Wei Xue, Shiyin Kang, and Yike Guo. Chatmusician: Understanding and generating music intrinsically with llm, 2024. URL https://arxiv.org/abs/2402.16153.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Timmy Liu, Vijay Thakkar, and Tri Dao. Flashattention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling, 2026. URL https://arxiv.org/abs/2603.05451.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.

A Appendix

A.1 Time resolution

For AMT-2, we change the subdivision of a second from AMT-1's value of 100 to 96. This eliminates rounding errors induced by common rhythms. For example, in the common tempo of 120 beats per minute, placing eight evenly spaced notes (sixteenth notes) in one second causes timing error because 100 is not divisible by 8. Other subdivisions have the same issue, with eighth note triplets needing $33.\overline{3}$ units for proper timing and thirty-second notes needing 3.125 units – any non-integer timings produce errors. Subdividing the second into 96 time units mitigates this error, as half notes cleanly divide into 48 units, quarter notes into 24, eighth notes into 12, eighth note triplets into 32, and thirty-second notes into 3. Timing in multiples of 96 expresses common subdivisions without rounding and is used in other contemporary tokenizations Pasquier et al. (2025); Lenz & Mani (2024).

A.2 Train dataset

A.2.1 Dataset mixture

Since the publication of AMT-1, two large symbolic music datasets have been published: GigaMIDI (Lee et al., 2025) and Aria-MIDI (Bradshaw & Colton, 2025). GigaMIDI is a superset of Lakh MIDI (Raffel, 2016), the AMT-1 training corpus, and consolidates within it other canonical symbolic music datasets like POP909 (Wang et al., 2020), MAESTRO (Hawthorne et al., 2019), ComMU (Lee et al., 2022), PDMX (Long et al., 2025), *inter alia*. GigaMIDI represents a diverse collection of multi-track full compositions, short isolated loops, and expressive performances. Aria-MIDI (Bradshaw & Colton, 2025) is a dataset of automatically transcribed (Benetos et al., 2019) solo-piano recordings with over 800 thousand files. We use the 'pruned' subset of Aria-MIDI because the authors recommended it for foundation model pretraining (Bradshaw & Colton, 2025). We report the summary statistics of our combined dataset and its splits in Table 5. Further details on filtering settings are discussed in Appendix A.2.

Like AMT-1, we perform anticipation augmentation on each sample, which increases the total tokens. For each sample, we tokenize it without anticipation, and further apply five distinct types of anticipation per file: (span, cold start instrument, warm start instrument, generalized span infill cold, generalized span infill warm). This means the original music appears six times in different tokenizations. Unlike AMT-1, we remove the 'random anticipation' style, which randomly selects events in the sequence to anticipate.

A.3 Filtering

We remove files exceeding two mebibytes from our dataset in all splits. Unlike ABC Notation and quantized scores, MIDI files may contain pathological note density that a duration filter will not catch. In particular, the largest file in the GigaMIDI dataset (Lee et al., 2025) is named `739981a769e9bf278efcd893ebb12b4c.midi` is 8.76 MiB. This file contains 143 seconds of music and $1{,}486{,}725$ notes, which exceeds ten thousand notes per second on average. The file contains a *Black MIDI* composition, a genre of music that deliberately adds notes so that the written score appears black in color (Connor, 2013; Wikipedia contributors). We show a single measure of this note-dense file in Figure 6. Although this is a legitimate musical idiom, we exclude such files from our dataset by combining a file size and duration filter.

A.4 Split sizes

In Table 5 we detail the size of our train, test, and validation splits.

In Figure 7, we show how increasing the context length affects the average time expressed in a sequence in our proposed tokenization.

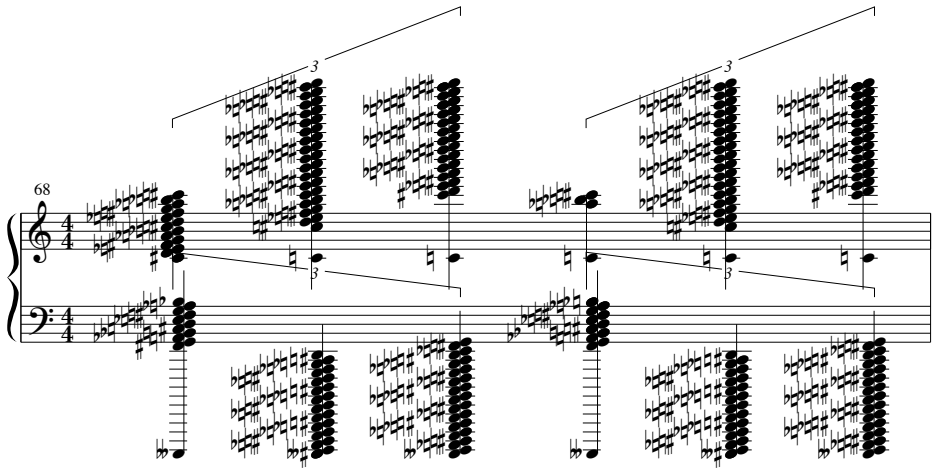

Figure 6: **Pathologically dense sample in the training split**. Measure 68 of a piano channel in `739981a769e9bf278efcd893ebb12b4c.midi`, the largest file in GigaMIDI Lee et al. (2025).

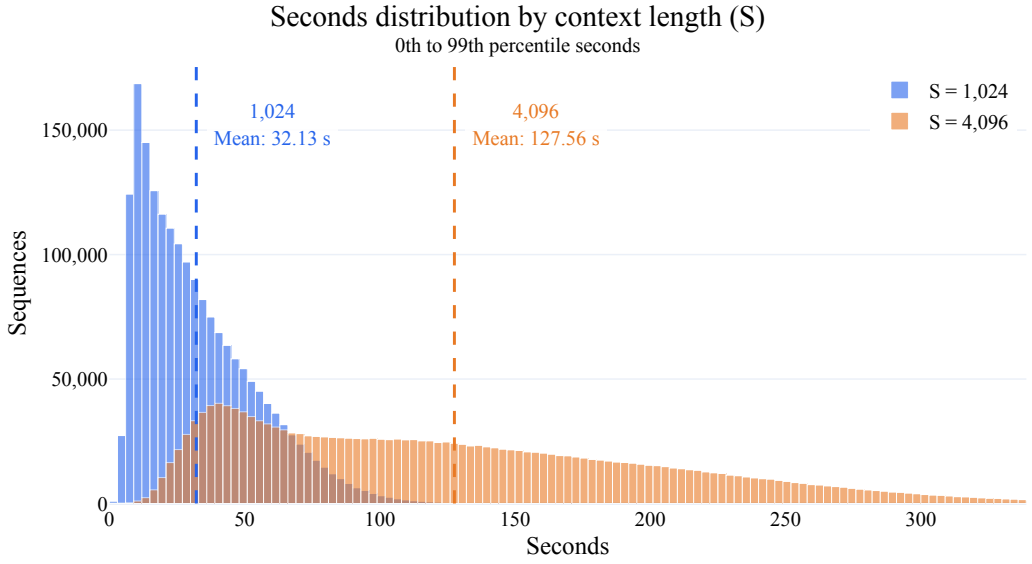

Figure 7: **Distribution of duration by context sequence length**. We measure the number of seconds contained in tokenized sequences of our training data without anticipation. Limiting the context size to 1,024 yields roughly 32 seconds of music on average. Increasing the context length by a factor of four roughly increases the average by a factor of four, though the pronounced right tail suggests our dataset has many sparse sequences.

A.5 Model architecture and optimizer

Our model is a modified LLaMA 2 (Touvron et al., 2023) model with changes as follows:

- Pre- and post-norm enclose the feed forward and the multihead attention components in transformer blocks, referred to as 'sandwich norm' (Team et al., 2024; Ding et al., 2021; Hashimoto, 2025). The norm implementation throughout remains RMSNorm (Zhang & Sennrich, 2019), consistent with LLaMA 2.

Table 5: **Dataset summary**. We apply filters to remove samples of extreme duration or note density. As a proxy for repetition, we report the total uncompressed MIDI file size in mebibytes (MiB) and the compression ratio of Zstandard (Collet & Kucherawy, 2021) with compression level 3.

Split	Files after filter	Filter Rate	Hours	MiB	Ratio Zstd-3	Tokens Autoregressive	Tokens Block Anticipation
Lakh Test (f)	10,663	4.96%	559	339	3.27	128,315,392	744,275,968
iid Validation	92,810	12.01%	3,844	1,029	2.53	427,659,264	1,547,243,520
iid Test	93,026	11.80%	3,839	1,030	2.52	427,188,224	1,550,249,984
Train	1,672,009	11.93%	68,980	18,519	2.53	7,687,221,248	27,858,149,376
Total	1,868,508	11.89%	77,222	20,917	2.53	8,670,384,128	31,699,918,848

- RMSNorm is applied to key and query tensors before RoPE (Su et al., 2023) in all multihead attention blocks, a design named QK Norm (Dehghani et al., 2023).
- The output of each transformer block is a learned weighting between the current layer and the globally first input layer, a design named "lambda scalars" proposed by the GPT-2 speedrun community for nanochat (Karpathy, 2025).
- Dropout (Srivastava et al., 2014) with $p = 0.1$ is applied to the embedding and residual path during training. We retain this design from AMT-1 to prevent overfitting (Thickstun et al., 2024).

Muon details

We follow the same Moonlight learning rate schedule but omit the final 'cooldown' phase because we do not have a set of higher quality data to save for the final stage of training – the schedule is a $2{,}000$ step linear warm-up followed by a single cycle cosine decay to 0.1 of peak. One can independently tune the learning rates for embedding and non-embedding layers to attain better test loss (Karpathy, 2025; Wen et al., 2025). We leave layer specific tuning to future work and follow (Liu et al., 2025), which applies the same peak learning rate, weight decay, and scheduler to all parameters.

A relevant finding also reported in Liu et al. (2025) is that models pretrained with Muon should also be fine-tuned with Muon for optimal performance. This means researchers who wish to fine-tune the AMT-2 checkpoint should also use Muon. We use the implementation included in PyTorch, which exists in version 2.14. To use the Moonlight RMS matching implementation, set the keyword argument: `adjust_lr_fn="match_rms_adamw"`.

A.6 Scaling laws

A.6.1 Data-constrained scaling law grid

We report our the grid we use for both single-epoch and multi-epoch samples in Table 6.

A.6.2 Data-constrained scaling law forms

In "Prescriptive Scaling Laws for Data Constrained Training" Lovelace et al. (2026), there are three variations of their scaling law equation. Each form adds new parameters to the previous and consequently can better fit their data. Their work contrasts their parametric forms with those in "Scaling Data-Constrained Language Models" Muennighoff et al. (2023), which present overfitting as exponential decay of additional epochs and additional model parameters N, yielding two parametric forms. Both works add parameters to the original *Chinchilla* scaling law equation. There are therefore five total data constrained scaling law parametric forms we consider. They are as follows:

Form A: the effective data form, proposed in Muennighoff et al. (2023)

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{(U_D + U_D R_D^* [1 - \exp(\frac{-R}{R_D^*})])^{\beta}} \tag{1}$$

Table 6: **Chinchilla-style, single-epoch fitting grid for data-constrained scaling laws.** All nine models are evaluated at all ten unique-token budgets with $e = 1$, giving 90 observations for fitting the baseline scaling law.

(a) Single-epoch observations ($e = 1$)

Our model	Unique-token budget U_D (approximate)										
N (M)	50M	100M	200M	400M	800M	1.5B	3B	6B	12B	24B	Obs.
17.1	✓	✓	✓	✓	✓*	✓*	✓*	✓*	✓*	✓*	10
26.4	✓	✓	✓	✓	✓	✓	✓	✓	✓*	✓*	10
37.3	✓	✓	✓	✓	✓*	✓*	✓*	✓*	✓*	✓*	10
50.6	✓	✓	✓	✓	✓	✓	✓	✓	✓*	✓*	10
139.0	✓	✓	✓	✓	✓	✓	✓	✓	✓*	✓*	10
252.0	✓	✓	✓	✓	✓	✓	✓	✓	✓*	✓*	10
496.5	✓	✓	✓	✓	✓	✓	✓*	✓*	✓*	✓*	10
787.8	✓	✓	✓	✓	✓	✓*	✓*	✓*	✓*	✓*	10
1,140.2	✓	✓	✓	✓	✓	✓*	✓*	✓*	✓*	✓*	10
Total									Single-epoch observations		90

✓ denotes an observation in our fitting dataset; ✓* marks a configuration absent from the Lovelace et al. (2026) non-weight decay study single-epoch grid (Table 8). We add larger unique token budgets of 12 and 24 billion to enable our fitting procedure to determine the entropy floor term E. This leads to 34 additional samples. "Obs." counts our observations (samples) that we use in fitting.

(b) Exact parameter counts and reference size classes

Our model N (M)	Lovelace size class	Total parameters	Transformer matrices	Non-Embedding
17.1	15M	17,061,324	3,544,320	10,302,924
26.4	25M	26,420,496	8,397,824	17,409,296
37.3	35M	37,286,610	14,758,272	26,022,610
50.6	50M	50,643,604	23,609,600	37,126,804
139.0	125M	139,041,048	84,973,056	112,007,448
252.0	250M	251,980,572	179,889,920	215,935,772
496.5	500M	496,544,040	406,430,720	451,488,040
787.8	750M	787,766,832	679,630,848	733,699,632
1,140.2	1B	1,140,193,588	1,014,034,944	1,077,115,188

M and B denote one million (10^6) and one billion (10^9), respectively. Our total parameter counts differ from Lovelace et al. (2026)'s Table 5 because we have modified the model design. Multi-epoch observations are listed separately in Table 7.

Table 7: **Multi-epoch fitting grid** ($e > 1$). Five models are evaluated at five unique-token budgets U_D and five epoch counts, giving 125 observations. Each checkmark represents one sample at the specified model size N, token budget U_D, and epoch count e

Our model N (M)	Unique tokens U_D (approximate)	Training epochs e				
		2	4	8	12	16
26.4	50M	✓	✓	✓*	✓*	✓
	100M	✓*	✓	✓	✓	✓
	200M	✓	✓	✓	✓	✓
	400M	✓*	✓*	✓*	✓*	✓*
	800M	✓*	✓*	✓*	✓*	✓*
50.6	50M	✓	✓	✓	✓	✓
	100M	✓	✓	✓	✓	✓
	200M	✓	✓	✓	✓	✓
	400M	✓*	✓*	✓*	✓*	✓*
	800M	✓*	✓*	✓*	✓*	✓*
139.0	50M	✓*	✓*	✓*	✓*	✓*
	100M	✓	✓	✓	✓	✓
	200M	✓	✓	✓	✓	✓
	400M	✓	✓	✓	✓	✓
	800M	✓	✓	✓	✓	✓
252.0	50M	✓*	✓*	✓*	✓*	✓*
	100M	✓	✓	✓	✓	✓
	200M	✓	✓	✓	✓	✓
	400M	✓	✓	✓	✓	✓
	800M	✓	✓	✓	✓	✓
496.5	50M	✓*	✓*	✓*	✓*	✓*
	100M	✓	✓	✓	✓	✓
	200M	✓	✓	✓	✓*	✓
	400M	✓	✓	✓	✓	✓
	800M	✓*	✓*	✓*	✓*	✓*
Observations per epoch count		25	25	25	25	25

✓* marks a configuration absent from the standard-study multi-epoch grid in Table 9 of Lovelace et al. (2026). We add 44 additional multi-epoch observations. Comparisons use the reference size classes in Table 6 sub-table (b) and nominal token budgets. For e epochs, $R_D = e - 1$ and the total number of processed tokens is $D = eU_D$. The second-stage, multi-epoch data-constrained fits use these 125 observations together with the 90 single-epoch observations, using 215 total samples, holding the baseline coefficients fixed.

Form B: The effective data, effective parameters form, proposed in Muennighoff et al. (2023)

$$L(N, D) = E + \frac{A}{(U_N + U_N R_N^* [1 - \exp(\frac{-R_N}{R_N^*})])^\alpha} + \frac{B}{(U_D + U_D R_D^* [1 - \exp(\frac{-R_D}{R_D^*})])^\beta} \tag{2}$$

Form C: The additive penalty form, proposed in Lovelace et al. (2026)

$$L\left(N, U_D, R_D\right) = E + \frac{A}{N^\alpha} + \frac{B}{\left(U_D \cdot (1 + R_D)\right)^\beta} + P \cdot R_D \cdot \frac{N}{U_D} \tag{3}$$

Form D: The Kappa ratio form, proposed in Lovelace et al. (2026)

$$L\left(N, U_D, R_D\right) = E + \frac{A}{N^\alpha} + \frac{B}{\left(U_D \cdot (1 + R_D)\right)^\beta} + P \cdot R_D \cdot \left(\frac{N}{U_D}\right)^\kappa \tag{4}$$

Form E: The four parameter form, proposed in Lovelace et al. (2026)

$$L\left(N, U_D, R_D\right) = E + \frac{A}{N^\alpha} + \frac{B}{\left(U_D \cdot (1 + R_D)\right)^\beta} + P \cdot R_D^\delta \cdot \left(\frac{N}{U_D^\gamma}\right)^\kappa \tag{5}$$

We report measures of goodness of fit for all five forms and styles of parameter counting in Table 9. The best fitting row is Form E with N equal to all parameters. We ultimately choose this form to extrapolate. However, Form A and B attain comparable R^2 values to this row and exceed the performance of Forms C and D. We report all parameters for all forms in Table 9.

Each form has substantially different optima, which we emphasize visually in Figure 3. For our release model, we chose to use Form E to predict the best run. This form does not exhibit the 'turn-back' behavior reported in Lovelace et al. (2026), though Form D does. The 'turn-back' is important because it suggests that compute is better spent on model capacity N rather than additional epochs past some point. In Table 9, we see that Form D and Form E have comparable R^2 and Adjusted R^2 values, though Form E is better. The optimal configurations, however, differ by over an order of magnitude in epochs, 4.2 epochs vs. 56.5 epochs.

The array of forms have conflicting messages, which is expected but confusing. Form A, B, and C suggest that the compute-optimal approach is to train a larger model for fewer epochs. Form D suggests that excessive repetition becomes harmful to performance past a specific threshold and compute is thereafter best spent on model capacity. Form E suggests the compute optimal approach is to attenuate model size under extreme repetition, though it also shows that excessive repetition is harmful.

A.6.3 Fitting procedure

We fit five data-constrained scaling laws using the two-stage procedure of (Lovelace et al., 2026), and quantify uncertainty by refitting both stages on 4,000 bootstrap resamples. The number of resamples follows (Besiroglu et al., 2024). The intervals reported here are pointwise, equal-tailed 95% confidence intervals for the fitted scaling-law loss. We specify the resampling design and numerical implementation below, including the choices that extend the cited procedures.

A.7 Out of Distribution Dataset

We procure a custom evaluation dataset to reduce train/test contamination. Several models to which we compare train on the entirety of Lakh MIDI or do not report their splits. Therefore, we collect music that was released after checkpoint release, i.e., it could not have been in the training data. We source our material from Shazam (Wang, 2003) because of its relationship with Apple Music; recent Shazam identifiers are equal to its Apple Music identifier. This is useful because the Apple Music style guide (Apple Inc., 2025) has and specific and strict metadata standards. We hypothesized these metadata standards would lead to more accurate descriptions of our data, though we do not systemically examine this. One could implement a similar collection procedure sourcing instead from Spotify charts or otherwise.

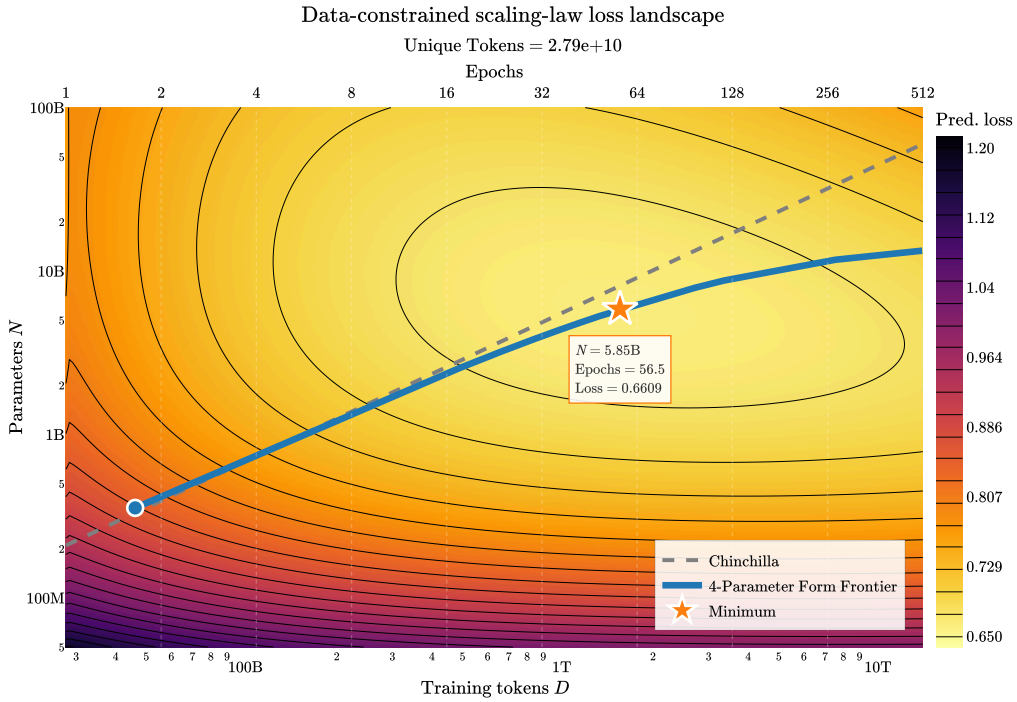

Figure 8: **Four parameter penalty loss prediction**. We plot the loss surface as a function of total model parameters N and total training tokens $D = U_D \cdot$epochs. The loss surface shows that excessive repetition leads to worse performance. The curvature of the efficient frontier does not match the 'turn-back' shape reported in the work that proposed this parametric form (Lovelace et al., 2026). We discuss this discrepancy in detail in Appendix A.6.

Table 8: **Prescriptive scaling law guided extrapolation run.** In the first row, we report the strongest model from AMT-1 (Thickstun et al., 2024) as a reference point. Our model attains superior performance in all categories even when context is limited to 1024. The reported loss is the test loss for the noted split. The final two rows do not have a prior comparison as our full dataset split is unique to this work albeit completely open. The perplexity metrics to the right of the BPS are note for reference, but are not directly comparable because the top row uses the AMT-1 tokenization and all others use the AMT-2 tokenization.

Split	Params	Step	Context	Loss	BPS ↓	ppl(e) ↓	ppl(t) ↓	ppl(d) ↓	ppl(n) ↓	ppl(tick) ↓
Lakh Test (f)	360m	800k	1024	0.781	69.7	10.4	1.49	3.29	2.12	
Lakh Test (f)	1.14b	371k	1024*	0.663	59.0	7.5	1.39	2.87	1.92	1.21
Lakh Test (f)	1.14b	371k	4096	0.486	44.0	4.4	1.29	2.28	1.52	1.16
iid Test	1.14b	371k	1024*	0.965	42.2	19.3	2.16	4.72	2.04	1.26
iid Test	1.14b	371k	4096	0.840	37.1	13.2	2.00	4.07	1.72	1.23

A.7.1 SHAZAM COLLECTION

We use the Shazam radio and country charts to source material released past a specific date.

The radio charts list the top two hundred most played songs on the radio anywhere in the world. The charts are subdivided by genres assigned by the Shazam platform. We collect songs from all available genres exposed by the platform on August 25th, 2026. At the time of collection, the charts surveyed were:

- All (any genre): `https://www.shazam.com/radiospins`

- Afrobeats: https://www.shazam.com/radiospins/genre/world/afrobeats
- Alternative: https://www.shazam.com/radiospins/genre/world/alternative
- Country: https://www.shazam.com/radiospins/genre/world/country
- Dance: https://www.shazam.com/radiospins/genre/world/dance
- Electronic: https://www.shazam.com/radiospins/genre/world/electronic
- French Pop: https://www.shazam.com/radiospins/genre/world/french-pop
- Hip-Hop/Rap: https://www.shazam.com/radiospins/genre/world/hip-hop-rap
- House: https://www.shazam.com/radiospins/genre/world/house
- Latin: https://www.shazam.com/radiospins/genre/world/latin
- Música Mexicana: https://www.shazam.com/radiospins/genre/world/m%C3%BAsica-mexicana
- Pop: https://www.shazam.com/radiospins/genre/world/pop
- R&B/Soul: https://www.shazam.com/radiospins/genre/world/randb-soul
- Reggae & Dancehall: https://www.shazam.com/radiospins/genre/world/reggae-dancehall
- Rock: https://www.shazam.com/radiospins/genre/world/rock
- Singer/Songwriter: https://www.shazam.com/radiospins/genre/world/singer-songwriter
- Worldwide: https://www.shazam.com/radiospins/genre/world/worldwide

The Shazam country charts track the songs most often identified by Shazam users in a nation. Some charts can be further subdivided by city, but we collect entries only country-wide. We collect entries from all charts available on August 26th, 2026. At the time of collection, the full list was:

- Algeria: https://www.shazam.com/charts/charts/top-200/algeria
- Argentina: https://www.shazam.com/charts/charts/top-200/argentina
- Australia: https://www.shazam.com/charts/charts/top-200/australia
- Austria: https://www.shazam.com/charts/charts/top-200/austria
- Azerbaijan: https://www.shazam.com/charts/charts/top-200/azerbaijan
- Belarus: https://www.shazam.com/charts/charts/top-200/belarus
- Belgium: https://www.shazam.com/charts/charts/top-200/belgium
- Brazil: https://www.shazam.com/charts/charts/top-200/brazil
- Bulgaria: https://www.shazam.com/charts/charts/top-200/bulgaria
- Cameroon: https://www.shazam.com/charts/charts/top-200/cameroon
- Canada: https://www.shazam.com/charts/charts/top-200/canada
- Chile: https://www.shazam.com/charts/charts/top-200/chile
- China: https://www.shazam.com/charts/charts/top-200/china

- Colombia: https://www.shazam.com/charts/charts/top-200/colombia
- Costa Rica: https://www.shazam.com/charts/charts/top-200/costa-rica
- Croatia: https://www.shazam.com/charts/charts/top-200/croatia
- Czech Republic: https://www.shazam.com/charts/charts/top-200/czechia
- Côte d'Ivoire: https://www.shazam.com/charts/charts/top-200/ivory-coast
- Denmark: https://www.shazam.com/charts/charts/top-200/denmark
- Egypt: https://www.shazam.com/charts/charts/top-200/egypt
- Finland: https://www.shazam.com/charts/charts/top-200/finland
- France: https://www.shazam.com/charts/charts/top-200/france
- Germany: https://www.shazam.com/charts/charts/top-200/germany
- Ghana: https://www.shazam.com/charts/charts/top-200/ghana
- Global: https://www.shazam.com/charts/charts/top-200/world
- Greece: https://www.shazam.com/charts/charts/top-200/greece
- Hungary: https://www.shazam.com/charts/charts/top-200/hungary
- India: https://www.shazam.com/charts/charts/top-200/india
- Indonesia: https://www.shazam.com/charts/charts/top-200/indonesia
- Ireland: https://www.shazam.com/charts/charts/top-200/ireland
- Israel: https://www.shazam.com/charts/charts/top-200/israel
- Italy: https://www.shazam.com/charts/charts/top-200/italy
- Japan: https://www.shazam.com/charts/charts/top-200/japan
- Kazakhstan: https://www.shazam.com/charts/charts/top-200/kazakhstan
- Kenya: https://www.shazam.com/charts/charts/top-200/kenya
- Malaysia: https://www.shazam.com/charts/charts/top-200/malaysia
- Mexico: https://www.shazam.com/charts/charts/top-200/mexico
- Morocco: https://www.shazam.com/charts/charts/top-200/morocco
- Mozambique: https://www.shazam.com/charts/charts/top-200/mozambique
- Netherlands: https://www.shazam.com/charts/charts/top-200/netherlands
- New Zealand: https://www.shazam.com/charts/charts/top-200/new-zealand
- Nigeria: https://www.shazam.com/charts/charts/top-200/nigeria
- Norway: https://www.shazam.com/charts/charts/top-200/norway
- Peru: https://www.shazam.com/charts/charts/top-200/peru
- Philippines: https://www.shazam.com/charts/charts/top-200/philippines
- Poland: https://www.shazam.com/charts/charts/top-200/poland
- Portugal: https://www.shazam.com/charts/charts/top-200/portugal
- Romania: https://www.shazam.com/charts/charts/top-200/romania
- Russia: https://www.shazam.com/charts/charts/top-200/russia

- Saudi Arabia: https://www.shazam.com/charts/charts/top-200/saudi-arabia
- Senegal: https://www.shazam.com/charts/charts/top-200/senegal
- Singapore: https://www.shazam.com/charts/charts/top-200/singapore
- South Africa: https://www.shazam.com/charts/charts/top-200/south-africa
- South Korea: https://www.shazam.com/charts/charts/top-200/south-korea
- Spain: https://www.shazam.com/charts/charts/top-200/spain
- Sweden: https://www.shazam.com/charts/charts/top-200/sweden
- Switzerland: https://www.shazam.com/charts/charts/top-200/switzerland
- Tanzania: https://www.shazam.com/charts/charts/top-200/tanzania
- Thailand: https://www.shazam.com/charts/charts/top-200/thailand
- Tunisia: https://www.shazam.com/charts/charts/top-200/tunisia
- Türkiye: https://www.shazam.com/charts/charts/top-200/t%C3%BCrkiye
- Uganda: https://www.shazam.com/charts/charts/top-200/uganda
- Ukraine: https://www.shazam.com/charts/charts/top-200/ukraine
- United Arab Emirates: https://www.shazam.com/charts/charts/top-200/united-arab-emirates
- United Kingdom: https://www.shazam.com/charts/charts/top-200/united-kingdom
- United States: https://www.shazam.com/charts/charts/top-200/united-states
- Uruguay: https://www.shazam.com/charts/charts/top-200/uruguay
- Uzbekistan: https://www.shazam.com/charts/charts/top-200/uzbekistan
- Venezuela: https://www.shazam.com/charts/charts/top-200/venezuela
- Vietnam: https://www.shazam.com/charts/charts/top-200/vietnam
- Zambia: https://www.shazam.com/charts/charts/top-200/zambia

After consolidating all listings, we remove recordings released before June 4th, 2026. This filtering retains songs *released* on digital streaming platforms past the cut-off date. Recently recorded cover songs of music *composed* before the cut-off as well as re-issued releases are not removed by this filter; some contamination is possible. We choose June 4th, 2026 as the cut-off date because this is the date created and date modified date of the released BEAT (Qian et al., 2026) multi-track checkpoint[2]. While file metadata is not completely reliable, our chosen cut-off date is well past the final submission deadline for the publication venue of BEAT, which was the *International Conference of Machine Learning* 2026. Even if the authors released their full training dataset, detecting musical duplicates could be challenging (Choi et al., 2025). Our full held-out evaluation set contains, in total 1,310 recordings. We will not release our dataset in compliance with copyright law. For completeness, we report summary statistics of its contents: the genre distribution in Table 10; the reported locality in Table 11; and whether there exist associated lyrics, the duration, and artist count in Table 12.

[2]The BEAT authors release a reference implementation with checkpoints here: https://github.com/Lekai-Qian/BEAT-code

Table 10: Distribution of Apple Music Genres. Each entity may be assigned more than one genre. The 'Percent of Genres' column specifies the percentage of all genres together that are a specific category. The 'Percent of Songs' column specifies how many songs in the dataset are associated with the given genre name. Each genre category was assigned by the recording artist representative in accordance with the Apple Music Style Guide (Apple Inc., 2025).

Genre	Count	Percent of Genres	Percent of Songs
Pop	386	17.9	29.5
African	215	10.0	16.4
Hip-Hop/Rap	196	9.1	15.0
Dance	120	5.6	9.2
Latin	108	5.0	8.2
Afro-Beat	75	3.5	5.7
Afro-Pop	65	3.0	5.0
Worldwide	60	2.8	4.6
Arabic	58	2.7	4.4
Afrobeats	51	2.4	3.9
Alternative	48	2.2	3.7
Urbano latino	45	2.1	3.4
R&B/Soul	43	2.0	3.3
Electronic	39	1.8	3.0
Reggae	29	1.3	2.2
Soundtrack	27	1.3	2.1
J-Pop	26	1.2	2.0
Rock	24	1.1	1.8
Christian	24	1.1	1.8
K-Pop	24	1.1	1.8
House	23	1.1	1.8
Música Mexicana	22	1.0	1.7
Amapiano	22	1.0	1.7
Indian	22	1.0	1.7
Singer/Songwriter	21	1.0	1.6
Maghreb Rai	21	1.0	1.6
Country	19	0.9	1.5
Arabic Pop	16	0.7	1.2
Indie Pop	14	0.6	1.1
Pop Latino	14	0.6	1.1
Regional Indian	14	0.6	1.1
Modern Dancehall	13	0.6	1.0
Folk	13	0.6	1.0
Hip-Hop	13	0.6	1.0
Tamil	12	0.6	0.9
Brazilian	11	0.5	0.8
Rap	10	0.5	0.8
South Africa	10	0.5	0.8
Egyptian Pop	10	0.5	0.8
Turkish Pop	9	0.4	0.7
New Age	7	0.3	0.5
Funk	7	0.3	0.5
African Dancehall	6	0.3	0.5
Música tropical	6	0.3	0.5
Jungle/Drum'n'bass	6	0.3	0.5
Bongo-Flava	6	0.3	0.5
French Pop	5	0.2	0.4
Afro House	5	0.2	0.4
Blues	5	0.2	0.4

Continued on next page

Table 10: Distribution of Apple Music Genres. Each entity may be assigned more than one genre. The 'Percent of Genres' column specifies the percentage of all genres together that are a specific category. The 'Percent of Songs' column specifies how many songs in the dataset are associated with the given genre name. Each genre category was assigned by the recording artist representative in accordance with the Apple Music Style Guide Apple Inc. (2025).

Genre	Count	Percent of Genres	Percent of Songs
Mandopop	5	0.2	0.4
Sertanejo	4	0.2	0.3
Alternative Rap	4	0.2	0.3
Classical	4	0.2	0.3
T-Pop	4	0.2	0.3
Indian Pop	4	0.2	0.3
Malaysian Pop	4	0.2	0.3
Jazz	3	0.1	0.2
Afrikaans	3	0.1	0.2
Egyptian Hip-Hop	3	0.1	0.2
Turkish	3	0.1	0.2
Folk-Rock	3	0.1	0.2
Baile Funk	3	0.1	0.2
Rock y Alternativo	3	0.1	0.2
Bollywood	3	0.1	0.2
Europe	3	0.1	0.2
Kizomba	3	0.1	0.2
Electronica	2	0.1	0.2
Forró	2	0.1	0.2
Easy Listening	2	0.1	0.2
Thai Pop	2	0.1	0.2
Techno	2	0.1	0.2
Gospel	2	0.1	0.2
Christian Pop	2	0.1	0.2
Punk	2	0.1	0.2
Hardcore	2	0.1	0.2
Indie Rock	2	0.1	0.2
German Pop	2	0.1	0.2
Anime	2	0.1	0.2
North African	2	0.1	0.2
Zouglou	2	0.1	0.2
Worldbeat	2	0.1	0.2
Maskandi	1	0.0	0.1
Afro Soul	1	0.0	0.1
Khaleeji	1	0.0	0.1
Khaleeji Hip-Hop	1	0.0	0.1
Khaleeji Pop	1	0.0	0.1
Electro-Cha'abi	1	0.0	0.1
Egyptian Tarab	1	0.0	0.1
Tarab	1	0.0	0.1
Turkish Hip-Hop/Rap	1	0.0	0.1
Arabesque	1	0.0	0.1
Vocal	1	0.0	0.1
Fantezi	1	0.0	0.1
Original Score	1	0.0	0.1
Pop in Russian	1	0.0	0.1
Baladas y Boleros	1	0.0	0.1
Devotional & Spiritual	1	0.0	0.1
Metal	1	0.0	0.1

Continued on next page

Table 10: Distribution of Apple Music Genres. Each entity may be assigned more than one genre. The 'Percent of Genres' column specifies the percentage of all genres together that are a specific category. The 'Percent of Songs' column specifies how many songs in the dataset are associated with the given genre name. Each genre category was assigned by the recording artist representative in accordance with the Apple Music Style Guide Apple Inc. (2025).

Genre	Count	Percent of Genres	Percent of Songs
Soul	1	0.0	0.1
Chinese	1	0.0	0.1
Samba	1	0.0	0.1
Fitness & Workout	1	0.0	0.1
Iraqi Folk	1	0.0	0.1
Acoustic Blues	1	0.0	0.1
Korean Hip-Hop	1	0.0	0.1
Kayokyoku	1	0.0	0.1
Bossa Nova	1	0.0	0.1
Neo-Soul	1	0.0	0.1
Roots Reggae	1	0.0	0.1
Classic Christian	1	0.0	0.1
Afro-fusion	1	0.0	0.1
Dubstep	1	0.0	0.1
Classical Crossover	1	0.0	0.1
Mbalax	1	0.0	0.1
Indonesian Religious	1	0.0	0.1
Alternative Folk	1	0.0	0.1
Instrumental	1	0.0	0.1
Latin Jazz	1	0.0	0.1
Latin Rap	1	0.0	0.1
Maghreb Dance	1	0.0	0.1
Garage	1	0.0	0.1
Pop/Rock	1	0.0	0.1

Table 11: Distribution of reported localities.

Locale	Name	Count	%
en-US	English (United States)	390	29.8
es-ES	Spanish (Spain)	107	8.2
ar	Arabic	104	7.9
fr-FR	French (France)	75	5.7
ru	Russian	47	3.6
pt-PT	Portuguese (Portugal)	40	3.1
zxx	No Linguistic Content	40	3.1
ja	Japanese	39	3.0
sw	Swahili	31	2.4
ro	Romanian	29	2.2
tr	Turkish	26	2.0
pl	Polish	24	1.8
ko	Korean	23	1.8
bg	Bulgarian	23	1.8
vi	Vietnamese	22	1.7
uk	Ukrainian	17	1.3
el	Greek	14	1.1
he	Hebrew	14	1.1
th	Thai	13	1.0

Continued on next page

Table 11: Distribution of reported localities.

Locale	Name	Count	%
ms	Malay	13	1.0
it	Italian	12	0.9
af	Afrikaans	12	0.9
de-DE	German (Germany)	12	0.9
da	Danish	12	0.9
hu	Hungarian	12	0.9
az	Azerbaijani	12	0.9
zu	Zulu	11	0.8
kk	Kazakh	11	0.8
ta	Tamil	9	0.7
uz	Uzbek	9	0.7
es-419	Spanish (Latin America)	8	0.6
hi	Hindi	8	0.6
zh-Hans-CN	Chinese (Simplified, China)	7	0.5
id	Indonesian	7	0.5
cs	Czech	7	0.5
swh	Swahili (Tanzania)	5	0.4
hr	Croatian	5	0.4
pa	Punjabi	4	0.3
ml	Malayalam	4	0.3
la	Latin	3	0.2
fi	Finnish	3	0.2
ha	Hausa	3	0.2
sv	Swedish	3	0.2
bm	Bambara	3	0.2
ve	Venda	2	0.2
xh	Xhosa	2	0.2
st	Southern Sotho	2	0.2
nso	Northern Sotho	2	0.2
sr	Serbian	2	0.2
yo	Yoruba	2	0.2
ig	Igbo	2	0.2
lg	Ganda	2	0.2
wo	Wolof	2	0.2
ny	Nyanja	2	0.2
ts	Tsonga	2	0.2
pcm	Nigerian Pidgin	1	0.1
sq	Albanian	1	0.1
ur	Urdu	1	0.1
bgc	Haryanvi	1	0.1
te	Telugu	1	0.1
tl	Filipino (Philippines)	1	0.1
fil	Filipino	1	0.1
ary	Moroccan Arabic	1	0.1
ak	Akan	1	0.1
tw	Akan (Ghana)	1	0.1
jam	Jamaican Creole English	1	0.1
nl	Dutch	1	0.1
fa	Persian	1	0.1
tg	Tajik	1	0.1
ca	Catalan	1	0.1

A.8 SAMPLING FOR PROMPT CREATION

We sample uniformly at random with replacement from our held-out Shazam recordings. From these samples, we extract their metadata, e.g., genre, number of artists, etc; the metadata reported in Tables 10, 11, 12 to use in the construction of text prompts for Suno. We collect prompts written by ChatGPT 5.6-Sol (OpenAI, 2026) with "high" thinking effort on August 30th, 2026. Our prompt is as follows:

```
I would like you to help me create a list of text
prompts for the text-to-music generation platform
Suno.  I'd like you to translate a list of JSON
objects into natural text that can be used as a prompt
in Suno.  These prompts must be crafted so that they
produce high-quality, musical outputs that agree with
the basic metadata given.

I will provide a list of 80 small JSON objects.
In each JSON, there are 4 keys:  "genres",
"minutes_approx", "total_artists", "language".
- The "genres" key is a list of Apple Music Genres
assigned to the song.
- The "minutes_approx" key is the approximate number
of minutes for the song.
- "total_artists" is the number of recording artists
that worked on the song.  When the total artists
value is greater than 1, this means that the song
is a collaboration.  A collaborative song can have
features, be a remix, or be the work of a 'supergroup'
or side project of prominent artists with their own
respective profiles on Digital Streaming Platforms.
- The "language" is a human readable language derived
from an ISO locality.  The language may or may not
include a region.  If the "language" key contains
"No Linguistic Content", then the composition is
instrumental with no vocals.  In this case, be sure
to mention that the song is instrumental.  The "No
Linguistic Content" value is common in dance or
electronic music, but may appear for any genre.

Note that the contents in the "language" and the
"genres" column might have redundancies or conflicts;
this is because there exist Apple Music genres named
after countries or languages.  For example, a value
in the "genres" key might be ["Pop in Russian"] and
the "language" might be "Russian".  The "genres"
key and "language" key might also conflict, e.g.
"genres" is equal to ["African"], but the "language"
key contains "English (United States)".  Both these
redundancies and conflicts are ok.  When encountering
instances like this, craft your prompt such that all
the information contained in both keys is preserved
while remaining natural and well suited for the Suno
platform.

Here are the JSON objects, please turn them into
prompts for me:

{'genres':  ['Techno', 'Dance', 'Pop'],
'minutes_approx':  3.16, 'total_artists':  1,
```

```
'language': 'English (United States)'}
{'genres': ['Afro-Beat', 'African'], 'minutes_approx':
1.82, 'total_artists': 1, 'language': 'English
(United States)'}
...
```

(full list omitted)

The response yields 80 natural language prompts that we then use to prompt Suno v6 Pro.

A.8.1 SUNO COLLECTION DETAILS

We manually enter the 80 natural language prompts into the Suno "Create" interface. We use the "Simple" configuration with their latest model, v6 Pro. See Figure 9 for a depiction of our entry method.

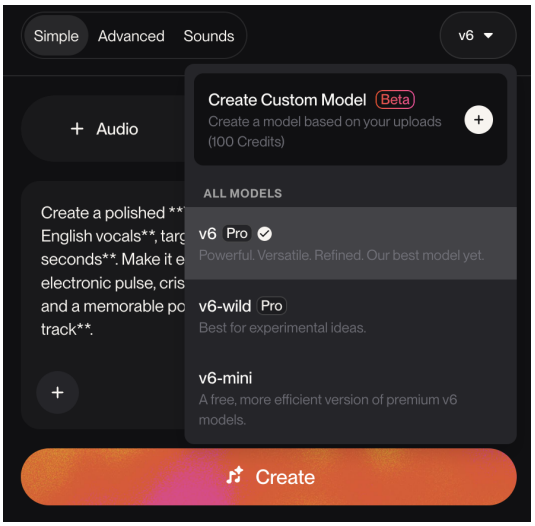

Figure 9: **Prompting routine for Suno.** Any **bold** instructions output by ChatGPT 5.6-Sol in the form of '*' marks are retained. We do not provide lyrics, sounds, or any instructions beyond our natural language prompt.

Upon entering the text prompt, the platform generates two outputs. Of the two, we choose the one that more closely matches the requested duration. If both prompts match the requested duration, we select the output that finished generating first. We download all 80 songs as lossless waveform files.

A.9 MULTI-TRACK TRANSCRIPTION

After obtaining the full audio recordings for the works collected from Shazam, we transcribe them with MuScriptor (Rouard et al., 2026). We use model settings that maximize transcription quality: the largest model, enable prelude-forcing, set the batch size to 1, and enable "best-effort" tempo detection. We do not specify which instruments are in the audio. A model call illustration is shown in Listing 1.

A.10 MODEL COMPARISON SAMPLING PROCEDURE

A.10.1 MUPT

To compare ABC notation with our tokenization, we compare against MuPT. To our knowledge, MuPT is the largest open weight symbolic music model, a 4.23 billion parameter transformer trained on a private 33.6 billion token dataset of music in ABC notation. However, only the 1.97 billion parameter checkpoint is available, which we for our comparison.

```python
import torch
from muscriptor import TranscriptionModel

model = TranscriptionModel.load_model(
    weights_path="large",
    device="cuda",
)
audio_path = Path("my_song.mp3")
with torch.inference_mode():
    # best quality settings possible
    midi_bytes = model.transcribe_to_midi(
        audio_path,
        prelude_forcing=True,
        batch_size=1,
        detect_tempo="best-effort",
    )
    write_to.write_bytes(midi_bytes)

```

Listing 1: Illustration of a call to MuScriptor.

We first sample unconditionally from their model to obtain 65 seconds of ABC notation. In ABC notation, chords may appear as annotations but are not always voiced. When we convert their outputs to MIDI, we ensure any silent chordal annotations are voiced. Using these MIDI files, we use the first five seconds as a prompt for our model and generate an additional 20 seconds of continuation. Doing this ensures the conditioning information given to our model is close to MuPT's learned distribution, which we do to ensure a fair comparison. We convert the samples from both models into audio using FluidSynth and truncate the waveform to 25 seconds.

A.10.2 AMT-1

As our work is based on the anticipatory music transformer (AMT-1) (Thickstun et al., 2024), we have withheld their test split from training in all experiments. The test split is folder `f` in the Lakh MIDI (Raffel, 2016) dataset and contains samples unseen during training for both models. We compare against the highest performing model reported, which is the medium (360 million parameter) anticipatory model trained for $800{,}000$ steps. For all tokenizable files, we randomly select regions in each that contain at least 25 seconds of music - this is to get an assessment of continuation quality not just at the start of a composition but anywhere. We further discard any files that have ambiguities when being converted to audio by FluidSynth, yielding $9{,}841$ total MIDI files, which we subsample to obtain $1{,}230$ prompts. We then use the first five seconds of these prompts as conditioning and perform autoregressive continuation on them for both models – this means both use the autoregressive token instead of the anticipatory δ token. For AMT-1, we use the settings reported as the highest performing, which is nucleus sampling with $p = 0.95$ (Holtzman et al., 2020). Both continue until 25 seconds are reached.

A.10.3 BEAT

With the exception of Lakh MIDI, the training and test splits of BEAT (Qian et al., 2026) are private. They report training on all of Lakh MIDI so we use our 2026 held-out dataset described in Section 5.1 to prevent contamination. We compare against the released multi-track model, which is a 150 million parameter LLaMA style transformer (Touvron et al., 2023). BEAT quantizes MIDI to a 16th note grid before tokenizing it. We use their released source code to apply this quantization to our MuScriptor outputs and source our conditioning prompts from the quantized files.

Similarly to our prompt sampling procedure for AMT-1, we uniformly sampled subsequences of at least 25 seconds from the BEAT-quantized 2026 held-out dataset. To prevent exceptionally sparse prompts, we reject subsequences with fewer than ten events in their first five seconds. We also refrain from sampling in the first five percent and last five percent of the duration of the file, since we found there is often silence at these regions.

Both models are given the first five seconds of these prompts as conditioning information and are asked to continue until 25 seconds are reached.

A.10.4 SUNO

We aim to procure samples that reflect our 2026 held-out dataset from Suno v6 Pro (Suno, Inc.). To do this, we randomly sample with replacement 80 instances of metadata from our 2026 held-out dataset. We then provide this metadata to ChatGPT 5.6, which we ask to convert to natural language prompts that we may input into Suno. We use these prompts to produce audio output from Suno and download them as lossless waveforms. The full details of this collection procedure are described in Appendix A.7. We obtain 80 audio files that we transcribe with MuSciptor.

We follow an identical subsequence sampling procedure as done in for BEAT, but without the quantization step. We take the first 25 seconds of the segment to be the Suno sample. We use the first five seconds of the segment as conditioning information for our model to continue and obtain samples from our model that share the same first five seconds but thereafter diverges. Both are converted to audio with FluidSynth.

A.11 HUMAN EVALUATION

A.11.1 PARTICIPANT DISTRIBUTION

We source our survey participants from Prolific. We require participants to have been active on Prolific within the last 90 days, to understand English, and to have at least a 50% submission approval rating. We do not restrict participants to a specific location and do not require participants to have a musical background.

Participants must pass a four question screening. The screening asks which of two 25 second prompts is more musically cohesive. For these screenings, we have taken MIDI from the Lakh MIDI dataset, a deliberately altered it to be tonally dissonant and rhythmically inconsistent to an extreme. If a participant answers any of these screening questions incorrectly, they cannot proceed and we do not collect ratings from them.

We pay participants $22 USD per hour, which is deemed ‘competitive’ by the Prolific platform.

A.11.2 SURVEY INTERFACE

We implement our survey through an interactive website. Figure 10 contains a screenshot of the screening instructions and Figure 11 shows a screening question. Once participants pass the screening, the instructions change slightly and ask them to decide which of two unlabeled audio prompts is more “conventionally musical”. A screenshot of the instructions for the main study is shown in Figure 12 and a screen depicting a survey question is shown in Figure 13.

A.12 TRAINING DETAILS

For completeness, we note additional software we use to implement our models. We use Flash Attention 4 (Zadouri et al., 2026) as our scaled dot production attention (Vaswani et al., 2017) implementation. We use the LigerKernel (Hsu et al., 2025) implementation `LigerRMSNorm` and their implementation of cross entropy Loss that avoids vocabulary materialization, `LigerFusedLinearCrossEntropyLoss`. We use NVIDIA Transformer Engine (NVIDIA, 2026) for the layers: `Linear`, `LayerNormMLP`, and for its FP8 low precision training recipe, i.e. `DelayedScaling(fp8_format=Format.HYBRID)`.

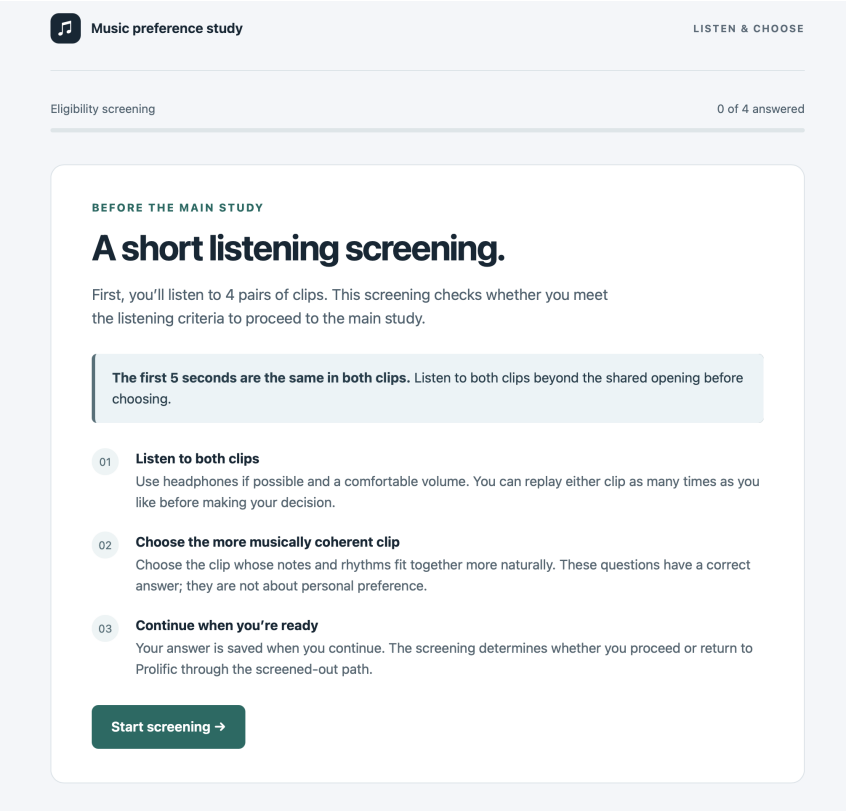

Figure 10: **Human evaluation screening instructions.**

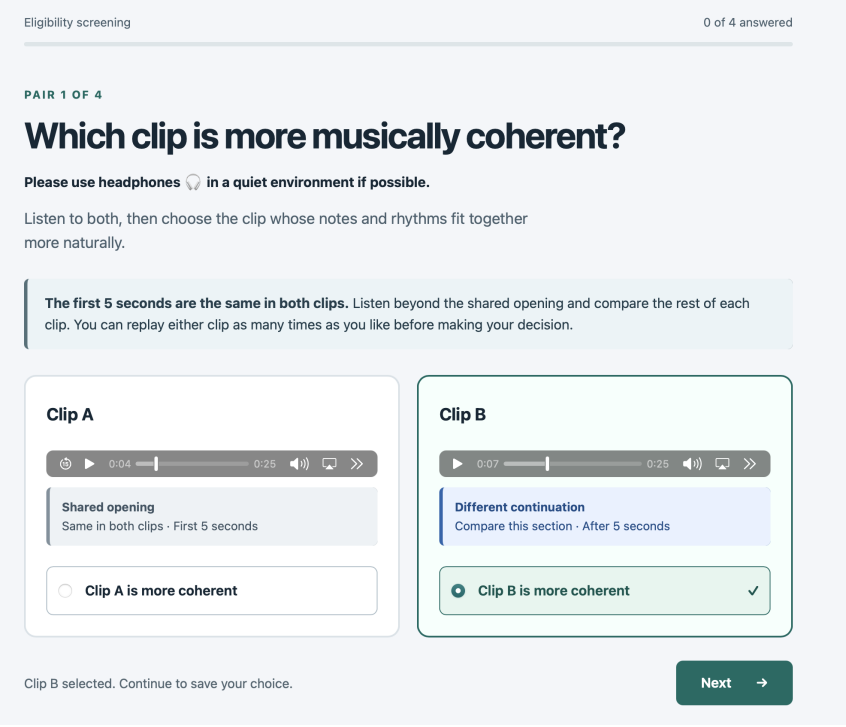

Figure 11: **Human evaluation screening example.**

Table 9: **Coefficients for all data-constrained scaling law forms.** Our result corroborates the claim in Lovelace et al. (2026) that their 4-parameter equation attains the best goodness of fit, which is why we use it for our extrapolation. Following Karpathy (2025), we count the parameters N in three ways: all parameters, transformer matrices only, and all parameters except the embeddings. We can see that using all model parameters *including* embeddings as N yields the best fit, which is consistent with the recommendation in (Pearce & Song, 2024)

	N counting Style		
Coefficient / Metric	All	Transformer matrices	All except embeddings
Chinchilla (shared baseline)			
A	12,550.94	60.88	1,750.15
B	64,372.58	64,895.71	64,424.41
E	0.546827	0.412780	0.524755
α	0.580660	0.276709	0.475068
β	0.526487	0.526882	0.526525
Huber ↓	0.007323	**0.007271**	0.007307
R^2 ↑	**0.923173**	0.923059	0.923137
Adj. R^2 ↑	**0.918600**	0.918480	0.918562
Muennighoff: Effective Data			
$R^\star$	4.212825	4.277208	4.230422
Huber ↓	0.018646	**0.018539**	0.018622
R^2 ↑	0.922800	**0.923227**	0.922905
Multi R^2 ↑	0.920815	**0.922339**	0.921208
Adj. R^2 ↑	0.922437	**0.922867**	0.922543
Muennighoff: Effective Data and Parameters			
$R_D^\star$	12.246752	13.261592	11.346657
$R_N^\star$	17.753233	2,426.628418	50.192036
Huber ↓	**0.017619**	0.018012	0.017679
R^2 ↑	**0.886316**	0.865885	0.884319
Multi R^2 ↑	**0.933525**	0.916133	0.931077
Adj. R^2 ↑	**0.885244**	0.864620	0.883228
Lovelace: 1-Parameter Form			
P	2.98289×10^{-2}	3.59804×10^{-2}	3.09903×10^{-2}
Huber ↓	**0.021108**	0.021787	0.021413
R^2 ↑	0.858805	0.857075	**0.865276**
Multi R^2 ↑	0.726981	0.721973	**0.746654**
Adj. R^2 ↑	0.858142	0.856404	**0.864643**
Lovelace: 2-Parameter Form			
P	3.86922×10^{-2}	4.38471×10^{-2}	4.13953×10^{-2}
κ	0.546348	0.513215	0.526099
Huber ↓	**0.017893**	0.018222	0.018004
R^2 ↑	**0.914276**	0.911763	0.913591
Multi R^2 ↑	**0.894999**	0.887614	0.892997
Adj. R^2 ↑	**0.913468**	0.910930	0.912776
Lovelace: 4-Parameter Form			
P	4.72161×10^2	1.22962×10^3	1.02041×10^3
δ	0.198928	0.210401	0.188228
κ	0.246172	0.218448	0.217307
γ	2.577289	2.990786	2.961478
Huber ↓	**0.016649**	0.016705	0.016655
R^2 ↑	**0.927375**	0.925513	0.927264
Multi R^2 ↑	**0.934956**	0.929490	0.934791
Adj. R^2 ↑	**0.925992**	0.924094	0.925879

Table 12: **Distributions of our recent music dataset.** Whether the song has lyrics reflects whether the lyrics are available in Apple Music, not whether a vocal exists.

(a) Lyrics

Has Lyrics	Count	%
False	563	43.0
True	747	57.0

(b) Duration

Duration (Seconds)	Count	%
(34, 231]	1,014	77.4
(231, 428]	270	20.6
(428, 625]	17	1.3
(625, 822]	2	0.2
(822, 1,019]	1	0.1
(1,216, 1,413]	2	0.2
(1,413, 1,610]	2	0.2
(2,399, 2,596]	1	0.1
(3,778, 3,975]	1	0.1

(c) Collaborations

Total Artists	Count	%
1	919	70.2
2	261	19.9
3	92	7.0
4	31	2.4
5	4	0.3
6	2	0.2
10	1	0.1

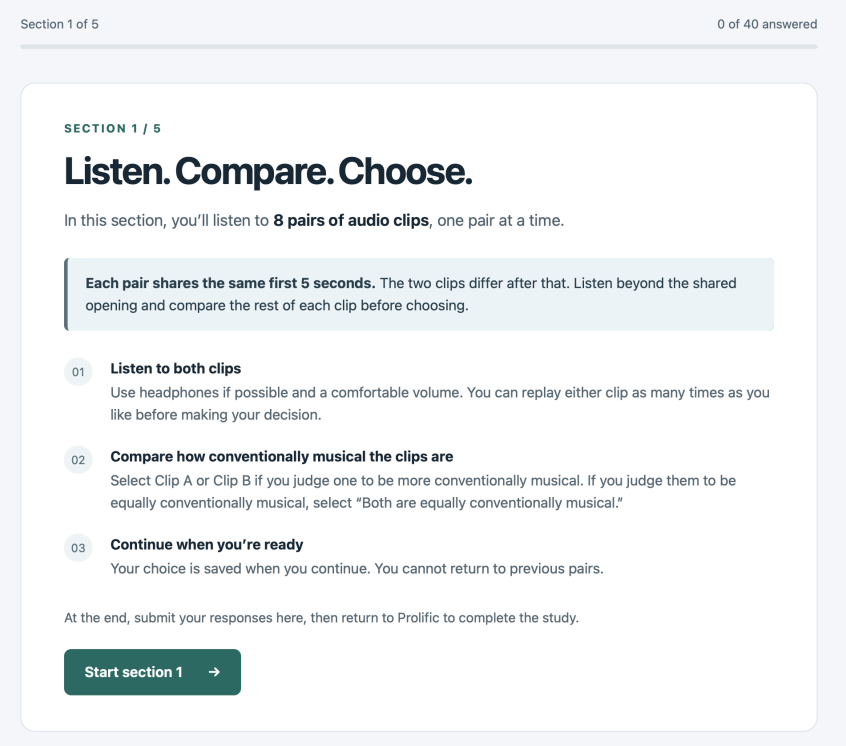

Figure 12: **Human evaluation instructions for real model output comparisons.**

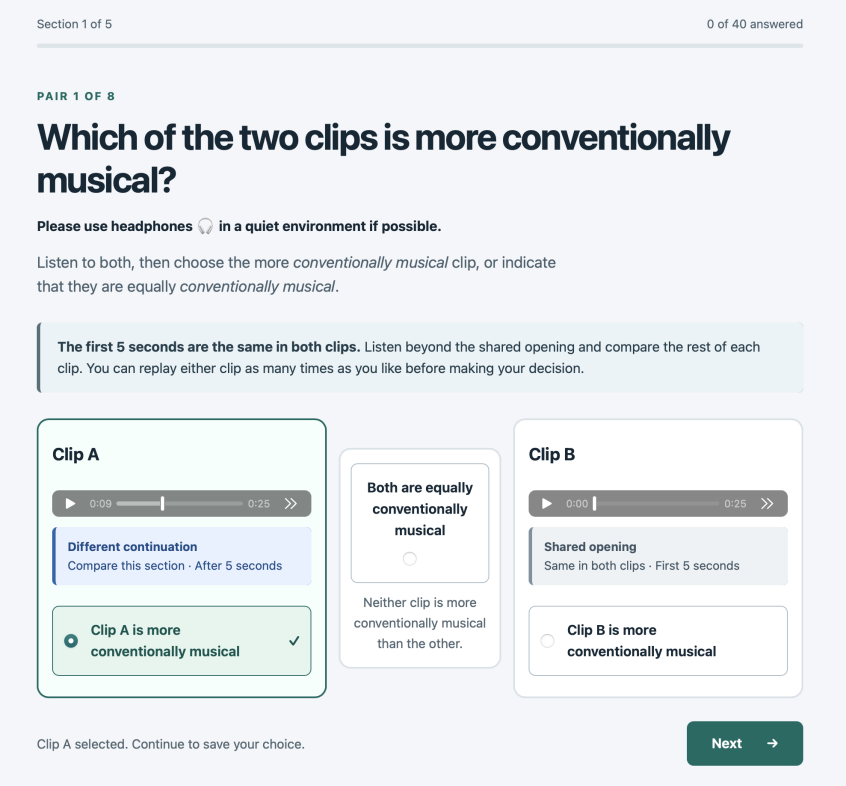

Figure 13: **Human evaluation comparison sample screen.** Model outputs are hidden and randomized. The popup below the audio playhead changes from grey to blue after five seconds have passed to emphasize that material past this point is different.