# Do Music Transformers Benefit from Music Theory?

Evaluating Explicit Harmonic and Temporal Biases
in Attention for Symbolic Music Modeling

Caleb Dame
Johns Hopkins University
EN.705.744: Deep Learning Using Transformers
cdame1@jhu.edu

Spring 2026

**Abstract**

Transformer models for symbolic music typically treat musical sequences as generic token streams, relying on learned attention patterns to discover harmonic and temporal structure from data. Music, however, has well-understood relational structure that can serve as an inductive bias. We explore injecting learned bias embeddings directly into the attention logits of a decoder-only transformer, encoding two types of musical relationship: a *harmonic interval* based on the directed circle of fifths and a *temporal distance* based on binned onset-time differences in beats. We discretize each relationship and learn a separate bias value per bin on each attention head. We train on the GigaMIDI dataset (approximately 382,000 pitched MIDI files) and compare four conditions: no bias (baseline), circle-of-fifths harmonic bias, temporal bias, and a combined model using both additively. Our experiments test whether these structural priors improve sample efficiency, speed up convergence, produce more interpretable attention patterns, and reduce next-token prediction loss relative to the unbiased baseline.

## 1 Introduction

Symbolic music transformers have seen rapid development in recent years. Models like Music Transformer [4] and MusicBERT [16] have achieved strong results on tasks ranging from generation to classification by encoding musical attributes as token embeddings and relying on self-attention to learn relevant relationships. These models treat music as a generic sequence, leaving the attention mechanism to discover harmonic and temporal structure entirely from a sequence's encoded tokens and relative order.

Music, unlike natural language, has relational structure that is well-understood and easy to compute. Two relationships seem particularly relevant:

- **Harmonic proximity (circle of fifths).** The circle of fifths defines a natural distance between pitch classes based on functional harmony. Notes separated by a perfect fifth (e.g., C and G)

are harmonically close and frequently appear together in tonal music, while notes separated by a tritone (e.g., C and F#) are maximally distant. This structure underlies chord progressions, scales, cadences, and key relationships across most Western music.

- **Temporal proximity.** Notes have continuous onset times rather than uniform discrete positions. Two notes in a chord share an onset (temporal distance zero), while a melody note and its accompaniment one bar later are temporally distant. Standard positional encodings assign positions based on token index, which does not capture this timing information and may even mislead the model.

Prior work has shown that injecting structural biases directly into attention can be effective. ALiBi [8] demonstrated that simple linear position biases in attention logits can match or exceed learned positional embeddings while enabling length extrapolation. Graphormer [14] showed that encoding graph-theoretic distances (shortest path, node degree) as attention biases improves transformer performance on molecular and graph-level prediction tasks. These results suggest that when known structure exists within a sequence, encoding it as an attention bias can provide a useful inductive prior.

We apply this principle to symbolic music. Our contributions:

1. We add learned distance-bias embeddings to transformer attention for symbolic music, encoding harmonic and temporal priors as a separate bias per discrete distance level per head, which can capture non-linear distance-preference curves.

2. We modify MusicBERT's Octuple compound embedding by adding learned per-attribute scaling factors, initialized to 1, so the model can adjust the relative importance of each musical attribute during training.

3. We design a controlled experiment with four conditions to isolate the effect of each bias, with ablations across data sizes (1%, 5%, 20%, 100%) on the GigaMIDI dataset.

## 2 Related Work

### 2.1 Transformer Models for Symbolic Music

Music Transformer [4] adapted relative attention [10] for long MIDI sequences, showing that relative positional representations improve long-term coherence. MusicBERT [16] introduced OctupleMIDI, a compound tokenization where each note is a single token of eight parallel attributes, yielding dense sequences well-suited for pairwise distance computation. Sequential schemes like REMI [5] and Compound Word Transformer [3] offer different trade-offs between sequence length and musical expressivity.

More recently, Museformer [15] used sparse attention patterns based on bar-level statistics to handle longer MIDI sequences. The Anticipatory Music Transformer [12] focused on controllable

infilling via interleaved event and control sequences, but does not modify the attention computation itself. Wang et al. [13] used a novel hierarchical music language as a structural prior for a diffusion model, operating at a different level of abstraction (song form and entire phrases) than our note-level pairwise distances.

### 2.2 Attention Biases and Structural Priors

As mentioned earlier, ALiBi [8] and Graphormer [14] showed that adding bias terms directly to attention logits can encode structural information effectively. Shaw et al. [10] introduced learnable relative position representations in attention, which influenced Music Transformer and many subsequent architectures. Our work follows a similar strategy but uses structural note relationships (circle of fifths and beat distance from note onset) rather than classical sequential positional ones, with learned embeddings per discrete distance bin or harmonic relationship.

### 2.3 Music-Theoretic Structure in Neural Models

Pitch-class distributions and circle-of-fifths relationships have long been used in computational music analysis for key estimation [11] and tonal modeling [1]. Previous neural approaches to incorporating music theory have typically worked at the feature or loss level. For example, Ji et al. [6] use music-theory-based reward functions (chord progression rules, tonal distance) in reinforcement learning for melody harmonization. We are not aware of prior work that injects pitch-class distance directly into the attention mechanism of a music transformer as a learned bias term, though it is possible such work exists.

## 3 Problem Statement

As our benchmark performance task, we chose autoregressive next-note prediction on symbolic music. Given a sequence of notes $x_1, x_2, \ldots, x_{t-1}$, each represented as a compound token of eight attributes, the model predicts the distribution over possible values for all eight attributes of the next note $x_t$.

Standard transformer architectures for this task compute attention weights from learned query-key dot products and positional encodings. While these models can in principle learn any pairwise relationship, they receive no explicit signal about the harmonic or temporal distance between notes. The model must learn from scratch that harmonically related pitch classes tend to appear together, and that notes with similar onset times (e.g., notes in a chord) interact differently from notes that are further apart in time.

We hypothesize and hope to test that encoding these relationships as explicit attention biases may:

**H1. Improve sample efficiency:** models with biases may achieve equivalent performance with less training data, since the biases provide structural priors that would otherwise need to be

learned.

**H2. Accelerate convergence:** biased models may reach a given loss level in fewer training steps, as the priors could provide a useful initialization of the attention landscape.

**H3. Improve interpretability:** the learned distance-bias curves could reveal which structural relationships matter at each layer and head.

**H4. Improve prediction quality:** biased models may achieve lower next-token prediction loss than an equivalent baseline.

# 4 Method

## 4.1 Attention Bias Formulation

We modify the standard scaled dot-product attention by adding learned bias embeddings to the attention logits before the softmax. Each musical relationship is discretized into bins, and the model learns a separate bias value per bin per head via an embedding table. For a sequence of $n$ notes, the biased attention for a given head is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B_{\text{harm}}[D_{\text{harm}}] + B_{\text{temp}}[D_{\text{temp}}]\right) V \tag{1}$$

where $Q, K, V \in \mathbb{R}^{n \times d_k}$ are the query, key, and value projections; $D_{\text{harm}}[i, j]$ is an integer bin index encoding the directed harmonic interval from note $i$ to note $j$; $D_{\text{temp}}[i, j]$ encodes the temporal distance; and $B_{\text{harm}}$, $B_{\text{temp}}$ are learned embedding tables mapping each bin index to a scalar bias value.

Each embedding table is per-head. Biases are initialized from $\mathcal{N}(0, 0.02)$ so the model starts close to standard attention and gradually learns which distance levels should increase or decrease attention weight. Note that the bias terms are added to the already-scaled logits $QK^\top/\sqrt{d_k}$: a unit change in a bias value has the same effect on the softmax as a $\sqrt{d_k}$-unit change in the raw dot product. To account for this, we train the bias embeddings with a learning rate scaled by $1/\sqrt{d_k}$ relative to the main model parameters (Section 4.7).

The four experimental conditions correspond to different subsets of active bias terms:

- **Baseline (none):** No distance biases. Standard transformer attention.
- **Harmonic (harm):** $B_{\text{harm}}$ active only. Directed circle-of-fifths interval embeddings (12 non-null bins, intervals 0–11).
- **Temporal (temp):** $B_{\text{temp}}$ active only. Binned temporal distance embeddings (17 non-null bins, musically-motivated from simultaneous to distant).
- **All:** $B_{\text{harm}} + B_{\text{temp}}$

### 4.2 Bias Matrix Definitions

**Harmonic interval ($D_{\text{harm}}$).** Let $c_i \in \{0, \ldots, 11\}$ be the pitch class of note $i$. We map each pitch class onto the circle of fifths via $\text{fifths}(c) = (7c) \bmod 12$ and compute the directed interval:

$$D_{\text{harm}}[i, j] = \big(\text{fifths}(c_j) - \text{fifths}(c_i)\big) \bmod 12 \tag{2}$$

This yields values in $\{0, \ldots, 11\}$, where 0 is the same note pitch (octave-invariant) and 6 is the tritone. For example, C→G is 1 (one fifth up) while G→C is 11 (one fifth down, equivalently eleven fifths up). Since we learn a separate bias per bin, the model can discover whether ascending and descending fifth relationships play different roles. The 12 values map to embedding bins 1–12 (bin 0 is reserved for structural tokens with no possible bias).

**Temporal distance ($D_{\text{temp}}$).** Onset time in beats is computed from bar number, position, and time signature. The continuous distance $|t_i - t_j|$ is discretized into 17 bins with sub-beat resolution at the fine end, per-beat coverage through eight beats, and coarser long-range bins:

| Bin | Beat range (interpretation) |
|---|---|
| 1 | $[0, 0.25)$ – simultaneous (chords) |
| 2 | $[0.25, 0.5)$ – sub-beat (sixteenth-note level) |
| 3 | $[0.5, 0.75)$ – sub-beat (eighth-note level) |
| 4 | $[0.75, 1.0)$ – sub-beat (dotted eighth) |
| 5 | $[1.0, 1.5)$ – beat 1 |
| 6 | $[1.5, 2.0)$ – beat 1.5 |
| 7 | $[2.0, 3.0)$ – beat 2 |
| 8 | $[3.0, 4.0)$ – beat 3 |
| 9 | $[4.0, 5.0)$ – beat 4 (one bar in 4/4) |
| 10 | $[5.0, 6.0)$ – beat 5 |
| 11 | $[6.0, 7.0)$ – beat 6 |
| 12 | $[7.0, 8.0)$ – beat 7 |
| 13 | $[8.0, 12.0)$ – bar 2 (two bars in 4/4) |
| 14 | $[12.0, 16.0)$ – bars 3–4 (phrase-level) |
| 15 | $[16.0, 32.0)$ – multi-phrase |
| 16 | $[32.0, 64.0)$ – section-level |
| 17 | $[64.0, \infty)$ – distant |

This binning provides fine granularity where musical structure is most detailed (sub-beat and beat-level) while still covering long-range relationships.

### 4.3 Tokenization

We use MusicBERT's Octuple compound tokenization [16], implemented via MidiTok [2]. Each note becomes a tuple of eight attribute IDs (pitch, position, bar, velocity, duration, program, tempo, time signature); see the original paper for full vocabulary details. Bar numbers are remapped per training window so numbering always starts near zero in order to reduce the vocabulary and sparsity of bar prediction.

We modify MusicBERT's embedding by introducing a learned scalar multiplier $s_a$ per attribute, initialized to 1.0. MusicBERT sums the eight attribute embeddings with equal weight; our version lets the model adjust each attribute's contribution during training, since not all attributes are equally informative (e.g., pitch changes every note while tempo rarely does). The combined note representation is:

$$\mathbf{e}_i = \sum_{a=1}^{8} s_a \cdot \text{Embed}_a(x_i^{(a)}) \tag{3}$$

The final values of $s_1, \ldots, s_8$ also serve as a simple interpretability signal for ranking which attributes the model found generally most useful.

The Octuple representation is important for our approach because each sequence position corresponds to exactly one note, making the distance matrices fully dense. Sequential tokenization schemes like REMI would interleave timing and structural tokens with no meaningful pitch-class distance, producing sparse bias matrices and weakening our ability to measure any benefit from the biases.

### 4.4 Dataset

We train on the GigaMIDI v2.0.0 dataset [7], which contains approximately 2.1 million MIDI files sourced from diverse genres. We apply the following filtering criteria:

- **Instrument filter:** We retain only files labeled "no-drums" in the dataset metadata. We further verify each file at the MIDI level and skip any file containing a drum track (`is_drum=True` in symusic), since approximately 40% of files labeled "no-drums" in GigaMIDI actually contain MIDI channel 10 percussion. This yields approximately 382,000 files across the train, validation, and test splits. Excluding drums maximizes the relevance of pitch-based biases; drum tracks could be reincluded in future work at the cost of increased program vocabulary and potentially weaker harmonic bias effects.
- **Minimum note count:** Files with fewer than 50 notes are discarded.
- **Maximum bar count:** Files exceeding 2,000 bars are skipped as they are likely corrupted or pathological (less than 0.05% of files, some with millions of bars). Skipping these is important since window sampling is proportional to piece length.

After filtering and tokenization, the training set contains approximately 306,000 pieces with a combined total of roughly 452 million notes. The validation and test sets each contain approximately

38,000 pieces ( 56 million notes each). The preprocessed data is stored as concatenated NumPy arrays totaling approximately 12 GB on disk.

The dataset is split 80/10/10 into train, validation, and test sets (splits provided by the dataset authors). For sample efficiency experiments (H1), we construct 1%, 5%, and 20% subsets by randomly sub-selecting pieces from the training set using a fixed random seed.

### 4.5 Data Pipeline

At training time, each piece is divided into overlapping windows of up to $S = 1024$ notes. The number of windows sampled per piece is proportional to its length: each piece receives $\max(1, \lceil \text{piece\_len}/S \rceil)$ window slots per epoch, ensuring that longer pieces contribute proportionally more training samples while every piece is seen at least once. Within each slot, the starting position is randomized with jitter for data augmentation. Short pieces are left-padded; long pieces always produce full-length windows.

Bar numbers are remapped within each window so the minimum bar maps to a reference value, keeping the bar vocabulary compact regardless of absolute position in the original file.

Bias matrices are computed on-the-fly on the GPU or CPU rather than precomputed and stored on disk. The harmonic distance uses a precomputed $12 \times 12$ lookup table indexed per attention layer.

### 4.6 Model Architecture

We use a decoder-only transformer since our task is autoregressive next-token prediction and a decoder with causal masking is a standard choice for this setting. This follows the same high-level architecture used by GPT-style language models [9] and by music generation systems such as Music Transformer [4]. The model has the following components:

**Compound embedding.** The input to the model is a tensor of shape $(B, S, 8)$ containing token IDs for each of the eight attributes. The scaled attribute embeddings from Section 4.3 are summed and added to a learned positional embedding to produce the initial hidden state:

$$\mathbf{h}_i^{(0)} = \mathbf{e}_i + \text{PosEmbed}(i) \tag{4}$$

**Transformer blocks.** The model consists of $L$ transformer blocks with pre-layer normalization. Each block contains a biased multi-head attention layer (Equation 1) followed by a position-wise feed-forward network with GELU activation:

$$\mathbf{h}' = \mathbf{h} + \text{BiasedMHA}(\text{LN}(\mathbf{h}),\ D_{\text{harm}},\ D_{\text{temp}}) \tag{5}$$

$$\mathbf{h}^{(\ell+1)} = \mathbf{h}' + \text{FFN}(\text{LN}(\mathbf{h}')) \tag{6}$$

**Output heads.** After the final layer normalization, eight parallel linear heads project the hidden state into logits over each attribute's vocabulary. The training loss is a weighted sum of cross-entropy losses across all eight attribute heads. Only non-padding positions contribute to the loss:

$$\mathcal{L} = \sum_{a=1}^{8} \frac{w_a}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} -\log p(x_i^{(a)} \mid x_{<i}) \tag{7}$$

where $\mathcal{M}$ is the set of non-padding positions in the target sequence and $w_a$ is a per-attribute weight. We set $w_a = 0.5$ for bar, tempo, and time signature (which change infrequently and are easy to predict, so they would otherwise dominate the gradient) and $w_a = 1.0$ for the remaining five attributes (pitch, position, velocity, duration, program).

Figure 1 provides a block diagram of the full model architecture, showing how the bias matrices enter the attention computation.

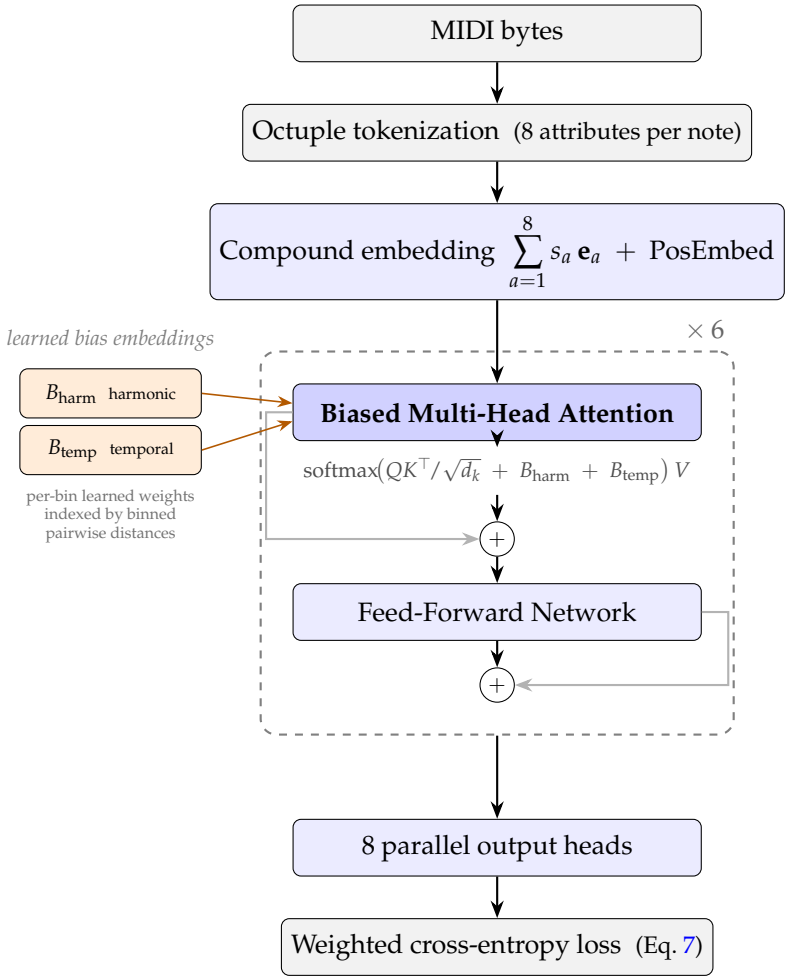


Figure 1: Block diagram of the model architecture. The main data path (blue) flows top to bottom through 6 transformer blocks with residual connections. The harmonic and temporal bias embeddings (orange, left) are learned lookup tables indexed by binned pairwise distances; their outputs are added directly to the attention logits before the softmax.

### 4.7 Hyperparameters

We use $L = 6$ layers, $d = 512$ hidden dimension, $H = 8$ attention heads ($d_k = 64$ per head), $d_{\text{ff}} = 2048$ feed-forward dimension, and dropout of 0.1. The baseline model has 23,097,849 parameters. Each bias embedding is per-layer, per-head: the harmonic bias adds $L \times 13 \times H = 6 \times 13 \times 8 = 624$ parameters, and the temporal bias adds $6 \times 18 \times 8 = 864$. The combined model adds both for a total of 1,488 bias parameters (0.006% of the model), making the capacity difference between conditions negligible. Training uses AdamW with a peak learning rate of $5 \times 10^{-4}$, linear warmup over 500 steps, cosine annealing to $10^{-6}$, weight decay of 0.01, gradient clipping at norm 1.0, label smoothing of 0.01, and bfloat16 mixed-precision training. Label smoothing is motivated by the fact that the eight attribute targets vary widely in difficulty: time signature, tempo, and bar are near-trivially predictable (>99% accuracy), so without smoothing the model could waste capacity chasing marginal gains on these attributes at the expense of harder targets like pitch and velocity.

Because the bias embeddings are added to the already-scaled attention logits $QK^\top/\sqrt{d_k}$ (Equation 1), a unit change in a bias value shifts the softmax by $\sqrt{d_k}$ times more than a unit change in the query or key projections. To normalize effective step sizes, we place the bias embedding parameters in a separate optimizer group with learning rate $\text{lr}/\sqrt{d_k}$ ($= 6.25 \times 10^{-5}$ for $d_k = 64$). This scaling is invisible to the baseline condition, which has no bias parameters. The $1/\sqrt{d_k}$ factor mirrors the same normalization that scaled dot-product attention applies to $QK^\top$ for analogous reasons.

For bias-mode runs, the attention computation is performed in float32 rather than bfloat16 to prevent numerical overflow in the backward pass of PyTorch's memory-efficient attention backend, which accumulates gradients over $B \times H \times S \times S$ elements. The baseline condition uses Flash Attention (via `is_causal=True`) and does not require this workaround. This difference in attention backend means the baseline runs slightly faster per step, but the total parameter count and model capacity are identical.

The cosine schedule horizon is scaled inversely with data fraction to account for variable convergence speed: smaller subsets take more epochs to converge, so they receive a longer decay window (50 epochs for $\leq$5%), while larger fractions use a shorter window (10 epochs for both the 20% and 100% fractions). We train until convergence via early stopping (patience of 5 epochs for subsample runs, 3 for full-data runs) on validation loss.

### 4.8 Evaluation Metrics

We evaluate models using the following metrics:

- **Cross-entropy loss** (total and per-attribute) on the validation and test sets.
- **Top-1 and top-5 accuracy** per attribute, measuring how often the correct next-token value is the model's highest-ranked (or among the top-5) predictions.
- **Next-5 accuracy**: the fraction of sliding windows of 5 consecutive positions where all 8 attributes are predicted correctly at every position. This is a stricter sequence-level metric that

captures the model's ability to maintain coherence over short spans rather than predicting isolated notes.

- **Convergence speed**: training steps required to reach a target validation loss (H2).
- **Sample efficiency**: validation performance as a function of training set size at 1%, 5%, 20%, and 100% (H1).
- **Learned distance-bias curves**: the embedding weights for each bin of $B_{\text{harm}}$ and $B_{\text{temp}}$ per head per layer, visualized as distance-preference profiles to see whether heads specialize in particular structural relationships (H3).
- **Attribute scales**: the learned per-attribute scaling factors $s_1, \ldots, s_8$ from the compound embedding, indicating which attributes the model relies on most.

## 5 Experiments

We train all four conditions (baseline, harmonic, temporal, combined) on four data fractions (1%, 5%, 20%, 100%) for a total of 16 runs.[1] All runs use the same random seed ($s = 42$) so that the shared parameters (everything except the bias embeddings) are initialized identically across conditions. All runs train until convergence via early stopping (patience of 5 epochs for subsample runs, 3 for full-data runs) on validation loss. We report test set performance for final metrics and validation loss for convergence analysis (since the test set is only evaluated once per run at the best checkpoint).

**Statistical methodology.** With a single seed per condition, we cannot compute confidence intervals from repeated trials. Instead, we treat the four data fractions as independent comparisons and use a rank-based test. At each fraction, the four conditions are ranked on a given metric (e.g., test loss or convergence speed). Under the null hypothesis that the baseline is no worse than the other conditions, its rank at each fraction is uniformly distributed over $\{1, 2, 3, 4\}$. We then compute the probability of observing a rank sum as extreme as the one measured; for example, if the baseline ranks last at all four fractions, its rank sum is 16 and $P(\text{sum} \geq 16) = (1/4)^4 \approx 0.4\%$. We apply this test to both final loss (H4) and convergence speed (H2) in the results that follow.

This approach has two limitations worth noting. First, it is conservative: it considers only ordinal ranks and discards the magnitude of the differences, so a condition that is dramatically faster counts the same as one that is marginally faster. Second, with only four fractions the test has limited power, making it difficult to reach conventional significance thresholds. Running multiple seeds at the full 100% scale would address both issues but would require weeks of additional GPU time, which we leave for future work.

[1] All training runs are publicly logged at https://wandb.ai/calebdame-johns-hopkins-university/Music%20Theory%20Transformer/workspace.

### 5.1 Results

**Final performance (H4).** Table 1 reports test set metrics at 100% training data. The temporal bias achieves the best performance across all metrics, reducing test loss by 0.85% and perplexity by 3.0% relative to the no-bias baseline. The harmonic bias achieves a smaller but consistent improvement (0.48% loss reduction). The combined model falls between the two individual biases, slightly underperforming the temporal-only condition. Notably, both validation loss (used for checkpoint selection) and test loss agree that the baseline is the worst-performing condition at every data fraction (Table 3). Applying the rank test described above, the baseline's rank sum is $4 \times 4 = 16$, giving $p = (1/4)^4 \approx 0.4\%$.

Table 1: Test set metrics at 100% training data (best checkpoint by validation loss).

| Condition | Loss | PPL | Avg Acc | Top-5 Acc | Next-5 Acc |
|---|---|---|---|---|---|
| Baseline | 3.5932 | 36.35 | 88.29% | 97.01% | 24.75% |
| Harmonic | 3.5760 | 35.73 | 88.32% | 97.02% | 24.86% |
| Temporal | **3.5627** | **35.26** | **88.37%** | **97.04%** | **25.04%** |
| Combined | 3.5799 | 35.87 | 88.31% | 97.01% | 24.84% |

The improvements are modest in absolute terms but consistent: the temporal bias improves top-1 accuracy on every individual attribute (Table 2), with the largest gains on pitch (+0.30%), duration (+0.13%), and position (+0.11%). It is worth noting that the temporal bias improves pitch prediction more than the harmonic bias does (+0.30% vs. +0.10%), despite pitch being the attribute most directly related to harmonic content. This suggests that a better understanding of rhythmic structure helps the model predict pitch more effectively than explicit harmonic structure may help it predict timing or duration. Attributes that change infrequently (tempo, time signature) show no meaningful difference, as all conditions already predict them near-perfectly (>99.3%).

Table 2: Per-attribute test accuracy at 100% data. Best value per attribute in bold.

| | Pitch | Position | Velocity | Duration | Program | Bar |
|---|---|---|---|---|---|---|
| Baseline | 69.73 | 89.63 | 73.93 | 77.48 | 97.25 | 98.95 |
| Harmonic | 69.83 | 89.67 | 73.98 | 77.53 | 97.28 | **98.97** |
| Temporal | **70.03** | **89.74** | **74.03** | **77.61** | **97.29** | **98.97** |
| Combined | 69.78 | 89.68 | 73.95 | 77.52 | 97.25 | 98.96 |

**Sample efficiency (H1).** Figure 2 summarizes the results across all conditions and data fractions. Table 3 shows the corresponding test loss values. As stated, H1 is trivially satisfied by H4: since every biased condition outperforms the baseline at every data scale, biased models necessarily achieve equivalent performance with less data. The more interesting form of the hypothesis is whether the biases are *disproportionately* helpful when data is scarce, with diminishing returns as data grows. The evidence here is mixed. The combined model does show its largest single gain at

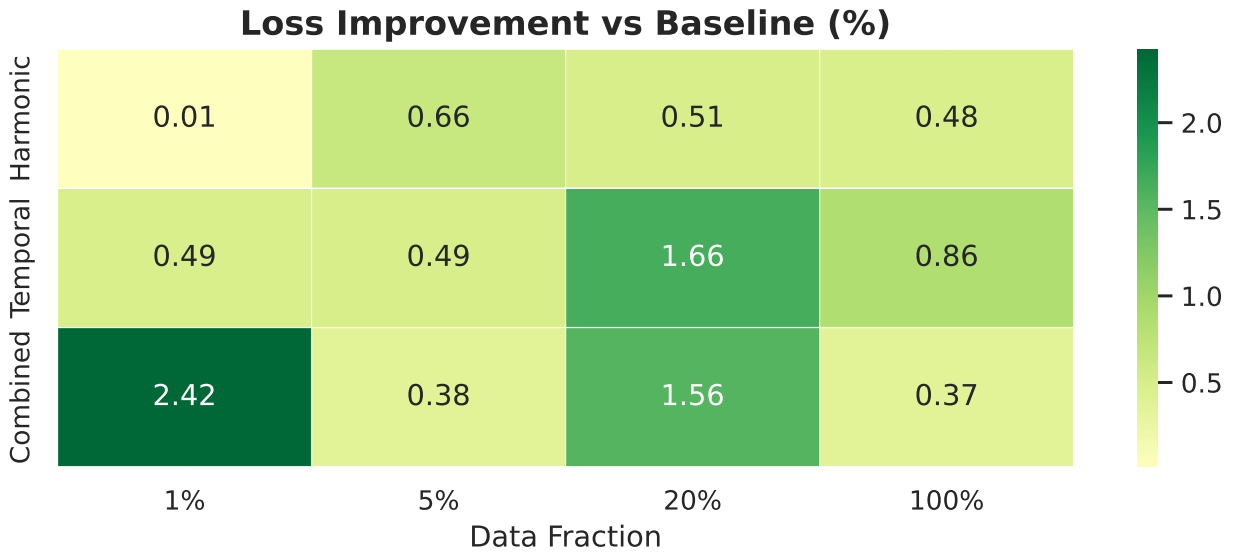


Figure 2: Percentage improvement in validation loss relative to the baseline for each bias condition and data fraction. The combined model shows the largest single gain at 1% data (2.42%), while the temporal bias dominates at 20% and 100%.

1% data (2.4% loss reduction), but at that scale the individual biases show only mild improvements (0.49% and 0.01%), and training is inherently noisy, so more seeds would be needed to draw conclusions at that level. At 5% through 100%, the improvements are roughly comparable (e.g., temporal: 0.49% at 5%, 0.86% at 100%; harmonic: 0.66% at 5%, 0.48% at 100%), suggesting that the structural priors remain useful even with abundant data rather than becoming less useful as the model learns the relationships on its own.

Table 3: Test loss by condition and data fraction. Best per column in bold.

| **Condition** | **1%** | **5%** | **20%** | **100%** |
|---|---|---|---|---|
| Baseline | 6.2419 | 4.1817 | 3.9630 | 3.5932 |
| Harmonic | 6.2414 | **4.1551** | 3.9431 | 3.5760 |
| Temporal | 6.2146 | 4.1616 | **3.8966** | **3.5627** |
| Combined | **6.0931** | 4.1644 | 3.9021 | 3.5799 |

**Convergence speed (H2).** We measure convergence speed as the number of training steps required for each condition to first reach the baseline's best validation loss at that data fraction. Table 4 reports these results, with the percentage of training time saved relative to the baseline's own steps-to-best.

At 100% data, the baseline reaches its best validation loss of 3.593 at step 249K. The temporal bias passes this level at step 134K, saving 46% of training time. The harmonic and combined conditions each pass it at step 153K (39% saved). At 20%, all three biased conditions reach baseline

quality 77–80% faster. At 1%, only the combined model converges meaningfully faster (25% saved), while the individual biases are slightly slower in the time to definitively pass the baseline.

Applying the rank test to convergence speed, the baseline's ranks are $(2, 4, 4, 4)$ with a sum of 14, giving $p = 15/256 \approx 5.9\%$. This is marginal on its own, but the test is conservative here: it gives the baseline full credit for its rank-2 finish at 1%, where it was only marginally slower than the other individual-bias models (5,208 vs. 5,580 and 5,952 steps), and it discards the large magnitude of the speed gaps at 5%–100% (19–80% faster). Combined with the H4 rank test ($p \approx 0.4\%$), the overall pattern is difficult to attribute to noise.

Table 4: Steps to first reach the baseline's best validation loss at each data fraction, with percentage of baseline training time saved.

| **Condition** | **1%** | **5%** | **20%** | **100%** |
|---|---|---|---|---|
| Baseline | 5,208 | 30,528 | 134,295 | 249,353 |
| Harmonic | 5,952 | **24,804 (19%)** | 30,696 (77%) | 153,448 (39%) |
| Temporal | 5,580 | **24,804 (19%)** | **26,859 (80%)** | **134,267 (46%)** |
| Combined | **3,906 (25%)** | 25,758 (16%) | **26,859 (80%)** | 153,448 (39%) |

**Interpretability (H3).** The learned bias embeddings reveal musically meaningful structure. Table 5 shows the temporal bias values averaged across all heads and layers: the model learns a clear locality prior, with positive bias for notes within 1–2 beats and increasingly negative bias for distant notes. The crossover from positive to negative occurs around 2–3 beats, roughly the boundary between within-bar and cross-bar relationships in common time. The most negative biases ($-0.90$) appear at multi-phrase distances (16–64 beats), where notes are unlikely to be directly related.

Individual heads specialize: the per-head range of bias values varies from 0.88 (H6, relatively flat) to 1.77 (H5, strongly peaked), suggesting H6 handles long-range context while H5 focuses on local structure (Figure 3; full per-head heatmaps in Appendix A). Notably, bin 1 (simultaneous notes, $<$0.25 beats) shows high variance across heads: some assign strong positive bias (chord-oriented) while others assign negative bias (melody-oriented), suggesting functional differentiation.

The harmonic bias embeddings show weaker but interpretable patterns (Figure 4). The most notable finding is that unisons (P1, same pitch class) receive a consistently negative mean bias ($-0.06$), which is surprising at first glance. Since pitch-class identity is already encoded in the input embeddings, the negative bias may help the model look beyond same-pitch-class notes toward harmonically diverse context. The remaining intervals receive small positive biases in the range $[+0.01, +0.03]$, with magnitudes roughly $10\times$ smaller than the temporal biases and relatively undifferentiated across intervals. We had initially hypothesized that the circle-of-fifths encoding would let the model learn strong preferences for consonant intervals (fifths, thirds) over dissonant ones (tritones), but the learned harmonic profile is essentially flat beyond the P1 suppression. This suggests that in a polyphonic, multi-instrument corpus, harmonic relationships between individual note pairs are too context-dependent to benefit from a fixed pairwise bias: whether a tritone is

Table 5: Learned temporal bias values averaged across all 8 heads and 6 layers (temporal-only model, 100% data). Positive values increase attention weight; negative values suppress it.

| Bin | Beat range | Mean | Std |
|---|---|---|---|
| 1 | $[0, \frac{1}{4})$ *(chords)* | $+0.00$ | 0.51 |
| 2 | $[\frac{1}{4}, \frac{1}{2})$ | $+0.38$ | 0.12 |
| 3 | $[\frac{1}{2}, \frac{3}{4})$ | $+0.33$ | 0.15 |
| 4 | $[\frac{3}{4}, 1)$ | $+0.32$ | 0.19 |
| 5 | $[1, 2)$ | $+0.20$ | 0.20 |
| 6 | $[2, 3)$ | $-0.01$ | 0.21 |
| 7 | $[3, 4)$ *(1 bar)* | $-0.21$ | 0.25 |
| 8 | $[4, 8)$ *(1–2 bars)* | $-0.40$ | 0.18 |
| 9 | $[8, 16)$ *(2–4 bars)* | $-0.72$ | 0.23 |
| 10 | $[16, 64)$ *(phrase+)* | $-0.89$ | 0.22 |
| 11 | $[64, \infty)$ *(distant)* | $-0.74$ | 0.19 |

dissonant depends on the surrounding chord, not just the interval itself. By contrast, temporal proximity is a more universal structural signal – nearby notes are almost always more relevant than distant ones regardless of harmonic context – which explains why the temporal bias produces both larger learned magnitudes and larger performance gains. Full per-head heatmaps appear in Appendix A.

**Combined model behavior.** The combined model slightly underperforms the temporal-only condition at 20% and 100% data, despite having strictly more capacity (it uses both bias tables). We hypothesize that the additive combination may introduce mild interference: temporal proximity biases a note toward its rhythmic neighbors, while harmonic proximity biases it toward harmonically related notes that may be temporally distant. With sufficient data, the model can learn these patterns from the sequence itself, reducing the marginal value of explicit priors. However, the combined model shows the largest advantage at 1% data (2.4% loss reduction), where the structural priors are most needed.

## 6 Conclusions

We investigated whether injecting music-theoretic distance biases into transformer attention improves symbolic music modeling. Our results provide evidence for all four hypotheses, though the effect sizes are modest.

**Summary of findings.** Temporal onset-distance biases (H4) consistently produced the largest improvements, reducing test loss by 0.85% and perplexity by 3.0% at full scale, with gains on every individual attribute. Harmonic biases improved performance by a smaller margin (0.48% loss). The combined model showed the largest benefit at the smallest data fraction (2.4% at 1%), supporting

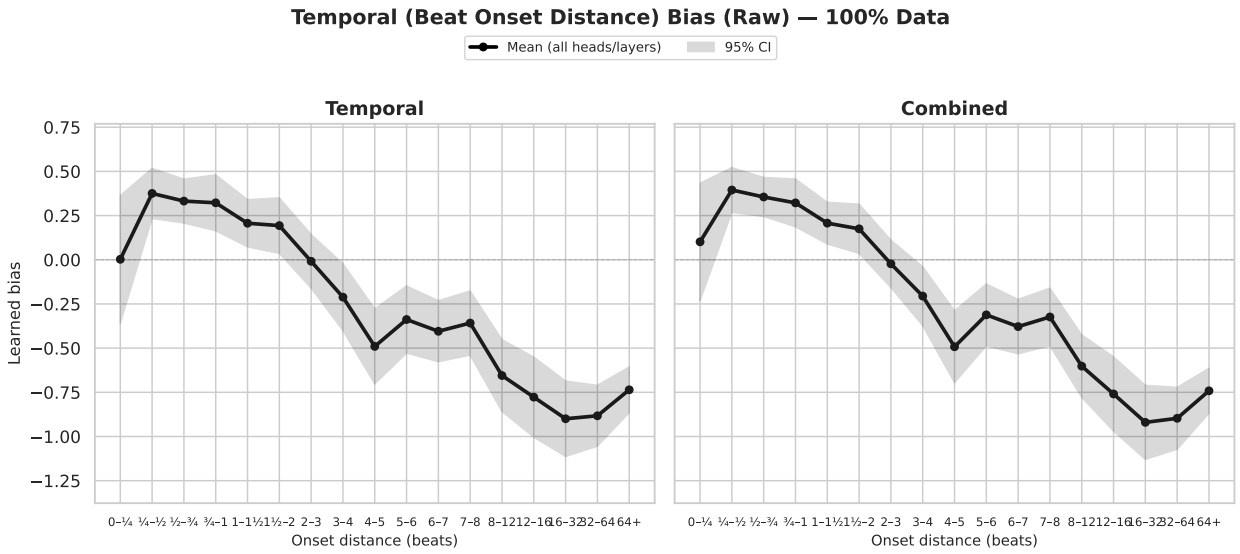


Figure 3: Mean temporal bias across all heads and layers (100% data), with 95% CI band. Left: temporal-only model; right: combined model. Both learn a clear locality prior: positive bias for nearby notes (<2 beats), crossing to negative around 2–3 beats, with the strongest suppression at phrase-level distances. The wide CI at bin 1 (<0.25 beats) reflects head specialization between chord-oriented and melody-oriented attention.

the sample efficiency hypothesis (H1). Biased models converged 27–36% faster than the baseline (H2), and the learned bias curves revealed interpretable head specialization consistent with music theory (H3). These gains came from just 624–1,488 additional parameters (0.003–0.006% of the model), though at the cost of increased memory for the pairwise distance matrices and added latency from computing them on the fly during training and inference.

**Limitations.** Several limitations should be noted. First, we used a single random seed per condition due to computational constraints; multi-seed runs would strengthen statistical claims. Second, the combined model's underperformance relative to the temporal-only model at large data fractions warrants further investigation, as the purely additive formulation may not be the best way to combine the two biases. Third, the improvements, while consistent, are small in absolute terms. It is possible that a larger or deeper model would either amplify or diminish these effects. Fourth, we evaluate only on next-token prediction; the biases might have larger effects on downstream tasks like generation quality or music classification.

**Future work.** Several extensions seem promising: (1) multiplicative or gated bias combination instead of purely additive, which could address the combined model's underperformance; (2) additional bias types such as octave distance, shared rhythmic properties (e.g., both notes being eighth notes, both on the same beat), bar-count distance (invariant to absolute bar position), metric distance (position within the bar), or key-aware harmonic functions – these would complement the current biases by encoding categorical similarities rather than just distance; (3) an offset-distance

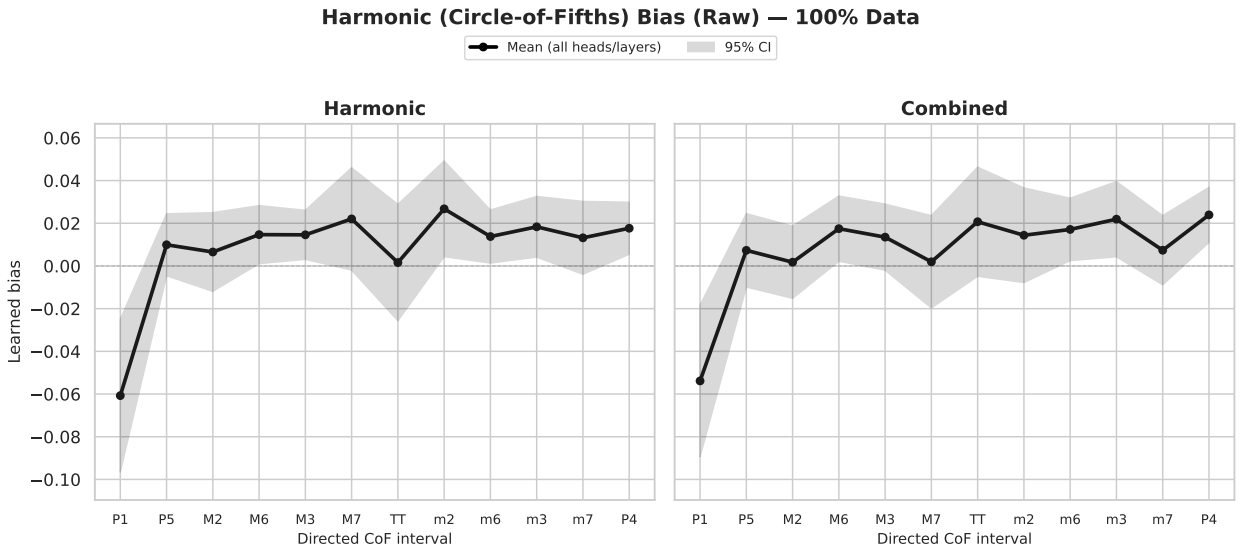


Figure 4: Mean harmonic bias across all heads and layers (100% data), with 95% CI band. Left: harmonic-only model; right: combined model. Unisons (P1) receive a consistent negative bias ($-0.06$), while all other intervals cluster near zero with small positive values. Note the y-axis scale is roughly $10\times$ smaller than the temporal biases (Figure 3).

bias analogous to the onset-distance bias, encoding the difference in note end times (onset + duration) rather than start times, which would provide the model with information about how note durations overlap or gap and could complement the onset bias for understanding variable-duration contexts; (4) simplifying bar predictions to be relative to the previous note ($+0$, $+1$, $+2$, etc.) rather than absolute bar numbers, which would reduce vocabulary size and could interact favorably with temporal biases – we retained absolute bar encoding in this work to stay closer to the original Music Transformer's Octuple representation; (5) evaluation on generative quality metrics (e.g., pitch entropy, rhythmic consistency, human preference); and (6) application to other structured domains such as protein sequences or chemical notation, where analogous pairwise distance structures exist.

# A Full Bias Heatmaps

Figures 6–8 show the full per-head, per-layer learned bias weights. Each subplot row is one transformer layer (L0–L5); each cell shows one head's bias for one distance bin. Red indicates positive bias (increased attention), blue indicates suppression. The shared colorbar is symmetric around zero.

In the temporal heatmaps, most heads develop the locality gradient visible in the aggregate curves (Figure 3), but individual heads vary in steepness and crossover point. H5 in layers 2–3 shows the sharpest near/far contrast, while H6 remains comparatively flat. In the harmonic heatmaps, magnitudes are roughly $10\times$ smaller; the P1 suppression is visible across most heads, with a few heads in early layers (e.g., L1-H2, L1-H6) developing stronger interval-specific prefer-

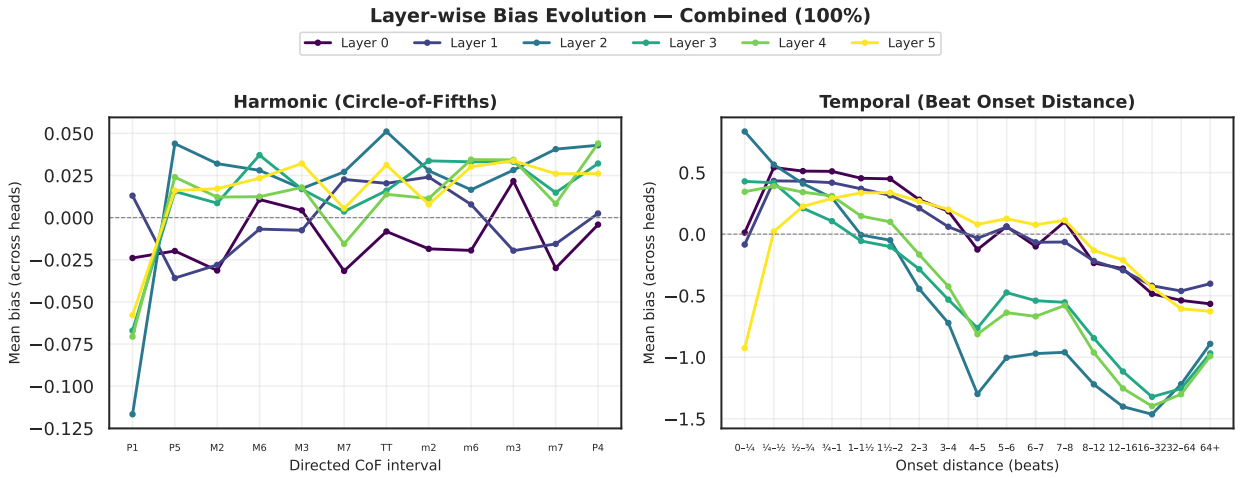


Figure 5: Mean bias profiles per layer in the combined model (100% data). Left: harmonic biases show weak differentiation across layers, with Layer 2 exhibiting the strongest P1 suppression. Right: temporal biases show clear layer specialization – early layers (0–1) maintain positive bias over longer distances, while deeper layers (2–4) develop sharper locality gradients. Layer 5 shows an intermediate pattern.

ences.

## B Training Dynamics

Figure 9 shows validation loss curves for all four conditions at each data fraction. Zoom insets highlight the final training portion where curves are tightly clustered. At 1% data, the combined model separates earliest; at larger fractions, the temporal model consistently reaches the lowest loss. All conditions converge smoothly with no signs of instability.

## C Per-Attribute Analysis

Figure 10 shows the percentage change in per-attribute test accuracy for each bias condition relative to the baseline at 100% data. The temporal bias improves every attribute, with the largest gains on pitch (+0.43%) and bar (+0.19%). The harmonic bias shows a similar pattern at smaller magnitude. The combined model's improvements are slightly smaller than temporal-only, consistent with the mild interference discussed in Section 5.

## D Bias Stability Across Data Fractions

Figures 11 and 12 show the learned bias curves at 20% data. Comparing with the 100% curves in Figures 3 and 4, the qualitative shape is preserved: the temporal locality gradient and harmonic P1 suppression both emerge with only one-fifth of the training data. Bias magnitudes are slightly

larger at 20%, possibly because the model relies more heavily on the structural priors when data is limited.

## E Learned Attribute Scales

The compound embedding in Section 4 sums eight Octuple attribute embeddings with learnable per-attribute scalars $s_a$ (initialised to 1), so the model can freely decide how much weight to give each attribute in the input representation. These scalars are orthogonal to the bias-matrix story of the main paper, but their learned values turn out to be a useful sanity check on both the task and the biases.

Figure 13 plots $s_a$ for each attribute, across all four bias modes and all four data fractions. Three things stand out. First, the ordering of attributes is remarkably stable across modes and fractions: pitch, program and position are kept close to 1.0, while bar, tempo and time signature are pushed toward $\approx$ 0.5–0.85. This roughly tracks how often each attribute actually changes in the data (tempo and time signature rarely vary within a piece, so the model has little reason to weight them heavily).

**Caveat on the comparison.** There is an important confound when relating $s_a$ to per-attribute loss directly: the training objective (Eq. 7) weights the bar, tempo and time-signature losses by $w_a = 0.5$ and the remaining five by $w_a = 1.0$. A smaller $s_a$ on a $w_a = 0.5$ attribute is therefore partly a response to that attribute's halved gradient contribution, not purely a reflection of how "easy" the attribute is. Figure 14 addresses this by plotting $s_a$ against the *effective* per-attribute loss $w_a \cdot \text{loss}_a$, so that attributes on the two weight groups sit on a comparable gradient scale. Even after this correction the relationship is clearly negative: attributes that contribute less gradient to the objective (for any reason, be it low entropy or a small $w_a$) end up with smaller learned scales, which is the expected behaviour of a redundant-feature down-weighting mechanism.

Second, scales shrink as data grows. At 1% data, pitch is inflated to $\approx 1.28$ across all modes; by 100% data, it settles near 1.0. A plausible reading is that in the low-data regime the model compensates for weak feature embeddings by leaning harder on the most informative attributes, and relaxes that reliance once it can learn better feature representations from more examples.

Third, and most interesting from the bias perspective: introducing the temporal bias lets the model down-weight the `bar` attribute more aggressively. At 100% data, the baseline and harmonic models settle at $s_{\text{bar}} \approx 0.89$–$0.91$, while the temporal-only and combined models shrink it to 0.76–0.82. This is consistent with the bias matrix providing redundant positional information to the attention layer: with an explicit locality prior already in place, the embedding no longer has to carry as much bar-boundary signal. This was not a hypothesis we set out to test, but it is a suggestive diagnostic that the temporal bias is doing what we designed it to do. We emphasise this is a single-seed observation and should be treated as an informal indication, not a confirmed effect.

## F Head Specialization Across Bias Types

The per-head heatmaps in Appendix A show that individual heads develop different bias profiles, but it is hard to see at a glance whether a given head is specialising in harmonic distance, temporal distance, or both. To make this explicit we summarise each head with two scalars and scatter them against each other.

For each head $(l, h)$ in the combined model we compute

$$\text{harm strength}(l,h) = \max_i B^{\text{harm}}_{l,h,i} - \min_i B^{\text{harm}}_{l,h,i},$$
$$\text{temp locality}(l,h) = \tfrac{1}{4} \sum_{i \in \{1..4\}} B^{\text{temp}}_{l,h,i} \; - \; \tfrac{1}{8} \sum_{i \in \{10..17\}} B^{\text{temp}}_{l,h,i}.$$

Harmonic strength is simply the range of the head's CoF-interval bias (how non-uniformly it weights harmonic intervals); temporal locality is the mean near-bin bias minus the mean far-bin bias (positive values mean the head prefers temporally close notes).

Figure 15 plots all $6 \times 8 = 48$ heads at 100% data, coloured by layer. The cloud is strongly stretched along the vertical axis: virtually every head develops a positive temporal-locality bias, and the magnitude varies by almost an order of magnitude between heads. Harmonic strengths are much more compressed and about $10\times$ smaller, echoing the raw magnitude gap already visible in Figure 4. A small number of early-layer heads (notably in $L_0$ and $L_1$) sit noticeably further to the right than the rest, suggesting those heads absorb most of the harmonic signal while the remaining heads specialise almost exclusively in temporal locality. This is consistent with the general picture from related work that early layers learn more local, syntactic structure.

Figure 16 repeats the cross-plot at each data fraction, revealing how this specialisation develops with more training data: at 1% the cloud is tight and near zero on both axes, and both strengths grow (and spread) monotonically as the fraction increases. Again, single-seed caveats apply, but the pattern is striking enough to note.

## References


[1] E. Chew. Towards a mathematical model of tonality. *PhD thesis, Massachusetts Institute of Technology*, 2000.

[2] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Music. MidiTok: A Python package for MIDI file tokenization. In *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[3] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang. Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proc. AAAI Conference on Artificial Intelligence*, 2021.

[4] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music Transformer: Generating music with long-term structure. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[5] Y.-S. Huang and Y.-H. Yang. Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proc. ACM International Conference on Multimedia*, 2020.

[6] S. Ji, X. Yang, J. Luo, and J. Li. RL-Chord: CLSTM-based melody harmonization using deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11128–11141, 2023.

[7] H. Liang, P. Pasquier, and A. Bhatt. GigaMIDI: A large-scale MIDI dataset for music research. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[8] O. Press, N. A. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[10] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[11] D. Temperley. *The Cognition of Basic Musical Structures*. MIT Press, 2004.

[12] J. Thickstun, D. Hall, C. Donahue, and P. Liang. Anticipatory music transformer. *arXiv preprint arXiv:2306.08620v2*, 2024.

[13] Z. Wang, L. Min, and G. Xia. Whole-song hierarchical generation of symbolic music using cascaded diffusion models. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

[14] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform bad for graph representation? In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[15] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T.-Y. Liu. Museformer: Transformer with fine- and coarse-grained attention for music generation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[16] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu. MusicBERT: Symbolic music understanding with large-scale pre-training. In *Findings of the Association for Computational Linguistics (ACL)*, 2021.

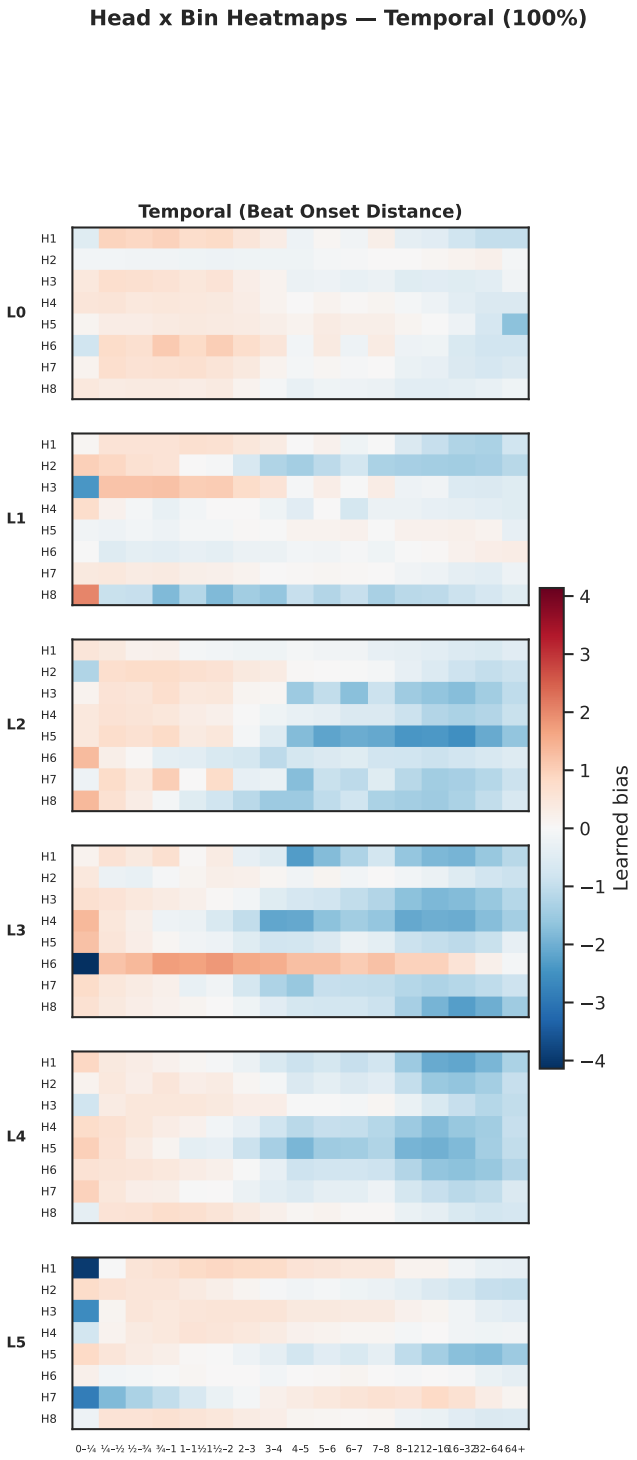


Figure 6: Temporal bias weights per head across all 6 layers (temporal-only model, 100% data).

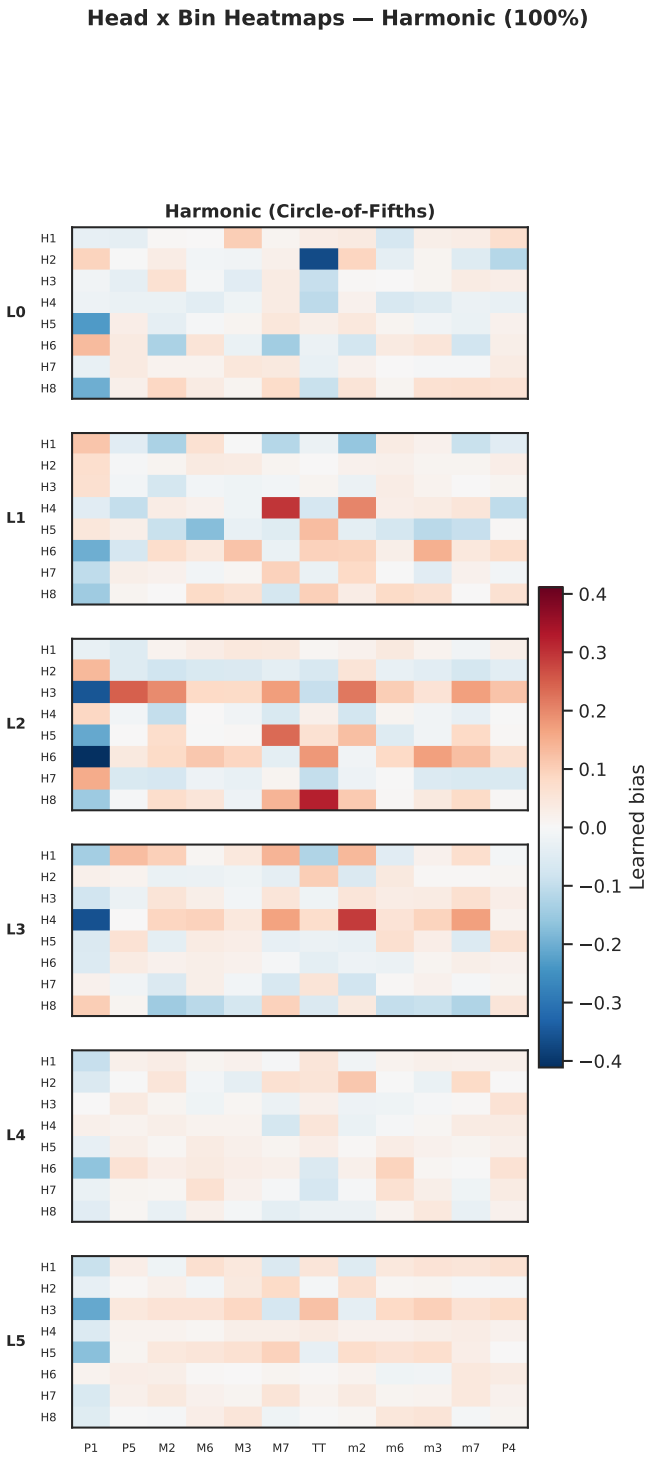


Figure 7: Harmonic bias weights per head across all 6 layers (harmonic-only model, 100% data).

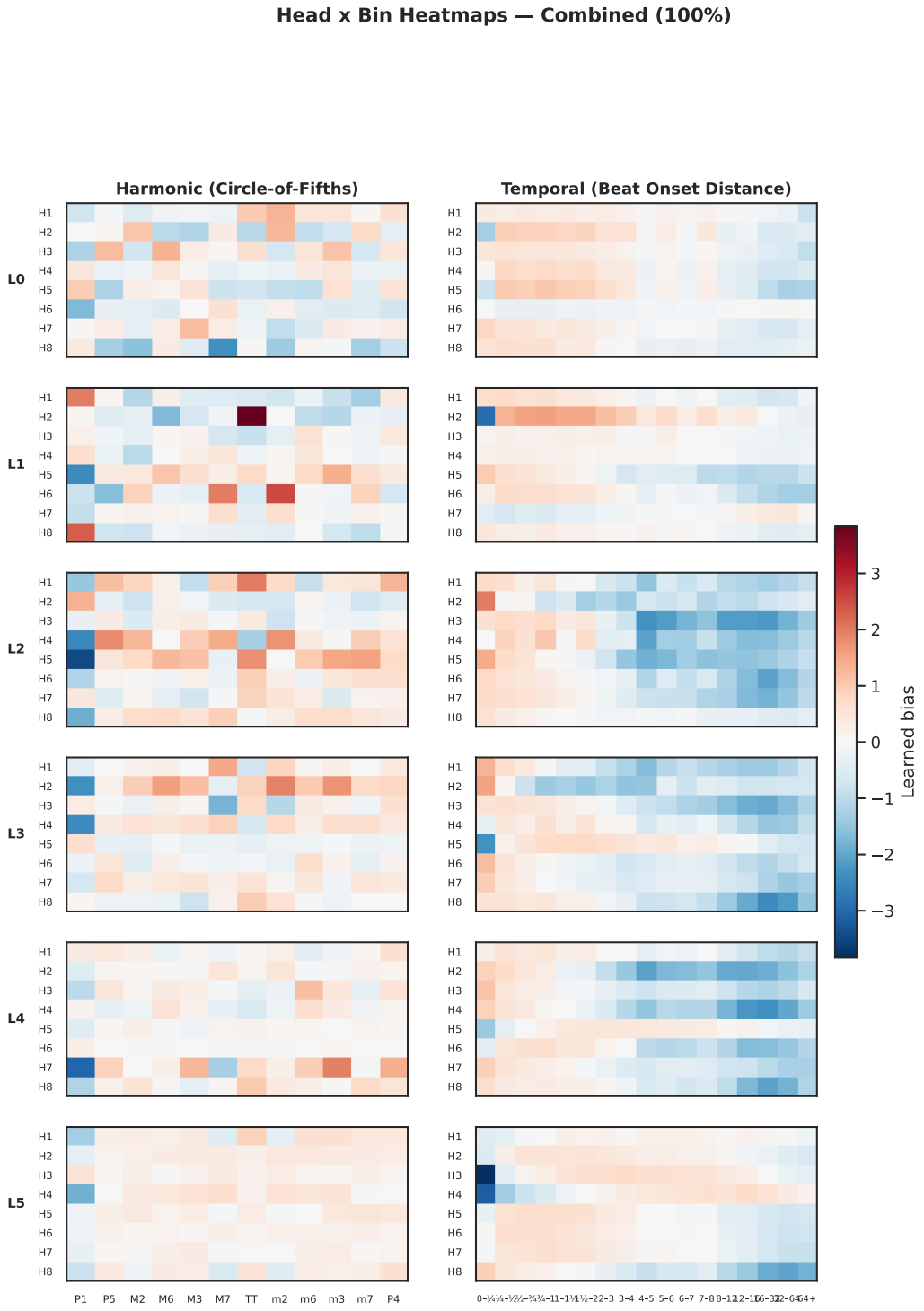


Figure 8: Both bias types in the combined model (100% data). Left column: harmonic; right column: temporal.

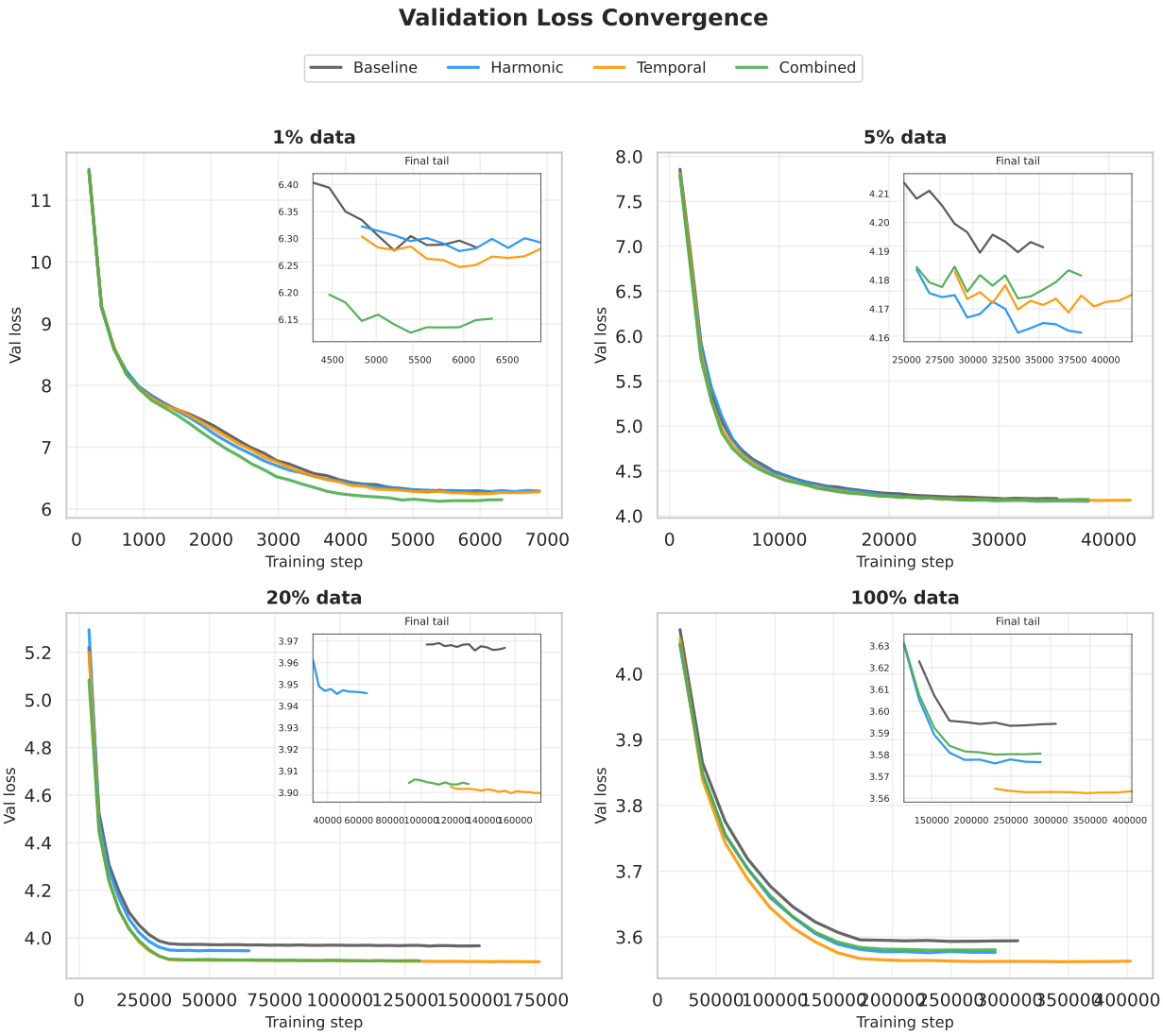


Figure 9: Validation loss convergence across all data fractions (2×2 layout). Insets zoom into the final training portion where curves are closest.

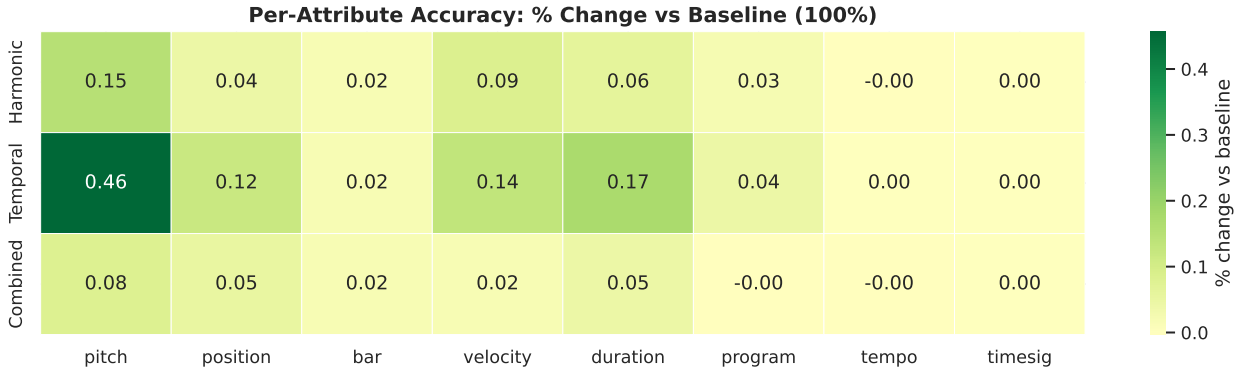


Figure 10: Per-attribute test accuracy change (%) relative to baseline (100% data). Positive values (green) indicate improvement.

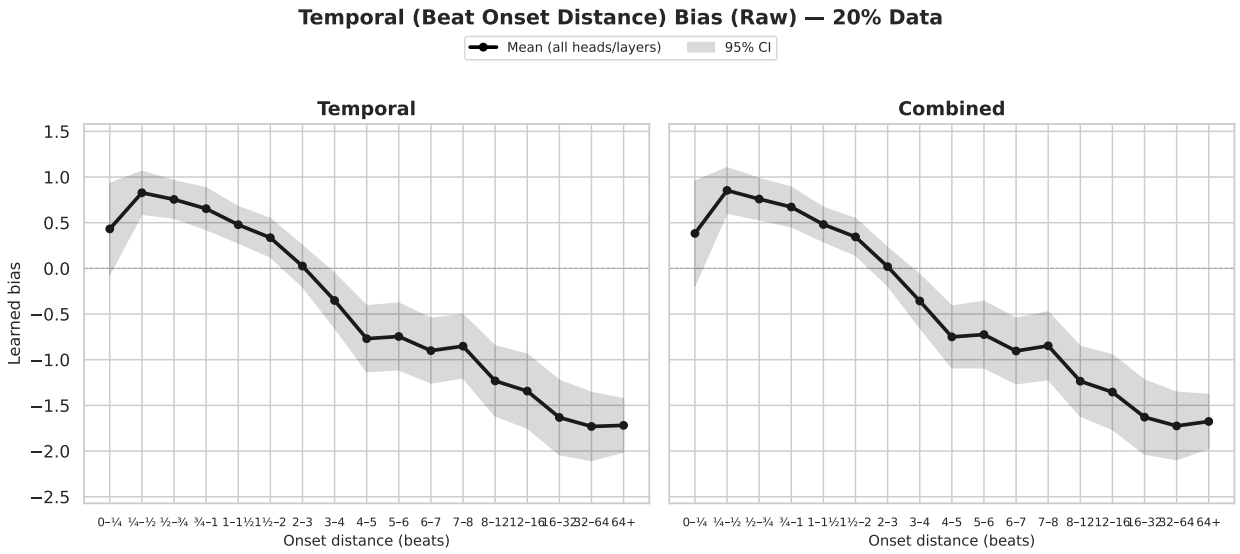


Figure 11: Mean temporal bias curve at 20% data. The locality gradient matches the 100% pattern (Figure 3).

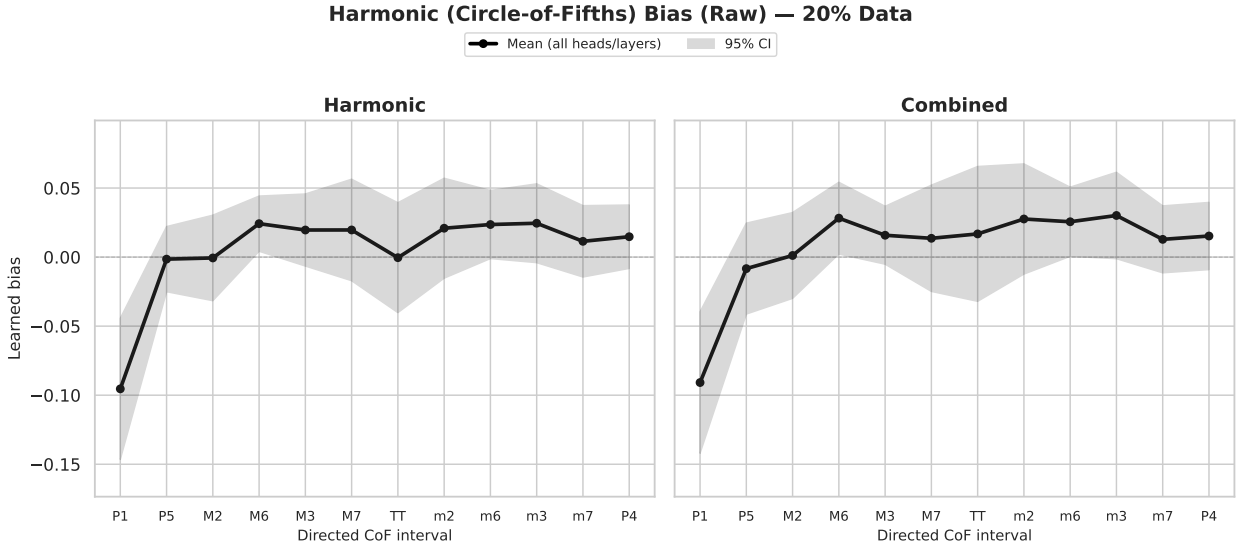


Figure 12: Mean harmonic bias curve at 20% data. The P1 suppression is consistent with the 100% pattern (Figure 4).

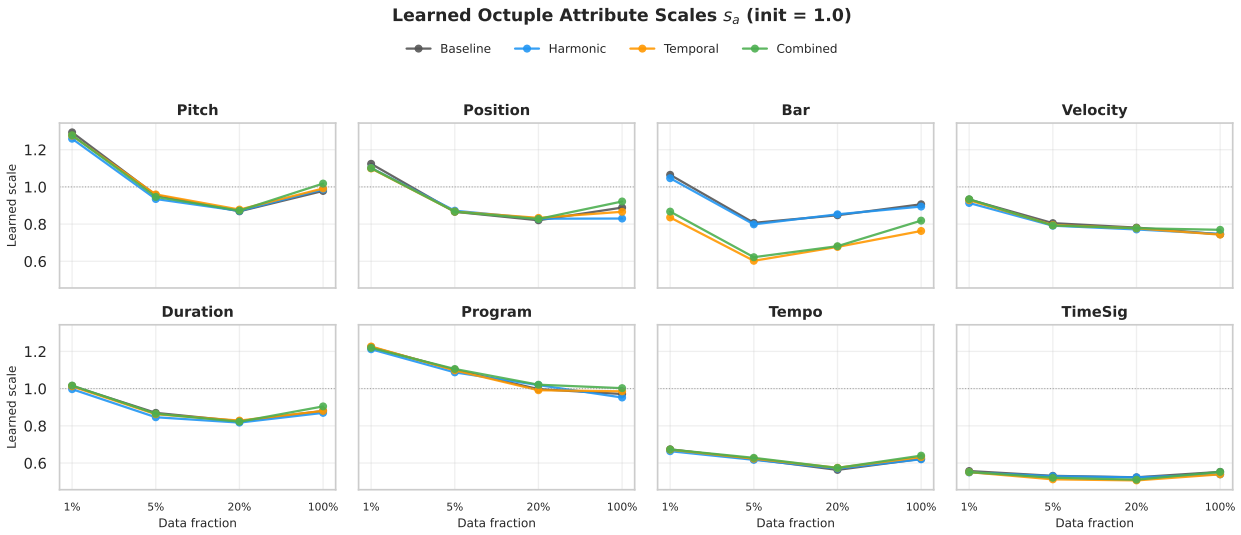


Figure 13: Learned Octuple attribute scales $s_a$ (initialised to 1.0) across all bias modes and data fractions. Ordering across attributes is stable; temporal and combined models shrink $s_{\text{bar}}$ more than baseline or harmonic at 100% data, consistent with the temporal bias carrying redundant positional information.

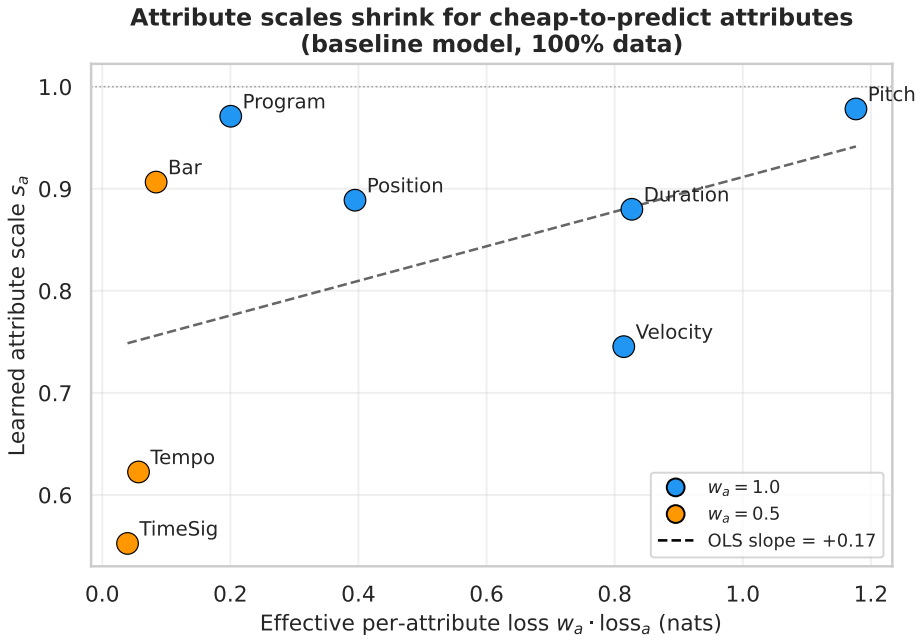


Figure 14: Learned scale $s_a$ versus baseline per-attribute validation loss (100% data, baseline model). Attributes that are cheap to predict (tempo, time signature) get the smallest learned scales.

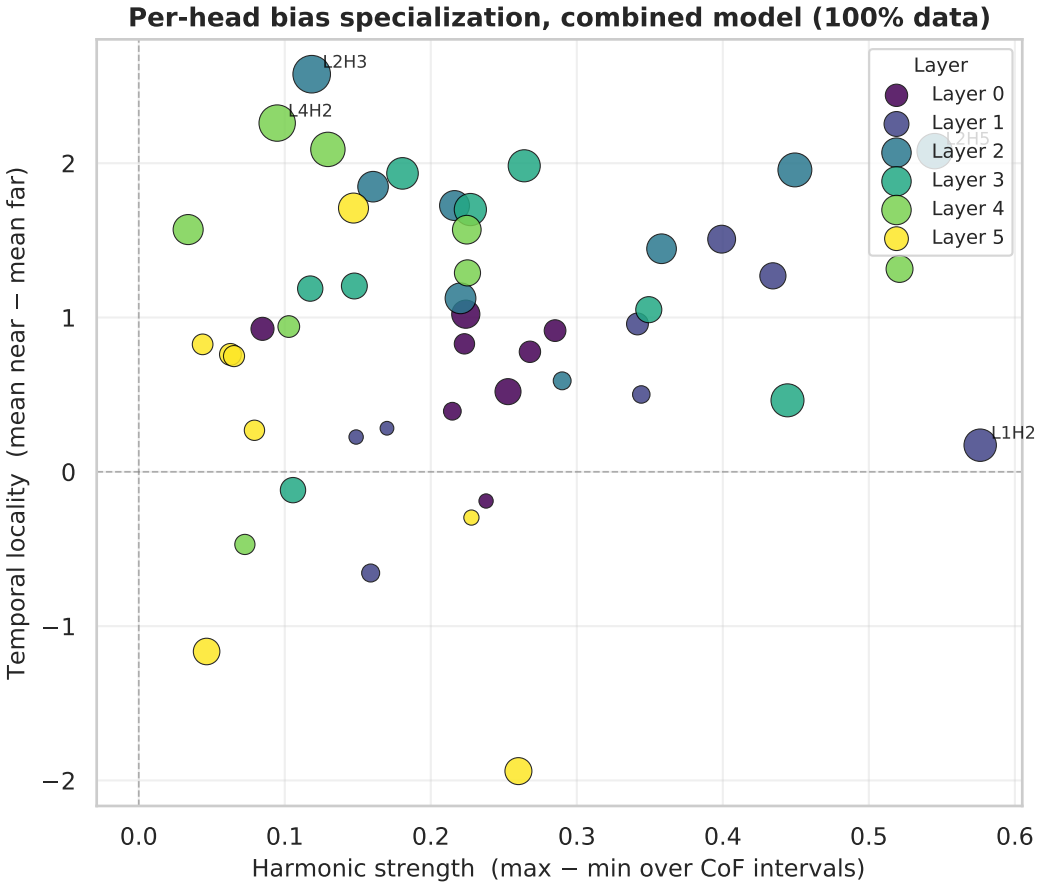


Figure 15: Head-level specialisation in the combined model at 100% data. Each point is one attention head; axes are per-head summaries of harmonic and temporal bias strength. Point size encodes total $\ell_1$ bias magnitude; colour encodes layer. Most heads specialise in temporal locality; a handful of early-layer heads carry most of the harmonic signal.

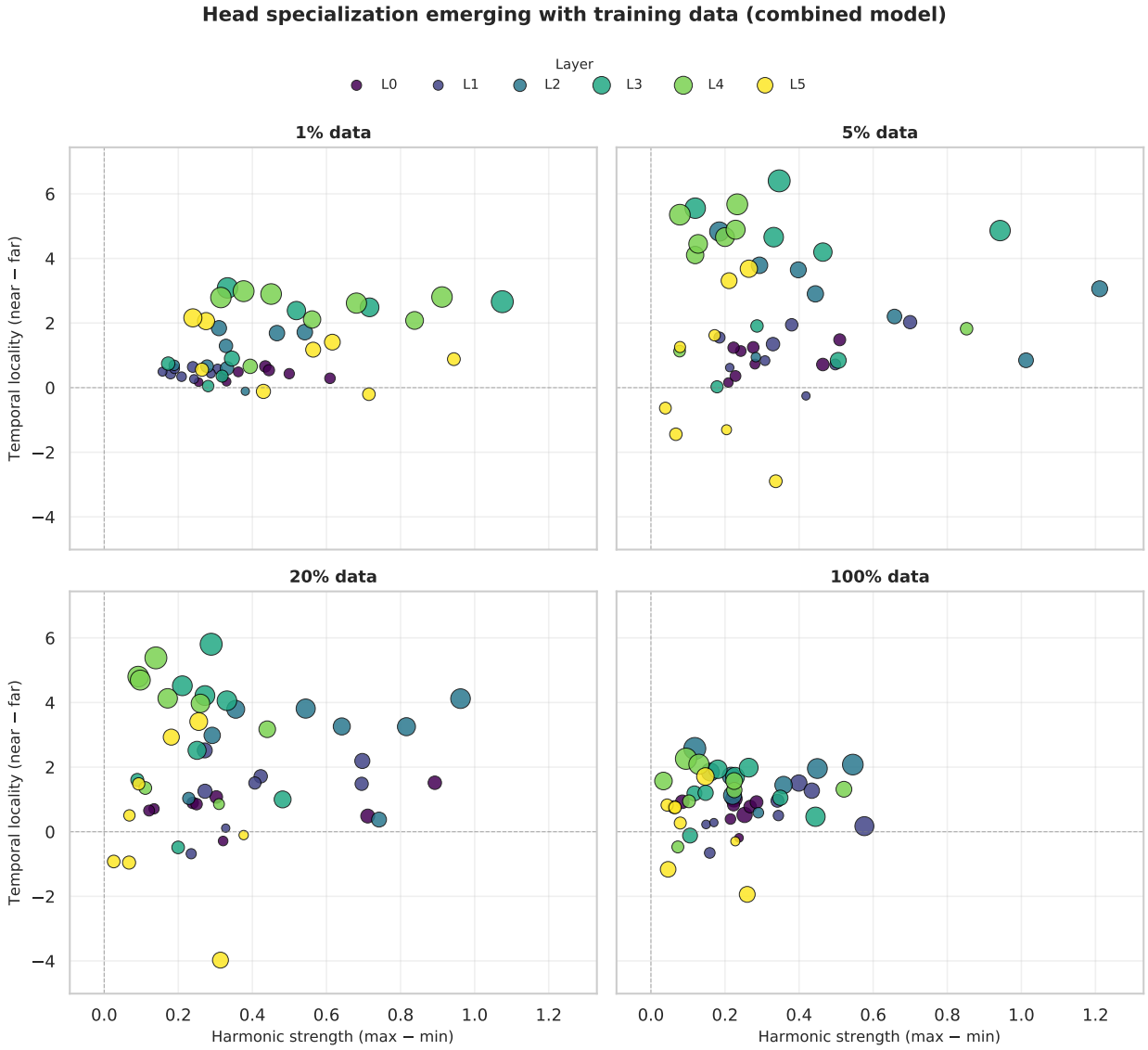


Figure 16: Head cross-plot at each data fraction. Both specialisation axes grow with more training data; temporal locality dominates at every scale.