F-StrIPE: Fast Structure-Informed Positional Encoding for Symbolic Music Generation

Manvi Agarwal, Changhong Wang, Gaël Richard
LTCI, Télécom Paris, Institut Polytechnique de Paris, France

***Abstract*—While music remains a challenging domain for generative models like Transformers, recent progress has been made by exploiting suitable musically-informed priors. One technique to leverage information about musical structure in Transformers is inserting such knowledge into the positional encoding (PE) module. However, Transformers carry a quadratic cost in sequence length. In this paper, we propose *F-StrIPE*, a structure-informed PE scheme that works in linear complexity. Using existing kernel approximation techniques based on random features, we show that F-StrIPE is a generalization of Stochastic Positional Encoding (SPE). We illustrate the empirical merits of F-StrIPE using melody harmonization for symbolic music.**

***Index Terms*—music generation, symbolic music, transformers, positional encoding, kernels.**

I. Introduction

Owing to their remarkable ability to produce realistic, high-quality samples, deep generative models are attracting significant interest. Two interdependent factors have been clearly established as being important for their superior performance: voluminous data and ever-increasing parameter counts [1]. Domains with abundant data – text, vision and speech – have successfully exploited these factors. In comparison, the limited size of publicly-available music datasets places an implicit limit on the size of the models that can be used, making music a challenging data domain.

The introduction of Transformers and attention has accelerated the advances in deep generative models and music has been no stranger to this phenomenon. However, generated music often lacks long-term coherence and organization, which are hallmarks of real music [2]. One way of improving music generation is to embed prior knowledge about musical structure into data-driven models [3], [4], for example, through the positional encoding (PE) module of Transformers [5]–[8]. This is an attractive option that provides a drop-in replacement for vanilla, structure-free PE without added complexity or training pipeline changes.

Despite their successes, Transformers bear a quadratic complexity in sequence length, which restricts their use on long sequences. Kernel approximations can be used to mitigate this cost [9], [10].

In this work, we unite these two strands of research, one of which aims to improve Transformers for music generation by using informative priors, and the other that employs kernel approximations to achieve low-complexity Transformers that are able to process long sequences. In particular, we propose F-StrIPE, a **f**ast **str**ucture-**i**nformed **p**ositional **e**ncoding method that works in linear complexity. We show that F-StrIPE is a generalization of Stochastic Positional Encoding (SPE) [11], an existing structure-free positional encoding technique, as sketched in Figure 1. We do this by using Random Fourier Features [12], thereby drawing on and providing a connection to previous work on kernel approximation for efficient attention. We empirically evaluate F-StrIPE on the symbolic music generation task of melody harmonization and show that, compared to the features used by SPE, Random Fourier Features are better-suited to structure-informed PE. We demonstrate how structure can be efficiently used in Transformers for music generation, giving us the coveted twin benefits of better performance and lower computational cost.

This work was funded by the European Union (ERC, HI-Audio, 101052978). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The accompanying website at *bit.ly/faststructurepe* contains music samples and code.

Fig. 1: A schematic showing our main contributions (best viewed in colour).

II. Background

A. Transformers and Positional Encoding

The Transformer [13] is a sequence-to-sequence architecture that processes all timesteps in a sequence parallely using *attention*. Given input sequence $[\mathbf{x}_1, ..., \mathbf{x}_T]$, each element of the output sequence is:

$$\mathbf{y}_m = \frac{\sum_n \mathsf{a}_{mn}\mathbf{v}_n}{\sum_n \mathsf{a}_{mn}} \text{ with } \mathsf{a}_{mn} = \exp\left(\frac{a_{mn}}{\sqrt{D}}\right) \tag{1}$$

where $a_{mn} = \mathbf{q}_m\mathbf{k}_n^\top$ is the attention coefficient or "similarity score" for a pair of timesteps (m, n), with $m, n \in \{1, ..., T\}$. The query, key and value vectors $\mathbf{q}_m, \mathbf{k}_m, \mathbf{v}_m$ are obtained by linearly transforming input $\mathbf{x}_m$. Since attention executes pairwise computation among all timesteps, Transformers are invariant to permutations in the temporal order of inputs. Hence, positional encoding (PE) is applied to provide the model with a sense of time. Positional information can be incorporated in two places: at the input or during attention computation. The latter is called Relative Positional Encoding (RPE) and we focus on this approach here.

B. Efficient Attention with Positional Information

Attention has quadratic complexity in sequence length. To address this, a kernelized form of attention was introduced:

$$\mathsf{a}_{mn} = \mathcal{K}(\mathbf{q}_m, \mathbf{k}_n) = \mathbb{E}\Big[\phi(\mathbf{q}_m)\phi(\mathbf{k}_n)^\top\Big] \tag{2}$$

where $\mathcal{K}$ is a positive (semi)definite kernel and $\phi(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^{D_\phi}$ defines a *randomized feature map* for $\mathbf{x}$ [10], [14]. With multiple instantiations, ϕ captures, on average, the relationship between $\mathbf{q}_m$ and $\mathbf{k}_n$, typified by $\mathcal{K}$. Thus, coefficients a_{mn} need not be computed explicitly, which produces linear-complexity Transformers.

The efficient formulation described above is not directly applicable to RPE which, as introduced in [15], requires the explicit computation

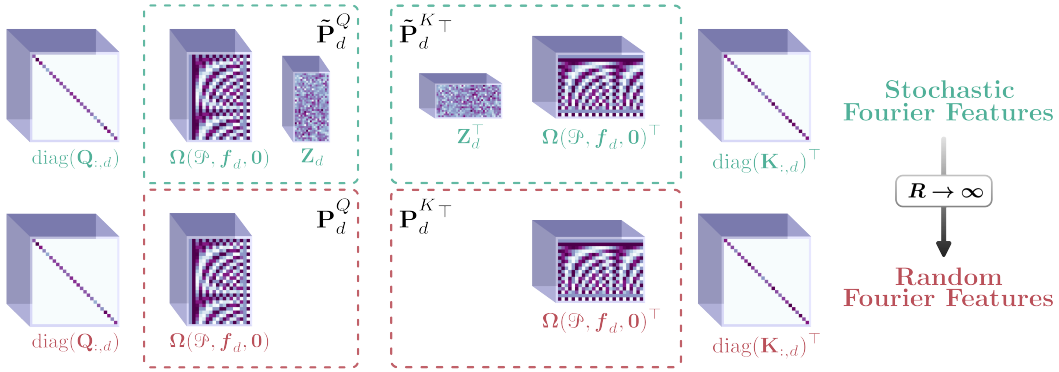

Fig. 2: A visual representation of the connection between Stochastic Positional Encoding and Random Fourier Features (best viewed in colour). The variables referenced here are detailed in Sections II and III.

of attention coefficients a_{mn}. This gap was addressed by Stochastic Positional Encoding [11], which approximates attention as:

$$a_{mn} \approx \left[\sum_{d=1}^{D} \overbrace{\text{diag}(\mathbf{Q}_{:,d})}^{\mathbf{Q}^{\text{SFF}}} \underbrace{\frac{\tilde{\mathbf{P}}_d^Q}{\sqrt{R}} \frac{\tilde{\mathbf{P}}_d^{K\top}}{\sqrt{R}}}_{\approx \mathbf{P}_d} \overbrace{\text{diag}(\mathbf{K}_{:,d}^\top)}^{\mathbf{K}^{\text{SFF}}} \right]_{mn} \tag{3}$$

In SPE, content information (**Q**/**K**) is combined with context information ($\tilde{\mathbf{P}}_d^Q$/$\tilde{\mathbf{P}}_d^K$) in the matrices $\mathbf{Q}^{\text{SFF}}$ and $\mathbf{K}^{\text{SFF}}$. Note that, for the sake of brevity, we use the convention '**Q**/**K**' as a shorthand to mean '**Q** (respectively **K**)' in this paper. SPE uses the feature maps ϕ proposed in earlier work [16], [17], but applies them to $\mathbf{Q}^{\text{SFF}}$ and $\mathbf{K}^{\text{SFF}}$ (3) instead of **Q** and **K** (2). In (3), $\mathbf{Q}_{:,d}/\mathbf{K}_{:,d}$ extracts a T_Q/T_K-dimensional vector containing the d^{th} dimension for all timesteps of the query/key matrix. The positional matrix $\mathbf{P}_d$ captures the relationship between all pairs (m, n) of timesteps that come from the positional index sequences $\mathcal{P}_Q = \{1, ..., m, ..., T_Q\}$ and $\mathcal{P}_K = \{1, ..., n, ..., T_K\}$. $\mathbf{P}_d$ is approximated by using R feature realizations of $\mathcal{P}_Q$ and $\mathcal{P}_K$. These realizations are collected in the positional representation matrices $\tilde{\mathbf{P}}_d^Q$ and $\tilde{\mathbf{P}}_d^K$, which are given as:

$$\tilde{\mathbf{P}}_d^{Q/K} = \frac{\mathbf{\Omega}\left(\mathcal{P}_{Q/K}, \boldsymbol{f}_d, \boldsymbol{\theta}_d^{Q/K}\right) \text{diag}\left(\ddot{\boldsymbol{\lambda}}_d\right) \mathbf{Z}_d}{\sqrt{2N_f}} \tag{4}$$

The first term, $\mathbf{\Omega}\left(\mathcal{P}_{Q/K}, \boldsymbol{f}_d, \boldsymbol{\theta}_d^{Q/K}\right)$, represents sinusoidal features for the index sequence $\mathcal{P}_{Q/K}$. These features are parameterized by N_f frequencies, collected in $\boldsymbol{f}_d$, and their phase shifts, collected in $\boldsymbol{\theta}_d$. The second term, diag $\left(\ddot{\boldsymbol{\lambda}}_d\right)$, consists of gains that apply to the sinusoidal features. Finally, the last term, $\mathbf{Z}_d$, consists of i.i.d. entries from a zero-mean, unit-variance Gaussian distribution. The sinusoidal features are given by:

$$[\mathbf{\Omega}(\mathcal{P}, \boldsymbol{f}, \boldsymbol{\theta})]_{ij} = \begin{cases} \cos\left(2\pi \mathbf{f}[\omega] p_i + \boldsymbol{\theta}[\omega]\right) & \text{if } j = 2\omega \\ \sin\left(2\pi \mathbf{f}[\omega] p_i + \boldsymbol{\theta}[\omega]\right) & \text{else} \end{cases} \tag{5}$$

where i runs over timesteps, such that p_i is the i^{th} timestep of index sequence $\mathcal{P}$ and ω runs over frequencies. The variable j allows us to apply the same frequency $\mathbf{f}[\omega]$ to a pair of cosine and sine features.

We call this feature representation technique, which uses (4) and (5), *Stochastic Fourier Features* (SFF). We visualize the different components of SFF in the first row of Figure 2. In particular, we illustrate the features $\tilde{\mathbf{P}}_d^Q$ and $\tilde{\mathbf{P}}_d^K$ when all gains are 1 and all phase shifts are 0.

III. Methods

A. Structure-informed positional encoding

As we hinted in Section II, positional encoding depends on a sequence $\mathcal{P} = [p_1, p_2, ..., p_T]$ of positional indices. In standard PE, which does not utilize structure, $p_i = i$, which makes $\mathcal{P}$ a linear grid. When structure is included in PE, $p_i = s_\ell(i)$, where $s_\ell(i)$ gives the structural label at level ℓ (e.g. chord) for timestep i. In this case, $s_\ell(i) = s_\ell(i')$ is possible for $i \neq i'$, making $\mathcal{P}$ a non-linear grid. In fact, we can consider vectorial positional indices with multiple resolutions of structural organization. Consequently, we obtain $p_i = \mathbf{s}(i)$, where $\mathbf{s}(i) = [s_1(i), ..., s_\ell(i), ..., s_L(i)]$ is a vector of L structural labels. Viewed in this way, PEs with structure can simply be used as a drop-in replacement for PEs without structure by replacing the form of p_i, giving us richer positional information. This is a way to flexibly represent domain-specific prior knowledge about the underlying data domain.

B. Adding structure to SPE: F-StrIPE:SFF

In order to use rich positional information in SPE, we can augment the sinusoidal feature matrix $\mathbf{\Omega}$ from (5) to be:

$$[\mathbf{\Omega}(\mathcal{P}, \boldsymbol{f}, \boldsymbol{\theta})]_{ij} = \begin{cases} \cos\left(2\pi \mathbf{f}[\omega, :]^\top p_i + \boldsymbol{\theta}[\omega]\right) & \text{if } j = 2\omega \\ \sin\left(2\pi \mathbf{f}[\omega, :]^\top p_i + \boldsymbol{\theta}[\omega]\right) & \text{else} \end{cases} \tag{6}$$

Here, we use the vectorial formulation of structure-aware positional indices $p_i = \mathbf{s}(i)$, unlike (5) where $p_i = i$ was a sequence of structure-free positional indices linked solely to the passage of time. Whereas in (5), $\mathbf{f}[\omega]$ was a single frequency, $\mathbf{f}[\omega, :]$ in (6) is a vector of frequencies. Therefore, each frequency $\mathbf{f}[\omega, \ell]$ in this vector acts on the ℓ^{th} structural label at timestep i. We can combine (3), (4) and (6) to obtain a fast structure-aware PE technique that uses Stochastic Fourier Features. We call this method *F-StrIPE:SFF*.

C. Asymptotic case of SFF: Random Fourier Features

Using Equations (3) and (4), we can express the SFF approximation of the positional matrix $\mathbf{P}_d$ for arbitrary timesteps m and n as:

$$\mathbf{P}_d[m, n] \approx \left[\Omega_{\mathbb{Q}}^d[m, :](\mathbf{Z}_d \mathbf{Z}_d^\top)\Omega_{\mathbb{K}}^{d\top}[:, n]\right]/R \tag{7}$$

where we use the abbreviation $\Omega_{\mathbb{I}}^d = \mathbf{\Omega}\left(\mathcal{P}_A, \boldsymbol{f}_d, \boldsymbol{\theta}_d^A\right) \text{diag}\left(\ddot{\boldsymbol{\lambda}}_d\right)$. We observe that $\mathbf{Z}_d \mathbf{Z}_d^\top = \widehat{\mathbf{C}}_d$ acts as an empirical covariance matrix for the features $\Omega_{\mathbb{Q}}^d$ and $\Omega_{\mathbb{K}}^d$. Since $\mathbf{Z}_d$ has zero mean and unit

variance, as $R \to \infty$, $\widehat{\mathbf{C}}_d$ approaches the theoretical covariance matrix $\mathbf{C}_d = \mathbf{I}_{2N_f}$. In the ideal case of $\mathbf{C}_d$, (7) simplifies to $\mathbf{P}_d[m,n] \approx \Omega^d_{\mathbb{C}}[m,:]\Omega^{d^\top}_{\mathcal{H}}[:,n]$, giving:

$$\mathbf{P}_d[m,n] \approx \frac{1}{N_f}\sum_{\omega=1}^{N_f} \Lambda_\omega \cos\left(f_\omega(\mathcal{P}_Q[m] - \mathcal{P}_K[n]) + \Theta_\omega\right) \quad (8)$$

where Λ_ω is the gain contributed by the matrices diag $\left(\ddot{\boldsymbol{\lambda}}_d\right)$ and Θ_ω is the phase-shift contributed by $\boldsymbol{\theta}^Q_d$ and $\boldsymbol{\theta}^K_d$. This representation has been studied in previous work, where it is called *Random Fourier Features* (RFF) [12], [18].

D. Generalizing F-StrIPE:SFF to F-StrIPE

Using this insight, we can redesign the positional feature matrices from (4) to be:

$$\mathbf{P}^{Q/K}_d = \boldsymbol{\Omega}\left(\mathcal{P}_{Q/K}, \boldsymbol{f}_d, \boldsymbol{\theta}^{Q/K}_d\right) \text{diag}\left(\ddot{\boldsymbol{\lambda}}_d\right)/\sqrt{N_f} \quad (9)$$

where the sinusoidal features $\boldsymbol{\Omega}$ uses structure-aware positional indices as given in (6). With this, we can now modify (3) to use $\mathbf{P}^{Q/K}_d$ in place of $\tilde{\mathbf{P}}^{Q/K}_d$, giving us $\mathbf{Q}^{\text{RFF}}/\mathbf{K}^{\text{RFF}}$ in place of $\mathbf{Q}^{\text{SFF}}/\mathbf{K}^{\text{SFF}}$. To signify that such a PE technique generalizes F-StrIPE:SFF to use RFF in place of SFF, we call this method *F-StrIPE*.

In the second row of Figure 2, similar to SFF, we show the different components of RFF in the case where gains are 1 and phase shifts are 0. RFF can be understood as the ideal case of SFF where $R \to \infty$. Seen in this way, RFF gives us a noiseless estimate of $\mathbf{P}_d$ with direct access to the theoretical covariance matrix $\mathbf{C}_d$.

IV. Experiments

We assess the merits of our approaches on the task of melody harmonization for symbolic music.

A. Dataset and Input Representation

We use the Chinese POP909 dataset [19] and three levels of structural labels with different resolutions [20]: melodic pitch (16^{th}-note), chord (quarter-note) and phrase (measure). Each MIDI file in this dataset consists of three tracks: melody, bridge (second melody) and piano (accompaniment). We use the POP909 alignment dataset [5] to correctly match the structural labels with the input. We convert the MIDI files to binary pianorolls $\mathbf{X} \in \mathbb{B}^{(n_{\text{tracks}}\times 128)\times n_{\text{time}}}, \mathbb{B} = \{0,1\}$, where n_{tracks} is the number of tracks and n_{time} is the number of timesteps in the pianoroll.

B. Task Setup

Given the sequence for the melody and bridge tracks $\left[\mathbf{x}_n \in \mathbb{B}^{(n_{\text{tracks}}-1)\times 128}\right]$ as input, with $n \in \{1,...,n_{\text{time}}\}$, the model must predict all tracks $\left[\mathbf{y}_n \in \mathbb{B}^{n_{\text{tracks}}\times 128}\right]$. We expect the model to produce the complete accompaniment track for all timesteps at once, without conditioning later predictions on earlier predictions. We use two settings: (16, 16) and (16, 64). The first number is the sequence length (in measures) for training and the second is that used for testing.

C. Model and Training

We use a 2-layer causal encoder Transformer with 4 heads and 512 model dimension. Training for 15 epochs with a batch size of 8, we use gradient clipping and curriculum learning [21]. We use two learning rate schedulers: a linear warmup and an epoch-wise decay. We do a grid-search for two hyperparameters: learning rate (choices: $\{1,5,10\} \times 0.0001$) and post-processing binarization strategy [5] (choices: thresholding, thresholding with merge). While the first binarization strategy uses a fixed threshold, the second additionally fills the gap between notes if the gap is less than a minimum distance.

D. Baselines and Our Methods

We consider three types of baselines: (i) Transformers without PE (NoPE [10], [22]), (ii) Transformers with efficient, approximate atttention but no structural information in PE (SPE [11]), and (iii) Transformers with structural information in PE but using inefficient, exact attention (S S-RPE [5]). From our methods, we use F-StrIPE with the three structural levels described in Section IV-A. We also assess the influence of different random features with F-StrIPE:SFF using all structural levels. We perform ablations on F-StrIPE by selecting one level at a time during training. Finally, we use the best-performing structural level from the F-StrIPE ablations to additionally do an ablation study with F-StrIPE:SFF.

E. Evaluation

We choose a collection of musically-motivated metrics from the literature, guided by four criteria.

To assess large- and small-scale structural properties, we use Self-Similarity Matrix Distance (SSMD) [23]. For both the target and the prediction, we calculate chroma vectors, giving us the number of onset occurrences per chroma in every half-measure. We then compose a self-similarity matrix (SSM) for each chroma vector by taking the pairwise cosine similarities between all elements of the vector. The SSMD is the mean absolute difference between the SSM of the target and the SSM of the prediction.

For melodic consistency, we use Chroma Similarity (CS) [23]. Using the aforementioned method of constructing chroma vectors, we compute the CS as the mean cosine similarity between corresponding entries of the target chroma vector and the prediction chroma vector.

For rhythmic consistency, we use Grooving pattern Similarity (GS) [2]. The grooving pattern of a piece of music is a vector that encodes a 1 for the quarter-notes where onsets occur and 0 for those where no onsets occur. After obtaining the grooving patterns of the target and prediction, we compute the GS as the percentage of quarter-notes where the corresponding pattern values match.

To gauge polyphonicity, we use Note Density Distance (NDD) [5], [24]. We calculate the total number of pitches in each 16^{th}-note of the target and prediction. The NDD is the average percentage of missing pitches in the prediction, with the number of pitches in the target giving us the maximum possible value.

V. Results and Discussion

In Table I, we report the mean and standard deviation on 3 seeds for each metric.

A. Utility of structural information in PE

When we compare PEs without structure (NoPE, SPE) against PEs with structure (S S-RPE, F-StrIPE), we observe that the latter perform better. This matches previous findings reported in the literature [5] which argued that using structure in PE boosts performance, particularly in underdetermined problems such as melody harmonization.

B. Influence of different random features

F-StrIPE:SFF adds structural information to SPE and F-StrIPE improves on this by using RFF in place of SFF. F-StrIPE:SFF yields marginal improvements over the performance of SPE in the (16, 16) scenario, in particular, on CS. This lends some additional support to our previous observation that structural information used in PE is useful. However, F-StrIPE, which uses a noise-free estimate of the positional matrix $\mathbf{P}_d$, gives us significant boosts over the performance of F-StrIPE:SFF and, by extension, that of SPE, on both (16, 16) and (16, 64). These improvements are particularly noticeable in CS, GS

Method	Train = 16 bars; Test = 16 bars				Train = 16 bars; Test = 64 bars			
	CS ↑	SSMD ↓	GS ↑	NDD ↓	CS ↑	SSMD ↓	GS ↑	NDD ↓
NoPE	2.68 ± 0.2	29.31 ± 0.0*	7.82 ± 0.1	93.94 ± 0.0*	2.67 ± 0.2	27.60 ± 0.0*	7.80 ± 0.2	92.49 ± 0.1
S S-RPE	14.52 ± 0.7	28.92 ± 0.1	21.58 ± 0.9	88.48 ± 0.5	\	\	\	\
SPE [11]	1.07 ± 0.0*	29.33 ± 0.0*	6.02 ± 0.0*	94.65 ± 0.0*	6.71 ± 0.3	27.59 ± 0.0*	12.62 ± 0.3	91.48 ± 0.1
F-StrIPE	11.84 ± 1.2	29.18 ± 0.0*	18.62 ± 1.4	90.93 ± 0.6	9.13 ± 1.2	27.53 ± 0.0*	14.70 ± 1.7	90.30 ± 0.3
F-StrIPE:M	1.9 ± 0.1	29.31 ± 0.0*	7.07 ± 0.1	94.27 ± 0.0*	2.22 ± 0.1	27.60 ± 0.0*	7.42 ± 0.2	92.69 ± 0.0*
F-StrIPE:C	**16.61 ± 1.5**	**28.71 ± 0.1**	**23.19 ± 2.7**	**86.42 ± 0.4**	**13.29 ± 1.1**	**27.15 ± 0.1**	**16.73 ± 0.9**	**85.61 ± 0.4**
F-StrIPE:P	2.07 ± 0.1	29.31 ± 0.0*	7.23 ± 0.1	94.20 ± 0.0*	2.31 ± 0.1	27.60 ± 0.0*	7.56 ± 0.1	92.62 ± 0.0*
F-StrIPE:SFF	2.75 ± 3.0	29.32 ± 0.0*	8.15 ± 3.5	94.18 ± 0.9	6.49 ± 0.3	27.58 ± 0.0*	12.09 ± 0.7	91.46 ± 0.1
F-StrIPE:SFF:C	4.72 ± 3.9	29.25 ± 0.1	10.22 ± 4.6	93.11 ± 1.5	10.35 ± 1.0	27.44 ± 0.1	12.09 ± 0.7	89.32 ± 0.6

TABLE I: Performance on melody harmonization. F-StrIPE:(M/C/P) are the ablations on F-StrIPE, described in Section IV-D, that apply RFF on only one structural level at a time - melodic pitch/chord/phrase. F-StrIPE:SFF:C applies SFF on chords, which is the best performing ablation setting with RFF. '0.0*' refers to standard deviations that are lower than 0.05. '\' refers to simulations where the given inference setting could not be accessed due to the heavy computational demands of the method.

and NDD and are especially pronounced in the (16, 16) setting. This emphasizes that the correct approximation techniques can strongly enhance the effect of augmenting our models with prior knowledge.

C. Ablations on F-StrIPE and F-StrIPE:SFF

As described in Section IV-D, we perform ablations on structural levels used during training, resulting in models that are specialized to use only melody (F-StrIPE:M), only chord (F-StrIPE:C) or only phrase (F-StrIPE:P) as structural information. When we compare these models against F-StrIPE, which uses all three structures, we see first that F-StrIPE:M and F-StrIPE:P do worse than F-StrIPE on all metrics and both task settings. In fact, their performance drops lower than even NoPE. In contrast, F-StrIPE:C brings a clear advantage over F-StrIPE, with significant improvements on all metrics and both task settings. This fits our intuition that the accompaniment in pop songs can be nicely characterized by chord progressions [25], [26]. Our results show that using only chord information is better than using all structures simultaneously. This shows that while task-specific structural information can boost performance, ill-founded and generic priors can prove counterproductive. Thus, how prior knowledge is selected and incorporated into a deep-learning model should be an important consideration while designing such systems.

D. Comparing complexities

These results should be viewed in the context of the complexity analysis presented in Table II. Compared to S S-RPE, SPE and F-StrIPE have linear complexity in sequence length, which makes a sizeable dent in the requirement for computational resources. Moreover, scaling up from SPE to F-StrIPE only adds a factor of $\mathfrak{s}$, corresponding to the number of structures we use in PE, which grows the slowest compared to all other variables. In fact, since the ablations use only a single structure at a time ($\mathfrak{s} = 1$), the complexity of our best-performing method, F-StrIPE:C, matches that of SPE.

Thus, on the one hand, in the worst case where multiple structures are needed, it only costs a small amount of additional computational resources. On the other hand, if we already know which structure is best-suited to our task, we can benefit from F-StrIPE and leverage prior knowledge in our model without needing any extra resources.

Method	Additional Parameters	Runtime Space Complexity
S S-RPE [5]	$\mathcal{O}(\mathfrak{s}(\hbar d)^2 + \ell(\hbar d)^2)$	$\mathcal{O}(\ell \mathfrak{t}^2 \hbar d)$
SPE [11]	$\mathcal{O}(\hbar d N_f)$	$\mathcal{O}(\ell \mathfrak{t} \hbar d N_f)$
F-StrIPE	$\mathcal{O}(\mathfrak{s} \hbar d N_f)$	$\mathcal{O}(\mathfrak{s} \ell \mathfrak{t} \hbar d N_f)$

TABLE II: Complexity analysis for different methods with ℓ layers, $\hbar$ heads, d head dimension, $\mathfrak{s}$ structures, and $\mathfrak{t}$ sequence length. A typical order for size is $\mathfrak{s} < (\hbar, \ell) << d <<< \mathfrak{t}$, with $\mathfrak{s}$ contributing the least and $\mathfrak{t}$ contributing the most, assuming equal growth rate.

E. Length Generalization

Finally, we see that models that are trained on 16 bars of music but tested on 64 bars of music reflect the same trends as seen in the models that were tested on 16 bars of music: structural information combined with RFF provides a sizeable improvement over baselines that either do not use structure or use structure but with SFF.

On CS and GS, the PEs that use SFF (SPE, F-StrIPE:SFF and F-StrIPE:SFF:C) show large improvements on the (16, 64) setting compared to the (16, 16) setting. This does not hold true for the RFF-based PEs, where some methods show small improvements and others show small deteriorations. Nevertheless, F-StrIPE:C outperforms all other PEs in the (16, 64) situation. This suggests that an in-depth comparative investigation of the characteristics of different approximation techniques is needed. Specifically, it would be interesting to understand which approximation is suited to what learning scenario and whether we can combine the strengths of different approximations to obtain a more robust, fast, structure-informed PE. Interestingly, in two metrics - SSMD and NDD - all the methods in Table I do marginally better in the (16, 64) scenario compared to the (16, 16) one.

The transfer of performance from the (16, 16) setting to the (16, 64) setting can be partly attributed to the presence of stereotypical structure and a high degree of repetition in pop songs [27], [28]. Thus, 16 measures of music could potentially contain much of the necessary information to generate much longer sequences. It would be interesting to quantatively assess whether this hypothesis is true.

VI. Conclusion

In this paper, we demonstrated how structural information can be used in linear-complexity positional encoding, thereby retaining superior performance without sacrificing efficiency. We did this by first extending SPE to accept multi-resolution, structure-aware positional indices, obtaining F-StrIPE:SFF. Then, we showed the connection between SPE [11] and Random Fourier Features [12] and developed a novel method, called F-StrIPE, framed as a structure-aware generalization of SPE. The combination of these two interventions — using structure and using Random Fourier Features — gave us a fast, structure-informed positional encoding method that outperformed SPE, F-StrIPE:SFF and other competitive baselines on melody harmonization for symbolic music.

REFERENCES

[1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020, preprint arXiv:2001.08361.

[2] S. Wu and Y. Yang, "The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures," *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2020.

[3] G. Richard, V. Lostanlen, Y.-H. Yang, and M. Müller, "Model-Based Deep Learning for Music Information Research," *IEEE Signal Processing Magazine*, 2024. [Online]. Available: https://hal.science/hal-04611461

[4] K. Bhandari and S. Colton, "Motifs, Phrases, and Beyond: The Modelling of Structure in Symbolic Music Generation," 2024, preprint arXiv:2403.07995.

[5] M. Agarwal, C. Wang, and G. Richard, "Structure-Informed Positional Encoding for Music Generation," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, preprint arXiV:2402.13301.

[6] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "PopMAG: Pop Music Accompaniment Generation," *ACM International Conference on Multimedia (MM)*, 2020.

[7] Z. Guo, J. Kang, and D. Herremans, "A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling," *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

[8] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, "Symphony Generation with Permutation Invariant Language Model," *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2022.

[9] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A Survey," *ACM Computing Surveys*, 2022.

[10] Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. Salakhutdinov, "Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel," *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[11] A. Liutkus, O. Cífka, S.-L. Wu, U. Simsekli, Y.-H. Yang, and G. Richard, "Relative Positional Encoding for Transformers with Linear Complexity," *International Conference on Machine Learning (ICML)*, 2021.

[12] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2007.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[14] K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller, "Rethinking Attention with Performers," *International Conference on Machine Learning (ICML)*, 2021.

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

[16] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention," *International Conference on Machine Learning (ICML)*, 2020.

[17] S. Zhuoran, Z. Mingyuan, Z. Haiyu, Y. Shuai, and L. Hongsheng, "Efficient Attention: Attention with Linear Complexities," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[18] D. J. Sutherland and J. Schneider, "On the error of random Fourier features," *Conference on Unicertainty in Artificial Intelligence (UAI)*, 2015.

[19] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "POP909: A Pop-song Dataset for Music Arrangement Generation," *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2020.

[20] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic Analysis and Influence of Hierarchical Structure on Melody, Rhythm and Harmony in Popular Music," *Joint Conference on AI Music Creativity (AIMC)*, 2020.

[21] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," *International Conference on Machine Learning (ICML)*, 2009.

[22] A. Haviv, O. Ram, O. Press, P. Izsak, and O. Levy, "Transformer Language Models without Positional Encodings Still Learn Positional Information," *Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[23] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with Just One Transformer VAE," *IEEE/ACM Transactions on Audio, Speech, Language Processing (TASLP)*, 2022.

[24] B. Haki, M. Nieto, T. Pelinski, and S. Jordà Puig, "Real-time drum accompaniment using transformer architecture," *Joint Conference on AI Music Creativity (AIMC)*, 2022.

[25] J.-F. Paiement, D. Eck, and S. Bengio, "A probabilistic model for chord progressions," *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2005.

[26] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[27] G. Sargent, F. Bimbot, and E. Vincent, "Estimating the Structural Segmentation of Popular Music Pieces Under Regularity Constraints," *IEEE/ACM Transactions on Audio, Speech, Language Processing (TASLP)*, 2017.

[28] S. Dai, H. Yu, and R. B. Dannenberg, "What is missing in deep music generation? a study of repetition and structure in popular music," *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2022.