DID: 2D AUTOREGRESSIVE MODELING WITH A DECODER-IN-DECODER ARCHITECTURE FOR SYMBOLIC MUSIC GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Symbolic music generation aims to create compositions from discrete musical events. Existing autoregressive methods flatten multi-bar events into a single sequence, losing the explicit 2D structure, while non-autoregressive approaches often rely on fixed grids, limiting flexibility for variable-length generation. In this work, we propose DID, a 2D autoregressive framework that models musical events across the time and note dimensions. This design preserves the two-dimensional organization of music while retaining autoregressive flexibility. DID follows a Decoder-in-Decoder architecture with two decoders: the time decoder updates a running musical history along the time dimension, while the note decoder uses this memory to generate events along the note dimension. To improve harmonic consistency, we introduce harmonic Plan events as intermediate tokens in the note decoder's output sequence, turning the given conditions into a soft harmonic constraint on the notes that follow. Experiments on multiple public datasets show that DID improves harmonic and rhythmic coherence over prior methods, while supporting flexible generation scenarios, including variable-length music continuation, chord-conditioned piano generation, and accompaniment generation. The generated music samples are available at `https://anonymous.4open.science/w/did-cases-785B/`.

1 INTRODUCTION

Listeners experience music as audio, while composers often work with symbolic representations of notes, chords, rhythm, and dynamics. These representations allow individual elements to be edited directly, but organizing them into a complete piece still takes considerable time and effort. Therefore, it is meaningful to develop AI systems for symbolic music generation that assist composers in the creative process.

Existing symbolic music generation methods can be broadly categorized into autoregressive and non-autoregressive paradigms. Autoregressive approaches typically rely on causal sequence models, which predict each event from preceding events and extend music step by step. Their causal decoding flexibly incorporates preceding context as the music grows, supporting variable-length generation. In contrast, non-autoregressive methods predict many musical elements in parallel or refine them iteratively. Recent diffusion models progressively denoise two-dimensional piano-roll representations, where local conditions such as chords and melodies can be aligned with corresponding time regions to support fine-grained control.

Despite significant progress, preserving the explicit two-dimensional structure of music within a flexible causal generation process remains challenging, as illustrated in Fig. 1. On the one hand, existing autoregressive methods typically flatten the time and note dimensions into a one-dimensional token stream, making their inherent two-dimensional organization implicit in the decoding order. On the other hand, diffusion-based methods preserve the time–note structure on two-dimensional grids, but each denoising process typically covers a fixed temporal window, limiting flexibility for variable-length generation. Extending music beyond this window often relies on overlapping-window inpainting, making it difficult to carry information from earlier windows forward without

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.

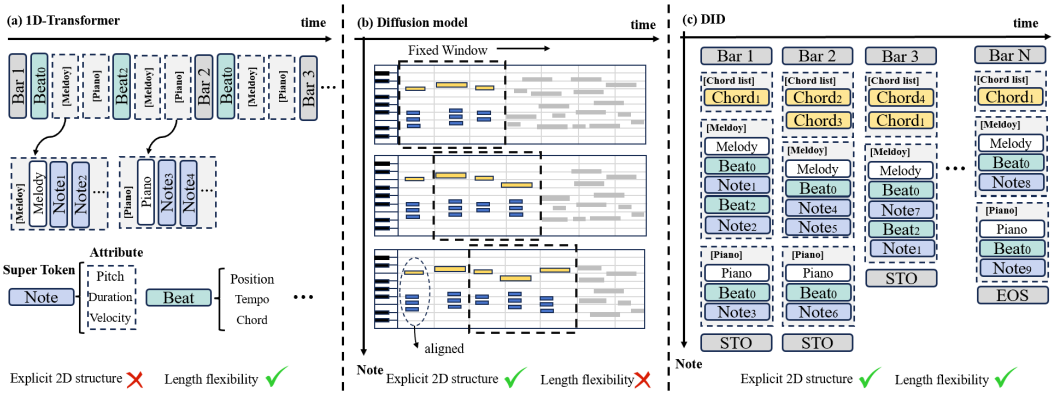

Figure 1: Comparison of generation paradigms. (a) 1D Transformers flatten musical events, retaining length flexibility but making 2D structure implicit. (b) Diffusion models preserve 2D structure within fixed windows and extend music through inpainting. (c) DID autoregressively decodes along the time and note dimensions, preserving explicit 2D structure and flexible length.

an additional memory or planning mechanism. These limitations leave a gap between explicit two-dimensional modeling and flexible autoregressive generation.

In this work, we propose DID, a framework for 2D autoregressive modeling of musical events along the time and note dimensions. DID treats each bar as one step along the time dimension. At each step, the model summarizes the generated history and uses this summary to autoregressively generate a row of events along the note dimension. Once the row is complete, DID updates the musical history with the generated events and advances to the next bar. This causal process can continue across a variable number of bars, preserving the explicit two-dimensional organization of music while retaining the flexibility of autoregressive generation.

Concretely, DID realizes this process through a Decoder-in-Decoder architecture with a time decoder and a note decoder. The time decoder summarizes completed bars into a global memory that follows musical progression over time. The note decoder combines this memory with the complete NOTE events from the immediately preceding bar to generate the current events. These two inputs serve different roles. The global memory provides longer context, while the NOTE events preserve exact local details. After the current bar is completed, its summary becomes an input to the time decoder at the next step to update the global memory, connecting the two decoders throughout generation.

Many conventional autoregressive models are built around sequential continuation, whereas DID extends the same 2D causal formulation to chord-conditioned generation and accompaniment. At each time step, the musical memory and aligned condition events form a prefix for the note decoder, and the remaining events are generated autoregressively along the note dimension. Chords are used for chord-conditioned piano generation, while chords and melody are used for accompaniment generation. To improve harmonic consistency in both settings, we introduce Harmonic Plan events that outline local harmony with a bass pitch class and a pitch-class set. Each Plan appears before the target notes, providing a soft harmonic constraint for subsequent generation.

We evaluate DID using POP1k7 and POP909. Music continuation is evaluated on both datasets, while chord-conditioned piano generation and accompaniment generation from given chords and melody are evaluated on POP909. The results show that DID improves harmonic and rhythmic coherence over prior methods while supporting all three generation settings. Our main contributions are summarized as follows:

- We propose DID, a 2D autoregressive modeling paradigm that directly generates musical events across the time and note dimensions, preserving explicit score organization while supporting flexible variable-length generation.
- We realize this paradigm with a Decoder-in-Decoder architecture: the time decoder maintains musical memory across bars, while the note decoder generates events within each bar.

We further introduce harmonic Plan events that transform aligned musical conditions into soft harmonic constraints on subsequent note generation, providing intermediate harmonic guidance.

- We evaluate DID on three generation tasks: music continuation on POP1k7 and POP909, chord-conditioned piano generation on POP909, and accompaniment generation from given chords and melody on POP909. The results show improved harmonic and rhythmic coherence over prior methods and demonstrate the flexibility of DID across variable-length and conditional generation.

2 RELATED WORK

Symbolic Music Generation. Symbolic music generation represents music as discrete events, enabling explicit control of musical attributes such as pitch, rhythm, harmony, and dynamics. Recent work has advanced long-term structure modeling and controllable generation in symbolic music and related music-creation tasks Roberts et al. (2018); Liu et al. (2022b;a); von Rütte et al. (2023); Li et al. (2026). Current approaches can be broadly divided into autoregressive and non-autoregressive paradigms. We focus on how these paradigms organize musical structure during generation.

Autoregressive Methods. Autoregressive symbolic music generation has been extensively studied through improved context modeling and event representations Huang et al. (2018); Guo et al. (2023); Fradet et al. (2023). REMI uses beat-based events Huang & Yang (2020), CP groups multiple attributes into compound tokens Hsiao et al. (2021), NMT further models dependencies among attributes within each compound token through nested decoding Yoo et al. (2024), and MuseTok explores music-specific tokenization Huang et al. (2026); these methods serve as our main autoregressive baselines. Several related works also exploit hierarchical musical structure, but their hierarchy is introduced mainly through attention, representation, or staged decoding rather than by explicitly autoregressing over both score-level time and note dimensions. For example, Museformer organizes attention with bar summaries Yu et al. (2022), Bar Transformer builds note-to-bar hierarchical representations Qin et al. (2023), and Sawada et al. autoregressively generate bar-level features before decoding individual bars Sawada et al. (2025). DID instead treats time and note as two coupled autoregressive dimensions, with the time decoder advancing over completed bars and the note decoder generating within-bar events; their distinction from hierarchical designs is detailed in Appendix D.

Non-Autoregressive Methods. Non-autoregressive approaches Dong et al. (2018); Mittal et al. (2021); Plasser et al. (2023); Huang et al. (2024) often preserve the two-dimensional organization of music explicitly. Polyffusion Min et al. (2023) performs conditional generation over a piano-roll-like representation, while GETMusic Lv et al. (2023) organizes symbolic tokens along track and time dimensions within a diffusion framework. WholeSongGen Wang et al. (2024) introduces hierarchical diffusion for full-song generation. FGG Zhu et al. (2025) and D3PIA Choi et al. (2026) further improve fine-grained conditioning for accompaniment generation. These methods preserve musical dimensions explicitly, but generation is generally performed over a fixed time span and extended through additional procedures when longer outputs are required. DID explores a different formulation by introducing explicit time–note organization into causal generation: the time decoder maintains cross-bar history, while the note decoder generates events within each bar. This allows the two dimensions to remain explicit while the generation process proceeds autoregressively.

3 METHODOLOGY

In this section, we detail our proposed DID framework, which integrates explicit two-dimensional modeling with flexible autoregressive generation. As illustrated in Figure 2(a), events within each bar are first embedded and summarized into a bar-level representation. Using these representations, Figure 2(b) performs 2D decoding along two autoregressive dimensions: the time decoder advances across bars along the time dimension, while at each time step the note decoder autoregressively expands the corresponding bar along the note dimension. Once a bar is completed, its summary is fed back to the time decoder, forming the Decoder-in-Decoder process. For conditional generation, Figure 2(c) introduces Harmonic Plan events to provide intermediate harmonic guidance, while Figure 2(d) summarizes the task-specific inputs for the three generation settings.

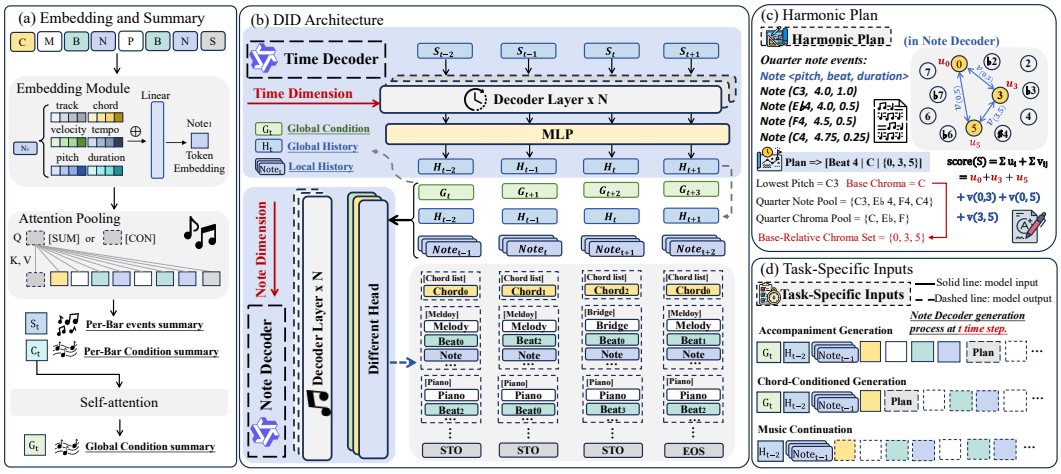

Figure 2: Overview of DID. (a) Compound-event embedding and attention-based bar summaries. (b) Decoder-in-Decoder modeling with global and local history. (c) Harmonic Plan construction and prediction. (d) Task-specific inputs for the three generation settings.

3.1 2D Autoregressive Formulation

Conventional event-based autoregressive modeling represents multi-bar music by flattening events into a single sequence, whereas DID retains bar progression and within-bar event order as two separate structural axes. After embedding, the two formulations can be written as

$$\mathbf{X}^{\mathrm{1D}} \in \mathbb{R}^{B\times L\times d_{\mathrm{n}}}, \qquad \mathbf{X}^{\mathrm{DID}} \in \mathbb{R}^{B\times T\times N_{\max}\times d_{\mathrm{n}}}. \tag{1}$$

Here, L denotes the flattened sequence length, T the number of bars, and $N_{\max}$ the maximum number of events within a bar. The two structural axes T and $N_{\max}$ correspond to the time and note dimensions, respectively, allowing DID to preserve the 2D organization of music explicitly during autoregressive modeling.

Based on this 2D formulation, we construct the bar-level representations in Figure 2(a) to connect the note and time dimensions. Each compound event is first projected into the note-decoder space, and a learned summary query then aggregates the events of a completed bar:

$$\boldsymbol{x}_{t,n} = W_{\mathrm{emb}}\left[\mathrm{Emb}(e^1_{t,n}); \ldots ; \mathrm{Emb}(e^K_{t,n})\right], \tag{2}$$

$$\boldsymbol{h}^s_{t,0:N_t} = \mathrm{BarEnc}\left([\boldsymbol{q}_{\mathrm{sum}}; \boldsymbol{x}_{t,1}; \ldots ; \boldsymbol{x}_{t,N_t}]\right), \qquad \boldsymbol{s}_t = \boldsymbol{h}^s_{t,0}. \tag{3}$$

Here, $e^k_{t,n}$ denotes the k-th field of compound event $\boldsymbol{e}_{t,n}$. The resulting $\boldsymbol{s}_t$ represents the completed bar along the time dimension. For conditional generation, all conditioning events are available before decoding. We therefore summarize the conditions within each bar using a separate query and apply self-attention across the resulting bar-level representations:

$$\boldsymbol{h}^c_{t,0:M_t} = \mathrm{BarEnc}\left([\boldsymbol{q}_{\mathrm{cond}}; \boldsymbol{x}^c_{t,1}; \ldots ; \boldsymbol{x}^c_{t,M_t}]\right), \qquad \boldsymbol{c}_t = \boldsymbol{h}^c_{t,0}, \tag{4}$$

$$(\mathbf{G}_1, \ldots, \mathbf{G}_T) = \mathrm{SelfAttn}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_T). \tag{5}$$

Here, $\boldsymbol{x}^c_{t,m}$ denotes the embedding of the m-th conditioning event in bar t. The resulting $\mathbf{G}_t$ provides global conditioning information for bar t.

3.2 Decoder-in-Decoder Architecture

With the 2D formulation and bar-level representations defined above, DID generates a piece by alternating along the time and note dimensions, as illustrated in Figure 2(b). At time step t, the time decoder advances across bars and provides context for the current bar, while the note decoder expands bar t event by event along the note dimension. To provide the preceding information required for this note-level decoding, we divide the musical history into two non-overlapping paths. The time decoder receives delayed bar summaries up to $t-2$ as long-range global history, while the complete

NOTE events of bar $t-1$ are retained separately as local history $\boldsymbol{L}_{t-1}$. This separation preserves both long-range context and precise neighboring details without representing bar $t-1$ twice.

The time decoder applies causal self-attention over the bar-summary stream and produces a hidden state $\boldsymbol{h}_{t-2}$. We then expand this state into Q global-history tokens in the note-decoder space:

$$\boldsymbol{h}_{t-2} = \mathcal{D}_{\text{time}}\left(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-2}\right)_{t-2}, \qquad \boldsymbol{H}_{t-2} = \text{Reshape}_Q\left(\text{MLP}(\boldsymbol{h}_{t-2})\right) \in \mathbb{R}^{Q \times d_{\text{n}}}. \tag{6}$$

The MLP maps the time-decoder state to Qd_{n} dimensions before reshaping it into Q tokens, and we set $Q = 4$ in all experiments. The resulting $\boldsymbol{H}_{t-2}$ provides compressed long-range history, while $\boldsymbol{L}_{t-1}$ preserves the complete NOTE events of the immediately preceding bar. Together, they provide the historical context for decoding bar t along the note dimension. For conditional generation, the global condition $\mathbf{G}_t$ is further added to this context. The note-decoder prefix is defined as:

$$\boldsymbol{P}_t = \begin{cases} [\mathbf{G}_t; \boldsymbol{H}_{t-2}; \boldsymbol{L}_{t-1}], & \text{conditional generation,} \\ [\boldsymbol{H}_{t-2}; \boldsymbol{L}_{t-1}], & \text{continuation.} \end{cases} \tag{7}$$

While $\boldsymbol{P}_t$ provides the historical and global context for bar t, task-specific conditions are supplied directly as known events before the target events in the current row. At time step t, the prefix and these task-specific conditions together provide the context for note-level decoding of the current bar. As illustrated in Figure 2(d), chord-conditioned generation uses chord events, accompaniment generation uses chords and melody, and continuation uses no conditional events.

With this context fixed at time step t, the note decoder autoregressively generates the events of the current bar along the note dimension:

$$P(\boldsymbol{b}_t \mid \boldsymbol{P}_t) = \prod_{n=1}^{N_t} P\left(\boldsymbol{e}_{t,n} \mid \boldsymbol{P}_t, \boldsymbol{e}_{t,<n}\right). \tag{8}$$

At each position n, DID first predicts the event Type and then activates the corresponding attribute heads shown in Figure 2(b). The event order follows the musical structure within the current bar, and an STO token terminates the note-level decoding step.

Once $\boldsymbol{b}_t$ is completed along the note dimension, DID returns to the time dimension. The NOTE events of $\boldsymbol{b}_t$ become the new local history $\boldsymbol{L}_t$, while the bar encoder summarizes the completed bar as $\boldsymbol{s}_t$ for subsequent time-level modeling. When generating bar $t+1$, $\boldsymbol{L}_t$ provides the immediate local context, whereas the global history contains summaries only up to $\boldsymbol{s}_{t-1}$. The summary $\boldsymbol{s}_t$ enters the global history when generating bar $t+2$, after the local history has advanced to $\boldsymbol{L}_{t+1}$. This delayed update keeps the two history paths non-overlapping.

Repeating this process across time steps gives the complete 2D autoregressive formulation:

$$P(\mathcal{X}) = \prod_{t=1}^{T} \prod_{n=1}^{N_t} P\left(\boldsymbol{e}_{t,n} \mid \boldsymbol{P}_t, \boldsymbol{e}_{t,<n}\right). \tag{9}$$

The outer product advances across bars along the time dimension, while the inner product advances across events along the note dimension. This nested factorization follows the 2D organization introduced above and summarizes the decoding process in Algorithm 1.

Algorithm 1: DID 2D Decoding.

Require: $\{\boldsymbol{p}_t\}, T_{\max}$; optional $\{\boldsymbol{g}_t, \mathcal{C}_t\}$
$\boldsymbol{L}_0 \leftarrow$
Use empty $\boldsymbol{g}_t, \mathcal{C}_t$ if conditions are absent
for $t = 1, \ldots, T_{\max}$ **do**
  $\boldsymbol{h}_{t-2} \leftarrow \mathcal{D}_{\text{time}}(\boldsymbol{p}_{\le t}, \boldsymbol{s}_{\le t-2})$
  $\boldsymbol{H}_{t-2} \leftarrow \text{MLP}(\boldsymbol{h}_{t-2})$ *// global history*
  $\boldsymbol{P}_t \leftarrow [\boldsymbol{g}_t; \boldsymbol{H}_{t-2}; \boldsymbol{L}_{t-1}], \boldsymbol{b}_t \leftarrow \mathcal{C}_t$
  repeat
    $\boldsymbol{z} \leftarrow \mathcal{D}_{\text{note}}(\boldsymbol{P}_t, \boldsymbol{b}_t)$
    Decode and append $(\tau, \boldsymbol{e})$ to $\boldsymbol{b}_t$
  until τ is STO or EOS
  $\boldsymbol{s}_t \leftarrow \text{BarEnc}(\boldsymbol{b}_t)$
  $\boldsymbol{L}_t \leftarrow \text{NOTE}(\boldsymbol{b}_t)$ *// local history*
  if EOS **then return** $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_t)$
end for
return $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{T_{\max}})$

3.3 Harmonic Plan Events

In conditional generation, the given chords or melodies constrain the target music but do not determine its exact note-level realization. DID therefore inserts Harmonic Plan events between the given conditions and target events, as illustrated in Figure 2(c). Each Plan describes a quarter-note region using a bass pitch class and a bass-relative pitch-class set. For quarter region q in bar t, the bass $b_{t,q}$ is determined by the lowest sounding note, and the relative set is constructed from the pitch classes of note onsets:

$$\mathcal{S}_{t,q} = \left\{ \left((p \bmod 12) - b_{t,q} \right) \bmod 12 \mid p \in \mathcal{A}_{t,q} \right\}, \tag{10}$$

where $\mathcal{A}_{t,q}$ denotes the onset pitches in the region. Using bass-relative pitch classes removes octave and exact voicing information while retaining local harmonic structure, allowing the Plan to guide harmony without fixing its note-level realization. During decoding, Harmonic Plan events are generated after the given conditions and before the target notes. Given the note-decoder state $\boldsymbol{z}$, DID first predicts the bass and then the corresponding bass-relative pitch-class set:

$$P(b_{t,q}, \mathcal{S}_{t,q} \mid \boldsymbol{z}) = P(b_{t,q} \mid \boldsymbol{z})\, P(\mathcal{S}_{t,q} \mid b_{t,q}, \boldsymbol{z}), \tag{11}$$

$$\mathrm{score}(\mathcal{S}) = \sum_{i \in \mathcal{S}} u_i + \sum_{\{i,j\} \subseteq \mathcal{S}} v_{ij}. \tag{12}$$

Rather than predicting pitch classes independently, the set head scores each candidate set using individual interval preferences and pairwise compatibility, and normalizes these scores to obtain $P(\mathcal{S} \mid b, \boldsymbol{z})$. The predicted Plan remains visible to subsequent note decoding, providing local harmonic guidance while leaving detailed note realization to the autoregressive decoder. We use Harmonic Plan events for chord-conditioned generation and accompaniment generation.

4 Experiments

4.1 Tasks and datasets

For our experiments, we consider three tasks: music continuation, chord-conditioned piano generation, and accompaniment generation. For continuation, we provide a four-bar prompt and generate the following 28 bars; chord-conditioned generation and accompaniment generation are likewise evaluated over 32 bars. We evaluate continuation on POP1k7 and POP909, and the other two tasks on POP909. POP1k7 Hsiao et al. (2021) contains 1,747 transcribed pop-piano performances, while POP909 Wang et al. (2020) contains 909 pop songs with separated melody, secondary-melody, and accompaniment tracks, together with aligned beat and chord annotations. Both datasets are split by song into training, validation, and test sets at an 80/10/10 ratio. Detailed architecture, training, and inference configurations are provided in Appendix C.

Table 1: Continuation generation results on POP909 and POP1k7. Best and second-best results are shown in bold and underlined.

Dataset	Method	$\mathrm{Sim}_{\mathrm{PCE}} \uparrow$	$\mathrm{Sim}_{\mathrm{Chr}} \uparrow$	$\mathrm{Sim}_{\mathrm{Grv}} \uparrow$	PD $\uparrow$	$\mathrm{Sim}_{\mathrm{SC}} \uparrow$	Avg$\uparrow$
POP909	REMI	94.95	91.00	88.93	<u>67.25</u>	96.33	87.69
	CP	<u>95.42</u>	91.27	90.12	65.66	97.85	<u>88.06</u>
	NMT	94.89	<u>91.69</u>	<u>91.35</u>	63.77	**98.47**	88.04
	MuseTok	93.31	87.77	84.90	66.94	94.37	85.46
	DID (Ours)	**96.37**	**92.10**	**92.49**	**68.42**	<u>98.29</u>	**89.53**
POP1k7	REMI	**95.18**	91.40	84.11	**62.06**	<u>97.11</u>	<u>85.97</u>
	CP	94.12	89.88	82.74	58.35	96.09	84.24
	NMT	94.25	<u>91.80</u>	85.77	60.75	97.03	85.92
	MuseTok	93.89	86.67	<u>86.68</u>	60.28	95.23	84.55
	DID (Ours)	<u>95.12</u>	**91.80**	**87.86**	<u>61.81</u>	**97.37**	**86.79**

4.2 Evaluation Metrics

Objective Evaluation. For music continuation, we evaluate tonal similarity using SimChr, SimPCE, and SimSC, rhythmic similarity using SimGrv, and pitch-distribution similarity using PD. For conditional generation, we evaluate harmonic consistency using CA and CS, with OS additionally used to measure rhythmic similarity in accompaniment generation.

Subjective Evaluation. For music continuation, we evaluate Coherence, Rhythmic Consistency, Harmonic Consistency, and Overall Quality. For accompaniment generation, we evaluate the same perceptual aspects and additionally include Accompaniment-Pattern Stability to assess the consistency of accompaniment patterns. Detailed criteria and protocols are provided in Appendix B.

Table 2: Results for accompaniment generation on POP909. Best and second-best results are shown in bold and underlined, respectively. Inference time is reported per eight generated bars.

Method	Base	Params (M)	Inf. Time (s)↓	CA↑	CS↑	OS↑	AVG↑
Polyffusion	Diff.	41.1M	21.4	52.00	62.57	77.57	64.05
WholeSongGen	Diff.	41.6M	79.0	79.72	78.74	<u>83.29</u>	80.58
FGG	Diff.	36.7M	0.4	75.44	69.78	100.00†	81.74
D3PIA	Diff.	2.2M	1.7	<u>81.61</u>	**81.23**	82.74	<u>81.86</u>
DID (Ours)	Trans.	34.3M	2.84	**83.52**	<u>79.30</u>	**85.00**	**82.61**

Base: **Diff.** denotes diffusion-based models, and **Trans.** denotes Transformer-based models. † FGG uses explicit onset information as additional rhythmic conditioning and is excluded from the OS ranking.

Table 3: Human evaluation on POP909 accompaniment generation and music continuation. Values are mean scores with 95% confidence intervals shown below. Best and second-best means are shown in **bold** and underlined, respectively.

(a) Accompaniment Generation

Model	Coh.	Harm.	Rhy.	Patt.	Corr.	Overall
Polyff.	2.32 ±0.41	2.55 ±0.35	2.50 ±0.57	2.42 ±0.43	2.40 ±0.42	2.17 ±0.33
WSG	2.82 ±0.40	3.25 ±0.54	2.98 ±0.49	<u>3.07</u> ±0.38	<u>3.00</u> ±0.32	<u>2.80</u> ±0.27
FGG	2.77 ±0.47	2.72 ±0.40	3.03 ±0.66	3.03 ±0.59	2.25 ±0.39	2.40 ±0.35
D3PIA	<u>2.90</u> ±0.40	<u>3.33</u> ±0.55	<u>3.13</u> ±0.54	3.03 ±0.52	2.87 ±0.23	<u>2.80</u> ±0.33
DID	**3.25** ±0.39	**3.35** ±0.35	**3.75** ±0.27	**3.68** ±0.26	**3.20** ±0.09	**3.27** ±0.19

(b) Music Continuation

Model	Coh.	Rhy.	Harm.	Overall
REMI	2.40 ±0.53	<u>2.58</u> ±0.44	2.68 ±0.61	<u>2.38</u> ±0.37
CP	<u>2.44</u> ±0.43	2.54 ±0.29	2.78 ±0.31	2.36 ±0.26
NMT	2.24 ±0.62	2.46 ±0.38	<u>2.84</u> ±0.32	2.18 ±0.30
MuseTok	1.80 ±0.68	2.42 ±0.58	1.36 ±0.34	1.24 ±0.29
DID	**3.04** ±0.45	**3.44** ±0.40	**2.94** ±0.34	**2.98** ±0.52

WSG denotes WholeSongGen; Polyff. denotes Polyffusion.

4.3 Baselines

For music continuation, we compare DID with representative autoregressive methods, including **REMI** Huang & Yang (2020), **Compound Word (CP)** Hsiao et al. (2021), **Nested Music Transformer (NMT)** Yoo et al. (2024), and **MuseTok** Huang et al. (2026). Their causal decoding naturally supports flexible music extension while continuously incorporating the generated history.

For accompaniment generation, we compare DID with recent conditional generation methods, including **Polyffusion** Min et al. (2023), **WholeSongGen** Wang et al. (2024), **FGG** Zhu et al. (2025), and **D3PIA** Choi et al. (2026). These methods largely build on explicit two-dimensional musical representations, which facilitate fine-grained conditioning over corresponding temporal regions. Among the more recent baselines, FGG also supports chord-conditioned generation, so we further compare DID with FGG on this task.

4.4 Performance Analysis

For music continuation, the 1D autoregressive baselines remain strong across both datasets, especially on SimPCE, SimSC, and PD, as shown in Table 1. DID achieves the highest SimGrv on both POP909 and POP1K7, together with the best SimChr on POP909 and a tied-best result on POP1K7. These results suggest that the 2D time–note organization helps preserve rhythmic structure while the compressed global history retains useful chromatic context. DID also remains competitive on the other metrics, leading to the highest average score on both datasets. We next examine whether this consistency is maintained as the continuation grows longer.

To examine long-range continuation, we compare DID with CP and NMT, which support autoregressive generation, and use the common subset of test examples for which all three methods generate beyond 80 bars. REMI is excluded because its non-compound representation produces longer token

sequences for the same musical duration, making comparison over the same long generation span less suitable.

As shown in Figure 3, DID maintains a nearly stable SimGrv as the continuation grows, whereas CP and NMT show clearer decreases. SimChr also remains relatively stable for DID and stays close to NMT over longer continuations, while CP declines more consistently with length. These trends suggest that DID can preserve rhythmic structure over extended generation while maintaining competitive chromatic consistency. This behavior is consistent with its history design, where compressed long-range summaries and full-resolution information from the preceding bar jointly condition each new bar.

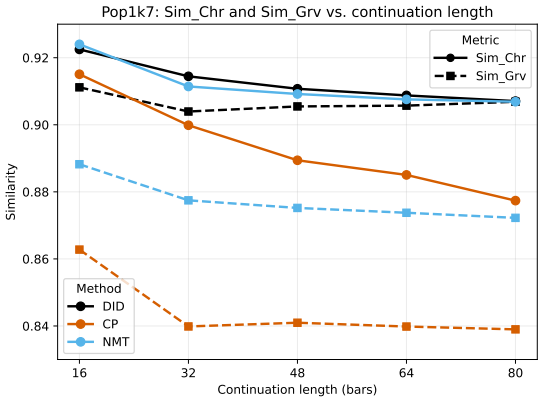

Figure 3: SimChr and SimGrv at increasing continuation lengths on POP1k7.

For accompaniment generation, Table 2 shows that the diffusion-based baselines show strong performance under fine-grained conditions. FGG further uses onset information as an additional input, giving it an advantage in rhythmic alignment; among methods without explicit onset conditioning, DID achieves the highest OS, suggesting that its explicit time–note organization can also support rhythmic consistency. On harmonic metrics, DID achieves the highest CA and ranks second on CS. This difference may partly reflect D3PIA's iterative sampling refinement, whereas DID follows causal decoding without repeated global correction. The chord-conditioned results in Table 4 further show consistent gains over FGG, supporting the effectiveness of the proposed conditional decoding framework.

Table 3 shows that DID generally receives higher subjective ratings in both generation settings. For continuation, the clearest gain appears in Rhythmic Consistency. In accompaniment generation, DID also performs favorably in Rhythmic Consistency and Accompaniment-Pattern Stability, suggesting that its 2D decoding process may help maintain rhythmic and structural continuity. DID further receives favorable Harmonic Consistency ratings, broadly agreeing with the harmonic trends observed in the objective evaluation.

Table 4: Chord-conditioned generation results.

Method	CA↑	CS↑	Avg↑
FGG	74.45	68.32	71.39
DID (Ours)	**83.37**	**75.61**	**79.49**

4.5 Ablation Analysis

To comprehensively assess the effectiveness of the key components in DID, we conduct ablation studies on the accompaniment generation task with the following variants. (1) **DID (w/o Plan)** removes the Harmonic Plan events and directly generates target notes from the given conditions. (2) **DID (w/o Global Cond.)** removes the global condition $\mathbf{G}_t$ obtained from all bar-level condition summaries, while retaining the task-specific condition events of the current bar. The full DID model incorporates both components.

Table 5: Ablation results on accompaniment generation on POP909. Avg. denotes the arithmetic mean of OS, CA, and CS.

Variant	Params (M)	Train Time (h)	OS↑	CA↑	CS↑	Avg.↑
DID (w/o Harmonic Plan)	34.25M	0.92h	84.41	81.19	77.60	81.07
DID (w/o Global Cond.)	**30.32M**	**0.81h**	84.85	82.93	77.89	81.89
DID (Full)	34.31M	1.08h	**85.00**	**83.52**	**79.30**	**82.61**

Table 5 shows that both conditional components contribute to generation quality. Removing the Harmonic Plan degrades the harmonic metrics, suggesting that the Plan may help transfer harmonic

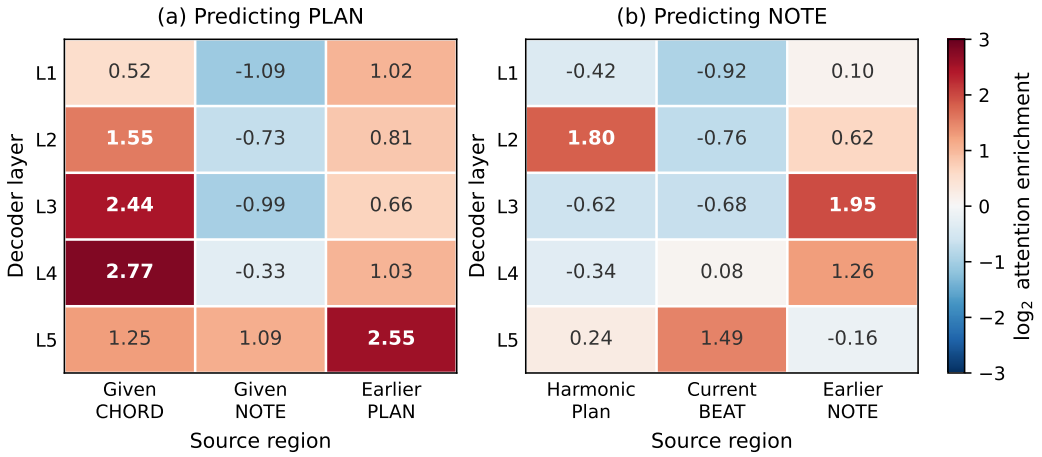

Figure 4: Attention enrichment across note-decoder layers for accompaniment generation. Positive values indicate attention above the uniform baseline.

information from the given conditions to subsequent note generation. Removing the global condition also lowers performance, suggesting that the full conditioning sequence provides useful context beyond the current-bar conditions and preceding musical history. Together, these results support the complementary roles of the Harmonic Plan and global condition in DID. We next examine how harmonic information is routed through the Harmonic Plan.

4.6 Information Flow through the Harmonic Plan

The ablation results show that the Harmonic Plan contributes to harmonic consistency. To trace this information flow, we analyze self-attention in the note decoder under the POP909 accompaniment setting. Attention is summarized by predicted event Type and source region, and reported as enrichment relative to uniform attention, as shown in Figure 4.

Figure 4 shows a clear CHORD–PLAN–NOTE information flow. From the second to the fourth decoder layer, the Plan attends strongly to the current-bar CHORD, indicating that it draws harmonic information from the given condition. This information then reaches NOTE prediction, where the Plan receives strong attention in the second layer. In later layers, attention shifts toward current BEAT and NOTE events, placing more weight on rhythm and note details. The Plan thus links the given harmony to note generation without directly determining the final notes. Together with the ablation results, this pattern supports the role of the Harmonic Plan in improving harmonic consistency while leaving note realization to the autoregressive decoder.

5 Conclusion

We presented DID, a 2D autoregressive framework that organizes symbolic music generation along the time and note dimensions. DID adopts a Decoder-in-Decoder architecture that combines Global History and Local History to connect cross-bar context with within-bar generation. For conditional generation, we introduce Harmonic Plan events to improve harmonic consistency by providing intermediate guidance between the given conditions and generated notes. Attention analysis further shows a clear CHORD–PLAN–NOTE information flow. Experiments on music continuation, chord-conditioned generation, and accompaniment generation show that DID achieves strong rhythmic and harmonic consistency across objective and subjective evaluations, while the ablation results support the proposed conditional components. Overall, these results suggest that explicitly modeling two-dimensional structure within an autoregressive framework may be a useful direction for symbolic music generation with flexible causal decoding across tasks.

AI USE STATEMENT

During the experimental stage, generative AI tools were used to assist with specific coding issues and to discuss parts of the experimental process. During manuscript preparation, they were used to improve language and organization and to help with literature search. The authors remained responsible for the research design, experimental decisions, interpretation of the results, and the final manuscript.

REFERENCES

Eunjin Choi, Hounsu Kim, Hayeon Bang, Taegyun Kwon, and Juhan Nam. D3pia: A discrete denoising diffusion model for piano accompaniment generation from lead sheet. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 15082–15086. IEEE, 2026.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

Nathan Fradet, Nicolas Gutowski, Fabien Chhel, and Jean-Pierre Briot. Byte pair encoding for symbolic music. In Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 2001–2020, 2023.

Zixun Guo, Jaeyong Kang, and Dorien Herremans. A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 5070–5077, 2023.

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. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 178–186, 2021.

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. arXiv preprint arXiv:1809.04281, 2018.

Jingyue Huang, Zachary Novack, Phillip Long, Yupeng Hou, Ke Chen, Taylor Berg-Kirkpatrick, and Julian McAuley. Musetok: Symbolic music tokenization for generation and semantic understanding. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3956–3960. IEEE, 2026.

Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In Proceedings of the 28th ACM international conference on multimedia, pp. 1180–1188, 2020.

Yujia Huang, Adishree Ghatare, Yuanzhe Liu, Ziniu Hu, Qinsheng Zhang, Chandramouli S Sastry, Siddharth Gururani, Sageev Oore, and Yisong Yue. Symbolic music generation with non-differentiable rule guided diffusion. arXiv preprint arXiv:2402.14285, 2024.

Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Kejun Zhang, Xiangyang Li, Tao Qin, and Tie-Yan Liu. Telemelody: Lyric-to-melody generation with a template-based two-stage method. arXiv preprint arXiv:2109.09617, 2021.

Shuyu Li, Dooho Choi, and Yunsick Sung. Midilm: A dual-path model for controllable text-to-midi generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 23160–23168, 2026.

Jiafeng Liu, Yuanliang Dong, Zehua Cheng, Xinran Zhang, Xiaobing Li, Feng Yu, and Maosong Sun. Symphony generation with permutation invariant language model. arXiv preprint arXiv:2205.05448, 2022a.

Nayu Liu, Wenjing Han, Guangcan Liu, Da Peng, Ran Zhang, Xiaorui Wang, and Huabin Ruan. ChipSong: A controllable lyric generation system for Chinese popular song. 2022b.

Ang Lv, Xu Tan, Peiling Lu, Wei Ye, Shikun Zhang, Jiang Bian, and Rui Yan. Getmusic: Generating any music tracks with a unified representation and diffusion framework. arXiv preprint arXiv:2305.10841, 2023.

Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. arXiv preprint arXiv:2307.10304, 2023.

Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. arXiv preprint arXiv:2103.16091, 2021.

Matthias Plasser, Silvan Peter, and Gerhard Widmer. Discrete diffusion probabilistic models for symbolic music generation. arXiv preprint arXiv:2305.09489, 2023.

Yang Qin, Huiming Xie, Shuxue Ding, Benying Tan, Yujie Li, Bin Zhao, and Mao Ye. Bar transformer: a hierarchical model for learning long-term structure and generating impressive pop music. Applied Intelligence, 53(9):10130–10148, 2023. doi: 10.1007/s10489-022-04049-3.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In International conference on machine learning, pp. 4364–4373. PMLR, 2018.

Keito Sawada, Wen-Chin Huang, and Tomoki Toda. Hierarchical symbolic music generation with variational autoencoder-based bar-wise feature sequences. In 2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 299–304. IEEE, 2025. doi: 10.1109/APSIPAASC65261.2025.11249414.

Hongju Su, Ke Li, Lan Yang, Honggang Zhang, and Yi-Zhe Song. Amadeus: Autoregressive model with bidirectional attribute modelling for symbolic music. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 40910–40928, 2026.

Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. Figaro: Controllable music generation using learned and expert features. In The Eleventh International Conference on Learning Representations, 2023.

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

Ziyu Wang, Lejun Min, and Gus Xia. Whole-song hierarchical generation of symbolic music using cascaded diffusion models. In International Conference on Learning Representations, volume 2024, pp. 13388–13408, 2024.

HaeJun Yoo, Hao-Wen Dong, Jongmin Jung, and Dasaem Jeong. Nested music transformer: Sequentially decoding compound tokens in symbolic music and audio generation. arXiv preprint arXiv:2408.01180, 2024.

Botao Yu, Peiling Lu, Rui Wang, Wei Hu, Xu Tan, Wei Ye, Shikun Zhang, Tao Qin, and Tie-Yan Liu. Museformer: Transformer with fine-and coarse-grained attention for music generation. Advances in neural information processing systems, 35:1376–1388, 2022.

Tingyu Zhu, Haoyu Liu, Ziyu Wang, Zhimin Jiang, and Zeyu Zheng. Efficient fine-grained guidance for diffusion model based symbolic music generation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 79938–79962. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhu25m.html.

A LIMITATIONS

DID provides a two-dimensional autoregressive framework that separates cross-bar history modeling from within-bar event generation, allowing the model to maintain long-range musical context while preserving detailed information from the previous bar. The current history design supports stable rhythmic and harmonic structure over extended generation, but richer ways of representing and passing earlier musical context remain worth exploring. Future work may further improve the use of long-range history so that more musical cues can be retained across distant bars, while keeping the balance between structural consistency and local variation. In addition, our experiments mainly focus on pop-piano settings, and the effectiveness of DID for richer multi-instrument arrangements remains to be further studied.

B EVALUATION METRIC DETAILS

B.1 OBJECTIVE METRICS

For music continuation, **Chroma Similarity (SimChr)** Huang et al. (2026) measures chromatic similarity using bar-wise chroma vectors, while **Groove Similarity (SimGrv)** Huang et al. (2026) compares bar-level onset patterns. For each generated bar, we take its highest similarity to a reference bar and average the scores over the output. **Pitch-Class Entropy Similarity (SimPCE)** and **Scale Consistency Similarity (SimSC)** Su et al. (2026) compare pitch-class entropy and scale consistency with the reference, respectively. **Pitch Distribution Similarity (PD)** Ju et al. (2021) measures the overlap between normalized pitch distributions. **Average Score (Avg)** is the arithmetic mean of these five metrics.

For conditional generation, **Chord Accuracy (CA)**Choi et al. (2026) and **Chord Similarity (CS)**Choi et al. (2026) evaluate harmonic agreement at different levels. CA first recognizes beat-level chords from the generated and reference MIDI files, reduces each chord to its root and major/minor quality, and reports the proportion of matching beats. CS provides a softer progression-level comparison by encoding corresponding two-bar chord sequences with a pretrained chord encoder and averaging their cosine similarities. For accompaniment generation, **Onset Similarity (OS)** additionally measures rhythmic agreement and is defined as $\mathrm{OS} = 1 - \mathrm{OXD}$, where OXD denotes the Onset XOR Distance. Higher values indicate better performance for all metrics.

B.2 SUBJECTIVE METRICS

For subjective evaluation, we conduct a blind listening test with five evaluators possessing substantial expertise in music production, arrangement, instrumental performance, and music education. All evaluators also have experience with AI-based music systems. We select 10 POP909 pieces for each of the two tasks: 32-bar music continuation and accompaniment generation. For each source piece, the outputs generated by the five systems are presented together as one evaluation group in anonymized order. Evaluators assign separate ratings to all five versions within a group before proceeding to the next group, rather than rating a globally shuffled collection of clips. Thus, each task contains 10 groups and 50 generated clips, resulting in 100 clips in total. Each evaluator rates every clip on a five-point scale (1–5) according to the criteria described below, while the identity of the model used to generate each clip remains concealed throughout the evaluation.

For music continuation, **Coherence** measures how naturally the continuation connects to the four-bar prompt and preserves its musical style. **Rhythmic Consistency** evaluates the continuity and stability of rhythmic patterns, density, and tempo. **Harmonic Consistency** assesses whether the generated pitches and harmonic relationships remain natural and coherent. **Overall Quality** evaluates the overall naturalness, fluency, musicality, and listening quality.

For accompaniment generation, **Coherence** measures how naturally the accompaniment coordinates with and supports the given melody. **Harmonic Consistency** evaluates whether the accompaniment conforms to the given chord progression without noticeable harmonic conflicts. **Rhythmic Consistency** assesses the continuity and stability of rhythmic patterns, density, and groove. **Accompaniment-Pattern Stability** evaluates whether accompaniment patterns and textures remain

coherent with natural repetition and variation. **Overall Quality** evaluates the overall naturalness, fluency, coordination, and suitability of the accompaniment for the given melody.

C Implementation Details

Architecture Configuration. DID employs separate Transformer decoders for the time and note dimensions, each with five layers, a hidden size of 384, and six attention heads. The bar encoder uses two layers with six attention heads. For chord-conditioned and accompaniment generation, DID additionally uses a two-layer condition encoder with six attention heads and Harmonic Plan events. The remaining architecture and task-specific configurations are summarized in Table 6.

Training Configuration. We train a separate DID model for each task, with the main configurations summarized in Table 6. The time and note decoders use the same architecture across all tasks, while chord-conditioned and accompaniment generation additionally use the condition encoder and Harmonic Plan. All models are trained on a single NVIDIA RTX 5090 using AdamW with a batch size of 16. For continuation, POP1k7 is trained for 48 epochs because each epoch contains more optimization steps than POP909; this corresponds to approximately 6.8k updates, close to the 7.2k updates obtained from 150 epochs on POP909. During training, we apply online pitch transposition and shift chord roots by the same interval. For the conditional tasks, Harmonic Plan events are constructed after augmentation to remain consistent with the transposed notes and chords.

Table 6: Training configuration of DID for the four experimental settings.

Configuration	POP909 Cont.	POP1k7 Cont.	Chord Cond.	Accomp.
Time decoder layers	5	5	5	5
Time decoder hidden size	384	384	384	384
Time decoder heads	6	6	6	6
Note decoder layers	5	5	5	5
Note decoder hidden size	384	384	384	384
Note decoder heads	6	6	6	6
FFN size	1536	1536	1536	1536
Bar encoder layers	2	2	2	2
Bar encoder heads	6	6	6	6
Condition encoder layers	–	–	2	2
Condition encoder heads	–	–	6	6
Global-history tokens Q	4	4	4	4
Harmonic Plan	No	No	Yes	Yes
Epochs	150	48	150	150
Batch size	16	16	16	16
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Weight decay	0.1	0.1	0.1	0.1
LR schedule	Cosine	Cosine	Cosine	Cosine
Warm-up ratio	0.1	0.1	0.1	0.1
Precision	BF16	BF16	BF16	BF16
Gradient clipping	1.0	1.0	1.0	1.0

Inference Configuration. We use grammar-constrained autoregressive sampling for all tasks, with the configurations summarized in Table 7. The same decoding configuration is used for both continuation datasets, and another shared configuration is used for chord-conditioned and accompaniment generation. These settings are fixed across all test examples and are not adjusted for individual samples or evaluation metrics.

For continuation, standard 1D autoregressive models directly condition on the complete preceding token sequence, whereas DID provides the note decoder with compressed long-range history together with the complete NOTE events of the previous bar. This compression may lead to relatively flatter predictive distributions for some structural fields. We therefore use more conservative sampling for Type, Beat, and Track ($p = 0.5$) and Chord ($p = 0.7$), restricting sampling to higher-

Table 7: Inference configuration used for evaluation. "Conditional" denotes chord-conditioned and accompaniment generation.

Configuration	Continuation	Conditional
General		
Prompt bars	4	0
Generated bars	28	32
Temperature	1.0	1.0
Structure and Control		
Type top-p	0.5	0
Beat top-p	0.5	0.5
Track top-p	0.5	0.5
Tempo top-p	0	0
Plan-Bass top-p	–	0
Plan-Set top-p	–	0
Harmony		
Chord top-p	0.7	0.5
Note Realization		
Pitch top-p	0.9	0.5
Duration top-p	0.9	0.5
Velocity top-p	0.9	0.5

probability candidates. Pitch, Duration, and Velocity instead use $p = 0.9$ to preserve variation in open-ended continuation.

For conditional generation, the provided musical conditions already constrain the generation space. Consistent with the task formulation of D3PIA Choi et al. (2026), which emphasizes adherence to melody and chord conditions together with structural coherence, we prioritize condition fidelity and consistency during decoding. Type and Harmonic Plan fields are therefore decoded deterministically, while Chord, Beat, Track, Pitch, Duration, and Velocity use $p = 0.5$. This narrower sampling still retains stochasticity in the generated realization while reducing deviations from the given conditions.

Tempo is decoded deterministically in all settings. Since tempo typically changes infrequently within a piece, repeatedly sampling tempo events may introduce unnecessary fluctuations. Moreover, the accompaniment baselines are evaluated under a unified tempo setting. Deterministic tempo decoding therefore avoids introducing tempo variation unrelated to the generation task or model comparison.

D Relation to Hierarchical and Nested Autoregressive Modeling

Several prior works incorporate hierarchical or nested structures into symbolic music generation, but organize autoregressive decoding differently from DID. Museformer Yu et al. (2022) inserts bar-level summary tokens to support fine- and coarse-grained attention, while the musical events are still organized as a single token sequence. NMT Yoo et al. (2024) introduces nested autoregressive decoding, where the main decoder models the sequence of compound tokens and the sub-decoder models the attributes within each token; thus, its additional autoregressive dimension concerns intra-token attributes rather than the score-level time–note organization. Bar Transformer Qin et al. (2023) introduces a note-to-bar hierarchy through note-level and bar-level representations, rather than formulating bar progression and within-bar event generation as two autoregressive axes. Sawada et al. Sawada et al. (2025) adopt a staged hierarchical process: a sequence of bar-wise features is first generated autoregressively, after which the individual bars are decoded from these features.

DID focuses on a different organization of causal decoding. It explicitly retains bar progression and within-bar events as the time and note dimensions, respectively, and alternates autoregressive decoding between them: the events of the current bar are generated before advancing to the next bar

along the time dimension. Thus, the distinction is not the use of hierarchy itself, but that DID retains the score-level time–note organization as two interacting autoregressive dimensions throughout generation.

E Understanding DID through Attention

We analyze self-attention in the note decoder on POP909 to study the mechanisms of 2D decoding and the role of the Harmonic Plan. We examine these two components through two generation settings. In music continuation, we study how the decoder combines global history from earlier bars with the complete note events of the previous bar. In accompaniment generation, we trace how information flows from the given musical conditions through the Harmonic Plan to the generated accompaniment notes. To make the two analyses directly comparable, we use the same ten test pieces and the same five target bars from each piece in both settings, resulting in 50 matched positions per setting.

E.1 Layer-wise History Routing in 2D Decoding

As described in the Methodology section, DID performs 2D decoding by modeling musical history along the time dimension and using this history to generate events along the note dimension. To examine how this history is used across the decoding process, we analyze self-attention in the five note decoder layers under the POP909 continuation setting. We group the attention weights by their source and sum the weights within each group. Figure 5(a) first compares Cross-bar History with Current-bar History, where Cross-bar History combines the four Global History tokens representing bars up to $t-2$ with all NOTE events from bar $t-1$ (Local History). Figure 5(b) further divides Cross-bar History into Global History and Local History to reveal how the two sources are used across decoder layers. For each group, we average the summed attention over all prediction queries and six attention heads.

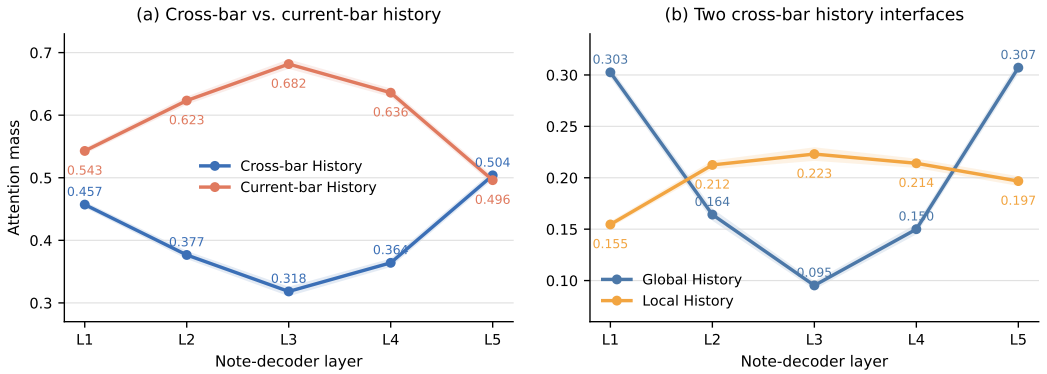

Figure 5: Layer-wise history attention mass in the POP909 continuation setting: (a) Cross-bar History versus Current-bar History; (b) Global History versus Local History.

Figure 5(a) shows a clear U-shaped pattern for Cross-bar History, with higher attention in the early (L1) and final (L5) layers and lower attention in the middle layers (L2–L4). Current-bar History follows the opposite trend, receiving more attention in the middle layers. This layer-wise shift suggests a three-stage decoding pattern: the early layers first retrieve context from previous bars, the middle layers focus more on organizing events within the current bar, and the final layer reconnects the current generation with earlier musical context. By alternating between cross-bar context and current-bar events, this routing pattern may help DID maintain rhythmic organization within each bar while improving continuity across bars.

Figure 5(b) further shows that Global History follows a pronounced U-shaped pattern, whereas Local History exhibits a weaker opposite trend. This separation suggests that Global History contributes more strongly to long-range structural context in the early and final layers, while Local History is used more heavily in the middle layers to provide exact note-level information from the previous

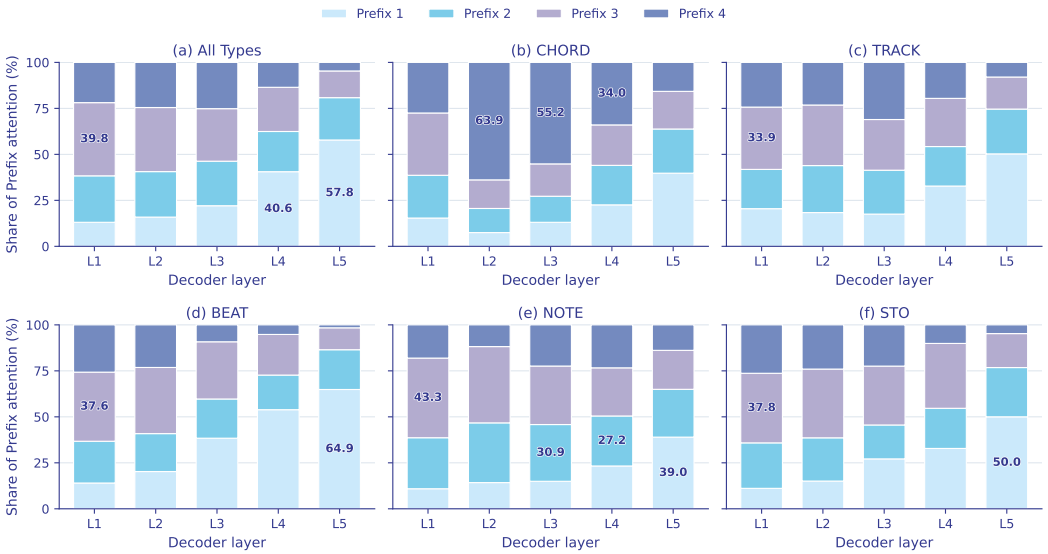

Figure 6: Mean distribution of Prefix attention across the four Global History tokens in the POP909 continuation setting. Each 100% stacked bar shows, for a given decoder layer and event Type, the mean fraction of total Prefix attention assigned to Prefix 1–Prefix 4. Labels indicate the key values discussed in the text.

bar. These complementary roles may further support long-range musical coherence while preserving local harmonic and melodic consistency between neighboring bars.

Figure 5 treats the four Global History tokens as a single group, so it cannot reveal whether the individual tokens take on different roles. To examine this possibility, Figure 6 summarizes the attention assigned to Prefix 1–Prefix 4 across all analyzed cases using 100% stacked bars. For each decoder layer and event Type, each bar shows the mean share of total Prefix attention assigned to the four Global History tokens.

Figure 6 reveals soft roles that vary across layers and event Types. Prefix 3 is most prominent in the early stage: its overall share decreases from 39.8% in L1 to 14.5% in L5, while remaining high across all event Types in L1. This pattern suggests that Prefix 3 provides a shared initial context for subsequent decoding. Prefix 4 has a smaller overall share, but its attention reaches 63.9% and 55.2% for CHORD in L2 and L3, respectively, indicating a stronger association with harmonic information. Prefix 2 shows a different preference: its share for NOTE reaches 30.9% in L3 and 27.2% in L4, both higher than its overall share in the corresponding layers, suggesting a greater role in providing note-level details. In contrast, Prefix 1 becomes increasingly important toward the final layer, rising from 13.1% in L1 to 57.8% in L5 and remaining high for BEAT, NOTE, and STO in L5. This pattern suggests that Prefix 1 helps bring together rhythmic, note-level, and bar-ending information before the final predictions. Since the MLP does not assign predefined musical meanings to these four tokens, these patterns indicate learned soft specialization rather than fixed musical labels. This flexible division of roles allows different layers and event Types to draw on different aspects of the long-range history, which may help DID preserve global musical coherence while adapting historical information to the rhythmic, harmonic, and note-level needs of each prediction. For a more intuitive view of these learned preferences, Figure 7 provides a representative example of the token-level attention patterns across the five note decoder layers.

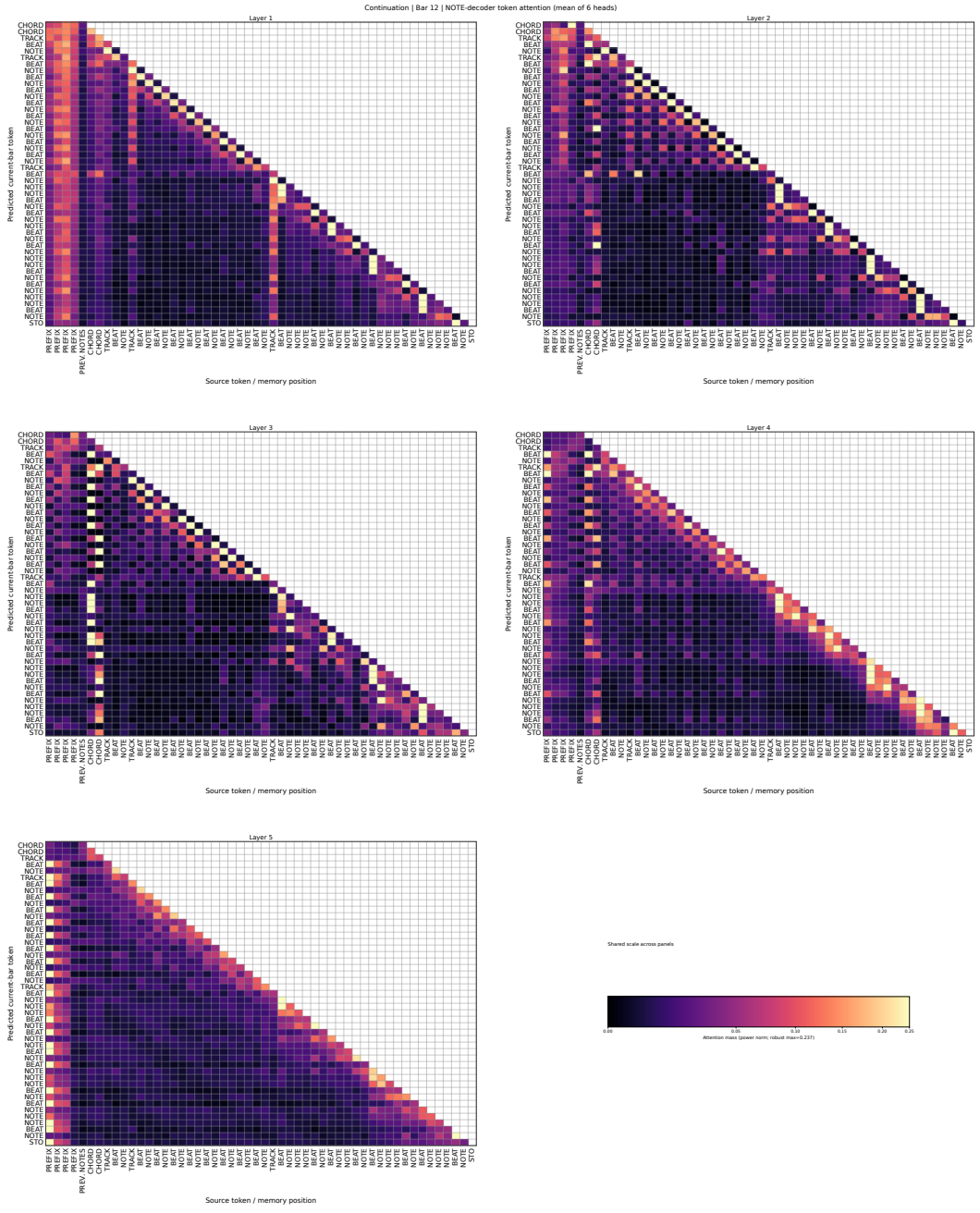

Figure 7: Token-level attention maps across the five note decoder layers for a representative POP909 continuation example. Rows denote predicted event Types, while columns include the four Global History tokens, Local History, and events already generated in the current bar. Attention is averaged over six heads, with the same color scale used across all layers.