U-MUST: UNIFIED CROSS-MODAL TRANSLATION OF SCORE IMAGES, SYMBOLIC MUSIC, AND PERFORMANCE AUDIO

Jongmin Jung[1] **Dongmin Kim**[1] **Sihun Lee**[1] **Seola Cho**[1]
Hyungjoon Soh[2] **Irmak Bukey**[3] **Chris Donahue**[3] **Dasaem Jeong**[1]
[1]Sogang University, Seoul, South Korea [2]Seoul National University, Seoul, South Korea
[3]Carnegie Mellon University, Pittsburgh, USA
jongmin@sogang.ac.kr, jdasaem@sogang.ac.kr

ABSTRACT

Traditional Music Information Retrieval (MIR) tasks like Optical Music Recognition (OMR) and Automatic Music Transcription (AMT) typically rely on specialized, single-task models. We challenge this paradigm by proposing a unified framework that trains a single Transformer on multiple cross-modal translation tasks simultaneously. Our approach is enabled by two key contributions: a novel large-scale dataset (YTSV) with over 1,300 hours of paired score-image and audio data, and a unified tokenization scheme that converts all music modalities into a common sequence format. Experiments show our multitask model significantly outperforms specialized baselines, reducing the OMR symbol error rate from 24.58% to a state-of-the-art 13.67%. Most notably, our framework achieves the first successful end-to-end generation of audio directly from a score image, marking a significant breakthrough in cross-modal music understanding and generation.

1. INTRODUCTION

Music can be represented in various forms, including score images, machine-readable formats like MusicXML, performance data such as MIDI, and audio recordings. Music Information Retrieval (MIR) has long focused on translating between these modalities through tasks like Automatic Music Transcription (AMT) and Optical Music Recognition (OMR), as shown in Figure 1. Historically, these tasks have been addressed with specialized models and datasets [1–4].

In this paper, we challenge this fragmented approach by proposing a unified framework that learns multiple translation tasks simultaneously within a single model. Our work is inspired by how human musicians perform sight-reading: directly translating a score image into an expressive audio performance without an explicit intermediate symbolic step. To this end, we introduce the novel task of direct score-image-to-performance-audio generation, a challenging feat that requires joint mastery of recognition and synthesis.

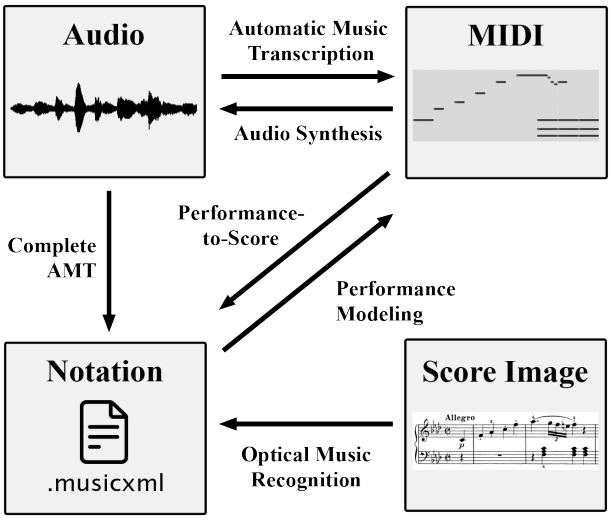


Figure 1: Conventional cross-modal conversion tasks in music information retrieval research.

Our unified approach is enabled by two key contributions. First, we introduce the YouTube Score Video (YTSV) dataset, a new large-scale collection of over 1,300 hours of paired score images and performance audio. Second, we employ a unified tokenization scheme that converts all modalities into a common sequence format, allowing a single Transformer model to treat diverse translation tasks as a sequence-to-sequence problem. Our experiments show that this multitask learning paradigm not only enables novel cross-modal generation but also enhances the performance of established subtasks like OMR, demonstrating a synergistic relationship between modalities.

2. PROBLEM FORMULATION AND RELATED WORK

We consider four primary music modalities: **Score Image** (raw sheet music pixels), **Symbolic Notation** (semantic information, as in MusicXML), **MIDI** (expressive performance timings), and **Audio** (recorded sound). As illus-

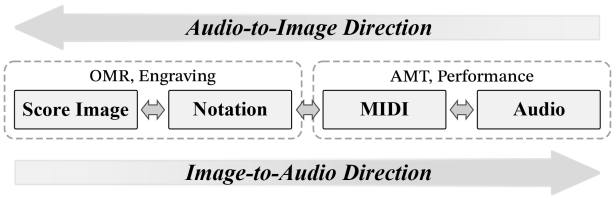


Figure 2: The four modalities of music representation used in this paper.

trated in Figure 2, these modalities exist on a spectrum. We group translation tasks into two directions: *Image-to-Audio* (**I2A**) and *Audio-to-Image* (**A2I**).

Recent work in AMT and OMR has shifted towards end-to-end, sequence-to-sequence models using Transformers [5–8]. These advances, however, still rely on task-specific models and are often constrained by the limited availability of paired training data, particularly for OMR [9]. Our work builds on this trend but extends it by unifying these disparate tasks into a single multitask framework, addressing data scarcity with our new YTSV dataset and exploring the synergistic potential of joint training.

3. METHODS

Our approach hinges on unifying multimodal music translation as a sequence-to-sequence task. This is achieved through a common tokenization framework and a multitask Transformer architecture.

3.1 Multimodal and Multitask Approaches

Inspired by unified models in other domains [10, 11], we train a single model on a diverse set of music translation tasks. Unlike previous works, we tackle novel challenges like direct score-image-to-audio generation and address data scarcity by incorporating our large-scale YTSV dataset (Table 1). We demonstrate that even without explicit note-level annotations, jointly training on image-audio pairs improves performance on related tasks like OMR and AMT by allowing the model to learn shared musical structures.

3.2 Tokenization

To create a unified input format, we convert all modalities into sequences of discrete tokens.

- **Image and Audio Tokens:** Continuous data like score images and audio are tokenized using a Residual Quantized VAE (RQVAE) [12] and a Descript Audio Codec (DAC) [13], respectively. These models discretize the raw data into compact token sequences. Further details on preprocessing and augmentation are in Appendix D.

- **Linearized MusicXML (LMX):** We use the concise, linearized MusicXML format proposed by [8] for representing symbolic notation.

Subset	Modalities				N	H
	Img	MXL	MIDI	Aud		
YTSV	√	-	-	√	433,920	1,341
GrandStaff	√	√	-	-	7,661	*23
OLiMPiC	√	√	-	-	17,945	*47
MusicNet	-	-	△	√	330	33
MAESTRO	-	-	√	√	1,276	199
SLakh	-	-	√	√	2,100	145
BPSD	√	√	△	√	32	14

Table 1: Data Distribution of Combined Datasets with Aligned Modalities.

Category	Videos	Segments	Duration (hrs)
Solo Piano	9,052	232,029	762.34
Accompanied Solo	912	47,373	141.83
String Quartet	594	48,470	138.48
Others (Chamber)	1,659	106,048	298.65
Total	12,217	433,920	1,341

Table 2: Data distribution of the YouTube Score Video dataset after filtering

- **MIDI-Like Tokens:** We adopt the event-based MIDI tokenization scheme from YourMT3+ [14], which quantizes performance data into discrete events.

3.3 Model Architecture

As depicted in Figure 3, we use a single encoder-decoder Transformer architecture for all tasks within a given direction (I2A or A2I). All modalities are mapped to a unified token space. To handle different tasks, we provide a target modality embedding as a hint to the model. For image and audio, which use multiple token streams (codebooks), we employ a sub-decoder module [11] to generate tokens for each codebook in parallel. The model is trained with a standard cross-entropy loss. Detailed mathematical formulations are available in Appendix A.

4. YOUTUBE SCORE VIDEO DATASET

A major bottleneck in multimodal music research is the lack of large-scale, aligned data. To address this, we introduce the YouTube Score Video (YTSV) dataset. We collected 12,217 score-following videos, where sheet music slides are synchronized with a performance audio, as seen in Figure 4. This provides a rich source of weakly aligned image-audio pairs.

After an extensive data processing pipeline—including slide segmentation, system cropping using a fine-tuned YOLOv8 model (Figure 9), and rigorous filtering—we curated a dataset of 433,920 segments, totaling over 1,300 hours (Table 2). While these pairs lack symbolic data, they are the cornerstone of our multitask training, enabling the model to learn direct image-to-audio translation. Full details on data collection and processing are in Appendix C.

5. EXPERIMENTS

We structure our experiments around the I2A and A2I directions, training two separate models. The complete train-

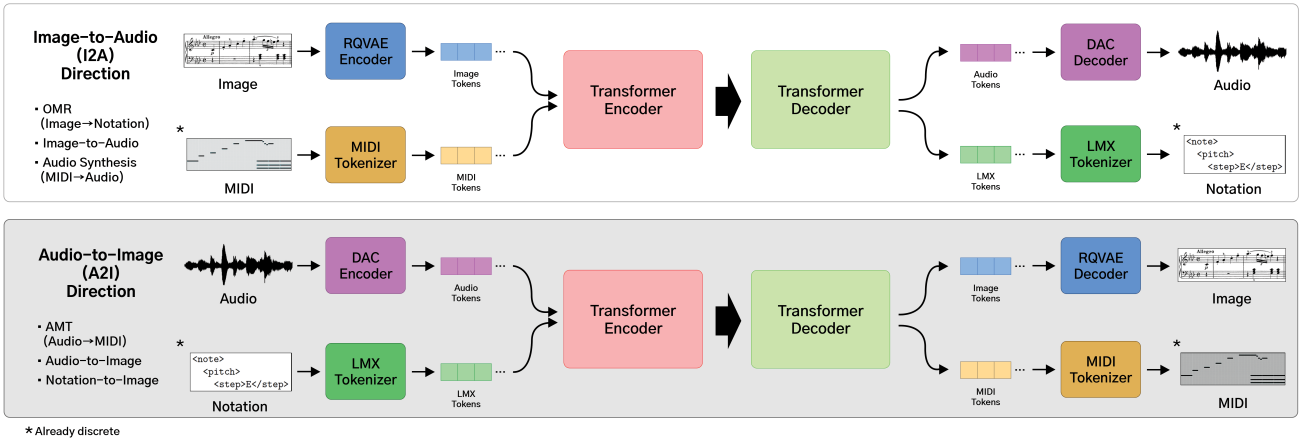


Figure 3: Overview of our proposed unified multimodal translation framework. We employ a single Transformer encoder-decoder model for each direction—one for *Image-to-Audio direction* (I2A) tasks and another for *Audio-to-Image direction* (A2I) tasks. Each model jointly handles multiple translation tasks. All modalities are discretised into token sequences, enabling end-to-end, multitask training entirely at the token level. Note that we train separate models for I2A and A2I directions; the two directions do not share weights.

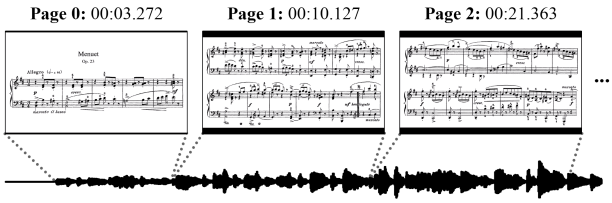


Figure 4: An example from one of the videos collected for the YouTube Score Video dataset. Slides of sheet music are aligned to the corresponding points in audio.

ing corpus combines our YTSV dataset with several public datasets like GrandStaff, OLiMPiC, MusicNet, MAESTRO, SLakh, and BPSD, as summarized in Table 1.

5.1 Implementation

Our Transformer models consist of 12 encoder and 12 decoder layers. To stabilize training, we employ a curriculum learning strategy, starting with data-rich tasks (e.g., OMR or AMT) before introducing the more challenging direct image-audio translation tasks. Model configurations and the curriculum schedule are detailed in Table 11. Further implementation details are in Appendix B.

5.2 Evaluation Metrics

We use modality-appropriate metrics for each task.

- **OMR:** Symbol Error Rate (SER) [8] on LMX tokens.
- **AMT:** Note-F_1 score from `mir_eval` [15].
- **Image-to-Audio:** We first transcribe the generated audio to MIDI with Onsets and Frames [16] and compute note onset F_1-score against the reference after dynamic time warping (DTW) alignment. We also report Fréchet Audio Distance (FAD) [17] for perceptual quality.

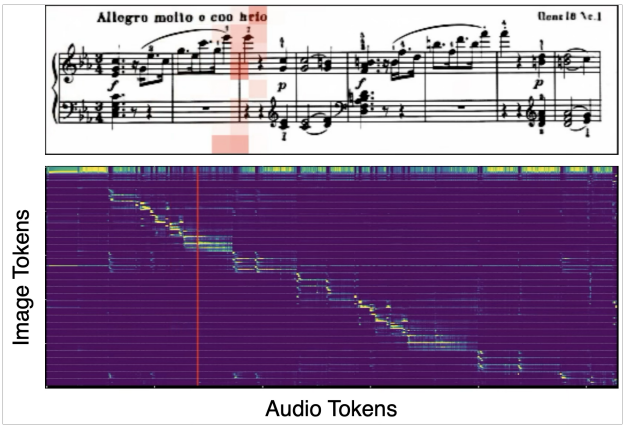


Figure 5: Attention patterns from a selected transformer head showing direct correlation between image token regions (top) and generated audio tokens (bottom).

- **Audio-to-Image:** We perform OMR on the generated image and compute the Earth Mover's Distance (EMD) [18] between the predicted and ground truth LMX token distributions (for pitch and duration).

Details on the DTW-F_1 and EMD calculations are in Appendix E.

6. RESULTS

6.1 Image-to-Audio Generation

As shown in Table 3, our unified model successfully generates audio from score images. The model trained only on the I2A task performs poorly. However, adding OMR and MIDI-to-audio synthesis tasks dramatically improves both note accuracy (F_1 score) and audio quality (FAD). This confirms that learning related subtasks is crucial. Our direct end-to-end approach achieves comparable note accuracy and superior audio quality (lower FAD) to a multi-stage pipeline (OMR $\rightarrow$ MIDI $\rightarrow$ Audio), which often suffers from propagated OMR errors that are musically

Method	Metric	F_1 Score ↑						FAD ↓	
	Dataset	BPSD			YTSV-T11			BPSD	YTSV-T11
	Onset Tolerance (ms)	50	100	200	50	100	200	–	–
Direct I2A: YTSV-P (Image-to-Audio Only)		23.49	34.51	44.15	27.05	43.32	53.02	0.422	0.317
Direct I2A: OMR + Image-to-Audio		48.67	64.30	74.01	51.60	67.92	75.98	0.098	0.056
Direct I2A: OMR + Image-to-Audio + MIDI-to-Audio		48.36	64.63	74.92	52.66	**68.45**	**76.24**	**0.081**	**0.055**
Multi-stage: OMR + Image-to-Audio + MIDI-to-Audio		**50.91**	**70.40**	**79.96**	–	–	–	0.137	–
Multi-stage: Zeus → VirtuosoNet → MSD		45.52	59.35	69.36	–	–	–	0.330	–
DAC Reconstruction (Upper-bound)		68.83	82.39	87.47	82.28	86.43	88.76	0.050	0.035

Table 3: Image-to-audio accuracy reported as onset F_1 (↑) and FAD (↓). The model jointly trained on all three I2A tasks achieves the best direct generation results.

Method	EMD ↓	
	Pitch	Duration
Audio-to-Image Only	4.6436	0.4873
+ AMT	2.8880	0.4377
+ LMX-to-Image	**2.6350**	**0.4317**
GT Random Pairing Baseline	3.4921	0.9936
RQVAE Reconstruction	0.8990	0.1301
GT Image	0.4865	0.1113

Table 4: Audio-to-image generation accuracy in EMD on BPSD.

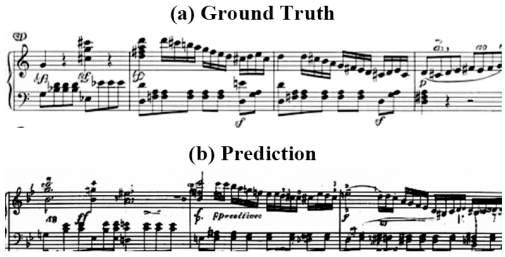


Figure 6: One example from audio-to-image translation.

jarring. Attention visualizations (Figure 5) confirm the model learns to read the score sequentially to generate corresponding audio.

6.2 Audio-to-Image Generation

The A2I task is inherently challenging. Results in Table 4 show that jointly training with auxiliary tasks (AMT and LMX-to-image rendering) significantly improves the model's ability to generate plausible notation, as measured by EMD. While the generated images (Figure 6) are not yet publication-quality, they demonstrate that the model captures key musical elements from the audio.

To evaluate the quality of the generated results, we strongly encourage readers to refer to the actual audio and video examples provided on our demo page [1] .

6.3 OMR, MIDI-to-Audio, and AMT

Our unified training approach yields significant benefits for subtasks. For OMR (Table 5), adding image-audio and even non-overlapping MIDI-audio tasks progressively improves performance, achieving a new state-of-the-art SER of 13.67% on the scanned OLiMPiC test set. This demonstrates strong synergistic learning. For MIDI-to-audio synthesis (Table 6), joint training improves both temporal precision (F_1) and audio quality (FAD). For AMT (Table 7), the improvements are modest, likely because existing datasets like MAESTRO are already large and provide high-quality supervision.

[1] https://sakem.in/u-must/

Method	OLiMPiC		BPSD
	Synth	Scanned	Scanned
OMR-only	15.90	24.58	45.39
+ Image-to-Audio	10.57	15.45	23.85
+ MIDI-to-Audio	**9.72**	**13.67**	**23.36**
Zeus	10.10	14.45	31.24

Table 5: OMR Results in SER. Lower is better.

Method	F_1 ↑			FAD ↓
	50ms	100ms	200ms	
MIDI-to-Audio Only	26.61	64.86	**88.20**	0.201
+ OMR + I2A	**39.37**	**66.63**	84.66	**0.143**

Table 6: MIDI-to-audio synthesis accuracy in F_1 and FAD on BPSD.

Method	MusicNet		MAESTRO
	Str	WW	
AMT-only	87.21	72.04	89.40
+ Audio-to-Image	**87.28**	72.61	89.38
+ LMX-to-Image	87.25	**75.52**	**89.45**
Maman *et al.*	81.8	84.2	89.7

Table 7: AMT results in note onset F_1 score for test set. Higher is better.

7. CONCLUSION

We presented the first unified model for music modal translation capable of successful score image-to-audio generation and state-of-the-art OMR performance. This highlights the potential of holistic, multitask learning for music information processing. We also introduced the large-scale YTSV dataset, emphasizing the value of scanned score images as a data source.

Limitations remain: A2I generation quality is not yet practical, and AMT improvements were modest. Future work will focus on improving audio/image codecs, exploring self-supervised alignment on the YTSV dataset, and developing a single, unified model for both I2A and A2I directions, potentially with a decoder-only architecture.

8. REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] F. Jamshidi, G. Pike, A. Das, and R. Chapman, "Machine learning techniques in automatic music transcription: A systematic survey," 06 2024.

[3] J. Calvo-Zaragoza, J. Martinez-Sevilla, C. Peñarrubia, and A. Ríos Vila, *Optical Music Recognition: Recent Advances, Current Challenges, and Future Directions*, 08 2023, pp. 94–104.

[4] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[5] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 246–253. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000030.pdf

[6] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: multi-task multitrack music transcription," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=iMSjopcOn0p

[7] A. Ríos-Vila, J. Calvo-Zaragoza, and T. Paquet, "Sheet music transformer: End-to-end optical music recognition beyond monophonic transcription," in *International Conference on Document Analysis and Recognition*. Springer, 2024, pp. 20–37.

[8] J. Mayer, M. Straka, J. Hajič, and P. Pecina, "Practical end-to-end optical music recognition for pianoform music," in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds. Cham: Springer Nature Switzerland, 2024, pp. 55–73.

[9] A. Ríos-Vila, D. Rizo, J. M. Iñesta, and J. Calvo-Zaragoza, "End-to-end optical music recognition for pianoform sheet music," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 26, no. 3, pp. 347–362, 2023.

[10] M. Kim, J.-w. Jung, H. Rha, S. Maiti, S. Arora, X. Chang, S. Watanabe, and Y. M. Ro, "Tmt: Tri-modal translation between speech, image, and text by processing different modalities as different languages," *arXiv preprint arXiv:2402.16021*, 2024.

[11] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, H. Guo, X. Chang, J. Shi, S. Zhao, J. Bian, Z. Zhao, X. Wu, and H. M. Meng, "UniAudio: Towards universal audio generation with large language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 56 422–56 447.

[12] D. Lee, C. Kim, S. Kim, M. Cho, and W. Han, "Autoregressive image generation using residual quantization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 11 513–11 522.

[13] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[14] S. Chang, E. Benetos, H. Kirchhoff, and S. Dixon, "Yourmt3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation," in *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2024, pp. 1–6.

[15] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "Mir_eval: A transparent implementation of common mir metrics." in *ISMIR*, vol. 10, 2014, p. 2014.

[16] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 50–57. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/19_Paper.pdf

[17] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *Proc. IEEE ICASSP*, 2024.

[18] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, Nov 2000. [Online]. Available: https://doi.org/10.1023/A:1026543900054

A. MATHEMATICAL FORMULATION OF MULTIMODAL TOKENIZATION AND UNIFIED TRANSLATION MODEL

A.1 Tokenizers and Unified Vocabulary

We denote a *modality* by the calligraphic symbol $\mathcal{X} \in \{\mathcal{I}, \mathcal{A}, \mathcal{N}, \mathcal{M}\}$ for **I**mage, **A**udio, musical **N**otation (LMX), and **M**IDI performance, respectively. The corresponding *raw* data X are written with roman capitals I, A, N, M. Each modality owns a *tokenizer encoder* $\mathcal{F}_{\mathcal{X}}$ and an inverse *tokenizer decoder* $\mathcal{G}_{\mathcal{X}}$ such that:

$$\mathcal{F}_{\mathcal{X}}(X) = z^{(\mathcal{X})}_{1:L_X}, \qquad \mathcal{G}_{\mathcal{X}}\big(z^{(\mathcal{X})}_{1:L_X}\big) \approx X. \tag{1}$$

Hence, $z^{(\mathcal{X})}_{1:L_X}$ is the discrete-token representation of X and L_X is its length. Every modality has its own vocabulary $\mathcal{V}_{\mathcal{X}}$, yet all tokens ultimately live in a *shared* space $\mathcal{V}$, allowing a single Transformer to translate between any pair of modalities. For the **continuous** modalities—score images and audio—we learn residual vector-quantised tokenizers. Both tokenizers employ $d = 4$ *unshared* codebooks, each of cardinality $\kappa = 1024$. Consequently, each time-step yields a *bundle* of d code indices $z_{t,1}, \ldots, z_{t,d} \in \{0, \ldots, \kappa-1\}$, so the discrete representation is a 2-D array of shape $L_X \times d$.

A.1.1 Score images, $\mathcal{X} = \mathcal{I}$

RQVAE compresses each image patch by a factor of $C = 16$. Given a score image that contains K musical systems, RQVAE produces token sequences which are then flattened in vertical reading order and concatenated with a separator token `[SEP]`.

A.1.2 Audio, $\mathcal{X} = \mathcal{A}$

All audio is resampled to $f_s = 44.1\,\text{kHz}$ (mono) and tokenized with DAC using hop size $h = 512$ samples. This results in a sequence of token bundles of length $L_A = \lceil \frac{T f_s}{h} \rceil$ for an audio of T seconds.

A.1.3 Linearized MusicXML, $\mathcal{X} = \mathcal{N}$

Notation data are stored as linearized MusicXML (LMX) [8]. For consistency, its single token stream is padded to match the $d = 4$ codebook structure of continuous modalities.

A.1.4 MIDI, $\mathcal{X} = \mathcal{M}$

We adopt the YourMT3+ MIDI-like event vocabulary (10ms quantisation) [14]. This token stream is also padded to $d = 4$ codebooks.

A.1.5 Unified Vocabulary, $\mathcal{V}$

The total vocabulary $\mathcal{V}$ is the union of all modality-specific vocabularies and special control tokens (`[SOS]`, `[EOS]`, `[SEP]`, `[PAD]`).

A.2 Model Architecture

A.2.1 Input embedding

For source modality $\mathcal{X}$ and target modality $\mathcal{Y}$, each input token z_i is embedded as:

$$e_i = \underbrace{\text{TokEmb}(z_i^{(\mathcal{X})})}_{\text{token}} + \underbrace{\text{PosEmb}_{\mathcal{X}}(i)}_{\text{modality-specific pos. enc.}} + \underbrace{\text{TgtEmb}_{\mathcal{Y}}}_{\text{target hint}}. \tag{2}$$

A.2.2 Sequence-to-Sequence Model

The encoder $\mathcal{E}$ processes the source token sequence to produce hidden states H; the decoder $\mathcal{D}$ autoregressively generates the target token sequence.

A.2.3 Sub-Decoder Module $\mathcal{D}_{sub}$

To generate the $d = 4$ parallel tokens for image and audio, a one-layer Transformer *sub-decoder* $\mathcal{D}_{sub}$ is used at each timestep of the main decoder.

A.3 Training Objective

The model is trained with a standard cross-entropy loss, maximizing the likelihood of the ground truth target sequence given the source sequence. During softmax calculation, tokens from non-target modalities are masked.

B. EVALUATION DETAILS

B.1 DTW Alignment for Onset F_1 Computation

To evaluate onset detection accuracy for the Image-to-Audio (I2A) task while accounting for potential temporal misalignments, we employ Dynamic Time Warping (DTW). We first transcribe both the ground truth and the model-generated audio into MIDI piano rolls. The DTW algorithm then operates in one dimension, warping the time axis of the generated piano roll to best match the ground truth, which handles variations in tempo and timing while preserving pitch information. This alignment allows for a more accurate computation of onset F_1 scores by correctly pairing musical events. Figure 7 illustrates this process.

B.2 Token Distribution Histograms for EMD Computation

For Audio-to-Image (A2I) evaluation, we compute the Earth Mover's Distance (EMD) between token distributions of the generated and ground truth notation. After generating a score image from audio, we use an OMR model to extract its Linearized MusicXML (LMX) token sequence. We then create frequency histograms for pitch tokens and duration tokens separately. To account for potential meter interpretation errors (e.g., a model outputting a piece in 2/2 time instead of 4/4), we compute the EMD for duration tokens with temporal shifts of -1, 0, and +1 on the histogram bins and select the minimum value, allowing for a more flexible evaluation. Figure 8 shows examples of these histograms.

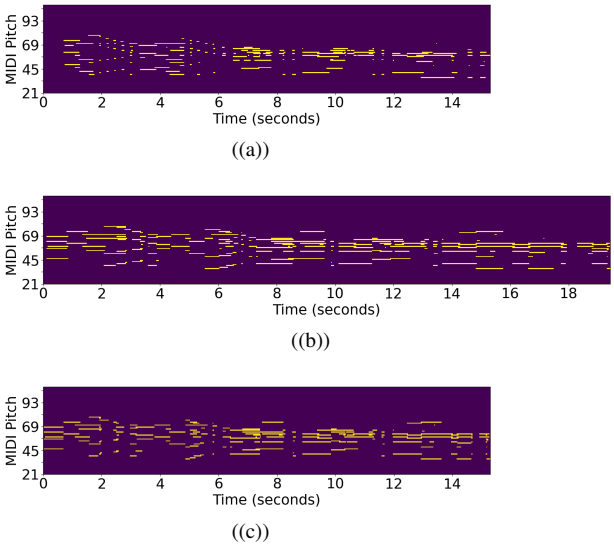


Figure 7: Piano roll visualizations of 22 Ground Truth MIDI, (b) MIDI transcribed from our I2A model's output, and (c) the result of DTW-aligning (b) to (a) for evaluation.

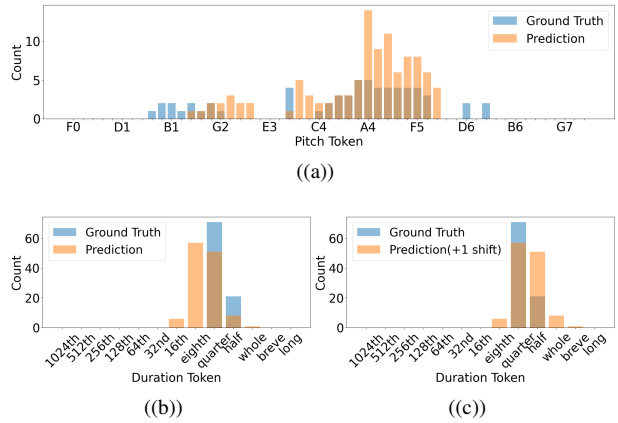


Figure 8: Token distribution histograms for EMD calculation: (a) pitch tokens, (b) duration tokens (no shift), and (c) duration tokens with a +1 shift applied to the prediction to improve alignment.

C. YOUTUBE SCORE VIDEO DATASET DETAILS

C.1 Metadata Extraction and Standardization

To enable robust data filtering, we used the Claude-3.5-Sonnet large language model to extract structured metadata (e.g., composer, instrumentation, year) from the titles of all 12,217 collected videos. This allowed us to programmatically filter the dataset based on musical characteristics. We also performed extensive standardization to unify piece titles and normalize composer names, resolving duplicates and variations. The aggregated results of the categories from extracted metadata are shown in the Table 9.

C.2 Data Processing Pipeline

Our processing pipeline involves three main steps:

1. **Slide Segmentation:** A rule-based algorithm analyzes video frames to detect the precise timing of slide transitions, accommodating both instantaneous cuts and animated effects like crossfades. This extracts individual score slides and their corresponding audio segments. Silent segments (e.g., title cards) are subsequently filtered out.

2. **System Cropping:** To handle visual inconsistencies, we fine-tuned a YOLOv8 model on manually labeled data to detect and crop each musical system (a line of music) from the score slides. A second YOLOv8 model detects staff-line height for normalization, ensuring all systems are resized consistently.

3. **Statistical Filtering:** We apply rigorous filtering based on statistical properties. This includes removing poor-quality or color-inverted scans using pixel intensity metrics, discarding segments with anomalous dimensions or significant overlap between detected systems, and enforcing temporal constraints (3-20 seconds) on audio duration to eliminate outliers.

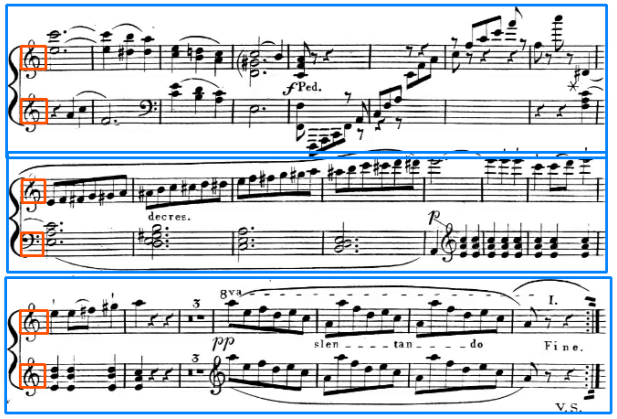


Figure 9: Illustration of the music system detection pipeline using the fine-tuned YOLOv8. Music systems detected by fine-tuned YOLOv8 are notated with blue boxes, and detected staff lines are notated with red boxes. Note that the red boxes detect the staff height near clefs, not the clefs themselves.

C.3 Test Set: YTSV-T11

To evaluate performance on in-the-wild scanned scores, we manually curated a test set, **YTSV-T11**, consisting of 11 diverse piano pieces from the YTSV dataset, for which we verified there were no duplicates in the training set.

D. RQVAE AND DAC TOKENIZATION DETAILS

D.1 RQVAE Model for Score Images

Our RQVAE model was adapted specifically for sheet music tokenization.

D.1.1 Architecture

It processes single-channel grayscale images and uses four unshared codebooks, each with 1024 codes, and a model dimension of 256. We removed attention blocks to ensure

Field	Description	Example Value
YT Id	Unique YouTube video identifier	`0oRyPLnPeFw`
Title of Video	Original video title as displayed on YouTube	`Walton - Passacaglia (1982) for solo cello [w/ score]`
Duration	Video length in `MM:SS` format	`10:06`
Composer Full Name	Complete name of the composer	William Walton
Title of piece	Name of the musical composition	Passacaglia
Opus number	Catalog number of the composition (null if unavailable)	null
Instrumentation	Categorization of musical forces from predefined set: `[orchestral, concerto, solo, duet, trio, quartet, quintet, larger chamber music, choral, wind band, non-classical, vocal, unknown]`	solo
Category	Specific genre or form description	cello solo
Piano Included	Boolean indicating presence of piano part	False
String Included	Boolean indicating presence of string instruments	True
Wind Included	Boolean indicating presence of wind instruments	False
Voice Included	Boolean indicating presence of vocal parts	False
Year	Year of composition	1982
Staff Count	Two numbers indicating single-melody instrument staves and piano staves, separated by hyphen	1-0

Table 8: Content Metadata Fields

Figure 10: Example patches showing the 16×16 pixel resolution of individual tokens

the model focuses on local features like noteheads rather than global image structure.

D.1.2 Compression

We use a 16x compression strategy, which ensures that each token corresponds to a small image patch, fine-grained enough to capture musical details (Figure 10).

D.1.3 Training

We implemented a resolution-adaptive training strategy, using different crop sizes and batch sizes for different input resolutions to handle the wide variety of score layouts (Figure 11). Instead of a standard perceptual loss, we used a weighted MSE loss between activations of an OMR model's encoder to better preserve musically relevant features.

D.1.4 Augmentation

To enhance robustness, we generated 32 augmented versions of image tokens for each image using pixel shifts (4 vertical shifts and 8 horizontal shifts) (Figures 12 and 13).

D.2 DAC Model for Audio

For audio tokenization, we retrained a Descript Audio Codec (DAC) model.

D.2.1 Configuration

The model uses a hop size of 512 samples on 44.1kHz mono audio, resulting in 86 tokens per second. It employs four unshared codebooks with 1024 codes each.

D.2.2 Training

We retrained the model specifically on classical music audio, rather than diverse sounds, to achieve a richer and more efficient representation for our target domain.

D.2.3 Augmentation

To improve robustness to slight temporal variations, we generated nine variants through temporal shifts ranging from -20 to +20 samples at five-sample intervals, with each five-sample shift corresponding to approximately 0.113 milliseconds (Figure 17).

E. MODEL AND TRAINING DETAILS

E.1 Architecture and Initialization

Our Transformer models feature 12 encoder and 12 decoder layers, a model dimension of 1024, a feed-forward hidden size of 4096, and 16 attention heads. For image and audio tokens, the embedding matrix was initialized with the learned codebook embeddings from the pretrained RQ-VAE and DAC models, respectively.

Category	Description	Videos	Segments	Duration (hrs)
Piano Solo	Solo piano compositions	9052	232029	762.34
Accompanied Solo	Solo compositions for a non-piano instrument with piano accompaniment	912	47373	141.83
String Quartet	Compositions for two violins, viola, and cello	594	48470	138.48
Others	Compositions not classified under predefined categories	454	24912	69.13
Unaccompanied Solo	Solo compositions for a single non-piano instrument	207	3542	11.24
Guitar Solo	Solo compositions for classical guitar	192	1976	6.97
Piano Trio	Compositions for piano, violin, and cello	254	22736	68.51
Organ Solo	Solo compositions for organ	161	5923	20.01
Piano Quintet	Compositions for piano and string quartet	109	13382	34.69
Piano Quartet	Compositions for piano, violin, viola, and cello	84	9168	26.07
Harpsichord Solo	Solo compositions for harpsichord	84	17419	43.93
Woodwind Ensemble	Ensembles consisting only of woodwind instruments	63	3784	10.05
Other Wind Ensemble	All kinds of wind ensembles beyond the woodwind family	51	3206	8.06

Table 9: Aggregated Category Counts and Durations of Metadata

Composer	Work	YouTube ID	Included Segments
Clara Schumann	Piano Sonata in G minor	`Pw4fMNMO90U`	74
Friedrich Gulda	Prelude and Fugue	`V2h23Dsw57A`	31
Alexander Borodin	Petite Suite	`7vBkBCa3n4o`	31
Lev Abeliovich	5 Pieces for Piano	`07zYLY1YTj0`	29
Carl Czerny	Studio No. 29 Op. 409	`J5WRTAYtaOg`	24
Mily Balakirev	Mazurka No. 5	`BqAWfT76pJY`	19
Alexander Borodin	Petite Suite	`38P9U3WRX9w`	15
Giovanni Sgambati	Vecchio minuetto	`EXNifef40vU`	14
Giovanni Sgambati	2 Concert Etudes	`olBJh_5rv2c`	10
Carl Czerny	Album élégant des Dames Pianistes Vol.3	`n_Sn48u1t94`	3
Carl Czerny	Romance Op. 755 No. 12	`4Pa4x9SDNdw`	1

Table 10: List of the videos in YTSV-T11

E.2 Training Procedure

We trained each model for 600,000 updates using the AdamW optimizer with a total batch size of 24 sequence pairs on two NVIDIA H100 GPUs. The learning rate started at 1×10^{-4} and decayed to 1×10^{-5} following a cosine schedule with a 2,000-step linear warmup.

E.3 Curriculum Learning

To stabilize training, we employed a curriculum learning strategy where tasks were introduced gradually. **I2A Model**: We started with only OMR (image-to-notation) examples. After 15k steps, we added MIDI-to-audio synthesis. Finally, at 50k steps, we introduced the direct image-to-audio task. **A2I Model**: We began with only AMT (audio-to-MIDI). After 40k steps, we added notation-to-image rendering. The direct audio-to-image task was introduced at 70k steps.

This approach allows the model to learn representations on simpler, data-rich subtasks before tackling the more complex, end-to-end translation challenges.

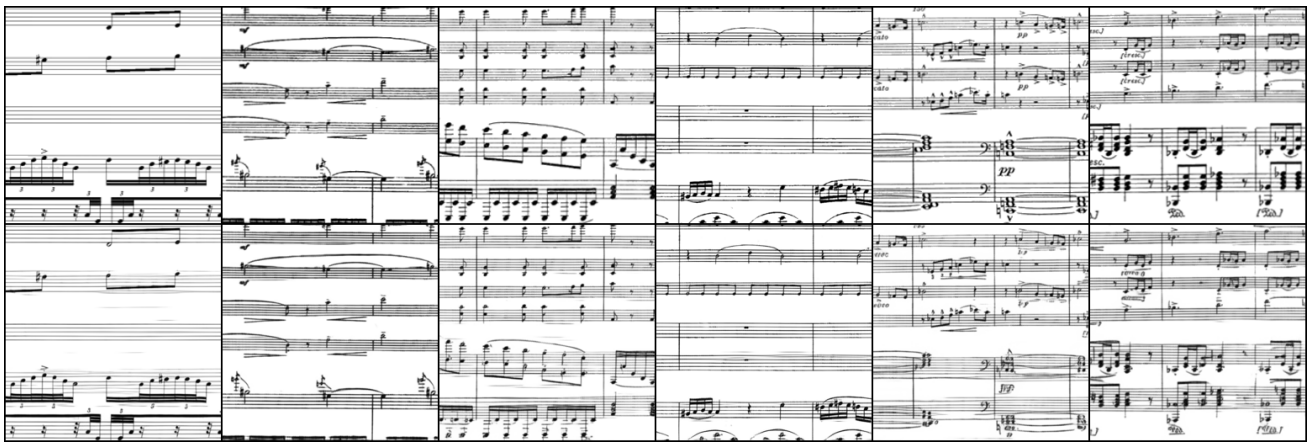
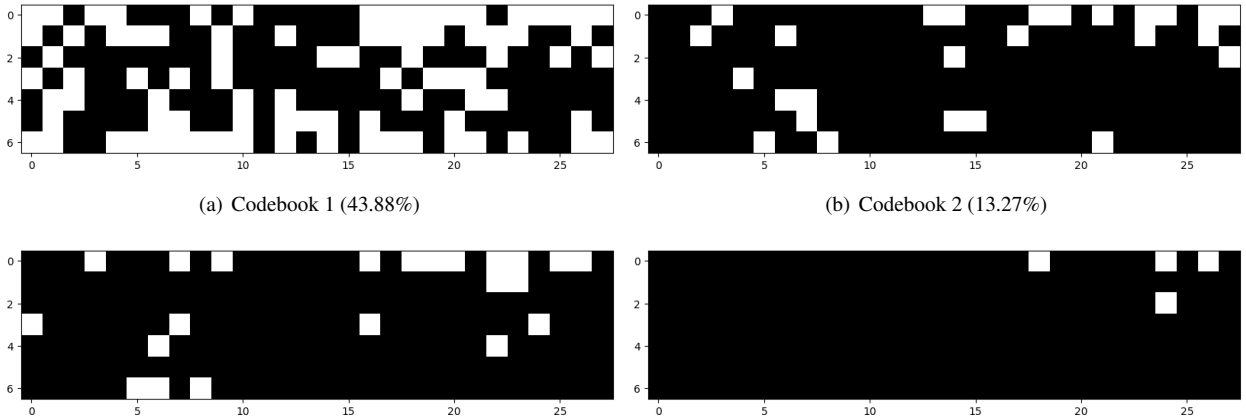


Figure 11: Comparison between input sheet music (top) and model reconstruction (bottom), demonstrating reconstruction artifacts in staff lines when a model trained only on 64×64 pixel image crops processes 256×256 pixel inputs(unseen image size).



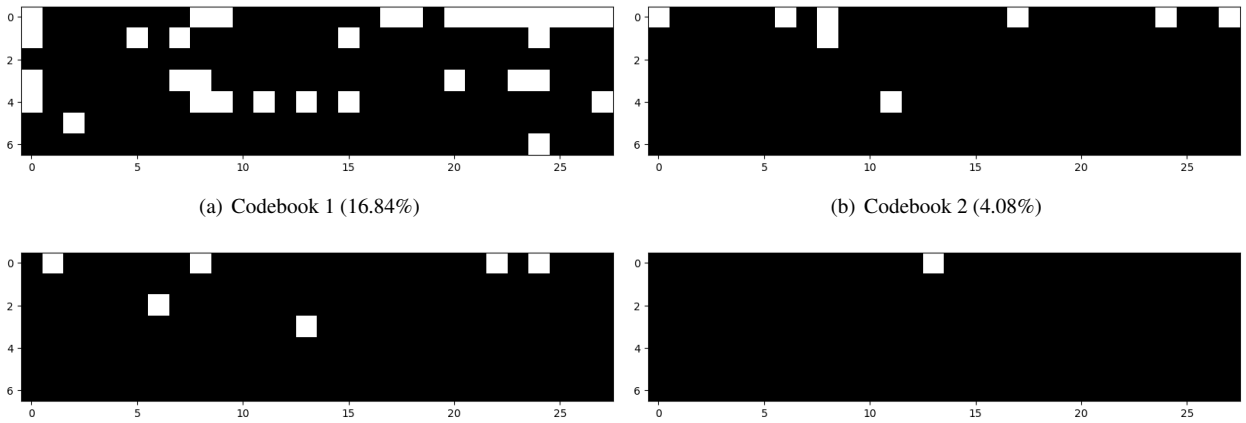
(a) Codebook 1 (43.88%)

(b) Codebook 2 (13.27%)

(c) Codebook 3 (11.22%)

(d) Codebook 4 (2.04%)

Figure 12: Token difference visualization for one-pixel horizontal shifts across codebooks. Each subplot shows the token changes for an individual codebook, with white indicating unchanged tokens and black indicating changed tokens. Percentages in parentheses represent the proportion of tokens that remained unchanged.



(a) Codebook 1 (16.84%)

(b) Codebook 2 (4.08%)

(c) Codebook 3 (3.06%)

(d) Codebook 4 (0.51%)

Figure 13: Token difference visualization for one-pixel vertical shifts across codebooks. Each subplot shows the token changes for an individual codebook, with white indicating unchanged tokens and black indicating changed tokens. Percentages in parentheses represent the proportion of tokens that remained unchanged.

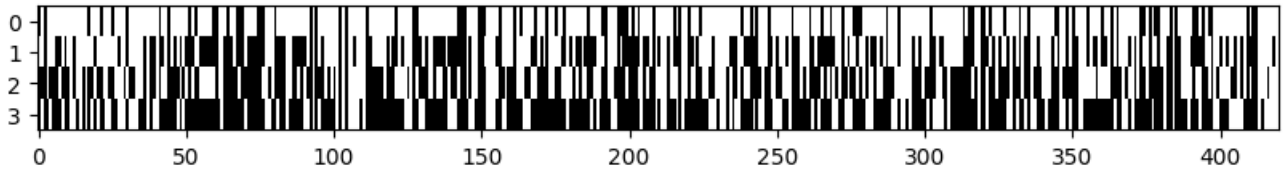


Figure 14: 1 sample shifted
1: 75.71% 2: 52.38% 3: 45.71% 4: 33.33%
Total: 51.79%

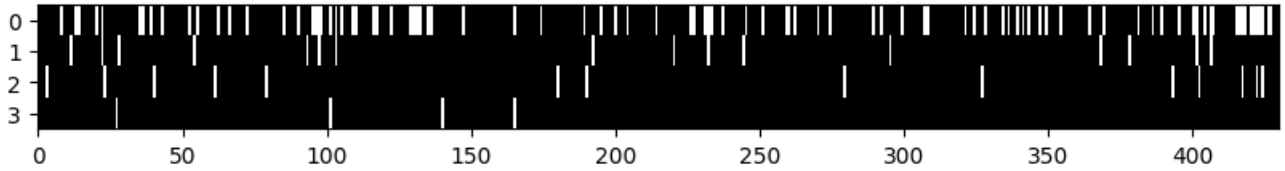


Figure 15: 5 sample shifted
1: 21.63% 2: 3.72% 3: 3.26% 4: 0.93%
Total: 7.38%

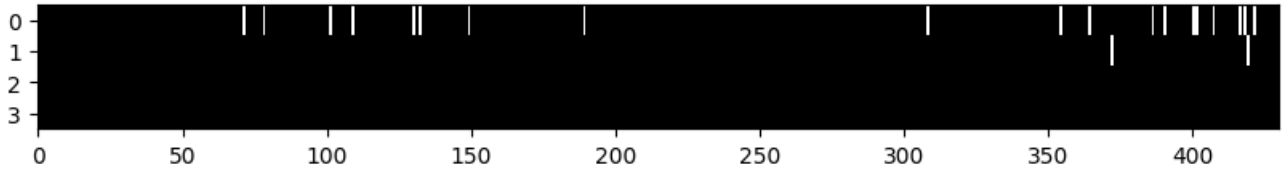


Figure 16: 10 sample shifted
1: 4.42% 2: 0.47% 3: 0.00% 4: 0.00%
Total: 1.22%

Figure 17: Visualization of token differences for varying temporal shifts in audio samples. Each subplot demonstrates the impact of different shift magnitudes. The visualization displays token changes across each of the four codebooks, with white representing unchanged tokens and black indicating changed tokens. Reported percentages show the token retention rate for each codebook (1-4) individually, followed by the total retention rate across all codebooks.

Models	Model Size				Task Introduction (trainging step)		
	Dimension	Enc/Dec Layers	Heads	Sub-Dec Heads	OMR/AMT	M2A/L2I	I2A/A2I
OMR Only	512	12	8	8	0	–	–
Image-to-Audio Only	768	12	10	10	–	–	0
MIDI-to-Audio Only	512	4	8	8	–	0	–
OMR + Image-to-Audio	1024	12	16	8	0	–	15,000
OMR + Image-to-Audio + MIDI-to-Audio	1024	12	16	8	0	15,000	50,000
AMT Only	768	12	12	8	0	–	–
Audio-to-Image Only	768	12	10	10	–	–	0
AMT + Audio-to-Image	1024	12	16	8	0	–	40,000
AMT + Audio-to-Image + LMX-to-Image	1024	12	16	8	0	40,000	70,000

Table 11: Transformer model configurations and task introduction steps. The upper block contains I2A models, the lower contains A2I models. Numbers in "Task Introduction" columns indicate the training step at which a task is introduced.