ENGRAVINGGNN: A HYBRID GRAPH NEURAL NETWORK FOR END-TO-END PIANO SCORE ENGRAVING

Emmanouil Karystinaios[1] **Francesco Foscarin**[2] **Gerhard Widmer**[1,3]
[1] Johannes Kepler University, Linz, Austria
[2] Moises Systems Inc., Salt Lake City, USA.
[3] LIT AI Lab, Linz Institute of Technology, Austria
firstname.lastname@jku.at

ABSTRACT

This paper focuses on automatic music engraving, i.e., the creation of a humanly-readable musical score from musical content. This step is fundamental for all applications that include a human player, but it remains a mostly unexplored topic in symbolic music processing. In this work, we formalize the problem as a collection of interdependent subtasks, and propose a unified graph neural network (GNN) framework that targets the case of piano music and quantized symbolic input. Our method employs a multi-task GNN to jointly predict voice connections, staff assignments, pitch spelling, key signature, stem direction, octave shifts, and clef signs. A dedicated postprocessing pipeline generates print-ready MusicXML/MEI outputs. Comprehensive evaluation on two diverse piano corpora (J-Pop and DCML Romantic) demonstrates that our unified model achieves good accuracy across all subtasks, compared to existing systems that only specialize in specific subtasks. These results indicate that a shared GNN encoder with lightweight task-specific decoders in a multi-task setting offers a scalable and effective solution for automatic music engraving.

1. INTRODUCTION

The musical score encodes a variety of information that we can organize into two classes: the *musical content*, which specifies the sounds that a musician should produce, and the *score engraving*, i.e., the notation symbols which don't influence the music, but are employed for its graphical representation [1]. Examples of engraving information include staff separation, clefs, pitch spelling, grouping into chords and voices, octave shifts, and stem directions. A high-quality engraving is a fundamental component of a musical score as it allows musicians and musicologists to efficiently read it and get the composer's intentions with precision.

The situation changes significantly when we exclude the human component and consider music that must be played or composed by a machine, such as a sequencer. In this scenario, formats that only encode the musical content, like a MIDI-file, are much more efficient and easier to handle. This is visible in the field of music information retrieval (MIR) where the vast majority of systems, for example, automatic music transcription or automatic composition systems, work on MIDI-like representations. This task of *automatic music engraving* is then a necessary bridge to allow musicians to play computer-processed music.

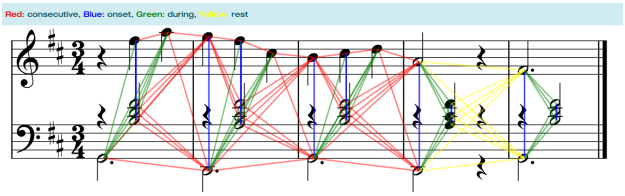

Figure 1. The score graph, nodes are notes and edges are marked in different color for each type.

Some systems were recently proposed to address individual engraving subtasks such as pitch spelling and key estimation [2, 3], voice and staff separation [4]. We use the latter as a basis for our system, and improve it by extending it to more engraving tasks without reducing its performance. Another related work is the one from Beyer and Dai [5], which targets in a single end-to-end framework the tasks of note quantization and music engraving. This is an interesting direction for automatic music transcription (AMT), but we keep our model more generic to be able to employ it in diverse scenarios (e.g., with a music generation system).

In this paper, we focus specifically on piano music and quantized inputs, introducing EngravingGNN: a unified framework for automatic score engraving. At its core, EngravingGNN employs a Hybrid-GNN encoder, a combination of heterogeneous graph convolutions and stacked GRU layers [6], to fuse relational note interactions with long-term temporal context. From this shared embedding, lightweight decoder heads simultaneously predict all of the following:

- *Voice edges* and *staff labels*, organizing notes into independent streams on the appropriate stave,
- *Pitch spelling* and *key signature*, ensuring correct accidentals and tonal context,
- *Stem direction* and *octave shift* markings, adhering to engraving conventions for clarity,

* Equal contribution.

- *Symbolic duration* predicting dots, triplets, and notehead types (whole, half, eight, etc.),
- *Clef signs*, producing the initial clefs and capturing mid-piece clef changes when they occur.

In summary, our contributions are 3-fold:

1. A end-to-end model network architecture for music engraving, with an *Hybrid-GNN* encoder integrating heterogeneous graph convolutions with GRU stacks, plus 10 decoder heads for all core engraving tasks.
2. A postprocessing and export pipeline yielding fully engraved MusicXML/MEI scores.
3. Evaluation and comparison with existing systems on two piano corpora, with detailed metrics for voice, staff, pitch spelling, key, stem direction, symbolic duration, octave shifts, and clefs.

2. METHODOLOGY

Figure 2 provides a schematic overview of our approach. Starting from a quantized score, we construct an input graph, which is then encoded using a graph neural network (GNN). Multiple tasks are decoded in parallel from this encoded representation, followed by postprocessing and export of the final musical score.

2.1 Input Graph Construction

Given quantized notes, we create a directed heterogeneous graph $G_{in} = (V, E_{in}, \mathcal{R}_{in})$:

- **Nodes** V: one per note, with features:
  - pitch class (12-way one-hot),
  - octave (scalar in $[1, 7]$),
  - normalized duration $d \in (0, 1)$ via $\tanh(\frac{\text{note } d}{\text{bar } d})$,
  - onset fraction within bar and downbeat positions.
- **Edge types** $\mathcal{R}_{in}$: onset, during, follow, silence, plus inverse edges, as in [4]

In the current version of our system, grace notes are removed as they pose further challenges due to the lack of a quantized duration.

2.2 Output Predictions

Our model produces a rich set of engraving annotations by formulating each attribute as either a node-classification or an edge-prediction task. Given the GNN-computed embedding $\mathbf{h}_v$ for each note v, and embeddings $(\mathbf{h}_u, \mathbf{h}_w)$ for each candidate note-pair (u, w), we jointly predict the following:

1. **Voice edges.** We treat voice assignment as an edge-prediction problem over a restricted set of candidate pairs Λ_v as in [4]. Each candidate $(u, w) \in \Lambda_v$ satisfies that u and w lie in the same bar and $\text{offset}(u) \leq \text{onset}(w)$, ensuring temporal consistency. A 2-layer MLP, followed by a sigmoid function, takes the concatenated pair $[\mathbf{h}_u; \mathbf{h}_w]$ and outputs a probability $p_{uw} \in (0, 1)$ of w immediately following u in the same voice stream.
2. **Staff label.** Each note is assigned to one of two staves (upper or lower) via a binary node classifier. A 2-layer MLP, followed by a sigmoid function, maps $\mathbf{h}_v$ to a probability s_v. At inference time, we threshold s_v at 0.5 to decide staff membership. This separation is crucial for later beam grouping and rest insertion.
3. **Pitch spelling.** Correct notation requires exact accidentals. We predict one of 35 classes per node (7 natural pitch classes $\{A, B, C, D, E, F, G\}$, each with optional $\sharp$, $\flat$, $\times$, $\flat\flat$, or $\natural$ sign). A 2-layer softmax MLP takes $\mathbf{h}_v$ and outputs a distribution over these classes, trained via categorical cross-entropy against MusicXML ground-truth spellings.
4. **Key signature.** MusicXML represents key signatures as an integer in $[-7, +7]$ (number of fifth-circle sharps or flats). We add a sixteenth node-classification head: a 2-layer MLP with softmax over 15 classes. Local key changes (a.k.a. modulations) occur in Romantic repertoire, so we predict this attribute per note and later smooth them within measures.
5. **Stem direction.** Stem orientation can be up or down or no-stem. We model this as a ternary classification per note: Postprocessing enforces consistency within chords and aligns with beam direction.
6. **Octave shift.** Octave transposition indications (e.g. 8va, 8vb, 15ma) are common in piano music. We predict a multiclass label per note among $\{\texttt{none}, 8\texttt{va}, 8\texttt{vb}, 15\texttt{ma}\}$ via a softmax MLP. Consecutive notes sharing the same label are later merged into a single octave-shift bracket in the score.
7. **Clef sign.** Piano scores occasionally employ clef changes (e.g. switch from G- to F-clef on the upper staff). We predict one of three classes: G-, F-, or C-clef (this is rare in piano music, but we still support it for completeness) per note via a softmax head. During export, we insert clef-change tokens at the first note of each predicted clef region.
8. **Symbolic duration.** In addition to the quantized onset and offset, we must determine the proper notehead type ($\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, \ldots$), the number of augmentation dots, and any tuplet bracket (e.g. triplet) attributes. We discretize "note-type" into classes $\{\texttt{whole}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{6}, \ldots\}$, "dots" into $\{0, 1, 2, 3\}$, and "tuplet" into common ratios $\{1, 3, 5\}$ (i.e. no tuplet, triplet, quintuplet). Each is predicted by its 2-layer softmax MLP on $\mathbf{h}_v$. Postprocessing then groups adjacent notes with identical tuplet flags into a single bracket and assigns dot placement.

All node-classification heads input the same latent embedding from the GNN encoder. Each head has its respective

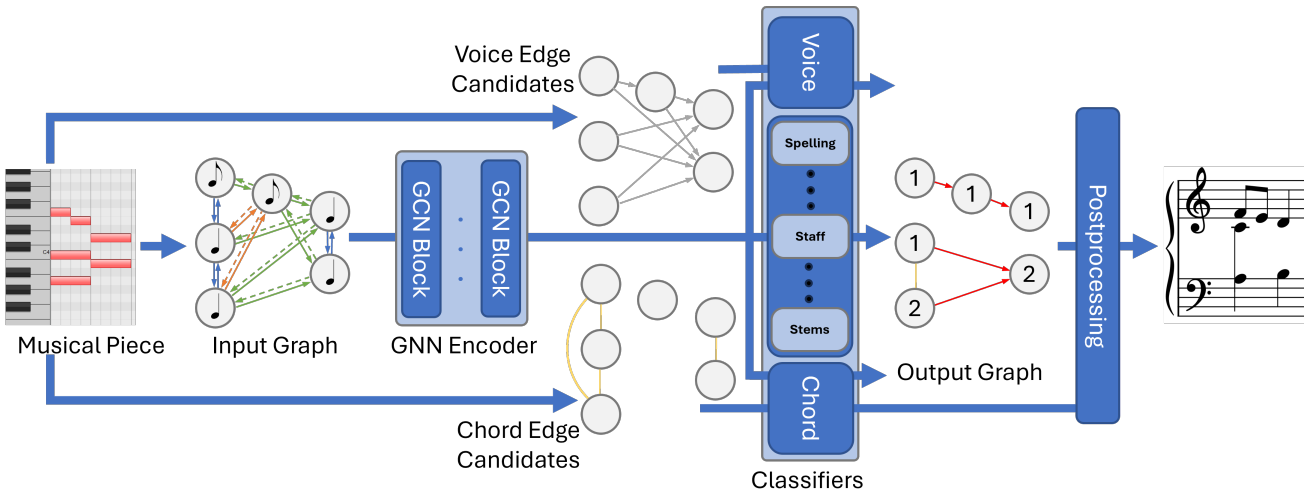

Figure 2. System overview: Input graph → GNN encoder → multi-task decoders → postprocessing → engraved score.

cross-entropy loss, and the voice-edge head uses binary cross-entropy over Λ_v. We sum these losses for end-to-end training.

2.3 GNN Encoder

Our backbone encoder is a *HybridGNN* that interleaves heterogeneous graph convolutions with recurrent GRU updates, combining the strengths of relational message-passing and sequential modeling [6]. By stacking these hybrid blocks, we capture both the rich, structured interactions between notes (via the graph convolutions) and the long-term temporal persistencies across the piece (via the GRU stacks).

Formally, let each node u in our input graph $G_{in} = (V, E_{in}, \mathcal{R}_{in})$ have an initial feature vector $\mathbf{h}_u^{(0)} \in \mathbb{R}^d$ encoding pitch class, octave, normalized duration, bar index, etc. We apply $L = 3$ identical hybrid layers. At layer l, we first perform a heterogeneous GraphSAGE convolution [7]:

$$\tilde{\mathbf{h}}_u^{(l)} = \sigma\Big(\mathbf{W}_0^{(l)}\,\mathbf{h}_u^{(l-1)} + \sum_{r\in\mathcal{R}_{in}}\sum_{v\in\mathcal{N}_r(u)} \mathbf{W}_r^{(l)}\,\mathbf{h}_v^{(l-1)}\Big),$$

where $\mathcal{N}_r(u)$ is the set of neighbors of u connected by relation type r, and $\{\mathbf{W}_r^{(l)}\}$ are relation-specific weight matrices. This step integrates multi-relation context such as onset, during, follow, and silence edges.

In parallel, we feed the initial feature vectors $\tilde{\mathbf{h}}_u^{(0)}$ into a gated recurrent unit (GRU) cell, the vectors are structured timely thanks to GraphMuse sampling process [6].

Because we stack one GRU per graph convolution layer, the network can accumulate and preserve information over long note sequences, learning temporal dependencies that pure GNN layers might lose. The GRU update also allows each node to "remember" its evolving embedding across layers, reinforcing global structure.

After $L = 3$ such hybrid blocks, each note u has a final embedding $\mathbf{h}_u^{(L)} \in \mathbb{R}^{256}$ that fuses local relational context with long-range sequential patterns. These embeddings serve as the shared inputs to all downstream decoders (voice-edge prediction, staff label, pitch spelling, etc.), enabling coordinated multi-task learning.

2.4 Multi-Task Decoders

Each task uses a lightweight 2-layer MLP on $\mathbf{h}_u$ (or $[\mathbf{h}_u; \mathbf{h}_v]$ for edge tasks):

- **Binary heads**: voice-edge candidate and chord-edge candidates (sigmoid output + BCE loss).
- **Softmax heads**: staff, stem, pitch spelling, key signature, octave shift, clef, and symbolic duration (categorical cross-entropy).

2.5 Training

We train end-to-end, minimizing the (unweighted) sum of all task losses. We employ Adam optimizer (lr=1e-3, weight decay 5×10^{-4}), 100 epochs. The train hyperparameters are taken from [4].

2.6 Implementation Details

For the encoder, we use a hidden size of 256 for both the heterogeneous graph convolution outputs and the GRU hidden state. Each convolutional layer is followed by a ReLU activation, and we apply 50% dropout to the convolution output before it enters the GRU. To stabilize training, layer normalization is applied both inside each GRU cell and between successive graph convolution blocks.

2.7 Postprocessing

A naive way to turn our voice-edge probabilities into a voice separation would be to simply apply a threshold and keep all edges above it. In practice, this can still produce three kinds of engraver's nightmares:

1. Distinct voices collapse into one,
2. A single voice splits into multiple disjoint streams,
3. Notes that belong together in the same chord end up in different voices.

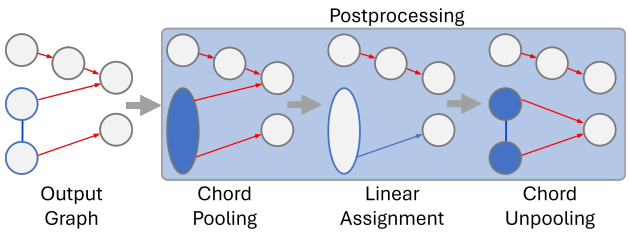

Figure 3. Postprocessing pipeline: chord pooling → voice assignment → unpooling → beaming & rests.

To guarantee musically valid output, we therefore follow prediction with a small postprocessing pipeline that enforces engraving constraints, as proposed in [4].

As shown in Figure 3, we perform the following steps in order:

1. **Chord pooling**: cluster synchronous notes (predicted chord edges) into virtual nodes.

2. **Voice assignment**: Once chords are fused, the score reduces to purely monophonic streams, so we can solve a linear assignment via the Hungarian algorithm [8] on pooled nodes.

3. **Unpooling**: restore original notes, distributing assigned voices.

4. **Rest infilling**: insert rests to fill each bar per voice.

We note that some node-level labels are altered by the chord-postprocessing to guarantee cohessiveness and proper engraving throughout predictions that need to carry the same label. Such labels include: symbolic duration, stem direction, and key signature. The logits of the notes that are pooled together for these tasks are averaged. All other node-level attributes, i.e. octave shift, staff, pitch spelling, and clef sign, are independent of the post-processing operations.

3. EXPERIMENTS

3.1 Datasets

We evaluate on two piano corpora that span simple to highly complex repertoire. The *J-Pop* set [9] (811 scores, 20% held out) features straightforward pop arrangements—lower-staff chords with simple upper-staff melodies. In contrast, the *DCML Romantic Corpus* [10] (393 scores, 20% held out) includes 17th–20th-century works with virtuosic textures, cross-staff beaming, multiple overlapping voices, and complex engravings. We convert all sources to MusicXML (using MuseScore and Partitura [11]) and split each corpus into 80 % train/validation and 20% test sets, ensuring that our model's ability to handle intricate engraving scenarios does not come at the cost of simpler pieces.

In line with Foscarin et al. [4], we train EngravingGNN as a single, unified model on the combined training sets of both J-Pop and DCML Romantic.

3.2 Evaluation Metrics

We evaluate each engraving subtask with measures tailored to its output type. Voice separation is judged using the homophonic F1 metric from Hiramatsu et al. [12]. Chord prediction is essentially a binary decision for each pair of synchronous notes: either they belong to the same chord or not. We report the standard F1 score over all candidate chord-edges.

All remaining engraving attributes—staff assignment, stem direction, clef sign, key signature, pitch spelling, octave shift, and symbolic duration features (note head type, augmentation dots, tuplet bracket)—are treated as per-note classification tasks. For each attribute, we compute the simple classification accuracy, i.e., the percentage of notes for which the predicted label exactly matches the ground truth. In the case of symbolic duration, which is decomposed into three separate predictions (note type, number of dots, tuplet flag), we measure both each individual head's accuracy and the overall rate at which all three sub-predictions are simultaneously correct.

All results are aggregated across the test set using micro-averaging, ensuring that pieces with more notes or edges contribute proportionally to the final scores.

3.3 Results

Table 1 presents the test-set performance of EngravingGNN alongside the baselines on both the J-Pop and DCML Romantic corpora. On the J-Pop dataset, EngravingGNN achieves a voice-separation F1 of 96.8—slightly above the previous GNN-only result (96.6) and well ahead of Shibata et al. (92.2). Chord grouping also improves to 96.9, and staff assignment climbs to 97.6%. While the specialized spelling model of Karystinaios and Widmer [6] attains 99.4% pitch and 96.9% key-signature accuracy, EngravingGNN's integrated predictions still perform strongly at 96.3% and 80.6%, respectively. We believe that the reason for the degraded performance is the absence of data augmentation, which may be crucial for pitch spelling on a dataset without much key variability. Stem direction (85.1%), octave shifts (99.9%), clef sign (96.2%), and symbolic duration (87.8%) likewise demonstrate high end-to-end engraving quality.

On the more demanding DCML Romantic corpus, which is characterized with attributes such as cross-staff beaming, frequent modulations, and dense polyphony, EngravingGNN again leads against the other models. Voice F1 rises marginally to 90.6 (vs. 89.9), chord F1 to 81.1 (vs. 79.5), and staff accuracy to 91.9% (vs. 91.0%). Notably, it matches the standalone spelling model on pitch (93.5%) and even surpasses it on key signature (50.5% vs. 46.5%), highlighting robust handling of chromatic passages and possible positive inter-task transfer. Stem direction (73.6%) and symbolic duration (83.3%) reflect the corpus's notational complexity. Overall, EngravingGNN delivers consistent, high-fidelity results across both straightforward and intricate piano repertoires.

We remind the reader that the above results are from a single unified model, trained on the combined training sets of both J-Pop and DCML Romantic. As Table 1 shows,

Model	Voice F1	Chord F1	Staff Acc	Pitch Spelling Acc	Key Acc	Stem Acc	Octave Acc	Clef Acc	Sym. Dur. Acc
J-Pop									
Spelling Model [6]	-	-	-	**99.4**	**96.9**	-	-	-	-
Shibata et al. [9]	92.2		92.8	-	-	-	-	-	-
Forscarin et al. [4]	96.6	94.9	96.3	-	-	-	-	-	-
EngravingGNN	**96.8**	**96.9**	**97.6**	96.3	80.6	85.1	99.9	96.2	87.8
DCML Romantic									
Spelling Model [6]	-	-	-	93.5	46.5	-	-	-	-
Shibata et al. [9]	84.9		88.5	-	-	-	-	-	-
Forscarin et al. [4]	89.9	79.5	91.0	-	-	-	-	-	-
EngravingGNN	**90.6**	**81.1**	**91.9**	93.5	**50.5**	73.6	100	90.0	83.3

Table 1. Comparison of the test-set results for our EngravingGNN model versus the Foscarin et al. [4] and Shibata et al. [9] on the voice and staff tasks and versus the Spelling model of [6] for pitch spelling and Key signature prediction.

this shared model delivers strong, consistent performance on both test sets, demonstrating EngravingGNN's ability to generalize across diverse piano repertoires without per-dataset tuning.

We do not evaluate against the model of Beyer and Dai [5] since it would not be a fair comparison, as the quantization task they perform is very complex and diminishes the performance of the score engraving subpart of their system.

3.4 Qualitative Analysis

Figure 4 illustrates our predictions on the first two bars of the first movements of Mozart's Sonata K310. Our method correctly captures symbolic duration, key, voices, staff, stem-direction, and spelling for each single note. We can also see a wrong prediction on the second half of the first beat, where the staff of an E which causes a cross staff assignment.

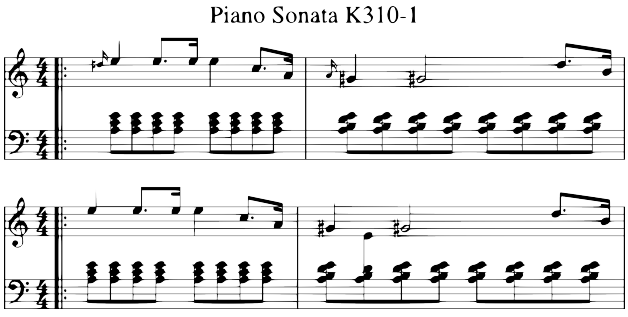

Figure 4. Engraving prediction on Sonata K310-1 by Mozart. Ground truth score on the top and EngravingGNN prediction on the bottom. Note that EngravingGNN, is not able to predict grace notes.

4. DISCUSSION AND LIMITATIONS

4.1 Treating Tied Notes

In its current version, our framework does not handle ties across multiple noteheads. In traditional engravings, ties connect notes across bars, across strong beats for better readability, or to form durations that cannot be expressed by one symbol alone. Since EngravingGNN's duration head predicts only one symbol per input duration, it cannot generate the necessary tie-edges or composite duration encodings. For this, autoregressive systems like the one of Bayer and Dai [5], which are naturally able to handle the "one-to-many" case, have a clear advantage. We believe that adding an autoregressive routine for the notehead task would be the way to improve in this direction.

4.2 Including Grace Notes

Our system does not currently support grace notes because grace notes lack quantized durations. However, grace notes are of crucial importance in many Baroque, Classical, and jazz passages, as they convey stylistic nuance. Proper engraving of grace notes would require flagging them in the input graph, linking them to their principal notes, and rendering them with small-noteheads with appropriate slurs. Extending EngravingGNN to handle these ornaments is not trivial but it would be a key step toward fully authentic and stylistically complete piano scores.

4.3 Postprocessing

While our postprocessing pipeline successfully pools chords, assigns voices, groups beams, and infills rests to produce valid MusicXML/MEI output, several limitations remain. First, symbolic-duration handling is limited to single-symbol durations and fixed tuplet brackets, so complex ties, composite rhythms, or wrong predictions can cause rendering errors in symbolic music visualizers. Second, global attributes such as key signature may exhibit measure-to-measure discontinuities, as we apply predictions per note without sequence-level smoothing. Finally, our beam grouping and rest-infilling steps rely on deterministic, rule-based algorithms rather than learned patterns, which can result in suboptimal engraving choices in corner-case scenarios (e.g. irregular beaming across voices or nonstandard rhythmic groupings).

4.4 Training and Optimization

Hyperparameters were selected from prior work [4]. We did not perform an extensive hyperparameter search over learning rates, layer depths, or batch sizes was conducted. Consequently, while our off-the-shelf settings are enough for our goal of showcasing the potential of our approach,

further gains may be possible through systematic hyperparameter search.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented EngravingGNN, a unified Hybrid-GNN framework that, with a single encoder and multi-head decoders, simultaneously predicts all essential piano engraving attributes, such as voice edges, staff assignment, pitch spelling, key signature, stem direction, octave shifts, and clef changes. After post-processing, our system generates ready-to-print MusicXML/MEI scores directly from quantized input, significantly reducing the manual effort traditionally required for high-quality engraving.

Despite these advances, our current model assumes each note's duration fits a single symbol in our predefined vocabulary, and does not handle tied notes or composite rhythms that span multiple noteheads or measures. It also leaves layout optimization, system breaks, staff spacing, and beam slopes, to external tools. Future work will extend the decoder to detect and represent ties and composite durations. Furthermore, we aim to explore end-to-end training with differentiable rendering feedback (e.g., via MEI/Verovio) to further close the gap between automated and professional human engraving.

6. ACKNOWLEDGEMENTS

This work was supported by the European Research Council (ERC) under Horizon 2020 grant #101019375 "Whither Music?".

7. REFERENCES

[1] F. Foscarin, P. Rigaux, and V. Thion, "Data quality assessment in digital score libraries: The gioqoso project," *International Journal on Digital Libraries*, vol. 22, no. 2, pp. 159–173, 2021.

[2] F. Foscarin, N. Audebert, and R. Fournier S'niehotta, "PKSpell: Data-driven pitch spelling and key signature estimation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[3] A. Bouquillard and F. Jacquemard, "Engraving oriented joint estimation of pitch spelling and local and global keys," *arXiv preprint arXiv:2402.10247*, 2024.

[4] F. Foscarin, E. Karystinaios, E. Nakamura, and G. Widmer, "Cluster and separate: a gnn approach to voice and staff prediction for score engraving," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[5] T. Beyer and A. Dai, "End-to-end piano performance-midi to score conversion with transformers," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[6] E. Karystinaios and G. Widmer, "Graphmuse: A library for symbolic music graph processing," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[7] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, vol. 40, no. 3, pp. 52–74, 2017.

[8] R. E. Burkard and E. Cela, "Linear assignment problems and extensions," in *Handbook of combinatorial optimization*. Springer, 1999, pp. 75–149.

[9] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.

[10] J. Hentschel, Y. Rammos, F. C. Moss, M. Neuwirth, and M. Rohrmeier, "An annotated corpus of tonal piano music from the long 19th century," *Empirical Musicology Review*, vol. 18, no. 1, pp. 84–95, 2023.

[11] C. E. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A python package for symbolic music processing," in *Proceedings of the Music Encoding Conference (MEC)*, Halifax, Canada, 2022.

[12] Y. Hiramatsu, E. Nakamura, and K. Yoshii, "Joint estimation of note values and voices for audio-to-score piano transcription." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 278–284.